

7.8.3.2.22 Timeout Control Register (I3Cm_TMOCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 0090h														
Initial Value :		0000_0030h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TOMDS[1:0]		TOHCTL	TOLCTL	-	-	TODTS[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7, 6	TOMDS[1:0]	0h	RW	Timeout Operation Mode Selection 00b: Timeout is detected during the following conditions: The bus is busy (BCST.BFREF = 0) in master mode. I3C's own slave address is detected and the bus is busy in slave mode. The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1). 01b: Timeout is detected while the bus is busy. 10b: Timeout is detected while the bus is free. 11b: Setting prohibited
5	TOHCTL	1h	RW	Timeout H Count Control 0b: Count is disabled while the SCL30 line is at a high level. 1b: Count is enabled while the SCL30 line is at a high level.
4	TOLCTL	1h	RW	Timeout L Count Control 0b: Count is disabled while the SCL30 line is at a low level. 1b: Count is enabled while the SCL30 line is at a low level.
3, 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1, 0	TODTS[1:0]	0h	RW	Timeout Detection Time Selection 00b: 16-bit timeout 01b: 14-bit timeout 10b: 8-bit timeout 11b: 6-bit timeout

TODTS[1:0] bits (Timeout Detection Time Selection)

These bits are used to select for the timeout detection time when the timeout function is enabled (BSTE.TODE bit = 1b).

When these bits are set to 00b, the timeout detection internal counter functions as a 16-bit counter.

When these bits are set to 01b, the counter functions as a 14-bit counter.

When these bits are set to 10b, the counter functions as an 8-bit counter.

When these bits are set to 11b, the counter functions as a 6-bit counter.

While the SCL line is in the state that enables this counter as specified by bits TOHCTL and TOLCTL, the counter counts up in synchronization with the internal reference clock (I3C ϕ) as a count source.

For details on the timeout function, see **(3) Timeout Error Detection**.

TOLCTL bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (BSTE.TODE = 1b).

TOHCTL bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (BSTE.TODE = 1b).

TOMDS[1:0] bits (Timeout Operation Mode Selection)

These bits are used to select the detection condition for timeout when the timeout function is enabled.

7.8.3.2.23 Wake-Up Unit Control Register (I3Cm_WUCTL)

Access Size : 32 bits

Address : <I3C_base> + 0098h

Initial Value : 0000_0041h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	WUFE	WUFSYNE	-	WUANFS	-	-	-	WUACKS
Initial Value	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7	WUFE	0h	RW	Wake-Up Function Enable 0b: Wake-up function disabled 1b: Wake-up function enabled Do not set WUFE = 0 during wake-up operation.
6	WUFSYNE	1h	RW	Wake-Up Function Synchronous Enable 0b: I3C asynchronous circuit enabled 1b: I3C synchronous circuit enabled
5	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	WUANFS	0h	RW	Wake-Up Analog Noise Filter Selection* ¹ 0b: Do not add the Wake-Up analog filter. 1b: Add the Wake-Up analog filter.
3 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	WUACKS	1h	RW	Wake-Up Acknowledge Selection* ¹ Choice of four response modes with a combination of RSTCTL.INTLRST bit and WUACKS bit. Shown in the Table 7.8-7 .

Note 1. This bit supports I²C mode.

Table 7.8-7 Wake-Up Mode

INTLRST	WUACKS	Operation Mode	Description
0	0	Normal Wake-up mode 1	ACK response at 9th SCL and SCL low-hold after at 9th SCL.
0	1	Normal Wake-up mode 2	No ACK response immediately and SCL low-hold between 8th and 9th SCL. Release SCL low-hold and ACK response at 9th SCL.
1	0	Command recovery mode	ACK response at 9th SCL and not SCL low-hold.
1	1	EEP response mode	NACK response at 9th SCL and not SCL low-hold.

Note: In Wake-up mode 2, HS mode cannot be used.

WUFSYNE bit (Wake-Up Function Synchronous Enable)

This bit is used to switch between the I3C_0_PCLK synchronous operation and the I3C_0_PCLK asynchronous operation.

The bit is used in combination with the WUASYNF flag when Wake-up function is effective (WUCTL.WUFE bit = 1b).

- When switching from the I3C_0_PCLK synchronous operation to the I3C_0_PCLK asynchronous operation:

- I3C operation changes into the I3C_0_PCLK asynchronous operation while the BCST.BFREF flag = 1, when the WUASYNF flag set to 1 while the WUFSYNE bit = 0b.
- The reception can operate without depending on the state of operation of I3C_0_PCLK (with I3C_0_PCLK stopped) after it switches to the I3C_0_PCLK asynchronous operation (Wake-up event detection operation).
- When switching from the I3C_0_PCLK asynchronous operation to the I3C_0_PCLK synchronous operation:
 - I3C operation changes into the I3C_0_PCLK synchronous operation at the following conditions. (At the same timing when the WUASYNF flag becomes 0)
 - In the case Wake-up event is detected: right after the WUFSYNE bit is set to 1b.
 - In the case Wake-up event is not detected: when STOP condition is detected after the WUFSYNE bit is set to 1b.

[Setting conditions]

- When 1b is written to the WUFSYNE bit.
- WUCTL.WUFE=0b

[Clearing condition]

- When 0b is written to the WUFSYNE bit.

7.8.3.2.24 Acknowledge Control Register (I3Cm_ACKCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 00A0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	ACKT WP	ACKT	ACKR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	ACKTWP	0h	W	ACKT Write Protect 0b: The ACKT bit are protected. 1b: The ACKT bit can be written (when writing simultaneously with the value of the target bit). This bit is read as 0b.
1	ACKT	0h	RW	Acknowledge Transmission 0b: A 0 is sent as the acknowledge bit (ACK transmission). 1b: A 1 is sent as the acknowledge bit (NACK transmission).
0	ACKR	0h	R	Acknowledge Reception 0b: A 0 is received as the acknowledge bit (ACK reception). 1b: A 1 is received as the acknowledge bit (NACK reception).

Note: This register supports I²C mode.

ACKR bit (Acknowledge Reception)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1b is received as the acknowledge bit with the PRSST.TRMD bit set to 1b.

[Clearing condition]

- When 0b is received as the acknowledge bit with the PRSST.TRMD bit set to 1b.

ACKT bit (Acknowledge Transmission)

[Setting condition]

- When 1b is written to the ACKT bit and 1b is written to the ACKTWP bit at the same time.

[Clearing conditions]

- When 0b is written to the ACKT bit and 1b is written to the ACKTWP bit at the same time.
- When a STOP condition is detected. (when a STOP condition is detected with the CNDCTL.SPCND bit set to 1b.)

Note: Set the ACKT bit to 0b in I²C Slave mode.

ACKTWP bit (ACKT Write Protect)

This bit is used to control the modification of the ACKT bit.

When changing the ACKT bit, setting this bit to 1b at the same time can change the ACKT bit.

When this bit is read, 0b is always read.

7.8.3.2.25 SCL Stretch Control Register (I3Cm_SCSTRCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 00A4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RWE	ACKTWE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	RW	Reserved Whenever it is read, 0 is read. The written value should always be 0b.
1	RWE	0h	RW	Receive Wait Enable 0b: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1b: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading NTDTBP0.
0	ACKTWE	0h	RW	Acknowledge Transmission Wait Enable 0b: NTST.RDBFF0 is set at the rising edge of the ninth SCL30 clock cycle. (The SCL30 line is not held low at the falling edge of the eighth clock cycle.) 1b: NTST.RDBFF0 is set at the rising edge of the eighth SCL30 clock cycle. (The SCL30 line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKCTL.ACKT bit.

Note: This register supports I²C mode.

ACKTWE bit (Acknowledge Transmission Wait Enable)

This bit is used to select the NTST.RDBFF0 flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When ACKTWE = 0b, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the NTST.RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When ACKTWE = 1b, the NTST.RDBFF0 flag is set to 1b at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKCTL.ACKT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKCTL.ACKT = 0b) or NACK (ACKCTL.ACKT = 1b) according to receive data.

RWE bit (Receive Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (NTDTBP0) is completely read each time single-byte data is received in receive mode.

When RWE = 0b, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the ACKTWE and RWE bits = 0b, continuous receive operation is enabled with the double buffer.

When RWE = 1b, the SCL line is held low from the falling edge of the ninth clock cycle until the NTDTBP0 value is read each time single-byte data is received.

This enables receive operation in byte units.

Note: When the value of the RWE bit is to be read, be sure to read the NTDTBP0 beforehand.

7.8.3.2.26 SCL Stalling Control Register (I3Cm_SCSTLCTL)

When setting this register, follow Chapter 5.1.2.5 Master Clock Stalling of MIPI I3C Spec V1.0, and use it only when necessary because of its negative impacts on bus performance.

Access Size :		32 bits														
Address :		<I3C_base> + 00B0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ACKPE	PARPE	-	AAPE	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STLCYC[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	ACKPE	0h	RW	ACK phase Enable Stall enable bit during ACK/NACK phase 0b: Does not stall the SCL30 clock during the ACK/NACK phase. 1b: Stall the SCL30 clock during the ACK/NACK phase.
30	PARPE	0h	RW	Parity Phase Enable Stall enable bit in parity bit period 0b: Does not stall the SCL30 clock during the parity bit period. 1b: Stall the SCL30 clock during the parity bit period.
29	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
28	AAPE	0h	RW	Assigned Address Phase Enable Enable bit that allows stall by the first bit at address assignment 0b: Does not stall the SCL30 clock during the address assignment phase. 1b: Stall the SCL30 clock during address assignment phase.
27 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 0	STLCYC[15:0]	0h	RW	Stalling Cycle Counter setting of stall period (I3C ϕ cycle). Common use for each phase.

Note: This register supports I3C master mode and I3C secondary master mode.

STLCYC[15:0] bits (Stalling Cycle)

These bits set the SCL stall period. The SCL stall period is counted by the internal reference clock (I3C ϕ). This is a counter common to the enable bits of each phase.

AAPE bit (Assigned Address Phase Enable)

The master can stall SCL during the low period of the first bit of the assigned address phase of the Enter Dynamic Address Assignment CCC command. It can gain time in assigning dynamic address to the device based on the BCR and DCR of the slave. However, because the Dynamic Address Assignment procedure sends the dynamic address set in the DATBASm register in sequence, it is not necessary to set this bit and it is prohibited.

PARPE bit (Parity Phase Enable)

The parity bit of the transmission data of I3C write transfer can be used for SCL stalling to avoid underrun of the transmission data FIFO. However, when the transmission data FIFO of the I3C master becomes empty, SCL stalling is

performed regardless of the setting of this bit, it is not necessary to set this bit and it is prohibited. It is necessary to set this bit when the I3C slave requires preparation time to receive data.

ACKPE bit (ACK phase Enable)

Determine the need to perform SCL stalling in the ACK/NACK phase based on the following criteria:

- It is necessary to set this bit when the I3C and I²C slaves connected to the bus require preparation time to receive or transmit data.
- In legacy I²C communication, if there is a possibility that the data FIFO of the I3C master might underrun or overflow, it is not necessary to set this bit because SCL Stalling is performed by FIFO Empty or Full regardless of the setting of this bit.
- Other than legacy I²C communication, the data FIFO of I3C master might underrun or overflow, and if SCL stalling is required in ACK phase, this bit can be set. However, it is necessary to build the software so that the FIFO does not underrun or overflow due to the interrupt generated according to the FIFO threshold setting (NQTHCTL, NTBTHCTL, NRQTHCTL).
- When I3C master responds ACK/NACK to IBI, it is not necessary to set this bit because ACK/NACK response can be set in advance by BCTL.HJACK, DATBASm.DVMRRJ and DATBASm.DVS IRRJ.
- It is necessary to set this bit when the I3C slave connected to the bus requires preparation time to transmit data for Direct GET CCC.

7.8.3.2.27 Slave Transfer Data Length Register (I3Cm_SVTDLG0)

Access Size :		32 bits														
Address :		<I3C_base> + 00C0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STDLG[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	STDLG[15:0]	0h	RW	Slave Transfer Data Length Indicates the number of bytes to be transferred.
15 to 0	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.28 Synchronous Timing Control Register (I3Cm_STCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 0120h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STOE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 1	-		All 0		RW		Reserved									
							Whenever it is read, 0b is read. The written value should always be 0b.									
0	STOE		0h		RW		Synchronous Timing Output Enable									
							0b: Disable									
							1b: Enable									

Note: This register supports all I3C mode.

7.8.3.2.29 Asynchronous Timing Control Register (I3Cm_ATCTL)

Access Size : 32 bits

Address : <I3C_base> + 0124h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDIV[7:0]							-	-	-	-	-	AMEOE	MREFOE	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 8	CDIV[7:0]	0h	RW	I3C_0_TCLK Counter Divide Setting* ¹
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	AMEOE	0h	RW	Additional Master-initiated bus Event Output Enable* ² 0b: Disable 1b: Enable
1	MREFOE	0h	RW	MREF Output Enable (Capture Event / Counter Overflow)* ² 0b: Disable 1b: Enable
0	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Note 1. This bit supports all I3C mode.

Note 2. This bit supports I3C master mode and I3C secondary master mode.

7.8.3.2.30 Asynchronous Timing Trigger Register (I3Cm_ATTRG)

Access Size :		32 bits														
Address :		<I3C_base> + 0128h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ATSTR G
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	W	Reserved The written value should always be 0b.
0	ATSTRG	0	W	The read value is undefined Asynchronous Timing Software Trigger 0b Write: Do nothing 1b Write: Software trigger (one-shot pulse) output This bit is always read as 0b

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.231 Asynchronous Timing Control Counter Enable Register (I3Cm_ATCCNTE)

Access Size :		32 bits														
Address :		<I3C_base> + 012Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ATCE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 1	-		All 0		RW		Description									
Reserved																
Whenever it is read, 0b is read. The written value should always be 0b.																
0	ATCE		0h		RW		Asynchronous Timing Counter Enable for MREF, MC2, SC1, SC2.									
0b: Disable																
1b: Enable																

Note: This register supports all I3C mode.

7.8.3.2.32 Condition Control Register (I3Cm_CNDCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 0140h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SPCND	SRCND	STCND
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	SPCND	0h	RW	STOP (P) Condition Issuance 0b: Does not request to issue a STOP condition. 1b: Requests to issue a STOP condition.
1	SRCND	0h	RW	Repeated START (Sr) Condition Issuance 0b: Does not request to issue a Repeated START condition. 1b: Requests to issue a Repeated START condition.
0	STCND	0h	RW	START (S) Condition Issuance 0b: Does not request to issue a START condition. 1b: Requests to issue a START condition.

Note: This register supports I²C mode.

STCND bit (START (S) Condition Issuance)

This bit is used to request transition to master mode and issuance of a START condition.

For details on the START condition issuance, see **(3) START Condition / Repeated START Condition / STOP Condition Issuing Function**.

[Setting condition]

- When 1b is written to the STCND bit

[Clearing conditions]

- When 0b is written to the STCND bit
- When a START condition has been issued (A START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1b

Note: Set the STCND bit to 1b (START condition issuance request) when the BCST.BFREF flag is set to 1b (bus free state).

Note that arbitration may be lost due to a START condition issuance error if the STCND bit is set to 1b (START condition issuance request) when the BFREF flag is set to 0b (bus busy state).

SRCND bit (Repeated START (Sr) Condition Issuance)

This bit is used to request that a Repeated START condition be issued in master mode.

When this bit is set to 1b to request to issue a Repeated START condition, a Repeated START condition is issued when the BFREF flag is set to 0b (bus busy state) and the PRSST.CRMS bit is set to 1b (master mode).

For details on the Repeated START condition issuance, see **(3) START Condition / Repeated START Condition / STOP Condition Issuing Function**.

[Setting condition]

- When 1b is written to the SRCND bit with the BCST.BFREF flag set to 0b

[Clearing conditions]

- When 0b is written to the SRCND bit
- When a Repeated START condition has been issued (A Repeated START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1b

Note: Do not set the SRCND bit to 1b while issuing a STOP condition.

If 1b (requests to issue a Repeated START condition) is written to the SRCND bit in slave mode, the Repeated START condition is not issued but the SRCND bit remains set to 1b.

If the operating mode changes to master mode with the bit not being cleared, note that the Repeated START condition may be issued.

SPCND bit (STOP (P) Condition Issuance)

This bit is used to request that a STOP condition be issued in master mode.

When this bit is set to 1b to request to issue a STOP condition, a STOP condition is issued when the BCST.BFREF flag is set to 0b (bus busy state) and the PRSST.CRMS bit is set to 1b (master mode).

For details on the STOP condition issuance, see **(3) START Condition / Repeated START Condition / STOP Condition Issuing Function**.

[Setting condition]

- When 1b is written to the SPCND bit with the BCST.BFREF flag set to 0b and the PRSST.CRMS bit set to 1b

[Clearing conditions]

- When 0b is written to the SPCND bit
- When a STOP condition has been issued (A STOP condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1b
- When a START condition and a Repeated START condition are detected

Note: Writing to the SPCND bit is not possible while the setting of the BCST.BFREF flag = 1b (bus free state).

Do not set the SPCND bit to 1b while a Repeated START condition is being issued.

7.8.3.2.33 Normal Command Queue Port Register (I3Cm_NCMDQP)

32-bit mailbox register NCMDQP contains a command descriptor structure that depends on the requested transfer type:

- Address Assignment Command (see **(1) Address Assign Command**)
- Immediate Data Transfer (see **(2) Immediate Transfer Command**)
- Regular Data Transfer (see **(3) Regular Transfer Command**)
- Write + Write/Read Combo Transfer (see **(4) Combo Transfer Command**)
- Internal Control Command (see **(5) Internal Control Command**)

Within the command descriptor, DWORDs appear starting with the Least Significant DWORD, in order until the Most Significant DWORD.

Access Size :		32 bits															
Address :		<I3C_base> + 0150h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	NCMDQP[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	NCMDQP[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NCMDQP[31:0]	-	W	Normal Command Queue Port

Note: This register supports all I3C mode.

7.8.3.2.34 Normal Response Queue Port Register (I3Cm_NRSPQP)

32-bit mailbox register NRSPQP contains a response structure (see 7.8.4.1.4 Receive Status Descriptor).

Access Size :		32 bits														
Address :		<I3C_base> + 0154h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NRSPQP[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRSPQP[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

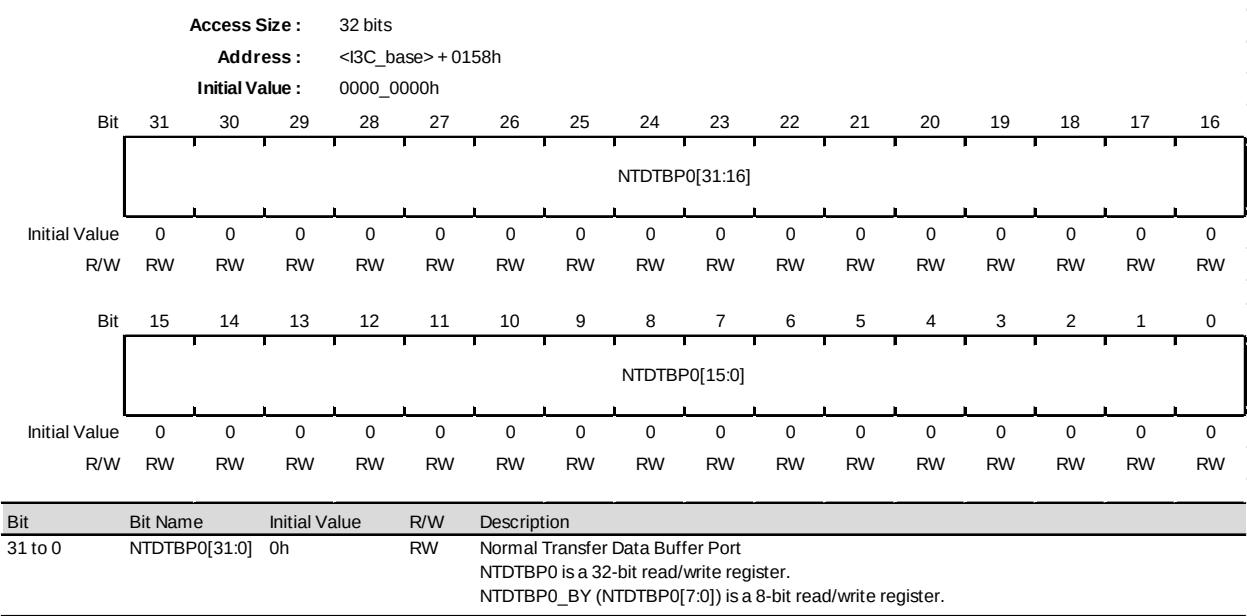
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRSPQP[31:0]	0h	R	Normal Response Queue Port

Note: This register supports all I3C mode.

7.8.3.235 Normal Transfer Data Buffer Port Register (I3Cm_NTDTBP0/NTDTBP0_BY)

32-bit mailbox register NTDTBP0 is a 32-bit bi-directional data transfer register which is used both to read from the Normal Receive Data Buffer, and to write to the Normal Transmit Data Buffer.

In other words, the Normal Receive Data Buffer and the Normal Transmit Data Buffer have the same offset, forming a single bidirectional port for transmitting or receiving I3C data.



Note: NTDTBP0 is for 32-bit access in I3C mode.
NTDTBP0_BY is for 8-bit access in I²C mode.

Read Operations:

[I3C protocol mode]

Data Read from the Normal Receive Data Buffer. It should be read based on Normal Queue Status Level indications. The Receive data is always aligned to a 4-byte boundary, and stored in the Normal Receive Data Buffer. If the length of the data transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. The valid data must be identified using the DATA_LENGTH field in the Response Descriptor.

[I²C protocol mode]

When 1 byte of data has been received, the received data is transferred from the internal shift register to NTDTBP0 to enable the next data to be received. The double-buffer structure of the internal shift register and NTDTBP0 allows continuous receive operation if the received data has been read from NTDTBP0 while the internal shift register is receiving data. Read data from NTDTBP0 once when a receive data full interrupt (INT_ri3c_rx_n) request is generated. If NTDTBP0 receives the next receive data before the current data is read from NTDTBP0 (while the RDBFF0 flag in NTST is 1b), this module automatically holds the SCL clock low one cycle before the RDBFF0 flag is set to 1b next. The lower 8 bits of the read 32-bit data are valid as received data.

Write Operations:

[I3C protocol mode]

Data Written to the Normal Transmit Data Buffer. Data DWORDs written to the Normal Transmit Data Buffer are placed onto the I3C bus one byte at a time, with the DWORD LSB first. Within each byte, bits are placed onto the I3C bus in big-endian order, with bit 7 going out first on the bus. The transmit data should always start aligned to a 4-byte

boundary, and written to the NTDTBP0 register. If the length of the transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. I3C shall only send the valid number of bytes indicated in the DATA_LENGTH field of the Command Descriptor.

[I²C protocol mode]

When NTDTBP0 detects a space in the internal shift register, it transfers the transmit data that has been written to NTDTBP0 to the internal shift register and starts transmitting data in transmit mode. The double-buffer structure of NTDTBP0 and the internal shift register allows continuous transmit operation if the next transmit data has been written to NTDTBP0 while the internal shift register data is being transmitted. Write transmit data to NTDTBP0 once when a transmit data empty interrupt (INT_ri3c_tx_n) request is generated. The lower 8 bits of the written 32-bit data are valid as transmission data.

7.8.3.2.36 Normal IBI Queue Port Register (I3Cm_NIBIQP)

Access Size :		32 bits														
Address :		<I3C_base> + 017Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIBIQP[31:16]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIBIQP[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NIBIQP[31:0]	0h	RW	Normal IBI Queue Port

Note: This register supports all I3C mode.

When receiving an IBI, 32-bit mailbox register NIBIQP is used for both:

- Read the IBI status descriptor (see **7.8.4.1.3 IBI Status Descriptor**)
- Read the IBI data (which is raw/opaque data).

The IBI status descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C bus.

Remark If the I3C HCI auto-read feature is used, then the IBI data includes the data received from the auto-generated private read operation.

Even if LAST_STATUS is set to 0b, the driver software still evaluates the data payload length by examining the CHUNKS field.

7.8.3.2.37 Normal Receive Status Queue Port Register (I3Cm_NRSQP)

32-bit mailbox register NRSQP contains a receive status structure (see 7.8.4.1.4 Receive Status Descriptor).

Access Size :		32 bits														
Address :		<I3C_base> + 0180h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NRSQP[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRSQP[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NRSQP[31:0]	0h	R	Normal Receive Status Queue Port

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.38 Normal Queue Threshold Control Register (I3Cm_NQTHCTL)

The Queue Threshold Control register controls the interrupt trigger thresholds for the Command Queue, the Response Queue, and the IBI Queue.

Access Size :		32 bits														
Address :		<I3C_base> + 0190h														
Initial Value :		0101_0101h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	IBIQTH[2:0]			IBIDSSZ[7:0]							
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RSPQTH[1:0]		-	-	-	-	-	-	CMDQTH[1:0]	
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
26 to 24	IBIQTH[2:0]	1h	RW	Normal IBI Queue Threshold* ¹ 000b: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is 1 or more. I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer is completely empty. Others: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is N + 1 or more. (N = IBIQTH[2:0]) I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer contains N empties. (N = IBIQTH[2:0])
23 to 16	IBIDSSZ[7:0]	1h	RW	Normal IBI Data Segment Size* ² Supported Values: Minimum: 1 (4 bytes) Maximum: 63 (252 bytes), provided that the configured IBI Queue depth is 64 or more. When ATCCNTEATCE = 1, restrict to the number of segments is 2 or more.
15 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9, 8	RSPQTH[1:0]	1h	RW	Normal Response Queue Threshold* ¹ 00b: Interrupt is issued when Response Queue contains 1 entry (DWORD). 01b: Interrupt is issued when Response Queue contains 2 entries (DWORD). 10b: Interrupt is issued when Response Queue contains 3 entries (DWORD). 11b: Interrupt is issued when Response Queue contains 4 entries (DWORD).
7 to 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1, 0	CMDQTH[1:0]	1h	RW	Normal Command Ready Queue Threshold* ¹ 00b: Interrupt is issued when Command Queue is completely empty. 01b: Interrupt is issued when Command Queue contains 1 empty. 10b: Interrupt is issued when Command Queue contains 2 empties. 11b: Interrupt is issued when Command Queue contains 3 empties.

Note: This register supports I3C secondary master mode and I3C slave mode.

Note 1. These bits support all I3C mode.

Note 2. These bits support I3C master mode and I3C secondary master mode.

CMDQTH[1:0] bits (Normal Command Ready Queue Threshold)

Controls the minimum number of Command Queue empties needed to trigger the INTCMD interrupt.

RSPQTH[1:0] bits (Normal Response Queue Threshold)

Controls the minimum number of Response Queue entries needed to trigger the INTRESP interrupt.

IBIDSSZ[7:0] bits (Normal IBI Data Segment Size)

This is the IBI data segment size, in DWORDs (4 bytes).

In PIO mode, this field allows the incoming IBI data to be sliced into multiple segments generating status individually, to support cutthrough readout of a long IBI payload data.

When Asynchronous Timing Control mode is supported, this field should be set to a value other than 1 or 3 to allow the single data segment to contain the entire Master time-stamp value (i.e., both MREF and MC2).

IBIQTH[2:0] bits (Normal IBI Queue Threshold)

For I3C protocol mode (Master): PRTS.PRTMD = 0b and PRSST.CRMS = 1b.

Controls generation of the INTIBI interrupt, based on the value of the IBI Queue's Outstanding IBI status count.

Each IBI status entry can represent either the complete IBI payload (if the IBI payload byte size is $4 \times \text{IBIDSSZ}$ or less), or a segment of the IBI payload (if the IBI payload byte size is more than $4 \times \text{IBIDSSZ}$).

For I3C protocol mode (Slave): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b.

Controls the minimum number of IBI Data Buffer empties needed to trigger the INTIBI interrupt.

7.8.3.2.39 Normal Transfer Data Buffer Threshold Control Register (I3Cm_NTBTHCTL0)

The Data Buffer Control register controls the interrupt trigger thresholds for the Receive Data Buffer Queue and the Transmit Data Buffer Queue.

Access Size :		32 bits														
Address :		<I3C_base> + 0194h														
Initial Value :		0101_0101h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	RXSTTH[2:0]			-	-	-	-	-	TXSTTH[2:0]		
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RXDBTH[2:0]			-	-	-	-	-	TXDBTH[2:0]		
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
26 to 24	RXSTTH[2:0]	1h	RW	Normal Receive Start Threshold* ² 000b: Wait for 2 empty DWORDs 001b: Wait for 4 empty DWORDs 010b: Wait for 8 empty DWORDs 011b: Wait for 16 empty DWORDs Others: Setting prohibited
23 to 19	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18 to 16	TXSTTH[2:0]	1h	RW	Normal Transmit Start Threshold* ² 000b: Wait for 2 entry DWORDs 001b: Wait for 4 entry DWORDs 010b: Wait for 8 entry DWORDs 011b: Wait for 16 entry DWORDs Others: Setting prohibited
15 to 11	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10 to 8	RXDBTH[2:0]	1h	RW	Normal Receive Data Buffer Threshold* ¹ 000b: Interrupt triggers at 2 Receive Buffer entries, in DWORDs 001b: Interrupt triggers at 4 Receive Buffer entries, in DWORDs 010b: Interrupt triggers at 8 Receive Buffer entries, in DWORDs Others: Setting prohibited
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2 to 0	TXDBTH[2:0]	1h	RW	Normal Transmit Data Buffer Threshold* ¹ 000b: Interrupt triggers at 2 Transmit Buffer empties, in DWORDs 001b: Interrupt triggers at 4 Transmit Buffer empties, in DWORDs 010b: Interrupt triggers at 8 Transmit Buffer empties, in DWORDs Others: Setting prohibited

Note 1. These bits support all I3C mode.

Note 2. These bits support I3C master mode and I3C secondary master mode.

TXDBTH[2:0] bits (Normal Transmit Data Buffer Threshold)

Minimum number of Transmit FIFO empties, in DWORDs, that will trigger the INT_ri3c_tx_n interrupt.

RXDBTH[2:0] bits (Normal Receive Data Buffer Threshold)

Minimum number of Receive FIFO entries in DWORDs that will trigger the INT_ri3c_rx_n interrupt.

TXSTTH[2:0] bits (Normal Transmit Start Threshold)

When preparing to initiate a Write Transfer on the I3C Bus, I3C shall wait until the Transmit Buffer has at least the indicated number of locations available.

Two optional configurable Modes are available:

(1) Store and Forward Mode

If the TXSTTH[2:0] field is set to the Transmit Buffer size (011b), then I3C shall delay initiation of the Write Command as follows:

- If the data length to be transferred is more than the Transmit Buffer size (16 DWORDs), then this module shall wait until the Transmit FIFO is completely full.
- If the data length to be transferred is less than the Transmit Buffer size (16 DWORDs), then I3C shall wait until enough Transmit FIFO locations are available to store the data to be transferred.

(2) Threshold Mode

If the TXSTTH[2:0] field value is less than the Transmit Buffer size (011b), then I3C shall initiate the Write Command as soon as the indicated number of Transmit FIFO locations are entries.

RXSTTH[2:0] bits (Normal Receive Start Threshold)

When preparing to initiate a Read Transfer on the I3C bus, I3C shall wait until the Receive Buffer has at least the indicated number of empty locations in DWORDs.

Two optional configurable Modes are available:

(1) Store and Forward Mode

If the RXSTTH[2:0] field is set to the Receive Buffer size (011b), then I3C shall delay initiation of the Read Command as follows:

- If the data length to be transferred is more than the Receive Buffer size (16 DWORDs), then this module shall wait until the Receive FIFO is completely empty.
- If the data length to be transferred is less than the Receive Buffer size (16 DWORDs), then I3C shall wait until enough Receive FIFO locations are available to store the data to be transferred.

(2) Threshold Mode

If the RXSTTH[2:0] field value is less than the Receive Buffer size (011b), then I3C shall initiate the Read Command as soon as the indicated number of Receive FIFO locations are empty.

7.8.3.2.40 Normal Receive Status Queue Threshold Control Register (I3Cm_NRQTHCTL)

Access Size :		32 bits														
Address :		<I3C_base> + 01C0h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RSQTH
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 1	-															
	Initial Value		All 0		R/W		Description									
							Reserved									
							Whenever it is read, 0b is read. The written value should always be 0b.									
0	RSQTH		1h		RW		Normal Receive Status Queue Threshold									
							0b: Interrupt is issued when Receive Status Queue contains 1 entry (DWORD).									
							1b: Interrupt is issued when Receive Status Queue contains 2 entries (DWORD).									

RSQTH bit (Normal Receive Status Queue Threshold)

Controls the minimum number of receive status queue entries needed to trigger the INTRCV interrupt.

7.8.3.2.41 Bus Status Register (I3Cm_BST)

Access Size : 32 bits

Address : <I3C_base> + 01D0h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	WUCNDDF	-	-	-	TODF	-	-	-	ALF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TENDF	-	-	-	NACKDF	-	-	SPCND DF	STCND DF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
24	WUCNDDF	0h	RW ^{*2}	Wake-Up Condition Detection Flag 0b: Wake-Up Condition is not detected. 1b: Wake-Up Condition is detected.
23 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TODF	0h	RW ^{*2}	Timeout Detection Flag 0b: Timeout is not detected. 1b: Timeout is detected.
19 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	ALF	0h	RW ^{*2}	Arbitration Lost Flag ^{*1} 0b: Arbitration is not lost 1b: Arbitration is lost.
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	TENDF	0h	RW ^{*2}	Transmit End Flag ^{*1} 0b: Data is being transmitted. 1b: Data has been transmitted.
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	NACKDF	0h	RW ^{*2}	NACK Detection Flag ^{*1} 0b: NACK is not detected. 1b: NACK is detected.
3, 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SPCND DF	0h	RW ^{*2}	STOP Condition Detection Flag 0b: STOP condition is not detected. 1b: STOP condition is detected.
0	STCND DF	0h	RW ^{*2}	START Condition Detection Flag 0b: START condition is not detected. 1b: START condition is detected.

Note 1. This bit supports I²C mode.

Note 2. Clearing (to 0) condition: Writing 0b after 1b is read.

STCND DF bit (START Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:

- 1) The BSTE.STCNDDE bit = 1b.
Thexxx
- 2) When a START condition (or a Repeated START condition) is detected.

[Clearing conditions]

- When 0b is written to the STCNDDF flag after reading STCNDDF flag = 1b.
- When a STOP condition is detected.

SPCNDDF bit (STOP Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The BSTE.SPCNDDE bit = 1b.
 - 2) When a STOP condition is detected.

[Clearing condition]

- When 0b is written to the SPCNDDF flag after reading SPCNDDF flag = 1b.

NACKDF bit (NACK Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The PRTS.PRTMD bit = 1b (I²C protocol mode).
 - 2) The BSTE.NACKDE bit = 1b (Enables NACK detection interrupt status logging).
 - 3) When acknowledge is not received (NACK is received) from the receive device in transmit mode.

[Clearing condition]

- When 0b is written to the NACKDF flag after reading NACKDF flag = 1b.

TENDF bit (Transmit End Flag)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The PRTS.PRTMD bit = 1b (I²C protocol mode).
 - 2) The BSTE.TENDE bit = 1b (Enables Transmit End Interrupt Status logging).
 - 3) At the rising edge of the ninth SCL clock cycle while the NTST.TDBEF0 flag = 1b. Excluding when sending an address.

[Clearing conditions]

- When 0b is written to the TENDF flag after reading TENDF flag = 1b.
- When data is written to the NTDTBP0 register.
- When a STOP condition is detected.

ALF bit (Arbitration Lost Flag)

[Setting conditions]

- When master arbitration-lost detection is enabled: BSTE.ALE bit = 1b, BFCTL.MALE = 1b.

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state)).
- All of the followings are satisfied.
 - 1) When the START condition is detected while the CNDCTL.STCND bit = 1b.
 - 2) When the internal SDA output state does not match the SDA line level.
- When the CNDCTL.STCND bit is set to 1 (START condition issuance request) while the BCST.BFREF flag = 0b.
- When NACK arbitration-lost detection is enabled: BSTE.ALE bit = 1b, BFCTL.NALE = 1b.
 - When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode.
- When slave arbitration-lost detection is enabled: BSTE.ALE bit = 1b, BFCTL.SALE = 1b.
 - When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0b is written to the ALF flag after reading ALF flag = 1b.

TODF bit (Timeout Detection Flag)

[Setting conditions]

- All of the followings are satisfied.
 - 1) The BSTE.TODE bit = 1b (Enables Timeout Detection Interrupt Status logging).
 - 2) When the master mode or the received slave address matches the slave address n in Slave mode.
 - 3) When the SCL line state remains unchanged for the period specified by TMOCTL register.

[Clearing condition]

- When 0b is written to the TODF flag after reading TODF flag = 1b.

WUCNDDE bit (Wake-Up Condition Detection Flag)

[Setting conditions]

- For I²C Protocol mode: PRTS.PRTMD bit = 1b.
When I3C_0_PCLK and I3C_0_TCLK are supplied after all of the following 1) to 4) are satisfied.
 - 1) The WUCTL.WUFE bit = 1b (Wake-up function is enabled).
 - 2) The BSTE.WUCNDDE bit = 1b (Enables Wake-up Condition Detection Status logging).
 - 3) The WUST.WUASYNF flag = 1b.
 - 4) When the address received in slave mode matches the address of slave enabled in the SVCTL.SVAEn bit (except for the Device-ID address).
- For I3C Protocol mode (Master): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 1b.
When I3C_0_PCLK and I3C_0_TCLK are supplied after all of the following 1) to 4) are satisfied.
 - 1) The WUCTL.WUFE bit = 1b (Wake-up function is enabled).
 - 2) The BSTE.WUCNDDE bit = 1b (Enables Wake-up Condition Detection Status logging).

- 3) The WUST.WUASYNF flag = 1b.
 - 4) When low level of the SDA line is detected (When the START condition is detected).
- For I3C Protocol mode (Slave) : PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b.
- When I3C_0_PCLK and I3C_0_TCLK are supplied after all of the following 1) to 4) are satisfied.
- 1) The WUCTL.WUFE bit = 1b (Wake-up function is enabled).
 - 2) The BSTE.WUCNDDE bit = 1b (Enables Wake-up Condition Detection Status logging).
 - 3) The WUST.WUASYNF flag = 1b.
 - 4) When the Broadcast Address (7Eh) is detected after a START (or Repeated START) condition and the own Dynamic Address is detected after the Repeated START condition following the Broadcast Address.

[Clearing condition]

- When 0b is written to the WUCNDDE flag after reading WUCNDDE flag = 1b while the WUST.WUASYNF flag = 0b.

7.8.3.2.42 Bus Status Enable Register (I3Cm_BSTE)

Access Size : 32 bits

Address : <I3C_base> + 01D4h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	WUCNDDE	-	-	-	TODE	-	-	-	ALE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TENDE	-	-	-	NACKDE	-	-	SPCNDDE	STCNDDE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
24	WUCNDDE	0h	RW	Wake-Up Condition Detection Enable 0b: Disables Wake-Up Condition Detection Status logging 1b: Enables Wake-Up Condition Detection Status logging
23 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TODE	0h	RW	Timeout Detection Enable 0b: Disables Timeout Detection Interrupt Status logging. 1b: Enables Timeout Detection Interrupt Status logging.
19 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	ALE	0h	RW	Arbitration Lost Enable* ¹ 0b: Disables Arbitration Lost Interrupt Status logging. 1b: Enables Arbitration Lost Interrupt Status logging.
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	TENDE	0h	RW	Transmit End Enable* ¹ 0b: Disables Transmit End Interrupt Status logging. 1b: Enables Transmit End Interrupt Status logging.
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	NACKDE	0h	RW	NACK Detection Enable* ¹ 0b: Disables NACK Detection Interrupt Status logging. 1b: Enables NACK Detection Interrupt Status logging.
3, 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SPCNDDE	0h	RW	STOP Condition Detection Enable 0b: Disables STOP condition Detection Interrupt Status logging. 1b: Enables STOP condition Detection Interrupt Status logging.
0	STCNDDE	0h	RW	START Condition Detection Enable 0b: Disables START condition Detection Interrupt Status logging. 1b: Enables START condition Detection Interrupt Status logging.

Note 1. This bit supports I²C mode.

STCNDDE bit (START Condition Detection Enable)

When this bit is 1b, operation of BST.STCNDDE is enabled. For the setting conditions and clearing conditions of the BST.STCNDDE flag, see the details of BST.STCNDDE.

SPCNDDE bit (STOP Condition Detection Enable)

When this bit is 1b, operation of BST.SPCNDDE is enabled. For the setting conditions and clearing conditions of the BST.SPCNDDE flag, see the details of BST.SPCNDDE.

NACKDE bit (NACK Detection Enable)

When this bit is 1b, the operation of BST.NACKDE is enabled. This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1b. For the setting conditions and clearing conditions of the BST.NACKDE flag, see the details of BST.NACKDE.

TENDE bit (Transmit End Enable)

When this bit is 1b, the operation of BST.TENDE is enabled. For the setting conditions and clearing conditions of the BST.TENDE flag, see the details of BST.TENDE.

ALE bit (Arbitration Lost Enable)

When this bit is 1b, the operation of BST.ALE is enabled. For the setting conditions and clearing conditions of the BST.ALE flag, see the details of BST.ALE.

TODE bit (Timeout Detection Enable)

When this bit is 1b, the operation of BST.TODE is enabled. For the setting conditions and clearing conditions of the BST.TODE flag, see the details of BST.TODE.

WUCNDDE bit (Wake-Up Condition Detection Enable)

When this bit is 1b, the operation of BST.WUCNDDE is enabled.

For the setting conditions and clearing conditions of the BST.WUCNDDE flag, see the details of BST.WUCNDDE.

7.8.3.2.43 Bus Interrupt Enable Register (I3Cm_BIE)

The BIE register enables signaling of outstanding bus interrupts received by I3C.

Access Size :		32 bits														
Address :		<I3C_base> + 01D8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	WUCN DDIE	-	-	-	TODIE	-	-	-	ALIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TENDIE	-	-	-	NACKD IE	-	-	SPCND DIE	STCND DIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
24	WUCNDDIE	0h	RW	Wake-Up Condition Detection Interrupt Enable 0b: Disables Wake-Up Condition Detection Interrupt Signal. 1b: Enables Wake-Up Condition Detection Interrupt Signal.
23 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TODIE	0h	RW	Timeout Detection Interrupt Enable 0b: Disables Timeout Detection Interrupt Signal. 1b: Enables Timeout Detection Interrupt Signal.
19 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	ALIE	0h	RW	Arbitration Lost Interrupt Enable* ¹ 0b: Disables Arbitration Lost Interrupt Signal. 1b: Enables Arbitration Lost Interrupt Signal.
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	TENDIE	0h	RW	Transmit End Interrupt Enable* ¹ 0b: Disables Transmit End Interrupt Signal. 1b: Enables Transmit End Interrupt Signal.
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	NACKDIE	0h	RW	NACK Detection Interrupt Enable* ¹ 0b: Disables NACK Detection Interrupt Signal. 1b: Enables NACK Detection Interrupt Signal.
3, 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SPCNDIE	0h	RW	STOP Condition Detection Interrupt Enable 0b: Disables STOP condition Detection Interrupt Signal. 1b: Enables STOP condition Detection Interrupt Signal.
0	STCNDIE	0h	RW	START Condition Detection Interrupt Enable 0b: Disables START condition Detection Interrupt Signal. 1b: Enables START condition Detection Interrupt Signal.

Note 1. This bit supports I²C mode.

STCNDIE bit (START Condition Detection Interrupt Enable)

This bit enables or disables the START Condition Detection interrupt requests when the BST.STCNDIE flag is set to 1.

SPCNDDIE bit (STOP Condition Detection Interrupt Enable)

This bit enables or disables the STOP Condition Detection interrupt requests when the BST.SPCNDDF flag is set to 1b.

NACKDIE bit (NACK Detection Interrupt Enable)

This bit enables or disables the NACK Detection interrupt requests when the BST.NACKDF flag is set to 1b.

TENDIE bit (Transmit End Interrupt Enable)

This bit enables or disables the Transmit End interrupt (I3C_TEND) requests when the BST.TENDF flag is set to 1b.

ALIE bit (Arbitration Lost Interrupt Enable)

This bit enables or disables the Arbitration Lost interrupt requests when the BST.ALF flag is set to 1b.

TODIE bit (Timeout Detection Interrupt Enable)

This bit enables or disables the Timeout Detection interrupt requests when the BST.TODF flag is set to 1b.

WUCNDDIE bit (Wake-Up Condition Detection Interrupt Enable)

This bit is used to enable or disable the Wake-up condition detection interrupt (INTWU) requests when the BST.WUCNDDF flag is set to 1b.

7.8.3.2.44 Bus Status Force Register (I3Cm_BSTFC)

Access Size : 32 bits

Address : <I3C_base> + 01DCh

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	WUCN DDFC	-	-	-	TODFC	-	-	-	ALFC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	W	R	R	R	W	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TENDF C* ²	-	-	-	NACKD FC	-	-	SPCND DFC	STCND DFC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	W	R	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
24	WUCNDDFC	0h	W	Wake-Up Condition Detection Force 0b: Not Force Wake-Up Condition Detection Interrupt for software testing. 1b: Force Wake-Up Condition Detection Interrupt for software testing.
23 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	TODFC	0h	W	The read value is undefined Timeout Detection Force 0b: Not Force Timeout Detection Interrupt for software testing. 1b: Force Timeout Detection Interrupt for software testing.
19 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	ALFC	0h	W	The read value is undefined Arbitration Lost Force* ¹ 0b: Not Force Arbitration Lost Interrupt for software testing. 1b: Force Arbitration Lost Interrupt for software testing.
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	TENDFC* ²	0h	W	The read value is undefined Transmit End Force* ¹ 0b: Not Force Transmit End Interrupt for software testing. 1b: Force Transmit End Interrupt for software testing.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	NACKDFC	0h	W	The read value is undefined NACK Detection Force* ¹ 0b: Not Force NACK Detection Interrupt for software testing. 1b: Force NACK Detection Interrupt for software testing.
3	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	-	0h	W	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SPCNDDFC	0h	W	The read value is undefined STOP condition Detection Force 0b: Not Force STOP condition Detection Interrupt for software testing. 1b: Force STOP condition Detection Interrupt for software testing.
0	STCNDDFC	0h	W	The read value is undefined START condition Detection Force 0b: Not Force START condition Detection Interrupt for software testing. 1b: Force START condition Detection Interrupt for software testing.

Note 1. This bit supports I²C mode.

Note 2. TENDFC does not work unless TDBEF0 = 1b.

7.8.3.2.45 Normal Transfer Status Register (I3Cm_NTST)

Access Size : 32 bits

Address : <I3C_base> + 01E0h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	RSQFF	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TEF	-	-	-	TABTF	RSPQFF	CMDQEF	IBIQEF	RDBFF	TDBEF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	RSQFF	0h	RW ³	Normal Receive Status Queue Full Flag ² 0b: The number of Receive Status Queue entries is less than or equal to the NRQTHCTL.RSQTH threshold. 1b: The number of Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold.
19 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	TEF	0h	RW ³	Normal Transfer Error Flag ¹ 0b: Transfer Error does not occur. 1b: Transfer Error occurs. To clear this bit, write 0 after 1 is read.
8 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	TABTF	0h	RW ³	Normal Transfer Abort Flag ¹ 0b: Transfer Abort does not occur. 1b: Transfer Abort occur. To clear this bit, write 0 after 1 is read.
4	RSPQFF	0h	RW ³	Normal Response Queue Full Flag ¹ 0b: The number of Response Queue entries is less than or equal to the NQTHCTL.RSPQTH threshold. 1b: The number of Response Queue entries is more than the NQTHCTL.RSPQTH threshold.
3	CMDQEF	0h	RW ³	Normal Command Queue Empty Flag ¹ 0b: If the NQTHCTL.CMDQTH = 0b: The number of Command Queue empties is less than the Command Queue size. If the NQTHCTL.CMDQTH is other than 0b: The number of Command Queue empties is less than the NQTHCTL.CMDQTH threshold. 1b: If the NQTHCTL.CMDQTH = 0b: The number of Command Queue empties is the Command Queue size. If the NQTHCTL.CMDQTH is other than 0b: The number of Command Queue empties is more than or equal to the NQTHCTL.CMDQTH threshold.

Bit	Bit Name	Initial Value	R/W	Description
2	IBIQEFF	0h	RW ^{*3}	Normal IBI Queue Empty/Full Flag^{*1} 0b: For I3C protocol mode (Master): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 1b. The number of IBI Status Queue entries is less than or equal to the NQTHCTL.IBIQTH threshold. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b. If the NQTHCTL.IBIQTH = 0b: The number of IBI Data Buffer empties is less than the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0b: The number of IBI Data Buffer empties is less than the NQTHCTL.IBIQTH threshold. 1b: For I3C protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1b. The number of IBI Status Queue entries is more than the NQTHCTL.IBIQTH threshold. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b. If the NQTHCTL.IBIQTH = 0b: The number of IBI Data Buffer empties is the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0b: The number of IBI Data Buffer empties is more than or equal to the NQTHCTL.IBIQTH threshold.
1	RDBFF0	0h	RW ^{*3}	Normal Receive Data Buffer Full Flag^{*1} 0b: For I2C protocol mode: PRTS.PRTMD bit = 1b. Normal Receive Data Buffer contains no receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0b. The number of entries in the Normal Receive Data Buffer is less than the NTBTHCTL0.RXDBTH[2:0] threshold. 1b: For I2C protocol mode: PRTS.PRTMD bit = 1b. Normal Receive Data Buffer contains receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0b. The number of entries in the Normal Receive Data Buffer is more than or equal to the NTBTHCTL0.RXDBTH[2:0] threshold.
0	TDBEF0	0h	RW ^{*3}	Normal Transmit Data Buffer Empty Flag^{*1} 0b: For I2C protocol mode: PRTS.PRTMD bit = 1b. Normal Transmit Data Buffer contains transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0b. The number of empties in the Normal Transmit Data Buffer is less than the NTBTHCTL0.TXDBTH[2:0] threshold. 1b: For I2C protocol mode: PRTS.PRTMD bit = 1b. Normal Transmit Data Buffer contains no transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0b. The number of empties in the Normal Transmit Data Buffer is more than or equal to the NTBTHCTL0.TXDBTH[2:0] threshold.

Note 1. This bit supports all I3C mode.

Note 2. This bit supports I3C secondary master mode and I3C slave mode.

Note 3. Clearing (to 0) condition: Writing 0b after 1b is read.

TDBEF0 bit (Normal Transmit Data Buffer Empty Flag)

[Setting conditions]

- For I²C Protocol mode: PRTS.PRTMD bit = 1b.

The following condition 1 is satisfied and any of the following conditions 2) to 4) are satisfied:

- 1) The NTSTE.TDBEE0 bit = 1b (enables Transmit Data Buffer Empty Interrupt Status logging).
- 2) When data has been transferred from the Normal Transmit Data Buffer to the Shift Register and the Normal Transmit Data Buffer becomes empty^{*1}.
- 3) When the PRSST.TRMD bit is set to 1b.
- 4) 4When the received slave address matches while the TRMD bit = 1b.

- For I3C Protocol mode: PRTS.PRTMD bit = 0b.

The following conditions 1) and 2) are satisfied:

- 1) The NTSTE.TDBEE0 bit = 1b (enables Transmit Data Buffer Empty Interrupt Status logging).

- 2) When the number of empties in the Normal Transmit Data Buffer is the NTBTHCTL0.TXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

- For I²C Protocol mode: PRTS.PRTMD bit = 1b.
 - When data is written to NTDTBP0.
 - When the TRMD bit in PRSST is set to 0b.
- For I3C protocol mode: PRTS.PRTMD bit = 0b.
 - Write 0b to this bit after 1b is read.
 - On completion of the last write access to Normal Transmit Data by DMAC.
 - When the number of empties in the Normal Transmit Data Buffer is less than the NTBTHCTL0.TXDBTH[2:0] threshold (see NTBTHCTL0 register).

Note 1. When the BST.NACKDF flag is set to 1b while the BSTE.NACKDE bit = 1b, I3C aborts data transmission/reception. If the TDBEF0 flag = 0b (next transmit data has been written), data is transferred to the Shift Register and the Normal Transmit Data Buffer register becomes empty at the rising edge of the 9th clock cycle, but the TDBEF0 flag is not set to 1b.

RDBFF0 bit (Normal Receive Data Buffer Full Flag)

[Setting conditions]

- For I²C Protocol mode: PRTS.PRTMD bit = 1b.

The following condition 1 is satisfied and any of the following condition 2) or 3) is satisfied:

- 1) The NTSTE.RDBFE0 bit = 1b (enables Receive Data Buffer Full Interrupt Status logging).
- 2) When receive data is transferred from Shift Register to Normal Receive Data Buffer.
The RDBFF0 flag is set to 1b on the rising edge of the 8th or 9th SCL clock cycle (selected in the ACKTWE bit in SCSTRCTL).
- 3) When the received slave address matches after a START (or Repeated START) condition is detected with the TRMD bit in PRSST set to 0b.

- For I3C Protocol mode: PRTS.PRTMD bit = 0b. The following conditions 1) and 2) are satisfied:

- 1) The NTSTE.RDBFE0 bit = 1b (enables Receive Data Buffer Full Interrupt Status logging).
- 2) When the number of Normal Receive Data Buffer entries is the NTBTHCTL0.RXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

- For I²C Protocol mode: PRTS.PRTMD bit = 1b.
 - When data is read from NTDTBP0.
- For I3C Protocol mode: PRTS.PRTMD bit = 0b.
 - Write 0b to this bit after 1b is read.
 - On completion of the last read access to Normal Receive Data by DMAC.
 - When the number of Normal Receive Data Buffer entries is less than the NTBTHCTL0.RXDBTH[2:0] threshold (see NTBTHCTL0 register).

IBIQEFF bit (Normal IBI Queue Empty/Full Flag)

[Setting conditions]

- For I3C Protocol mode (Master): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 1b.

The following 1) and 2) are satisfied.

- 1) The NTSTE.IBIQEFE bit = 1b. (Enables IBI Status Buffer Empty/Full Interrupt Status logging)
- 2) When the number of IBI Status Queue entries is more than the NQTHCTL.IBIQTH threshold (see register NQTHCTL).

- For I3C Protocol mode (Slave) : PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b.

If the NQTHCTL.IBIQTH = 0b:

The following 1) and 2) are satisfied.

- 1) The NTSTE.IBIQEFE bit = 1b. (Enables IBI Status Buffer Empty/Full Interrupt Status logging)
- 2) When IBI Data Buffer is completely empty

- If the NQTHCTL.IBIQTH is other than 0b:

The following 1) and 2) are satisfied.

- 1) The NTSTE.IBIQEFE bit = 1b. (Enables IBI Status Buffer Empty/Full Interrupt Status logging)
- 2) When the number of IBI Data Buffer empties is the NQTHCTL.IBIQTH threshold or more (see register NQTHCTL).

[Clearing conditions]

- For I3C protocol mode (master): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 1b.

- Write 0b to this bit after 1b is read.
- On completion of the last read access to IBI Status by DMAC.
- When the number of IBI Status Queue entries is the NQTHCTL.IBIQTH threshold or less (see NQTHCTL register).

- For I3C protocol mode (slave): PRTS.PRTMD bit = 0b, PRSST.CRMS bit = 0b.

- Write 0b to this bit after 1b is read.
- On completion of the last write access to IBI Status by DMAC.

- If the NQTHCTL.IBIQTH = 0b:

- When IBI Data Buffer is not completely empty.

- If the NQTHCTL.IBIQTH is other than 0b:

- When the number of IBI Data Buffer empties is less than the NQTHCTL.IBIQTH threshold (see NQTHCTL register).

CMDQEF bit (Normal Command Queue Empty Flag)

[Setting conditions]

- If the NQTHCTL.CMDQTH = 0b:

The following 1) and 2) are satisfied.

- 1) The NTSTE.CMDQEE bit = 1b. (Enables Command Buffer Empty Interrupt Status logging)
- 2) When Command Queue is completely empty

- If the NQTHCTL.CMDQTH is other than 0b:

The following 1) and 2) are satisfied.

- 1) The NTSTE.CMDQEE bit = 1b. (Enables Command Buffer Empty Interrupt Status logging)
- 2) When the number of Command Queue empties is the NQTHCTL.CMDQTH threshold or more (see register NQTHCTL).

[Clearing conditions]

- Write 0b to this bit after 1b is read.
- On completion of the last write access to Normal Command by DMAC.

If the NQTHCTL.CMDQTH = 0b:

- When Command Queue is not completely empty.

If the NQTHCTL.CMDQTH is other than 0b:

- When the number of Command Queue empties is less than the NQTHCTL.CMDQTH threshold (see NQTHCTL register).

RSPQFF bit (Normal Response Queue Full Flag)

[Setting conditions]

- The following 2 conditions are satisfied:
 - 1) The NTSTE.RSPQFE bit = 1b (enables Response Buffer Full Interrupt Status logging).
 - 2) When the number of Response Queue entries is more than the NQTHCTL.RSPQTH threshold (see NQTHCTL register).

[Clearing conditions]

- Write 0b to this bit after 1b is read.
- On completion of the last read access to Normal Receive Status by DMAC.
- When the number of Response Queue entries is the NQTHCTL.RSPQTH threshold or less (see NQTHCTL register).

TABTF bit (Normal Transfer Abort Flag)

[Setting conditions]

- The following 2 conditions are satisfied:
 - 1) The NTSTE.TABTE bit = 1b (enables Transfer Abort Interrupt Status logging).
 - 2) When any transfer is aborted.

[Clearing condition]

- Write 0b to this bit after 1b is read.

TEF bit (Normal Transfer Error Flag)

[Setting conditions]

- The following 2 conditions are satisfied:
 - 1) The NTSTE.TEE bit = 1b (enables Transfer Error Interrupt Status logging).
 - 2) When any transfer error occurs on the I3C bus. The Error type for this error is available in the Response or Receive Status structure corresponding to the Transfer command.

[Clearing condition]

- Write 0b to this bit after 1b is read.

RSQFF bit (Normal Receive Status Queue Full Flag)

[Setting conditions]

- The following 2 conditions are satisfied:
 - 1) The NTSTE.RSQFE bit = 1b (Normal Receive Status Queue Full Enable).
 - 2) When the number of Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold (see NRQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Receive Status by DMAC.
- When the number of Receive Status Queue entries is the NRQTHCTL.RSQTH threshold or less (see register NRQTHCTL).

7.8.3.2.46 Normal Transfer Status Enable Register (I3Cm_NTSTE)

Access Size : 32 bits

Address : <I3C_base> + 01E4h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	RSQFE	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TEE	-	-	-	TABTE	RSPQFE	CMDQEE	IBIQEFE	RDBFE0	TDBEE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	RSQFE	0h	RW	Normal Receive Status Queue Full Enable* ² 0b: Disables Receive Status Buffer Full Interrupt Status logging. 1b: Enables Receive Status Buffer Full Interrupt Status logging.
19 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	TEE	0h	RW	Normal Transfer Error Enable* ¹ 0b: Disables Transfer Error Interrupt Status logging. 1b: Enables Transfer Error Interrupt Status logging.
8 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	TABTE	0h	RW	Normal Transfer Abort Enable* ¹ 0b: Disables Transfer Abort Interrupt Status logging. 1b: Enables Transfer Abort Interrupt Status logging.
4	RSPQFE	0h	RW	Normal Response Queue Full Enable* ¹ 0b: Disables Response Buffer Full Interrupt Status logging. 1b: Enables Response Buffer Full Interrupt Status logging.
3	CMDQEE	0h	RW	Normal Command Queue Empty Enable* ¹ 0b: Disables Command Buffer Empty Interrupt Status logging. 1b: Enables Command Buffer Empty Interrupt Status logging.
2	IBIQEFE	0h	RW	Normal IBI Queue Empty/Full Enable* ¹ 0b: Disables IBI Status Buffer Empty/Full Interrupt Status logging. 1b: Enables IBI Status Buffer Empty/Full Interrupt Status logging.
1	RDBFE0	0h	RW	Normal Receive Data Buffer Full Enable 0b: Disables Receive Data Buffer Full Interrupt Status logging. 1b: Enables Receive Data Buffer Full Interrupt Status logging.
0	TDBEE0	0h	RW	Normal Transmit Data Buffer Empty Enable 0b: Disables Transmit Data Buffer Empty Interrupt Status logging. 1b: Enables Transmit Data Buffer Empty Interrupt Status logging.

Note 1. This bit supports all I3C mode.

Note 2. This bit supports I3C secondary master mode and I3C slave mode.

TDBEE0 bit (Normal Transmit Data Buffer Empty Enable)

When this bit is 1b, the operation of NTST.TDBEF0 is enabled.

For the setting conditions and clearing conditions of the NTST.TDBEF0 flag, see the details of NTST.TDBEF0.

RDBFE0 bit (Normal Receive Data Buffer Full Enable)

When this bit is 1b, the operation of NTST.RDBFF0 is enabled.

For the setting conditions and clearing conditions of the NTST.RDBFF0 flag, see the details of NTST.RDBFF0.

IBIQEFE bit (Normal IBI Queue Empty/Full Enable)

When this bit is 1b, the operation of NTST.IBIQEFF is enabled.

For the setting conditions and clearing conditions of the NTST.IBIQEFF flag, see the details of NTST.IBIQEFF.

CMDQEE bit (Normal Command Queue Empty Enable)

When this bit is 1b, the operation of NTST.CMDQEF is enabled.

For the setting conditions and clearing conditions of the NTST.CMDQEF flag, see the details of NTST.CMDQEF.

RSPQFE bit (Normal Response Queue Full Enable)

When this bit is 1b, the operation of NTST.RSPQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSPQFF flag, see the details of NTST.RSPQFF.

TABTE bit (Normal Transfer Abort Enable)

When this bit is 1b, the operation of NTST.TABTF is enabled.

For the setting conditions and clearing conditions of the NTST.TABTF flag, see the details of NTST.TABTF.

TEE bit (Normal Transfer Error Enable)

When this bit is 1b, the operation of NTST.TEF is enabled.

For the setting conditions and clearing conditions of the NTST.TEF flag, see the details of NTST.TEF.

RSQFE bit (Normal Receive Status Queue Full Enable)

When this bit is 1b, the operation of NTST.RSQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSQFF flag, see the details of NTST.RSQFF.

7.8.3.2.47 Normal Transfer Interrupt Enable Register (I3Cm_NTIE)

The PIO Interrupt Signal Enable register enables signaling of outstanding interrupts received by I3C.

Access Size :		32 bits														
Address :		<I3C_base> + 01E8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	RSQFIE	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TEIE	-	-	-	TABTIE	RSPQFIE	CMDQEIE	IBIQFIE	RDBFIE0	TDBEIE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	RSQFIE	0h	RW	Normal Receive Status Queue Full Interrupt Enable* ² 0b: Disables Receive Status Buffer Full Interrupt Signal. 1b: Enables Receive Status Buffer Full Interrupt Signal.
19 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	TEIE	0h	RW	Normal Transfer Error Interrupt Enable* ¹ 0b: Disables Transfer Error Interrupt Signal. 1b: Enables Transfer Error Interrupt Signal.
8 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	TABTIE	0h	RW	Normal Transfer Abort Interrupt Enable* ¹ 0b: Disables Transfer Abort Interrupt Signal. 1b: Enables Transfer Abort Interrupt Signal.
4	RSPQFIE	0h	RW	Normal Response Queue Full Interrupt Enable* ¹ 0b: Disables Response Buffer Full Interrupt Signal. 1b: Enables Response Buffer Full Interrupt Signal.
3	CMDQEIE	0h	RW	Normal Command Queue Empty Interrupt Enable* ¹ 0b: Disables Command Buffer Empty Interrupt Signal. 1b: Enables Command Buffer Empty Interrupt Signal.
2	IBIQFIE	0h	RW	Normal IBI Queue Empty/Full Interrupt Enable* ¹ 0b: Disables IBI Status Buffer Empty/Full Interrupt Signal. 1b: Enables IBI Status Buffer Empty/Full Interrupt Signal.
1	RDBFIE0	0h	RW	Normal Receive Data Buffer Full Interrupt Enable 0b: Disables Receive Data Buffer Full Interrupt Signal. 1b: Enables Receive Data Buffer Full Interrupt Signal.
0	TDBEIE0	0h	RW	Normal Transmit Data Buffer Empty Interrupt Enable 0b: Disables Transmit Data Buffer Empty Interrupt Signal. 1b: Enables Transmit Data Buffer Empty Interrupt Signal.

Note 1. This bit supports all I3C mode.

Note 2. This bit supports I3C secondary master mode and I3C slave mode.

TDBEIE0 bit (Normal Transmit Data Buffer Empty Interrupt Enable)

This bit is used to enable or disable the Normal Transmit Data buffer empty interrupt (INT_ri3c_tx_n) requests when the NTST.TDBEF0 flag is set to 1b.

RDBFIE0 bit (Normal Receive Data Buffer Full Interrupt Enable)

This bit is used to enable or disable the Normal Receive Data buffer full interrupt (INT_ri3c_rx_n) requests when the NTST.RDBFF0 flag is set to 1b.

IBIQFIE bit (Normal IBI Queue Empty/Full Interrupt Enable)

This bit is used to enable or disable the Normal IBI Status buffer full interrupt (INTIBI) requests when the NTST.IBIQE0 flag is set to 1b.

CMDQEIE bit (Normal Command Queue Empty Interrupt Enable)

This bit is used to enable or disable the Normal Command buffer empty interrupt (INTCMD) requests when the NTST.CMDQE0 flag is set to 1b.

RSPQFIE bit (Normal Response Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Response Status buffer full interrupt (INTRESP) requests when the NTST.RSPQFF flag is set to 1b.

TABTIE bit (Normal Transfer Abort Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Abort interrupt (INTABORT) requests when the NTST.TABTF flag is set to 1b.

TEIE bit (Normal Transfer Error Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Error interrupt (INTTERR) requests when the NTST.TEF flag is set to 1b.

RSQFIE bit (Normal Receive Status Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Receive Status buffer full interrupt (INTRCV) requests when the NTST.RSQFF flag is set to 1b.

7.8.3.2.48 Normal Transfer Status Force Register (I3Cm_NTSTFC)

The PIO Interrupt Force register is used to force specific interrupt. It can be used for debug purposes.

Access Size :		32 bits														
Address :		<I3C_base> + 01ECh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	RSQFFC	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TEFC	-	-	-	TABTFC	RSPQFFC	CMDQEFC	IBIQEFC	RDBFFC0	TDBEFC0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	W	Reserved The written value should always be 0b.
20	RSQFFC	0h	W	Normal Receive Status Queue Full Force* ² 0b: Not Force Receive Status Buffer Full Interrupt for software testing. 1b: Force Receive Status Buffer Full Interrupt for software testing.
19 to 10	-	All 0	W	Reserved The written value should always be 0b.
9	TEFC	0h	W	Normal Transfer Error Force* ¹ 0b: Not Force Transfer Error Interrupt for software testing. 1b: Force Transfer Error Interrupt for software testing.
8 to 6	-	All 0	W	Reserved The written value should always be 0b.
5	TABTFC	0h	W	Normal Transfer Abort Force* ¹ 0b: Not Force Transfer Abort Interrupt for software testing. 1b: Force Transfer Abort Interrupt for software testing.
4	RSPQFFC	0h	W	Normal Response Queue Full Force* ¹ 0b: Not Force Response Buffer Full Interrupt for software testing. 1b: Force Response Buffer Full Interrupt for software testing.
3	CMDQEFC	0h	W	Normal Command Queue Empty Force* ¹ 0b: Not Force Command Buffer Empty Interrupt for software testing. 1b: Force Command Buffer Empty Interrupt for software testing.
2	IBIQEFC	0h	W	Normal IBI Queue Empty/Full Force* ¹ 0b: Not Force IBI Status Buffer Full Interrupt for software testing. 1b: Force IBI Status Buffer Full Interrupt for software testing.
1	RDBFFC0	0h	W	Normal Receive Data Buffer Full Force 0b: Not Force Receive Data Buffer Full Interrupt for software testing. 1b: Force Receive Data Buffer Full Interrupt for software testing.
0	TDBEFC0	0h	W	Normal Transmit Data Buffer Empty Force 0b: Not Force Transmit Data Buffer Empty Interrupt for software testing. 1b: Force Transmit Data Buffer Empty Interrupt for software testing.

Note 1. This bit supports all I3C mode.

Note 2. This bit supports I3C secondary master mode and I3C slave mode.

TDBEFC0 bit (Normal Transmit Data Buffer Empty Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to TDBEE0 and TDBEIE0 configuration.

RDBFFC0 bit (Normal Receive Data Buffer Full Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to RDBFE0 and RDBFIE0 configuration.

IBIQEFC bit (Normal IBI Queue Empty/Full Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to IBIQEFE and IBIQEFIE configuration.

CMDQEFC bit (Normal Command Queue Empty Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to CMDQEE and CMDQEIE configuration.

RSPQFFC bit (Normal Response Queue Full Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to RSPQFE and RSPQFIE configuration.

TABTFC bit (Normal Transfer Abort Force)

For software testing, forces the corresponding interrupt, subject to TABTE and TABTIE configuration.

TEFC bit (Normal Transfer Error Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to TEE and TEIE configuration.

RSQFFC bit (Normal Receive Status Queue Full Force)

For software testing, when set to 1b, forces the corresponding interrupt, subject to RSQFE and RSQFIE configuration.

7.8.3.2.49 Bus Condition Status Register (I3Cm_BCST)

Access Size :		32 bits														
Address :		<I3C_base> + 0210h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	BIDLF	BAVLF	BFREF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read.
2	BIDLF	0h	R	Bus Idle Detection Flag ^{*1} 0b: Have not Detected Bus Idle 1b: Have Detected Bus Idle
1	BAVLF	0h	R	Bus Available Detection Flag ^{*1} 0b: Have not Detected Bus Available 1b: Have Detected Bus Available
0	BFREF	0h	R	Bus Free Detection Flag 0b: Have not Detected Bus Free 1b: Have Detected Bus Free

Note 1. This bit supports all I3C mode.

BFREF bit (Bus Free Detection Flag)

The Bus Free Condition is a period occurring after a STOP and before a START, and with the following duration:

- For Pure Bus: A duration of at least tCAS
- For Mixed Bus (at least one Legacy I²C is present on the I3C Bus): A duration of at least tBUF

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3C ϕ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3C ϕ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0b.

BAVLF bit (Bus Available Detection Flag)

The Bus Available Condition is a period during which the Bus Free Condition is sustained continuously for a duration of at least tAVAL. A Slave can only issue a START Request (for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3C ϕ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3C ϕ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0b.

BIDLF bit (Bus Idle Detection Flag)

The I3C Bus Idle Condition is in order to help ensure Bus stability during Hot-Join events. The Bus Idle Condition is a period during which the Bus Available Condition is sustained continuously for a duration of at least tIDLE.

If a Hot-Join Device is powered up onto the I3C Bus at the same time as the Main Master, then the Hot-Join Device may pull SDA Low after 1 ms if (1) the Main Master has SCL and SDA pulled up, and (2) the Master does not act on the I3C Bus within the same Idle period.

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3C ϕ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3C ϕ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0b.

7.8.3.2.50 Slave Status Register (I3Cm_SVST)

Access Size : 32 bits

Address : <I3C_base> + 0214h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	SVAF2	SVAF1	SVAF0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HOAF	-	-	-	-	-	-	-	-	DVIDF	HSMCF	-	-	-	-	GCAF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18	SVAF2	0h	RW ^{*1}	Slave Address Detection Flag 2 0b: Slave 2 not detected 1b: Slave 2 detected
17	SVAF1	0h	RW ^{*1}	Slave Address Detection Flag 1 0b: Slave 1 not detected 1b: Slave 1 detected
16	SVAF0	0h	RW ^{*1}	Slave Address Detection Flag 0 0b: Slave 0 not detected 1b: Slave 0 detected
15	HOAF	0h	RW ^{*1}	Host Address Detection Flag 0b: Host address not detected 1b: Host address detected This bit set to 1b when the received slave address matches the host address (0001_000b).
14 to 7	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
6	DVIDF	0h	RW ^{*1}	Device-ID Address Detection Flag 0b: Device-ID command not detected 1b: Device-ID command detected This bit set to 1b when the first frame received immediately after a START condition is detected matches a value of (device ID (1111_100) + 0[W]).
5	HSMCF	0h	RW ^{*1}	Hs-mode Master Code Detection Flag 0b: Hs-mode Master Code not detected 1b: Hs-mode Master Code detected
4 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	GCAF	0h	RW ^{*1}	General Call Address Detection Flag 0b: General call address not detected 1b: General call address detected

Note: This register supports I²C mode.

Note 1. Clearing (to 0) condition: Writing 0b after 1b is read.

GCAF bit (General Call Address Detection Flag)I²C Normal Wake-Up Mode1/2 sets GCAF to 1b when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.

- 1) The SVCTL.GCAE bit = 1b (General call address detection is enabled).
- 2) When the received slave address matches the general call address (0000 000 + 0 (write)).

[Clearing conditions]

- When 0b is written to the GCAF flag after reading GCAF flag to be 1b.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

HSMCF bit (Hs-mode Master Code Detection Flag)

I²C Normal Wake-Up Mode1/2 sets HSMCF to 1b when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
 - 1) The SVCTL.HSMCE bit = 1b (Hs-mode master code detection is enabled).
 - 2) When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXX) + 1 (NACK).

[Clearing conditions]

- When 0b is written to the HSMCF flag after reading HSMCF flag to be 1b.
- When a STOP condition is detected.

DVIDF bit (Device-ID Address Detection Flag)

[Setting conditions]

- This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
 - 1) The SVCTL.DVIDE bit = 1b (Device-ID address detection is enabled).
 - 2) When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of Device ID (1111 100) + 0 (write).

[Clearing conditions]

- When 0b is written to the DVIDF flag after reading DVIDF flag to be 1b.
- When a STOP condition is detected.
- This flag is set to 0b at the rising edge of the ninth SCL clock cycle in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
 - 1) The SVCTL.DVIDE bit = 1b (Device-ID address detection is enabled).
 - 2) When the first byte received immediately after a START condition or Repeated START condition is detected does not match a value of Device ID (1111 100).
 - 3) When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of (device ID (1111 100) + 0 [W]) and the second byte does not match any of slave addresses 0 to 2.

HOAF bit (Host Address Detection Flag)

I²C Normal Wake-Up Mode1/2 sets HOAF to 1 at the time of switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
 - 1) The SVCTL.HOAE bit = 1b (Host address detection is enabled).
 - 2) When the received slave address matches the host address (0001 000).

[Clearing conditions]

- When 0b is written to the HOAF flag after reading HOAF flag to be 1b.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

SVAFn bits (Slave Address Detection Flag n) (n = 0 to 2)

I²C Normal Wake-Up Mode1/2 sets SVAF2/1/0 to 1 when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- For 7-bit address format: SVDVADn.SADLG bit = 0b.

This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.

- 1) The SVCTL.SVAEn bit = 1b (Slave n enabled).
- 2) When the received slave address matches the SVDVADn.SVAD[6:0] bits value.

- For 10-bit address format: SVDVADn.SADLG bit = 1b.

This flag is set to 1b at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.

- 1) The SVCTL.SVAEn bit = 1b (Slave n enabled).
- 2) When the received slave address matches a value of 11110 + SVDVADn.SVAD[9:8] bits and the following address matches the SVDVADn.SVAD[7:0] value.

[Clearing conditions]

- When 0b is written to the SVAFn flag after reading SVAFn flag to be 1b.
- When a STOP condition is detected.

For 7-bit address format: SVDVADn.SADLG bit = 0b.

This flag is set to 0b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.

- 1) The SVCTL.SVAEn bit = 1b (Slave n enabled).
- 2) When the received slave address does not match SVDVADn.SVAD[6:0] bits value.

For 10-bit address format: SVDVADn.SADLG bit = 1b.

This flag is set to 0b at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.

- 1) The SVCTL.SVAEn bit = 1b (Slave n enabled).

- 2) When the received slave address does not match a value of $11110 + \text{SVDVADn.SVAD}[9:8]$ bits.
- This flag is set to 0b at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
 - 1) The SVCTL.SVAEn bit = 1b (Slave n enabled).
 - 2) When the received slave address matches a value of $11110 + \text{SVDVADn.SVAD}[9:8]$ bits and the following address does not match the SVDVADn.SVAD[7:0] value.

7.8.3.2.51 Wake-Up Unit Operating Status Register (I3Cm_WUST)

Access Size :		32 bits														
Address :		<I3C_base> + 0218h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WUAS YNF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read.
0	WUASYNF	0h	R	Wake-Up Function Asynchronous Operation Status Flag 0b: I3C synchronous circuit enable condition 1b: I3C asynchronous circuit enable condition

WUASYNF bit (Wake-Up Function Asynchronous Operation Status Flag)

This bit shows whether I3C is in the I3C_0_TCLK asynchronous operation (WUCTL.WUFE bit = 1b).

[Setting conditions]

- The following 1) and 2) are satisfied.
 - The WUCTL.WUFE bit = 1b (Wake-up function is enabled)
 - When the BCST.BFREF flag = 1b after 0b is written to the WUCTL.WUFSYNE bit

[Clearing condition: I²C slave]

- The WUCTL.WUFE bit = 0b (Wake-up function is disabled)
- All of the following 1) to 3) are satisfied.
 - The WUCTL.WUFE bit = 1b (Wake-up function is enabled)
 - Wake-up event is detected
 - When 1b is written to the WUCTL.WUFSYNE bit while the WUASYNF flag = 1b

[Clearing condition: I3C slave]

- The WUCTL.WUFE bit = 0b (Wake-up function is disabled)
- All of the following 1) to 4) are satisfied.
 - The WUCTL.WUFE bit = 1b (Wake-up function is enabled)
 - Wake-up event is detected
 - When 1b is written to the WUCTL.WUFSYNE bit while the WUASYNF flag = 1b
 - When a STOP condition is detected.

[Clearing condition: I²C/I3C slave]

- All of the following 1) to 5) are satisfied.

- 1) The WUCTL.WUFE bit = 1b (Wake-up function is enabled)
- 2) Wake-up event is not detected
- 3) The WUASYNF flag = 1b
- 4) The WUCTL.WUFSYNE bit = 1b
- 5) When a STOP condition is detected.

[Clearing condition: I3C master]

- The WUCTL.WUFE bit = 0b (Wake-up function is disabled)
- All of the following 1) to 4) are satisfied.
 - 1) The WUCTL.WUFE bit = 1b (Wake-up function is enabled)
 - 2) Wake-up event is detected
 - 3) The WUASYNF flag = 1b
 - 4) The WUCTL.WUFSYNE bit = 1b

7.8.3.2.52 MsyncCNT Counter Capture Register (I3Cm_MRCCPT)

Access Size :		32 bits														
Address :		<I3C_base> + 021Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRCCPT[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRCCPT[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MRCCPT[31:0]	0h	R	MSyncCNT Counter Capture Used in Async Mode 1, not used in Async Mode0.

Note: This register supports I3C master mode and I3C secondary master mode.

MRCCPT[31:0] bits (MSyncCNT Counter Capture)

- Async Mode 1 (Asynchronous Advanced Mode)

When ATCCNTE.ATCE is enabled, the counter starts counting. Its value is captured as MSyncCNT for each aME (SDA falling edge of START condition), and stored in the capture register.

7.8.3.2.53 Device Address Table Basic Register n (I3Cm_DATBASn) (n = 0 to 7)

Access Size :		32 bits															
Address :		<I3C_base> + 0224h + n x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	DVTYP	DVNACK[1:0]		-	-	-	-	-	DVDYAD[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	DVIBITS	DVMRRJ	DVSIRRJ	DVIBIPL	-	-	-	-	-	DVSTAD[6:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31	DVTYP	0h	RW	Device Type 0b: I3C Device 1b: I2C Device
30 to 29	DVNACK[1:0]	0h	RW	Device NACK Retry Count Device-specific retry count
28 to 24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
23 to 16	DVDYAD[7:0]	0h	RW	Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.
15	DVIBITS	0h	RW	Device IBI Time-stamp 0b: The Master shall not time-stamp IBIs from this Device with Master Time-stamps. 1b: The Master shall time-stamp IBIs for this Device with Master Time-stamps.
14	DVMRRJ	0h	RW	Device In-Band Master Request Reject 0b: This Device shall ACK Master Requests. 1b: This Device shall NACK Master Requests and send the auto-disable command.
13	DVSIRRJ	0h	RW	Device In-Band Slave Interrupt Request Reject 0b: This Device shall ACK the SIR. 1b: This Device shall NACK the SIR and send the auto-disable CCC.
12	DVIBIPL	0h	RW	Device IBI Payload 0b: IBIs from this Device do not carry a Data Payload. 1b: IBIs from this Device do carry a Data Payload.
11 to 7	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
6 to 0	DVSTAD[6:0]	0h	RW	Device Static Address I3C Static Address

Note: This register supports I3C master mode and I3C secondary master mode.

DVIBIPL bit (Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

DVSIRRJ bit (Device In-Band Slave Interrupt Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Slave Interrupt Requests from other Devices.

DVMRRJ bit (Device In-Band Master Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Master requests from other Devices. This bit is only valid if I3C declares Non-Current Master Capability.

DVIBITS bit (Device IBI Time-stamp)

Enables or disables IBI time-stamping for a specific Device.

Note: The IBI Status Descriptor for each IBI event indicates whether or not the individual IBI event was actually time- stamped. Set to 0 except for Async mode 0 and Async mode 1 of timing control.

DVNACK[1:0] bits (Device NACK Retry Count)

These bits set the number of retries when a NACK response is received from the slave for the transaction set in the Command Descriptor.

Note: When ENTDAAs is executed by Address Assign Command, the setting of this bit is ignored and the transaction ends when NACK is received once.

Note: I3C will retry according to the setting of DVNACK[1:0], even if it receives a NACK for the broadcast address.

Note: If DVNACK[1:0] is 00b, I3C will not retry even for Direct CCCs.

7.8.3.2.54 Slave Device Address Table Basic Register n (I3Cm_SDATBASn) (n = 0 to 2)

Access Size :		32 bits														
Address :		<I3C_base> + 02B0h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	SDDYAD[6:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SDIBI L	-	SDADL S	SDSTAD[9:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
22 to 16	SDDYAD* ¹ [6:0]	0h	RW	Slave Device I3C Dynamic Address* ⁵
15 to 13	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
12	SDIBIPL* ¹	0h	RW	Slave Device IBI Payload* ⁴ This bit is the mirror bit of the SVDCT.TBCR[2]. 0b: IBIs from this device do not carry a data payload. 1b: IBIs from this device carry a data payload.
11	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10	SDADLS	0h	RW	Slave Device Address Length Selection* ³ 0b: Slave device address length 7 bits selected. 1b: Slave device address length 10 bits selected. (I2C device only)
9 to 0	SDSTAD[9:0]	0h	RW	Slave Device Static Address* ² I3C / I2C Static Address

Note: SW write to the SDATBAS register of the main master is prohibited.

Note 1. This bit is valid only in SDATBAS0 register.

Note 2. These bits support I²C, I3C secondary master, and I3C slave mode.

Note 3. This bit supports I²C mode.

Note 4. This bit supports I3C secondary master mode and I3C slave mode.

Note 5. These bits support I3C secondary master mode and I3C slave mode.

SDSTAD[9:0] bits (Slave Device Static Address)

When the 7-bit address format is selected (SDADLS bit is 0b), the lower 7 bits of SDSTAD[9:0] function as the 7-bit address.

When the 10-bit address format is selected (SDADLS bit is 1b), the SDSTAD[9:0] function as the 10-bit address. While the SVCTL.SVAEn bit is 0b, the setting of this bit is ignored.

SDIBIPL bit (Slave Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

SDDYAD[6:0] bits (Slave Device I3C Dynamic Address)

[Update conditions]

- When writing Dynamic Address value.
- When Slave Address value is its own Static Address in receiving SETDASA CCC (Direct), these bits are updated to Dynamic Address value.*¹
- When Dynamic Address Assignment procedure that starts by receiving ENTDAACCC (Broadcast) is established.*¹
- When receiving RSTDAA CCC (Broadcast), all bits are cleared to 0.*¹
- When Slave Address value is its own Dynamic Address in receiving RSTDAA CCC (Direct), all bits are cleared to 0.*¹
- When Slave Address value is its own Dynamic Address in receiving SETNEWDA CCC (Direct), these bits are updated to the Dynamic Address value.*¹
- When receiving SETAASA CCC (Broadcast), these bits are updated to the value of SDSTAD[6:0] bits*².

Note 1. See the MIPI I3C Specification v1.0.

Note 2. See the MIPI I3C Basic Specification v1.0.

7.8.3.2.55 Master Device Characteristic Table Register n (I3Cm_MSDCTn) (n = 0 to 7)

Access Size :		32 bits														
Address :		<I3C_base> + 02D0h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBCR76[1:0]		-	-	RBCR3	RBCR2	RBCR1	RBCR0	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15, 14	RBCR76[1:0]	0h	RW	Device Role 00b: I3C Slave 01b: I3C Master* ³ Others: Setting prohibited
13, 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11	RBCR3	0h	RW	Offline Capable* ² 0b: Device will always respond to I3C bus commands. 1b: Device will not always respond to I3C bus commands.
10	RBCR2	0h	RW	IBI Payload 0b: No data byte follows the accepted IBI. 1b: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.
9	RBCR1	0h	RW	IBI Request Capable 0b: Not Capable 1b: Capable
8	RBCR0	0h	RW	Max Data Speed Limitation* ¹ 0b: No Limitation 1b: Limitation
7 to 0	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Note: This register supports I3C master mode and I3C secondary master mode.

Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.

Note 2. Offline Capable Devices retain the Dynamic Address.

Note 3. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01.

RBCRn bits (Received Bus Characteristic Register)

Each I3C Device that is connected to the I3C Bus shall have an associated read-only Bus Characteristics Register (BCR). This read-only register describes the I3C compliant Device's role and capabilities for use in Dynamic Address assignment and Common Command Codes.

Note: When RBCR[2] is 0b and when ACK response to Slave Interrupt Request from I3C Slave while DATBASm.DVSIRRJ is set to 0b, STOP Condition is issued after ACK response. When RBCR[2] is 1b and when ACK response to Slave Interrupt Request from I3C Slave while DATBASm.DVSIRRJ is set to 0b, IBI Payload is received after ACK response. STOP Condition is issued after end of IBI Payload.

[Update condition]

When receiving of Bus Characteristics Register (BCR) from Device in the Dynamic Address Assignment procedure starting by receiving ENTDAACCC (Broadcast).^{*1}

Note 1. See the MIPI I3C Specification v1.0

7.8.3.2.56 Extended Device Address Table Basic Register (I3Cm_EXDATBAS)

Access Size :		32 bits															
Address :		<I3C_base> + 0310h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	EDTYP	EDNACK[1:0]		-	-	-	-	-	EDDYAD[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	EDSTAD[6:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31	EDTYP	0h	RW	Extended Device Type 0b: I3C Device 1b: I2C Device
30, 29	EDNACK[1:0]	0h	RW	Extended Device NACK Retry Count Device-specific retry count
28 to 24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
23 to 16	EDDYAD[7:0]	0h	RW	Extended Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.
15 to 7	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
6 to 0	EDSTAD[6:0]	0h	RW	Extended Device Static Address I3C / I2C static address

Note: This register supports I3C master mode and I3C secondary master mode.

7.8.3.2.57 Slave Device Characteristic Table Register (I3Cm_SVDCT)

Access Size :		32 bits															
Address :		<I3C_base> + 0320h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TBCR7[1:0]		-	-	TBCR3	TBCR2	TBCR1	TBCR0	TDCR[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 14	TBCR7[1:0]	0h	RW	Device Role 0 0b: I3C Slave 0 1b: I3C Master* ³ 1 0b: Reserved for future definition by MIPI Sensor WG 1 1b: Reserved for future definition by MIPI Sensor WG
13, 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11	TBCR3	0h	RW	Offline Capable* ² 0b: Device will always respond to I3C bus commands. 1b: Device will not always respond to I3C bus commands.
10	TBCR2	0h	RW	IBI Payload 0b: No data byte follows the accepted IBI. 1b: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.
9	TBCR1	0h	RW	IBI Request Capable 0b: Not Capable 1b: Capable
8	TBCR0	0h	RW	Max Data Speed Limitation* ¹ 0b: No Limitation 1b: Limitation
7 to 0	TDCR[7:0]	0h	RW	Transfer Device Characteristic Register 255 available codes for describing the type of sensor, or device. Examples: Accelerometer, gyroscope, composite devices Default value is 0b: Generic Device

Note: This register supports I3C secondary master mode and I3C slave mode.

Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.

Note 2. Offline Capable Devices retain the Dynamic Address.

Note 3. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01.

TDCR[7:0] bits (Transfer Device Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Device Characteristics Register (DCR). This register describes the I3C compliant device type such as accelerometer and gyroscope, for use in Dynamic Address assignment and Common Command Codes.

TBCRn bits (Transfer Bus Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Bus Characteristics Register (BCR). This register describes the role and capabilities of the I3C compliant device for use in Dynamic Address assignment and Common Command Codes.

When I3C Slave issues IBI by Command Descriptor, the setting values of TBCR[7:0] are as follows:

[Slave Interrupt Request : No IBI Payload follow the accepted IBI]

- TBCR1 = 1b
- TBCR2 = 0b

Note: Set DATA_LENGTH[15:0] of Command Descriptor to 0.

[Slave Interrupt Request : IBI Payload follow the accepted IBI]

- TBCR1 = 1b
- TBCR2 = 1b

Note: Set DATA_LENGTH[15:0] of Command Descriptor to any value.

[Mastership Request]

- TBCR1 = 1b
- TBCR76[1:0] = 01b

[Hot-join Event]

- TBCR1 = 1b

When I3C Slave receives CCC from I3C Master, it performs the following operations according to the setting of TBCR[7:0]:

- When TBCR2 = 1b, CMRLG.IBIPSZ[7:0] is sent as the 3rd byte data to GETMRL CCC from I3C Master
- When TBCR0 = 0b, NACK responses to GETMXDS CCC from I3C Master
- When TBCR0 = 1, ACK responses to GETMXDS CCC from I3C Master and sends data from CMDSPW, CMDSPR, and CMDSP registers

7.8.3.2.58 Slave Device Characteristic Table Provisional ID Low Register (I3Cm_SDCTPIDL)

Access Size :		32 bits														
Address :		<I3C_base> + 0324h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDCTPIDL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCTPIDL[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 0	SDCTPIDL [31:0]															
	Initial Value															
	R/W															
	Description															
Transfer Device Provisional ID Low																
Bits 31 to 16 are read as 0b.																
Bits 15 to 0 are bits [15:0] of device's I3C PID.																

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.59 Slave Device Characteristic Table Provisional ID High Register (I3Cm_SDCTPIDH)

Access Size :		32 bits														
Address :		<I3C_base> + 0328h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDCTPIDH[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDCTPIDH[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SDCTPIDH [31:0]	0h	RW	Transfer Device Provisional ID High Bits [47:16] of device's I3C PID.

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.60 Slave Device Address Register n (I3Cm_SVDVADn) (n = 0 to 2)

Access Size :		32 bits																		
Address :		<I3C_base> + 0330h + n x 0004h																		
Initial Value :		0000_0000h																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
	SDYADV	SSTADV	-	-	SADLG	-	SVAD[9:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R				

Bit	Bit Name	Initial Value	R/W	Description
31	SDYADV* ⁴	0h	R	Slave Dynamic Address Valid* ³ 0b: Dynamic Address is disabled. 1b: Dynamic Address is enabled.
30	SSTADV	0h	R	Slave Static Address Valid* ¹ 0b: Slave address is disabled. 1b: Slave address is enabled.
29,28	-	All 0	R	Reserved Whenever it is read, 0b is read.
27	SADLG	0h	R	Slave Address Length* ² 0b: The 7-bit address format is selected. 1b: The 10-bit address format is selected.
26	-	0h	R	Reserved Whenever it is read, 0b is read.
25 to 16	SVAD[9:0]	0h	R	Slave Address* ¹ A slave address is set. When rewriting SVAD, change to SVAE = 0b and rewrite.
15 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read.

Note 1. These bits support I²C, I3C secondary master, and I3C slave mode.

Note 2. This bit supports I²C mode.

Note 3. This bit supports I3C secondary master mode and I3C slave mode.

Note 4. This bit is valid only in SVDVAD0 register.

SVAD[9:0] bits (Slave Address)

The SVAD[9:0] bits indicate a valid slave address.

[The SVDVAD0.SDYADV bit = 1b]

Note: This condition is only for SVDVAD0.SVAD[9:0].

- The SVAD[9:7] bits = 0b
- The SVAD[6:0] bits = the SDATBAS0.SDDYAD[6:0] bits

[The SVDVADn.SSTADV bit = 1b and the SVDVADn.SADLG bit = 0b]

- The SVAD[9:7] bits = 0b
- The SVAD[6:0] bits = the SDATBASn.SDSTAD[6:0] bits

[The SVDVADn.SSTADV bit = 1b and the SVDVADn.SADLG bit = 1b]

- The SVAD[9:0] bits = the SDATBASn.SDSTAD[9:0] bits

SADLG bit (Slave Address Length)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The PRTS.PRTMD bit = 1b (I²C Protocol mode)
 - 2) The SVCTL.SVAEn bit = 1b (Slave n is enabled)
 - 3) The SDATBASn.SDADLS bit = 1b (The address length is 10 bits)

[Clearing condition]

- [Setting condition] is not satisfied.

SSTADV bit (Slave Static Address Valid)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The SVCTL.SVAEn bit = 1b (Slave n is enabled)
 - 2) The SVDVAD0.SDYADV bit = 0b (Dynamic Address is disabled)

Note: This condition is only for SVDVAD0.SSTADV.

 - 3) If the SVDVADn.SADLG bit = 0b, the SDATBASn.SDSTAD[6:0] bits are not all 0
 If the SVDVADn.SADLG bit = 1b, the SDATBASn.SDSTAD[9:0] bits are not all 0

[Clearing condition]

- “[Setting condition]” is not satisfied.

SDYADV bit (Slave Dynamic Address Valid)

[Setting conditions]

- All of the followings are satisfied:
 - 1) The PRTS.PRTMD bit = 0b (I3C Protocol mode)
 - 2) The SVCTL.SVAEn bit = 1b (Slave n is enabled)
 - 3) The SDATBAS0.SDDYAD[6:0] bits are not all 0

Note: This condition is only for SVDVAD0.SDYADV.

[Clearing condition]

- “[Setting condition]” is not satisfied.

7.8.3.2.61 CCC Slave Events Command Register (I3Cm_CSECMD)

Access Size :		32 bits														
Address :		<I3C_base> + 0350h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	HJEVE	-	MSRQ E	SVIRQ E
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	HJEVE	0h	RW	Hot-Join Event Enable 0b: DISABLED: Slave-initiated Hot-Join is Disabled by the Master to control. 1b: ENABLED: Slave-initiated Hot-Join is Enabled by the Master to control.
2	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	MSRQE	0h	RW	Mastership Requests Enable 0b: DISABLED: Mastership requests from Secondary Masters is Disabled by the Current Master to control. 1b: ENABLED: Mastership requests from Secondary Masters is Enabled by the Current Master to control.
0	SVIRQE	0h	RW	Slave Interrupt Requests Enable 0b: DISABLED: Slave-initiated Interrupts is Disabled by the Master to control. 1b: ENABLED: Slave-initiated Interrupts is Enabled by the Master to control.

Note: This register supports I3C secondary master mode and I3C slave mode.

SVIRQE bit (Slave Interrupt Requests Enable)

This bit allows the Master to control when Slave-initiated Interrupts are allowed on the I3C Bus.

These four Direct (ENEC/DISEC Format 1) or Broadcast (ENEC/DISEC Format 2) CCCs allows the Master to control when Slave- initiated traffic is (Enable) vs. is not (Disable) allowed on the I3C Bus. This control governs a Slave's attempts to request an Interrupt (ENI), to request Mastership (ENMR), or to signify a Hot-Join event (ENHJ).

[Setting conditions]

- When writing 1b
- When receiving ENEC CCC (Broadcast) with ENINT bit = 1b.*¹
- When ENINT bit = 1b with own Slave Address in receiving ENEC CCC (Direct).*¹

[Clearing conditions]

- When writing 0b.
- When receiving DISEC CCC (Broadcast) with DISINT bit = 1b.*¹
- When DISINT bit = 1b with own Slave Address in receiving DISEC CCC (Direct).*¹

MSRQE bit (Mastership Requests Enable)

This bit allows the Current Master to control when Mastership requests from Secondary Masters are allowed on the I3C Bus.

[Setting conditions]

- When writing 1b.
- When receiving ENEC CCC (Broadcast) with ENMR bit = 1b.*¹
- When ENMR bit = 1b with own Slave Address in receiving ENEC CCC (Direct).*¹

[Clearing conditions]

- When writing 0b.
- When receiving DISEC CCC (Broadcast) with DISMR bit = 1b.*¹
- When DISMR bit = 1b with own Slave Address in receiving DISEC CCC (Direct).*¹

HJEVE bit (Hot-Join Event Enable)

This bit allows the Master to control when Slave-initiated Hot-Join is allowed on the I3C Bus. [Setting conditions]

- When writing 1b.
- When receiving ENEC CCC (Broadcast) with ENHJ bit = 1b.*¹
- When ENHJ bit = 1b with own Slave Address in receiving ENEC CCC (Direct).*¹

[Clearing conditions]

- When writing 0b.
- When receiving DISEC CCC (Broadcast) with DISHJ bit = 1b.*¹
- When DISHJ bit = 1b with own Slave Address in receiving DISEC CCC (Direct).*¹

Note 1. See the MIPI I3C Specification v1.0

7.8.3.2.62 CCC Enter Activity State Register (I3Cm_CEACTIONST)

Access Size :		32 bits															
Address :		<I3C_base> + 0354h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	ACTST[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3 to 0	ACTST[3:0]	0h	RW	Activity State 1h: ENTAS0 (1 μs: Latency-free operation) 2h: ENTAS1 (100 μs) 4h: ENTAS2 (2 ms) 8h: ENTAS3 (50 ms: Lowest-activity operation) Others: Setting prohibited

Note: This register supports I3C secondary master mode and I3C slave mode.

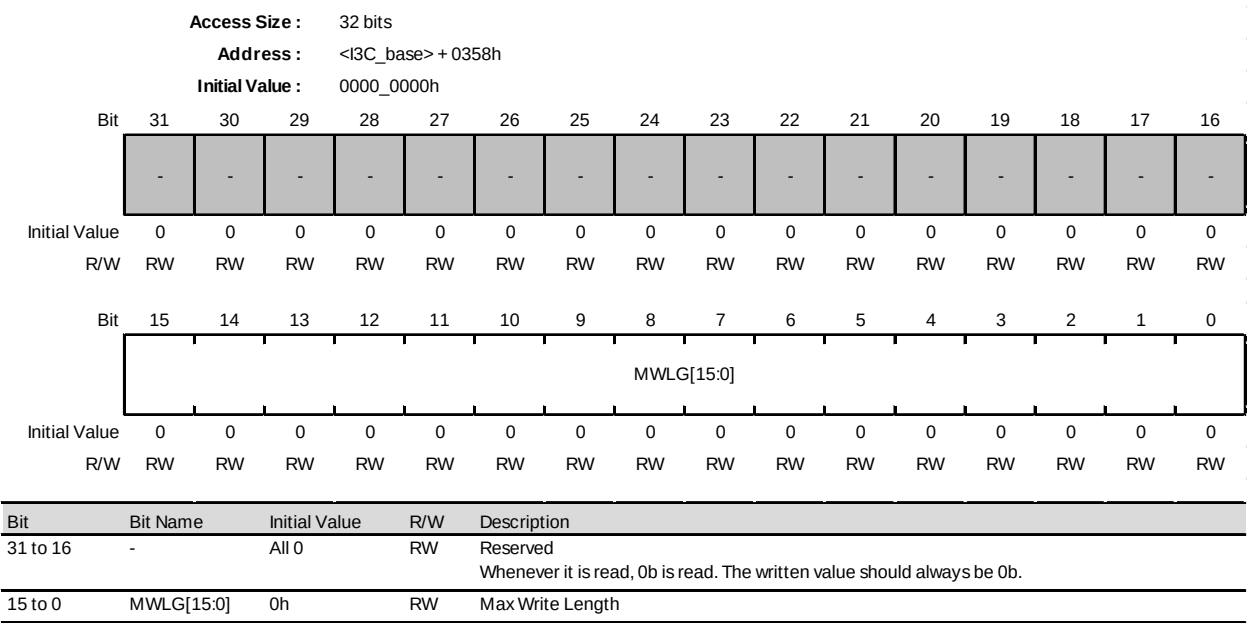
ACTST[3:0] bits (Activity State)

[Update conditions]

- When writing Activity State value.
- When receiving ENTAS0 CCC (Broadcast), these bits are updated to 1h.*¹
- When receiving ENTAS1 CCC (Broadcast), these bits are updated to 2h.*¹
- When receiving ENTAS2 CCC (Broadcast), these bits are updated to 4h.*¹
- When receiving ENTAS3 CCC (Broadcast), these bits are updated to 8h.*¹
- When Slave Address value is its own Slave Address in receiving ENTAS0 CCC (Direct), these bits are updated to 1h.*¹
- When Slave Address value is its own Slave Address in receiving ENTAS1 CCC (Direct), these bits are updated to 2h.*¹
- When Slave Address value is its own Slave Address in receiving ENTAS2 CCC (Direct), these bits are updated to 4h.*¹
- When Slave Address value is its own Slave Address in receiving ENTAS3 CCC (Direct), these bits are updated to 8h.*¹

Note 1. See the MIPI I3C Specification v1.0.

7.8.3.2.63 CCC Max Write Length Register (I3Cm_CMWLG)



Note: This register supports I3C secondary master mode and I3C slave mode.

MWLG[15:0] bits (Max Write Length)

These bits use for the I3C Master to set or get a maximum data write length in bytes for one Slave Device.

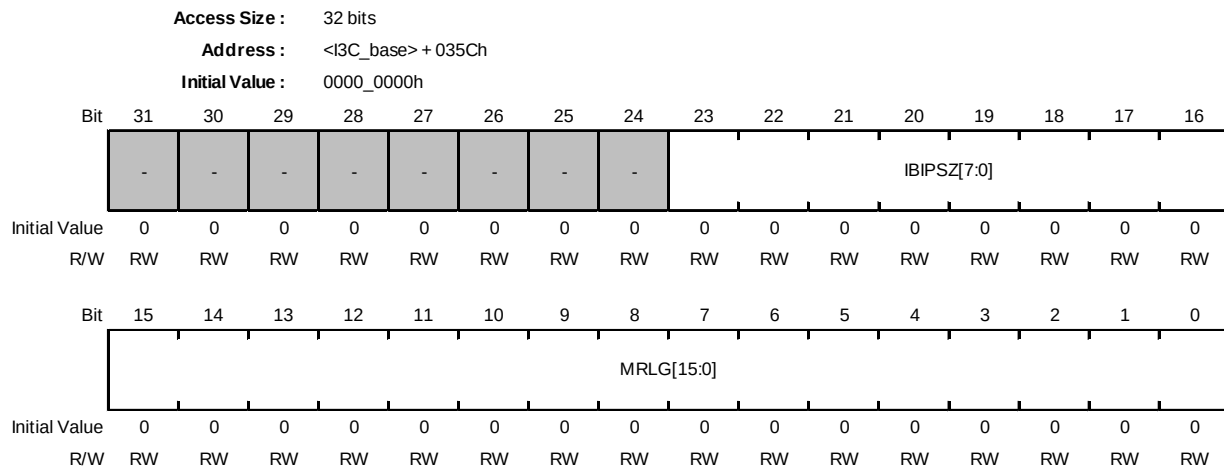
This Max Write Length does not affect data write lengths for Broadcast CCCs. The Set/Get Max Write Length value is transmitted over two bytes, with the most significant byte (MSB) transmitted first. The minimum value that Max Write Length can be set to is 8.

[Update conditions]

- When writing Max Write Length value.
- When receiving SETMWL CCC (Broadcast), these bits are updated to MWL value.*1
- When Slave Address value is its own Slave Address in receiving SETMWL CCC (Direct), these bits are updated to MWL value.*1

Note 1. See the MIPI I3C Specification v1.0

7.8.3.2.64 CCC Max Read Length Register (I3Cm_CMRLG)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
23 to 16	IBIPSZ[7:0]	0h	RW	IBI Payload Size
15 to 0	MRLG[15:0]	0h	RW	Max Read Length

Note: This register supports I3C secondary master mode and I3C slave mode.

MRLG[15:0] bits (Max Read Length)

These bits use for the I3C Master to set or get a maximum data read length for one Slave Device.

The Set/Get Max Read Length value is transmitted over the first two bytes, with most significant byte (MSB) transmitted first. The minimum value to which Max Read Length can be set is 16.

[Update conditions]

- When writing Max Read Length value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to MRL value.*1
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to MRL value.*1

IBIPSZ[7:0] bits (IBI Payload Size)

These bits use for the I3C Master to set or get optionally a maximum IBI payload size.

For devices with BCR bit 2 set to 1b, the Max IBI payload size value is added as a third-byte, where a value of 0 indicates an unlimited payload size. If Timing Control is used, then the minimum IBI payload size is either four bytes or five bytes. If Timing Control is not used, then the minimum IBI payload size is one (one byte).

This CCC is optional for the Slave, with two exceptions:

1. This CCC is required if both (a) any private Read Request Message (s) and/or any extended Read Request CCC (s) implemented by the Slave support a variable limit on the maximum number of data bytes that the Slave may return per Message, and (b) this limit is greater than 16 bytes.
2. This CCC is required if the Slave both (a) supports an IBI Payload (as indicated with BCR bit 1), and (b) will transmit more than one byte of private payload (not counting Timing Control bytes, when Timing Control is used).

[Update conditions]

- When writing Max IBI payload size value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to IBI payload size value.*¹
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to IBI payload size value.*¹

Note 1. See the MIPI I3C Specification v1.0.

7.8.3.2.65 CCC Enter Test Mode Register (I3Cm_CETSTMD)

Access Size :		32 bits																
Address :		<I3C_base> + 0360h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TSTMD[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read.
7 to 0	TSTMD[7:0]	0h	R	Test Mode 00h: Exit Test Mode This value removes all I3C devices from Test Mode. 01h: Vendor Test Mode This value indicates that I3C devices shall return a random 32bit value in the provisional ID during the Dynamic Address Assignment procedure. Others: MIPI reserved Reserved for future use by the MIPI Alliance

Note: This register supports I3C secondary master mode and I3C slave mode.

TSTMD[7:0] bits (Test Mode)

When these bits set to 00h, all I3C Devices remove from Test Mode.

When these bits set to 01h, I3C Devices shall return a random 32bit value in the Provisional ID during the Dynamic Address Assignment procedure.

The Broadcast CCC informs all I3C Devices that the Master is entering a specified Test Mode during manufacturing or Device test. The Enter Test Mode command Frame format includes a byte that specifies which Test Mode to enter.

Supporting I3C Devices shall enter the indicated Test Mode upon receipt of the Enter Test Mode CCC.

[Update condition]

- When receiving ENT TM CCC (Broadcast), these bits are updated to Test Mode Byte value.*1

Note 1. See the MIPI I3C Specification v1.0.

7.8.3.2.66 CCC Get Device Status Register (I3Cm_CGDVST)

Access Size :		32 bits														
Address :		<I3C_base> + 0364h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VDRSV[7:0]								ACTMD[1:0]		PRTE	-	PNDINT[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 8	VDRSV[7:0]	0h	RW	Vendor Reserved Reserved for vendor-specific meaning
7 to 6	ACTMD[1:0]	0h	RW	Slave Device's current Activity Mode 00b: Activity Mode 0 01b: Activity Mode 1 10b: Activity Mode 2 11b: Activity Mode 3
5	PRTE	0h	RW	Protocol Error 0b: The Slave has not detected a protocol error since the last Status read. 1b: The Slave has detected a protocol error since the last Status read.
4	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3 to 0	PNDINT[3:0]	0h	RW	Pending Interrupt Contains the interrupt number of any pending interrupt, or 0 if no interrupts are pending. This encoding allows for up to 15 numbered interrupts. If more than one interrupt is set, then the highest priority interrupt shall be returned.

Note: This register supports I3C secondary master mode and I3C slave mode.

PRTE bit (Protocol Error)

If this bit set to 1b, then the Slave detects a protocol error since the last Status read.

The Slave checks for such errors. Note that this value self-clears by the hardware upon every successful completion of a Master read of the Slave's Status.

The Direct CCC is a Get request for one I3C Slave Device to return its current Status, in the two-byte format detailed. Note that byte 0 is the LSB, and byte 1 is the MSB.

[Setting condition]

- When the Slave detected a protocol error.*¹

[Clearing condition]

- When transmission by own Slave Address is completed without error after receiving GETSTATUS CCC (Direct).*¹

ACTMD[1:0] bits (Slave Device's current Activity Mode)

Contains the two-bit ID of the Slave Device's current Activity Mode (readiness to support data read of sensor or related information).

Note 1. See the MIPI I3C Specification v1.0.

7.8.3.2.67 CCC Max Data Speed W (Write) Register (I3Cm_CMDSPW)

Access Size :		32 bits														
Address :		<I3C_base> + 0368h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MSWDR[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		RW		Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
2 to 0	MSWDR[2:0]		0h		RW		Maximum Sustained Write Data Rate 000b: fscI Max (default value) 001b: 8 MHz 010b: 6 MHz 011b: 4 MHz 100b: 2 MHz Others: Setting prohibited									

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.68 CCC Max Data Speed R (Read) Register (I3Cm_CMDSPR)

Access Size : 32 bits

Address : <I3C_base> + 036Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CDTTIM[2:0]			MSRDR[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5 to 3	CDTTIM[2:0]	0h	RW	Clock to Data Turnaround Time (TSCO) 000b: 8 ns or less (default value) 001b: 9 ns or less 010b: 10 ns or less 011b: 11 ns or less 100b: 12 ns or less 111b: TSCO is more than 12 ns, and is reported by private agreement. Others Setting prohibited
2 to 0	MSRDR[2:0]	0h	RW	Maximum Sustained Read Data Rate 000b: fscI Max (default value) 001b: 8 MHz 010b: 6 MHz 011b: 4 MHz 100b: 2 MHz Others: Setting prohibited

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.69 CCC Max Data Speed T (Turnaround) Register (I3Cm_CMDSPt)

Access Size :		32 bits														
Address :		<I3C_base> + 0370h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MRTE	-	-	-	-	-	-	-	MRTTIM[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MRTTIM[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	MRTE	0h	RW	Maximum Read Turnaround Time Enable 0b: Disables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 1: Without Turnaround) 1b: Enables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 2: With Turnaround)
30 to 24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
23 to 0	MRTTIM[23:0]	0h	RW	Maximum Read Turnaround Time 24-bit field can encode turnaround times from 0.0 seconds to 16 seconds. F40000h: 0 μs (minimum value) F40001h: 1 μs (resolution) ⋮ F42400h: 16 seconds (maximum value) F42401h: Setting prohibited ⋮ FFFFFFh: Setting prohibited

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.70 CCC Exchange Timing Support Information M (Mode) Register (I3Cm_CETSM)

Access Size : 32 bits

Address : <I3C_base> + 0374h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	INAC[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FREQ[7:0]								-	-	-	-	-	SPTAS YN1	SPTAS YN0	SPTSY N
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
23 to 16	INAC[7:0]	0h	RW	Inaccuracy Byte This byte represents the maximum variation of the Slave's internal oscillator in 1/10th percent (0.1%) increments, up to 25.5%. 00h: 0.0% 0Fh: 1.5% 1Fh: 3.1% 2Fh: 4.7% 3Fh: 6.3% 4Fh: 7.9% 5Fh: 9.5% 6Fh: 11.1% 7Fh: 12.7% 8Fh: 14.3% 9Fh: 15.9% AFh: 17.5% BFh: 19.1% CFh: 20.7% DFh: 22.3% EFh: 23.9% FFh: 25.5%
15 to 8	FREQ[7:0]	0h	RW	Frequency Byte This byte represents the Slave's internal oscillator frequency in increments of 0.5 MHz (500 kHz), up to 127.5 MHz. 00h: 32.0 KHz 0Fh: 7.5 MHz 1Fh: 15.5 MHz 2Fh: 23.5 MHz 3Fh: 31.5 MHz 4Fh: 39.5 MHz 5Fh: 47.5 MHz 6Fh: 55.5 MHz 7Fh: 63.5 MHz Others: Setting prohibited
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	SPTASYN1	0h	RW	Support Async Mode 1 0b: Async Mode 1 is not supported. 1b: Async Mode 1 is supported.
1	SPTASYN0	0h	RW	Support Async Mode 0 0b: Async Mode 0 is not supported. 1b: Async Mode 0 is supported.

Bit	Bit Name	Initial Value	R/W	Description
0	SPTSYN	0h	RW	Supports Sync Mode 0b: Sync Mode is not supported. 1b: Sync Mode is supported.

Note: This register supports I3C secondary master mode and I3C slave mode.

SPTASYN1 bit (Support Async Mode 1)

Bit mask indicating Async Mode 1 of Timing Control Mode(s) the target Slave supports.

If this bit set (has value 1), then that Slave supports Async Mode 1 of Timing Control Mode.

SPTASYN0 bit (Support Async Mode 0)

Bit mask indicating Async Mode 0 of Timing Control Mode(s) the target Slave supports.

If this bit set (has value 1), then that Slave supports Async Mode 0 of Timing Control Mode.

SPTSYN bit (Supports Sync Mode)

Bit mask indicating Sync Mode of Timing Control Mode(s) the target Slave supports.

If this bit set (has value 1), then that Slave supports Sync Mode of Timing Control Mode.

The Directed CCC provides the framework for the Master to query the Exchange Timing capabilities supported by the I3C Slaves. The Get Exchange Timing Support Information CCC causes the addressed Slave to return four data bytes containing key information on supported current state, and internal oscillator/clock frequency and inaccuracy.

7.8.3.2.71 CCC Exchange Timing Support Information S (State) Register (I3Cm_CETSS)

Bit mask indicating which Timing Control Mode (if any) is currently enabled for the target Slave, and whether any counter overflows have occurred since the most recent previous check. If a Timing Control Mode bit is set (has value 1), then that Slave has currently enabled the corresponding Timing Control Mode. If the Overflow bit is set (has value 1), then that Slave experienced a counter overflow since the most recent previous check.

Access Size :		32 bits														
Address :		<I3C_base> + 0378h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ICOVF	-	-	-	-	ASYNE[1:0]		SYNE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7	ICOVF	0h	RW	Internal Counter Overflow 0b: Slave has not experienced a counter overflow since the most recent previous check. 1b: Slave experienced a counter overflow since the most recent previous check.
6 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2, 1	ASYNE[1:0]	0h	RW	Async Mode Enable 00b: All Mode disabled 01b: Async Mode 0 enabled 10b: Async Mode 1 enabled 11b: Setting prohibited
0	SYNE	0h	RW	Sync Mode Enable 0b: Sync Mode disabled 1b: Sync Mode enabled

Note: This register supports I3C secondary master mode and I3C slave mode.

ASYNE[1] bit (Async Mode Enabled 1)

Slave Timing Control Async Mode 1 is enabled.

[Setting condition]

- When writing 1b.
- When CETSM.SPTASYN[1] bit = 1b and either of the following 1) or 2) is satisfied.
 - 1) When receiving SETXTIME CCC(Broadcast) with Defining byte value EFh.
 - 2) When Slave Address value is its own Slave Address in receiving SETXTIME CCC(Direct) with Defining byte value EFh.

[Clearing condition]

- When writing 0b.
- When CETSM.SPTASYN[1] bit = 1b and either of the following 1) or 2) is satisfied.

- 1) When receiving SETXTIME CCC(Broadcast) with Defining byte value DFh.
- 2) When Slave Address value is its own Slave Address in receiving SETXTIME CCC(Direct) with Defining byte value DFh.

ASYNE[0] bit (Async Mode Enabled 2)

Slave Timing Control Async Mode 0 is enabled.

[Setting condition]

- When writing 1b.
- When CETSM.SPTASYN[0] bit = 1b and either of the following 1) or 2) is satisfied.
 - 1) When receiving SETXTIME CCC(Broadcast) with Defining byte value DFh.
 - 2) When Slave Address value is its own Slave Address in receiving SETXTIME CCC(Direct) with Defining byte value DFh.

[Clearing condition]

- When writing 0b.
- When CETSM.SPTASYN[0] bit = 1 and either of the following 1) or 2) is satisfied.
 - 1) When receiving SETXTIME CCC(Broadcast) with Defining byte value EFh.
 - 2) When Slave Address value is its own Slave Address in receiving SETXTIME CCC(Direct) with Defining byte value EFh.

7.8.3.2.72 Bit Count Register (I3Cm_BITCNT)

Access Size :		32 bits														
Address :		<I3C_base> + 0380h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	BCNT[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read.
4 to 0	BCNT[4:0]	0h	R	Bit Counter Indicates the number of bits remaining to be transferred. For details on the values, see Table 7.8-8 and Table 7.8-9 .

BCNT[4:0] bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a sampling edge on the SCL line.

Table 7.8-8 I²C transfer / Legacy I²C transfer

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
00h	2 to 1 bits	2 to 1 bits	3 to 1 bits	2 to 1 bits
01h	3 bits	3 bits	4 bits	3 bits
02h	4 bits	4 bits	5 bits	4 bits
03h	5 bits	5 bits	6 bits	5 bits
04h	6 bits	6 bits	7 bits	6 bits
05h	7 bits	7 bits	8 bits	7 bits
06h	8 bits	8 bits	9 bits	8 bits
07h	9 bits	9 bits	—	9 bits

Table 7.8-9 I3C transfer

BCNT[4:0]	SDR*1		BCNT[4:0]	SDR*1	
	Transmission	Reception		Transmission	Reception
00h	1 bit	2 to 1 bits	05h	6 bits	7 bits
01h	2 bits	3 bits	06h	7 bits	8 bits
02h	3 bits	4 bits	07h	8 bits	9 bits
03h	4 bits	5 bits	08h	9 bits	—
04h	5 bits	6 bits			

Note 1. The address phase is the same as in **Table 7.8-8**.

7.8.3.2.73 Normal Queue Status Level Register (I3Cm_NQSTLV)

Access Size :		32 bits														
Address :		<I3C_base> + 0394h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	IBISCNT[4:0]					IBIQLV[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSPQLV[7:0]								CMDQFLV[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read.
28 to 24	IBISCNT[4:0]	0h	R	Normal IBI Status Count* ² Number of IBI Status entries currently in the IBI Queue.
23 to 16	IBIQLV[7:0]	0h	R	Normal IBI Queue Level* ¹ Number of buffer entries currently in the IBI Queue.
15 to 8	RSPQLV[7:0]	0h	R	Normal Response Queue Level* ¹ Number of buffer entries currently in the Response Queue.
7 to 0	CMDQFLV[7:0]	0h	R	Normal Command Queue Free Level* ¹ Number of free buffer entries currently in the Command Queue. Reset value is the depth of the Command Queue.

- Note 1. These bits support all I3C mode.
- Note 2. These bits support I3C master mode and I3C secondary master mode.

7.8.3.2.74 Normal Data Buffer Status Level Register (I3Cm_NDBSTLV0)

Access Size :		32 bits														
Address :		<I3C_base> + 0398h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDBLV[7:0]								TDBFLV[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 8	RDBLV[7:0]	0h	R	Normal Receive Data Buffer Level Indicates the number of Receive Data Buffer entries in the Receive Data Queue.
7 to 0	TDBFLV[7:0]	1h	R	Normal Transmit Data Buffer Free Level Indicates the number of free Transmit Data Buffer entries in the Transmit Data Queue. Reset value is the depth of the Transmit Data Queue.

Note: This register supports all I3C mode.

7.8.3.2.75 Normal Receive Status Queue Status Level Register (I3Cm_NRSQSTLV)

Access Size :		32 bits														
Address :		<I3C_base> + 03C0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RSQLV[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 8	-		All 0		R	Reserved Whenever it is read, 0b is read.										
7 to 0	RSQLV[7:0]		0h		R	Normal Receive Status Queue Level										

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.76 Present State Debug Register (I3Cm_PRSTDBG)

Access Size : 32 bits

Address : <I3C_base> + 03CCh

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SDOLV	SCOLV	SDILV	SCILV
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read.
3	SDOLV	0h	R	SDA30 Output Level 0b: I3C has driven the SDA30 pin low. 1b: I3C has released the SDA30 pin.
2	SCOLV	0h	R	SCL30 Output Level 0b: I3C has driven the SCL30 pin low. 1b: I3C has released the SCL30 pin.
1	SDILV	0h	R	SDA30 Line Signal Level This bit is used to check the SDA30 Line level, in order to recover from errors and for debugging.
0	SCILV	0h	R	SCL30 Line Signal Level This bit is used to check the SCL30 Line level, in order to recover from errors and for debugging.

SDOLV bit (SDA Output Level) and SCOLV bit (SCL Output Level)

When reading these bits, the state of signals output from I3C can be read.

7.8.3.2.77 Master Error Counters Register (I3Cm_MSERRCNT)

Access Size :		32 bits														
Address :		<I3C_base> + 03D0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	M2ECNT[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 8	-		All 0		R/W		Description									
							Reserved									
							Whenever it is read, 0b is read.									
7 to 0	M2ECNT[7:0]		0h		R		M2 Error Counter									
							Counts I3C Type M2 errors on the I3C Bus.									
							Cleared upon read out.									

Note: This register supports I3C secondary master mode and I3C slave mode.

7.8.3.2.78 SC1 Capture monitor Register (I3Cm_SC1CPT)

Access Size :		32 bits															
Address :		<I3C_base> + 03E0h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SC1CPT[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	SC1CPT[15:0]	0h	R	SC1 Capture

Note: This register supports I3C secondary master mode and I3C slave mode.

SC1C[15:0] bits (SC1 Capture)

- Async Mode 0 (Asynchronous Basic Mode)

After enabling ATCCNTE.ATCE, SC1 Counter counts up from SC1 count trigger*¹ to SCL rise edge next to ACK for the IBI, and capture it as SC1.

- Async Mode 1 (Asynchronous Advanced Mode)

After enabling ATCCNTE.ATCE, SC1 Counter counts up from SC1 count trigger*¹ to the first aME, and capture it as SC1.

Note 1. SW or external trigger can be selected by selection bits.

Remark As the timing control specification, the SC1 counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

7.8.3.2.79 SC2 Capture monitor Register (I3Cm_SC2CPT)

Access Size :		32 bits														
Address :		<I3C_base> + 03E4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC2CPT[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	SC2CPT[15:0]	0h	R	SC2 Capture

Note: This register supports I3C secondary master mode and I3C slave mode.

SC2C[15:0] bits (SC2 Capture)

- Async Mode 0 (Asynchronous Basic Mode)
After enabling ATCCNTE.ATCE, SC2 Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.
- Async Mode 1 (Asynchronous Advanced Mode)
After enabling ATCCNTE.ATCE, SC2 Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.

Remark As the timing control specification, the SC2 counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

7.8.4 Operation

7.8.4.1 Data Structures

7.8.4.1.1 Command Descriptor

The write-only Command Descriptor structure is 64 bits in length. The Command Descriptor is put to the Command Queue with writes to the Command Queue Port.

Write to the Command Queue Port in the following order:

1. First write: The least significant DWORD (Command Descriptor Structure Low).
2. Second write: The most significant DWORD (Command Descriptor Structure High).

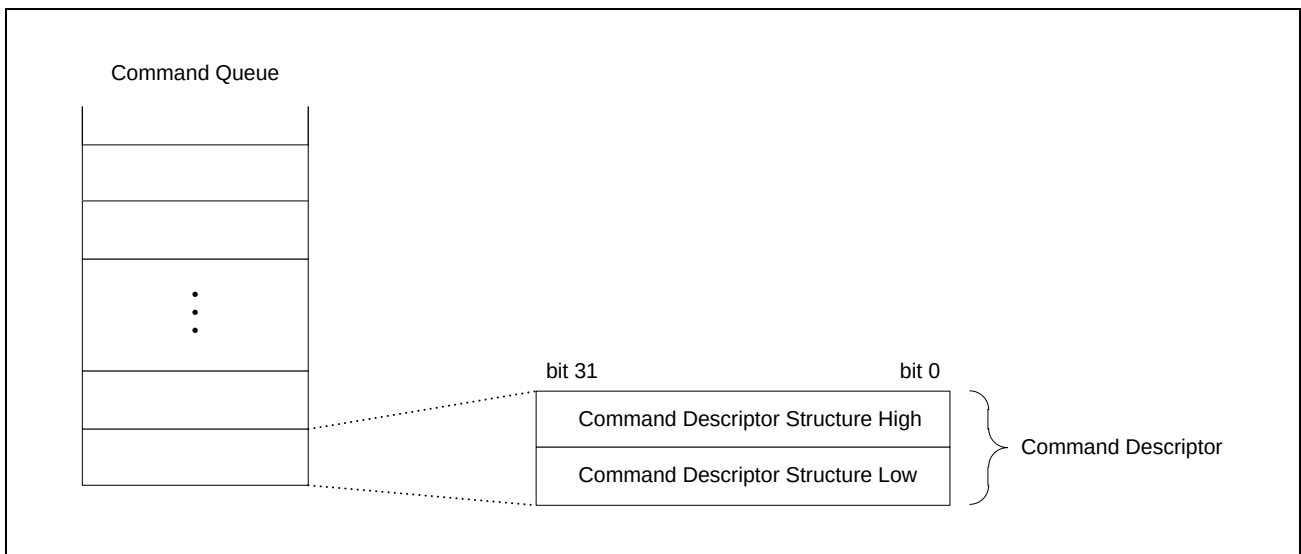


Figure 7.8-2 Command descriptor data structure

I3C provides a Command Descriptor structure for each command type as follows:

- Address Assign Command
- Immediate Transfer Command
- Regular Transfer Command
- Combo Transfer Command
- Internal Control Command

Details are explained in the following sections.

(1) Address Assign Command

This command is used for address assignment (ENTDAA, SETDASA).

Remark When issuing SETAASA CCC, use the Immediate Transfer command.

The I3C provides an address assign command for the following mode:

- I3C Master Mode

Details of the Address Assign command structure are as follows.

Bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field	TOC	ROC	DEV_COUNT[3:0]				—	—	—	—	EXT_D EVICE	DEV_INDEX[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	—	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attributes 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID	W
14:7	CMD[7:0]	Transfer Command CCC Value	W
15	—	The write value should be 0b.	W
20:16	DEV_INDEX[4:0]	Device Index	W
21	EXT_DEVICE	Extended Device Index 0b: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1b: Use the EXDATBAS table.	W
25:22	—	The write value should be 0b.	W
29:26	DEV_COUNT[3:0]	Device Count	W

Bit	Symbol	Function	R/W
30	ROC	Response on Completion 0b: NOT_REQUIRED: Response Status is not required. 1b: REQUIRED: Response Status is required.	W
31	TOC	Terminate on Completion 0b: RESTART: Issue Repeated START (Sr) at end of transfer 1b: STOP: Issue STOP (P) at end of transfer	W
63:32	—	The write value should be 0b.	W

CMD_ATTR[2:0] bits (Command Attributes)

Command Type, defining the format of the other fields.

TID[3:0] bits (Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CMD[7:0] bits (Transfer Command CCC Value)

Specifies CCC code indicating whether Address Assignment uses ENTDAAs or SETDASAs commands. The field comprises the entire command code (ENTDAAs or SETDASAs).

DEV_INDEX[4:0] bits (Device Index)

Indicates the DATBASm table index for the Slave device being addressed with the transfer. Static and device addressing related information are stored to this index in the DATBASm.

DEV_COUNT[3:0] bits (Device Count)

Indicates the number of devices that a dynamic address is assigned to.

ROC bit (Response on Completion)

Controls whether Response Status is sent after successful completion of the Transfer command. The successful completion is read from register NRSPQP. Upon unsuccessful transfer the Response Status is sent.

TOC bit (Terminate on Completion)

Controls what bus condition to issue after the Transfer command completes.

For ENTDAAs, a STOP condition is issued regardless of the setting value of TOC. It is meaningful for SETDASAs transfers.

When sending SETDASAs CCC by TOC = 0b (RESTART), the next command must be set to SETDASAs CCC with the Address Assign Command.

When the next command is not the same SETDASAs CCC flame, it must be set to TOC = 1b (STOP).

(2) Immediate Transfer Command

This structure directly contains data (max 4 bytes) to be transferred, and as a result is only useful for Transfers/CCCs that write data. This structure shall not be used for Read operations (for example, to receive data).

When transmitting data of 4 bytes or less, use this Immediate Transfer Command to communicate.

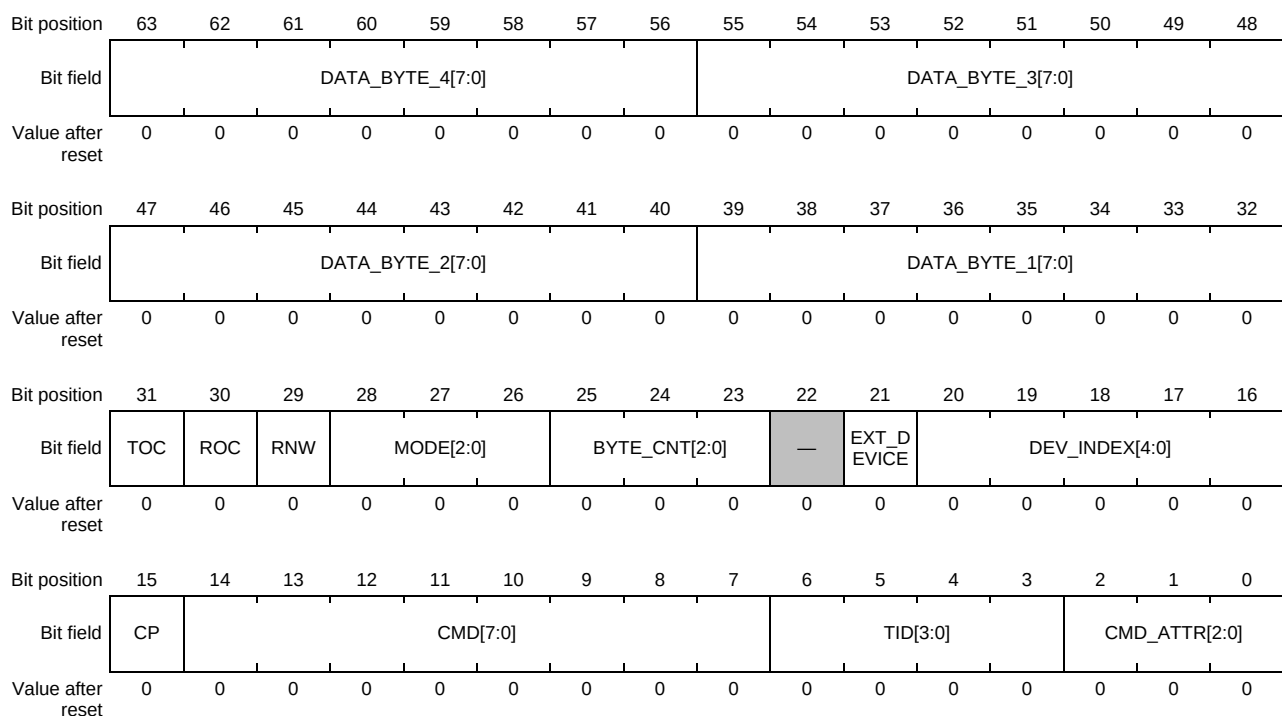
When transmitting data of 5 bytes or more, use the Regular Transfer Command to communicate.

For the Regular Transfer Command, see **(3) Regular Transfer Command**.

I3C provides an Immediate Transfer Command for the following mode:

- I3C Master Mode

Details of the Immediate Transfer Command Structure of each mode are shown in this section.



Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Immediate Data Transfer Command Attribute 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Immediate Data Transfer Transaction ID	W
14:7	CMD[7:0]	Immediate Data Transfer CCC Value	W
15	CP	Immediate Data Transfer Command Present 0b: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1b: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Immediate Data Transfer Device Index	W

Bit	Symbol	Function	R/W
21	EXT_DEVICE	Immediate Data Transfer Extended Device Index 0b: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1b: Use the EXDATBAS table.	W
22	—	The write value should be 0b.	W
25:23	BYTE_CNT[2:0]	Immediate Data Transfer Byte Count 0h: No payload 1h: 1 byte is valid. 2h: 2 bytes are valid. 3h: 3 bytes are valid. 4h: 4 bytes are valid. Others: Setting prohibited	W
28:26	MODE[2:0]	Immediate Data Transfer Mode and Speed Values 0h: I3C SDR0/Data rate: STDBR (I3C mode) I ² C Message 0/Data rate: STDBR (I ² C mode) 1h: I3C SDR1/Data rate: EXTBR (I3C mode) I ² C Message 0/Data rate: EXTBR (I ² C mode) 2h: I3C SDR2/Data rate: STDBR × 2 (I3C mode) Reserved (I ² C mode) 3h: I3C SDR3/Data rate: EXTBR × 2 (I3C mode) Reserved (I ² C mode) 4h: I3C SDR4/Data rate: EXTBR × 4 (I3C mode) Reserved (I ² C mode) Others: Setting prohibited	W
29	RNW	Immediate Data Transfer R/W 0b: WRITE: Write transfer 1b: READ: Read transfer	W
30	ROC	Immediate Data Transfer Response on Completion 0b: NOT_REQUIRED: Response Status is not required. 1b: REQUIRED: Response Status is required.	W
31	TOC	Immediate Data Transfer Terminate on Completion 0b: RESTART: Issue Repeated START (Sr) at end of data transfer 1b: STOP: Issue STOP (P) at end of data transfer	W
39:32	DATA_BYTE_1[7:0]	Immediate Data Transfer Data Byte 1 Direct argument	W
47:40	DATA_BYTE_2[7:0]	Immediate Data Transfer Data Byte 2 Direct argument	W
55:48	DATA_BYTE_3[7:0]	Immediate Data Transfer Data Byte 3 Direct argument	W
63:56	DATA_BYTE_4[7:0]	Immediate Data Transfer Data Byte 4 Direct argument	W

CMD_ATTR[2:0] bits (Immediate Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Immediate Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Immediate Data Transfer Command Present)

Indicates whether CMD field is valid for CCC Transfer.

DEV_INDEX[4:0] bits (Immediate Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

BYTE_CNT[2:0] bits (Immediate Data Transfer Byte Count)

Number of valid data bytes to use in this Immediate Data Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

MODE[2:0] bits (Immediate Data Transfer Mode and Speed Values)

Sets the mode and speed for the I3C or I²C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I²C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Immediate Data Transfer R/W)

Identifies direction of the transfer.

This field shall always be set to 0b, because Immediate transfers are valid for Write transactions only.

ROC bit (Immediate Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Immediate Data Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

When sending Direct CCC by TOC = 0b (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1b (STOP).

(3) Regular Transfer Command

This structure does not contain data to be transferred.

For Master Mode, the data buffer is available through Transfer Data Queue Port (Receive Data Queue Port and Transmit Data Queue Port).

When transmitting data of 5 bytes or more, use this Regular Transfer Command to communicate.

When transmitting data of 4 bytes or less, use the Immediate Transfer Command to communicate.

For the Regular Transfer Command, see **(2) Immediate Transfer Command**.

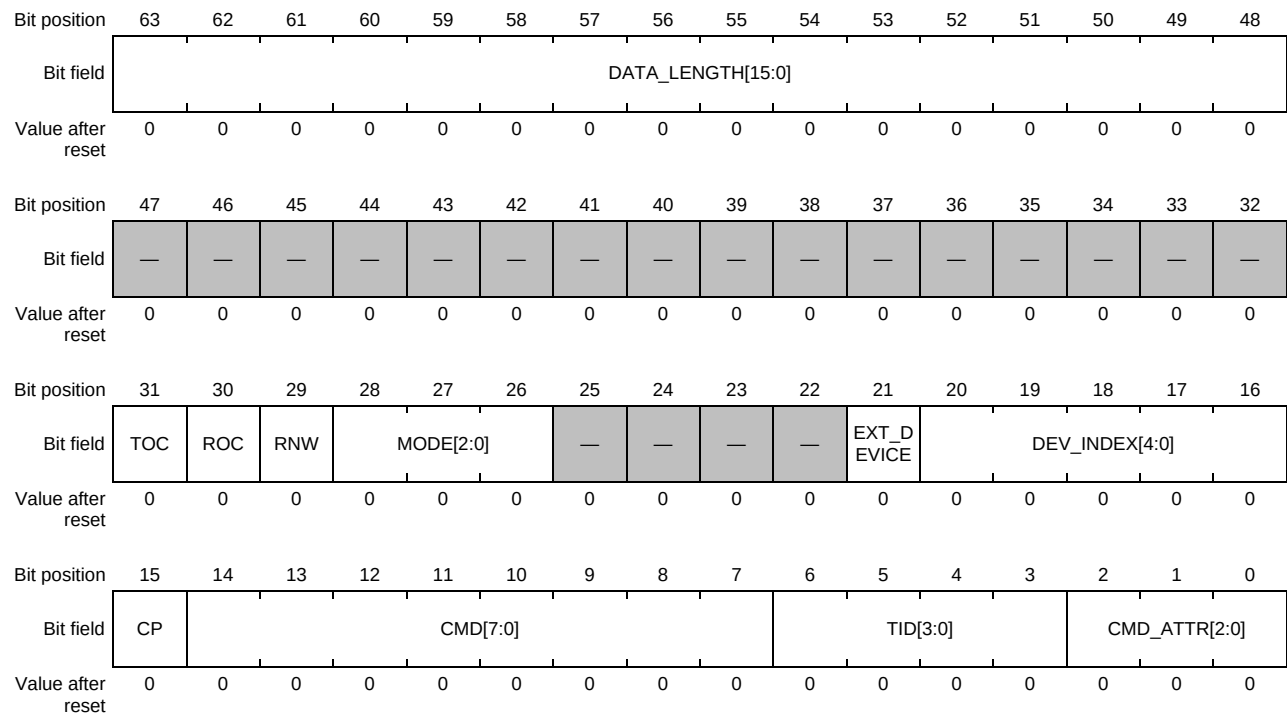
For I3C Slave Mode, the IBI Payload buffer is available through IBI Status Queue Port.

I3C provides a Regular Transfer Command for each mode below.

- I3C Master Mode
- I3C Slave Mode

Details of the regular transfer command structure of each mode are shown below.

(1) I3C Master Mode



Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W

Bit	Symbol	Function	R/W
14:7	CMD[7:0]	Data Transfer CCC Code Value Specifies the I3C Command code	W
15	CP	Data Transfer Command Present 0b: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1b: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Data Transfer Device Index	W
21	EXT_DEVICE	Data Transfer Extended Device Index 0b: Use the DATBASm Table indicated by DEV_INDEX[4:0]. 1b: Use the EXDATBAS table.	W
25:22	—	The write value should be 0b.	W
28:26	MODE[2:0]	Data Transfer Speed and Mode 0h: I3C SDR0/Data rate: STDBR (I3C mode) I ² C Message 0/Data rate: STDBR (I ² C mode) 1h: I3C SDR1/Data rate: EXTBR (I3C mode) I ² C Message 0/Data rate: EXTBR (I ² C mode) 2h: I3C SDR2/Data rate: STDBR × 2 (I3C mode) Reserved (I ² C mode) 3h: I3C SDR3/Data rate: EXTBR × 2 (I3C mode) Reserved (I ² C mode) 4h: I3C SDR4/Data rate: EXTBR × 4 (I3C mode) Reserved (I ² C mode) Others: Setting prohibited	W
29	RNW	Data Transfer R/W 0b: WRITE: Write transfer 1b: READ: Read transfer	W
30	ROC	Data Transfer Response on Completion 0b: NOT_REQUIRED: Response Status is not required. 1b: REQUIRED: Response Status is required.	W
31	TOC	Data Transfer Terminate on Completion 0b: RESTART: Issue Repeated START (Sr) at end of transfer 1b: STOP: Issue STOP (P) at end of transfer	W
47:32	—	The write value should be 0.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID Identification tag for this command)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Data Transfer Command Present)

Indicates whether the contents of the CMD field is valid for a CCC Transfer.

DEV_INDEX[4:0] bits (Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

MODE[2:0] bits (Data Transfer Speed and Mode)

Sets the mode and speed for the I3C or I²C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I²C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Data Transfer Terminate on Completion)

Controls what Bus condition will be issued after completion of the transfer.

When sending Direct CCC by TOC = 0b (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1b (STOP).

DATA_LENGTH[15:0] bits (Data Transfer Data Length)

Number of valid data bytes to use in this Regular Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

Length setting of GETMXDS command should be fixed to 5.

(2) I3C Slave Mode

Bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field	DATA_LENGTH[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field	—	ROC	RNW	—	—	—	—	—	—	ITS	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	HJ	—	—	—	—	—	—	—	—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W
14:7	—	The write value should be 0b.	W
15	HJ	Data Transfer Hot-Join Event 0b: Slave Interrupt Request or Mastership Request, so the RNW field is valid. 1b: Hot-Join Event, so the RNW field is not valid.	W
21:16	—	The write value should be 0b.	W
22	ITS	Include timestamp for Async Mode 0b: Do not include timestamp. 1b: Include timestamp.	W
28:23	—	The write value should be 0b.	W
29	RNW	Data Transfer R/W 0b: WRITE: Write transfer (Mastership Request) 1b: READ: Read transfer (Slave Interrupt Request)	W
30	ROC	Data Transfer Response on Completion 0b: NOT_REQUIRED: Response Status is not required. 1b: REQUIRED: Response Status is required.	W
47:31	—	The write value should be 0b.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID Identification tag for this command)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

HJ bit (Data Transfer Hot-Join Event)

Indicates whether Hot-Join Event is valid in this IBI Data transfer.

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

(4) Combo Transfer Command

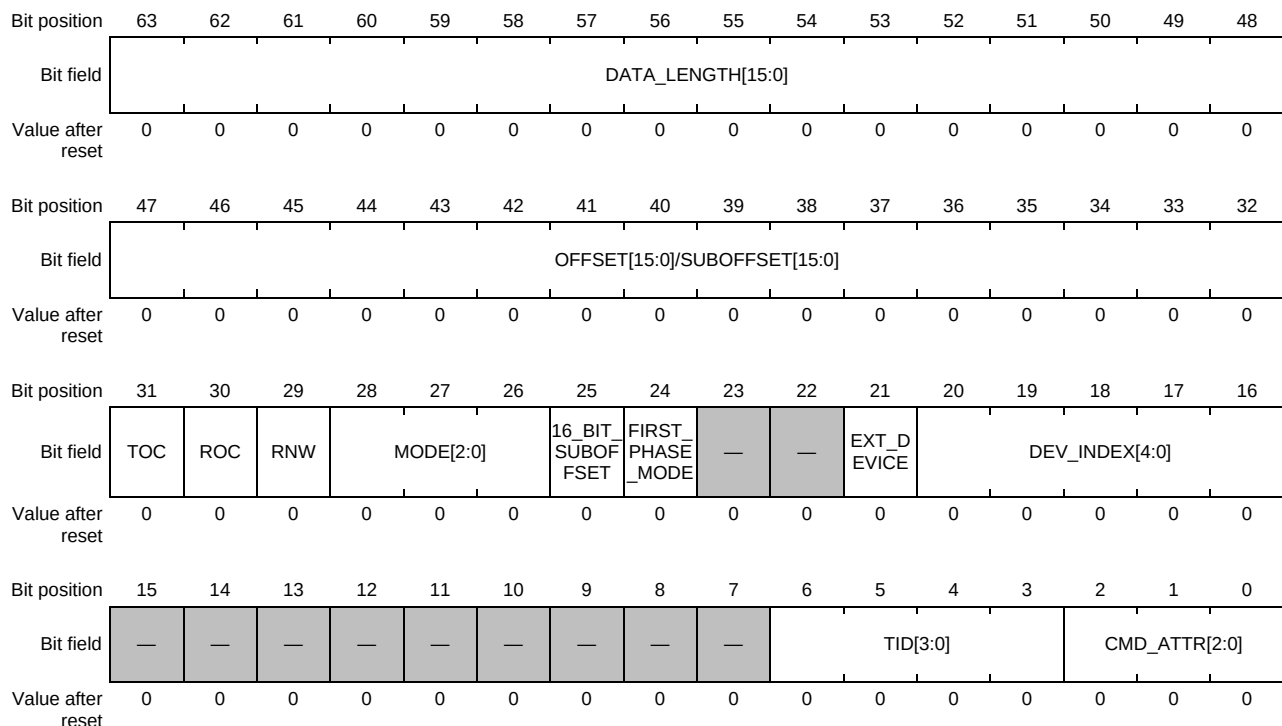
This structure contains a combined Write + Read/Write operation.

The data buffer is available through Transfer Data Queue Port (Receive Data Queue Port and Transmit Data Queue Port).

I3C provides a Combo Transfer Command for the following mode:

- I3C Master Mode

Details of the Combo Transfer Command Structure of each mode are as follows.



Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Combo Transfer Command Attribute Command Type, defining the format of the other fields. 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Combo Transfer Transaction ID Identification tag for the command	W
15:7	—	The write value should be 0b.	W
20:16	DEV_INDEX[4:0]	Combo Transfer Device Index	W
21	EXT_DEVICE	Combo Transfer Extended Device Index 0b: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1b: Use the EXDATBAS table.	W
23:22	—	The write value should be 0b.	W

Bit	Symbol	Function	R/W
24	FIRST_PHASE_MODE	Combo Transfer First Phase Mode 0b: SDR: First phase is executed in SDR mode. 1b: MODE: First phase is executed in the mode indicated by the MODE field.	W
25	16_BIT_SUBOFFSET	Combo Transfer Sub Offset Size 0b: 8_BIT_SUBOFFSET: Sub-offset is 8-bits long. Value is encoded in Lower Byte of OFFSET/SUBOFFSET field. 1b: 16_BIT_SUBOFFSET: Sub-offset is 16-bits long.	W
28:26	MODE[2:0]	Combo Transfer Speed and Mode Values for I3C Mode 0h: I3C SDR0/Data rate: STDBR 1h: I3C SDR1/Data rate: EXTBR 2h: I3C SDR2/Data rate: STDBR × 2 3h: I3C SDR3/Data rate: EXTBR × 2 4h: I3C SDR4/Data rate: EXTBR × 4 Others: Setting prohibited	W
29	RNW	Combo Transfer R/W Identifies direction of the transfer 0b: WRITE: Write transfer 1b: READ: Read transfer	W
30	ROC	Combo Transfer Response on Completion 0b: NOT_REQUIRED: Response Status is not required. 1b: REQUIRED: Response Status is required.	W
31	TOC	Combo Transfer Terminate on Completion 0b: RESTART: Issue Repeated START (Sr) at end of transfer 1b: STOP: Issue STOP (P) at end of transfer	W
47:32	OFFSET[15:0]/SUB OFFSET[15:0]	Combo Transfer Offset/Sub-Offset Offset of the target operation	W
63:48	DATA_LENGTH [15:0]	Combo Transfer Data Length Number of bytes to be transferred. This field must be set to non-zero value.	W

CMD_ATTR[2:0] bits (Combo Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Combo Transfer Transaction ID Identification tag for the command)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

DEV_INDEX[4:0] bits (Combo Transfer Device Index)

Indicates the DATBASm table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

FIRST_PHASE_MODE bits (Combo Transfer First Phase Mode)

Indicates whether the first phase of the Combo Transfer is executed in SDR Mode, vs. the Mode indicated by the MODE field.

MODE[2:0] bits (Combo Transfer Speed and Mode Values for I3C Mode)

Sets the mode and speed for the I3C or I²C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I²C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Combo Transfer R/W Identifies direction of the transfer)

Identifies direction of the transfer.

ROC bit (Combo Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Combo Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

DATA_LENGTH[15:0] bit (Combo Transfer Data Length Number of bytes to be transferred. This field must be set to non-zero value.)

Number of valid data bytes to use in this Combo Transfer Descriptor.

This field must be set to non-zero value.

(5) Internal Control Command

This structure is used for controlling I3C itself (not for transfer commands).

I3C provides an Internal Control Command for the following mode:

- I3C Master Mode

Details of the Internal Control Command Structure are as follows:

Bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	—	—	—	ON_OFF	MIPI_CMD[3:0]				—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attribute*2 Command Type, defining the format of the other fields. 0h: XFER: Regular Transfer 1h: IMMED_DATA_XFER: Immediate Data Transfer 2h: ADDR_ASSGN_CMD: Address Assignment Command 3h: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 7h: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID Identification tag for the command	W
7	—	The write value should be 0b.	W
11:8	MIPI_CMD[3:0]	MIPI Alliance Command 00h: NoOp, so the ON_OFF field is not valid. 02h: Include 7E (IBA), so the ON_OFF field is valid. Others: Setting prohibited	W
12	ON_OFF	Bus Instance 7E On/Off*1 Enables or disables automatic transmission of the I3C Broadcast Header after every START condition on this I3C Bus instance. 0b: IBA_INCLUDE off 1b: IBA_INCLUDE on	W
63:13	—	The write value should be 0b.	W

Note 1. The IBA_INCLUDE on state set by MIPI_CMD [3:0] = 2h and ON_OFF = 1 is cleared by setting RSTCTL.INTLRST to 1b.

Note 2. The Response descriptor is not stored when the Internal Control Command is executed.

7.8.4.1.2 Response Descriptor

The Response Descriptor is a read-only structure describing the success or failure of a command, and the amount of data transferred.

The Response Descriptor is read from Response Queue with reads from Response Queue Port.

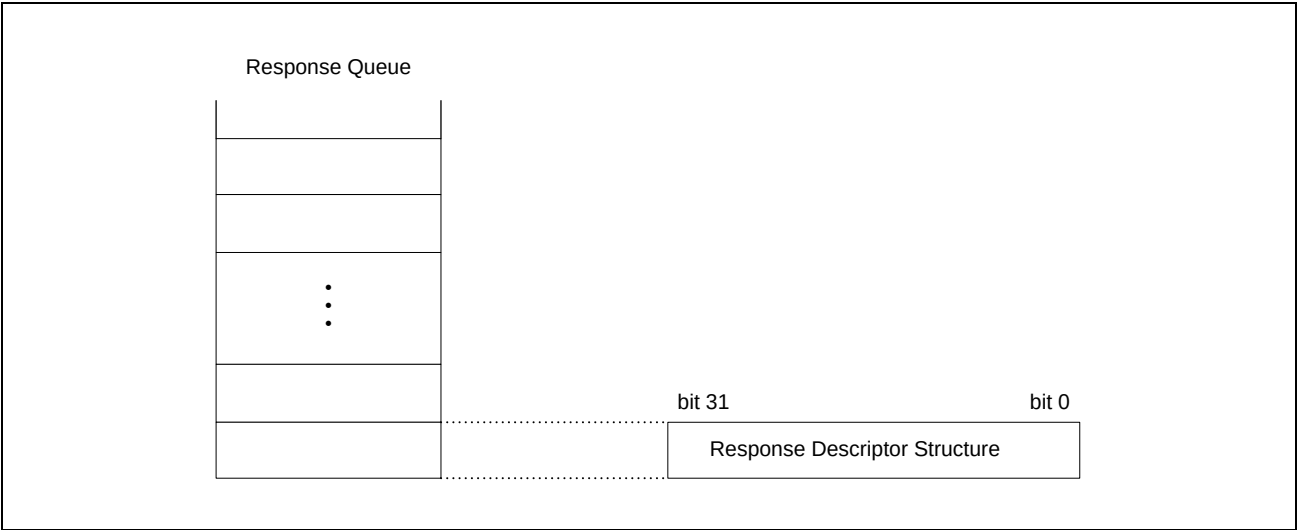


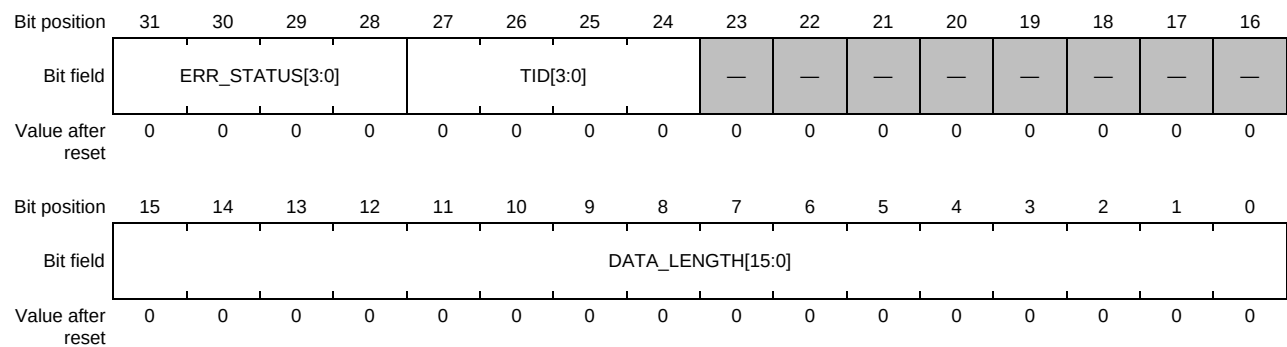
Figure 7.8-3 Response descriptor data structure

I3C provides a Response Descriptor for the following modes:

- I3C Master Mode
- I3C Slave Mode

Details of the Response Descriptor structure of each mode are shown in the following sections.

(1) I3C Master Mode

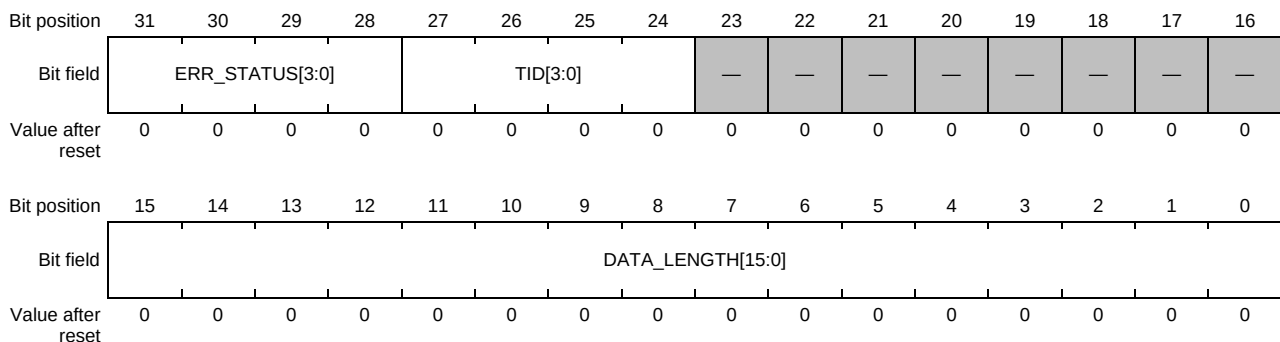


Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length/Device Count The meaning of this field depends on the context: For Write Transfer: Remaining data length (in bytes) For Read Transfer: Received data length (in bytes) For Address Assignment: Remaining Device count	R
23:16	—	These bits are read as 0b.	R

Bit	Symbol	Function	R/W
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value shall match one of commands sent on the Bus. 0h-7h: Valid Transaction IDs Others: Setting prohibited	R
31:28	ERR_STATUS[3:0]	MIPI Alliance Command 0h: SUCCESS: Transfer successful, no error 1h: CRC: CRC Error 2h: PARITY: Parity Error 3h: FRAME: Frame Error 4h: ADDR_HEADER: Address Header Error 5h: NACK: Address NACKed or Dynamic Address Assignment NACKed 6h: OVL: Receive Overflow or Transfer Underflow Error 8h: ABORTED: Aborted 9h: I ² C_WR_DATA_NACK: NACK received for the I ² C Write Data transfer Ah: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Setting prohibited	R

Note: In I3C Master mode, when an abnormal command with a specific parameter that is not supported is stored in Command Descriptor, it is indicated as NOT_SUPPORTED (Ah) in ERR_STATUS [3:0].

(2) I3C Slave Mode



Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length Remaining data length (in bytes) for Slave Interrupt Request	R
23:16	—	These bits are read as 0b.	R
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value matches one of commands sent on the bus. 0h-7h: Valid Transaction IDs Others: Setting prohibited	R

Bit	Symbol	Function	R/W
31:28	ERR_STATUS[3:0]	Response Error Status 0h: SUCCESS: Transfer successful, no error. 3h: FRAME: Frame Error 4h: ADDR_HEADER: Address Header Error 5h: NACK: Address NACK'ed or Dynamic Address Assignment NACK'ed 6h: OVL: Receive Overflow or Transfer Underflow Error 8h: ABORTED: Aborted Ah: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Setting prohibited	R

Note: In I3C Slave mode, it is indicated as NOT_SUPPORTED (Ah) in ERR_STATUS[3:0] in the following cases:

- When an abnormal command with a specific parameter that is not supported is stored in the Command Descriptor.
- When the IBI to be transmitted is disabled in the CSECMD register.
- After the normal command for IBI transmission is prepared in the Command Queue, when that IBI is disabled in the CSECMD register by the DISEC CCC frame from the I3C Master.

7.8.4.1.3 IBI Status Descriptor

The IBI Status Descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C Bus. The IBI Status Descriptor is read from IBI Status Queue with reads from IBI Status Queue Port.

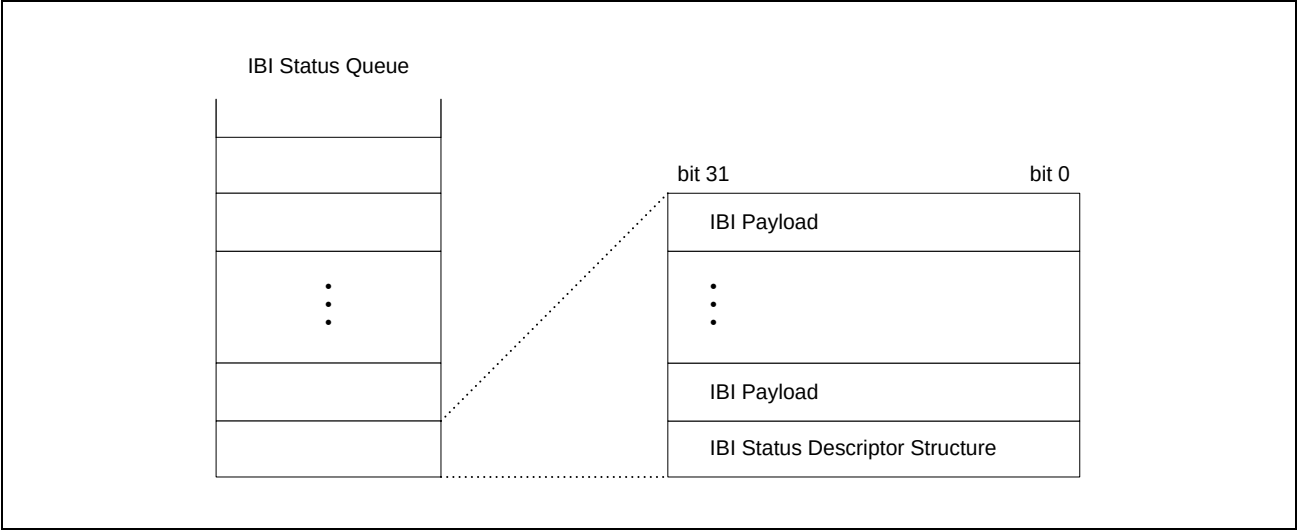


Figure 7.8-4 IBI status descriptor data structure

I3C provides a IBI Status Descriptor for the following mode:

- I3C Master Mode

Details of the IBI Status Descriptor Structure are as follows.

Bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field	IBI_ST	—	—	ERR_STATUS[2:0]			TS	LAST_S TATUS	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	IBI_ID[7:0]								DATA_LENGTH[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA_LENGTH [7:0]	IBI Data Length Number of data bytes in IBI Data.	R
15:8	IBI_ID[7:0]	IBI Received ID The meaning of this field depends on the context: For Slave Interrupt or Master Request: Bits 15:9 contain the Slave's Device Address, and bit 8 contains the R/W bit. For Hot-Join IBI: Bits 15:8 contain the Hot-Join ID for the IBI.	R
23:16	—	These bits are read as 0b. The write value should be 0b.	R
24	LAST_STATUS	Last IBI Status Last IBI status for the IBI transaction.	R

Bit	Symbol	Function	R/W
25	TS	IBI Time-stamp Present Indicates whether a time-stamp is available for the IBI. 0b: OFF: IBI is not time-stamped. 1b: ON: IBI is time-stamped.	R
28:26	ERR_STATUS[2:0]	IBI Error Status 0h: SUCCESS 3h: ERROR: FRAME (Frame Error) 4h: ERROR: ADDR_HEADER (Address Header Error) 5h: NACK: Address NACKed 7h: ERROR: ABORT (Aborted to Master) Others: Setting prohibited	R
30:29	—	These bits are read as 0b.	R
31	IBI_ST	IBI Received Status Indicates how the received IBI was handled. 0b: The IBI was handled with ACK. 1b: NACK: The IBI was handled with NACK, and then Auto-Disabled.	R

LAST_STATUS bits (Last IBI Status)

Even if LAST_STATUS is set to 0b, the software driver still evaluates the data payload length by examining the CHUNKS field.

7.8.4.1.4 Receive Status Descriptor

The Receive Status Descriptor is a read-only structure describing the success or failure of read/write operation from the master, and the amount of data transferred.

The Receive Status Descriptor is read from Receive Status Queue with reads from Receive Status Queue Port.

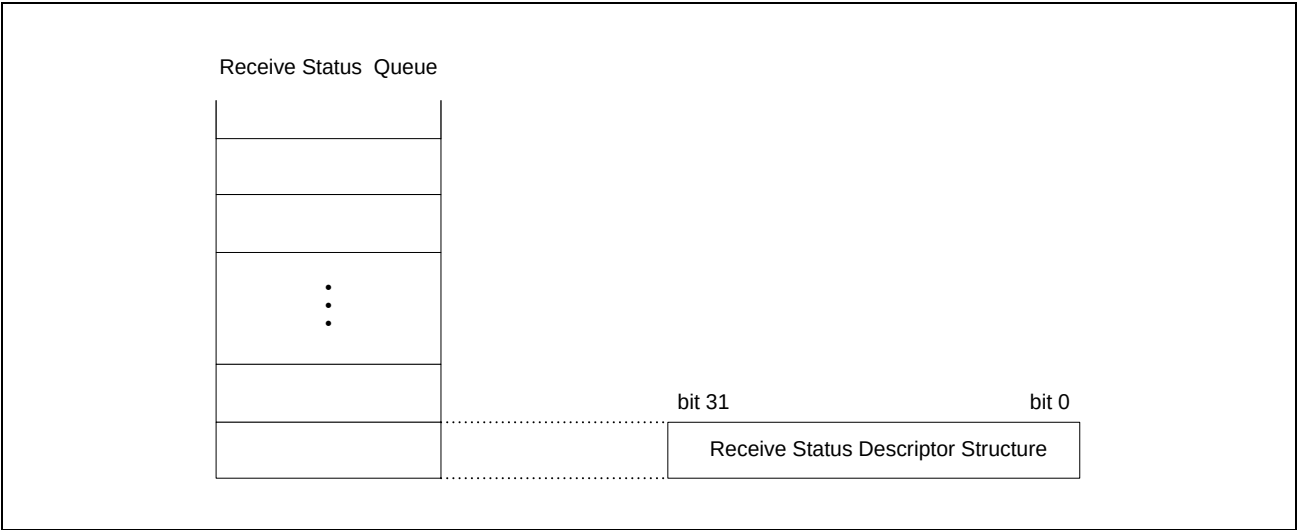


Figure 7.8-5 Receive status descriptor data structure

I3C provides a Receive Status Descriptor for the following mode:

- I3C Slave Mode

Details of the Receive Status Descriptor structure of each mode are as follows.

Bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field	DEV_INDEX[2:0]			TRANSFER_TY PE[1:0]		ERR_STATUS[2:0]			CMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	DATA_LENGTH[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length The meaning of this field depends on the context. For Write Transfer: Received data length (in bytes) For Read Transfer: Transmitted data length (in bytes)	R

Bit	Symbol	Function	R/W
23:16	CMD[7:0]	The contents are different depending on the operation mode. Details are follows: [SDR Private Message Mode] bit 23: R/W Type bit 22 to 20: Setting prohibited bit 19: I3C_I²C Type bit 18 to 16: Setting prohibited [SDR CCC Mode] CCC code[7:0]	R
26:24	ERR_STATUS[2:0]	Error Status 0h: SUCCESS 1h: ERROR: CRC (CRC Error) 2h: ERROR: PARITY (Parity Error) 3h: ERROR: FRAME (Frame Error) 4h: ERROR: ADDR_HEADER (Address Header Error) 5h: ERROR: NACK (Slave NACKed) 6h: ERROR: OVL (FIFO Overflow/Underflow) 7h: ERROR: ABORT (Aborted to Master)	R
28:27	TRANSFER_TYPE [1:0]	Transfer Type 0 0b: I3C SDR/I²C Message 0 1b: I3C CCC 1 0b: Setting prohibited 1 1b: Setting prohibited	R
31:29	DEV_INDEX[2:0]	Device Index Indicates the SVDVADn index for the response with the transfer.	R

7.8.4.2 Details of Function

7.8.4.2.1 Operation Mode

The support relationship between the mode select (I3C mode / I²C mode) and operation mode (Master / Slave) on the I3C bus or the I²C bus is shown in **Table 7.8-10**.

Table 7.8-10 Support of operating mode

I3C/I ² C Bus	I3C mode		I ² C mode	
	Master	Slave	Master	Slave
I3C Bus	✓	✓	—	✓
I ² C Bus	—	—	✓	✓

Note: ✓: Supported
—: Un-Supported

(1) Master Mode Operation

(a) I²C Master Operation

(a-1) Data Write Transfer (Single Buffer transfer)

In master transmit operation, I3C outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. **Figure 7.8-114** shows an example of usage of master transmission and **Figure 7.8-6** to **Figure 7.8-8** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

1. Initial settings. For details, see **7.8.4.3.1 Initial Setting Flow**.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1b (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag bit is automatically set to 0b, the BST.STCNDDEF flag is automatically set to 1b and the STCND bit is automatically set to 0b. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the STCND bit = 1b, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1b, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1b in response to setting of the TRMD bit to 1b.
3. Check that the NTST.TDBEF0 flag = 1b, and then write the value for transmission (the slave address and the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0b, the data are transferred from the Normal Transmit Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1b. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Because the BST.NACKDF flag being 1b at this time indicates that no slave device recognized the address or there was an error in communications, write 1b to the CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to the NTDTBP0 register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the NTDTBP0 register.

4. After confirming that the NTST.TDBEF0 flag = 1b, write the data for transmission to the NTDTBP0 register. I3C automatically holds the SCL line low until the data for transmission are ready or a STOP condition is issued.
5. After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then set the CNDCTL.SPCND bit to 1b (STOP condition issuance request). Upon receiving a STOP condition issuance request, I3C issues the STOP condition.
6. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDBEF0 and TENDF flags to 0b, and sets the BST.SPCNDDF flag to 1b.
7. After checking that the BST.SPCNDDF flag = 1b, set the BST.NACKDF and SPCNDDF flags to 0b for the next transfer operation.

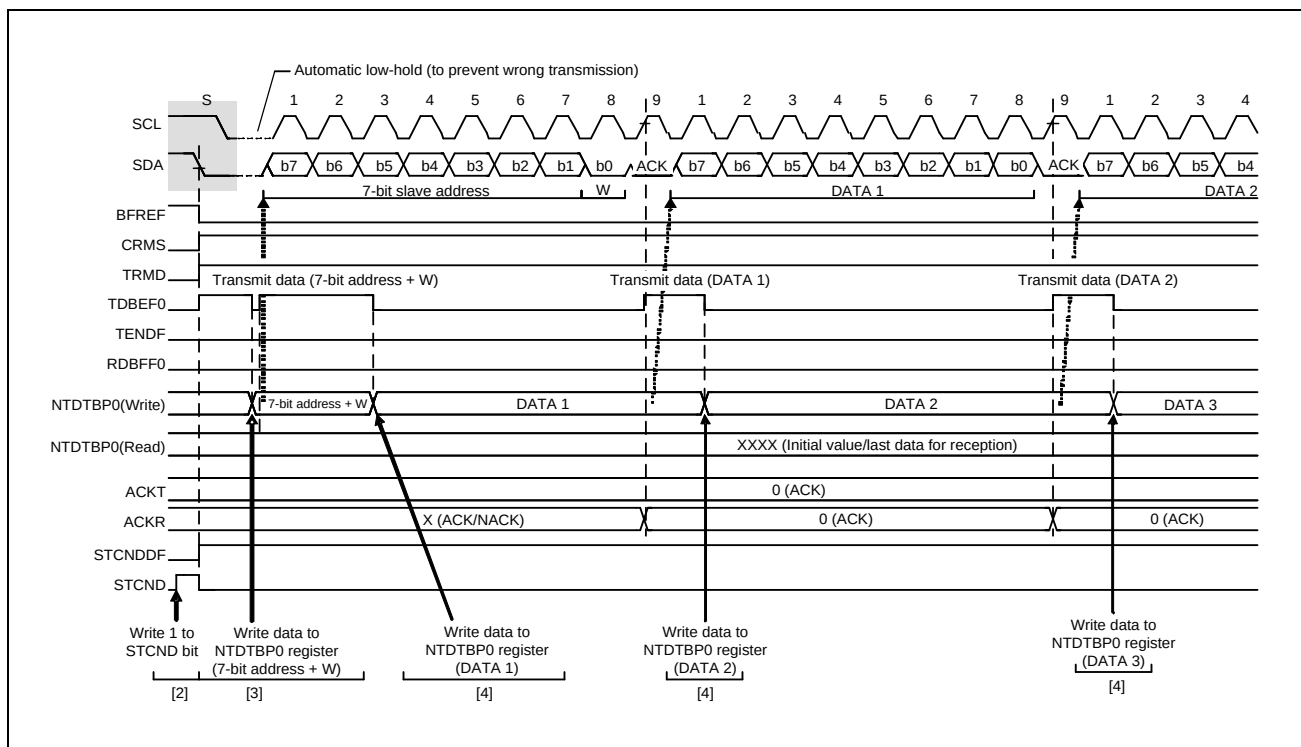


Figure 7.8-6 Master transmit operation timing (1) (7-bit address format)

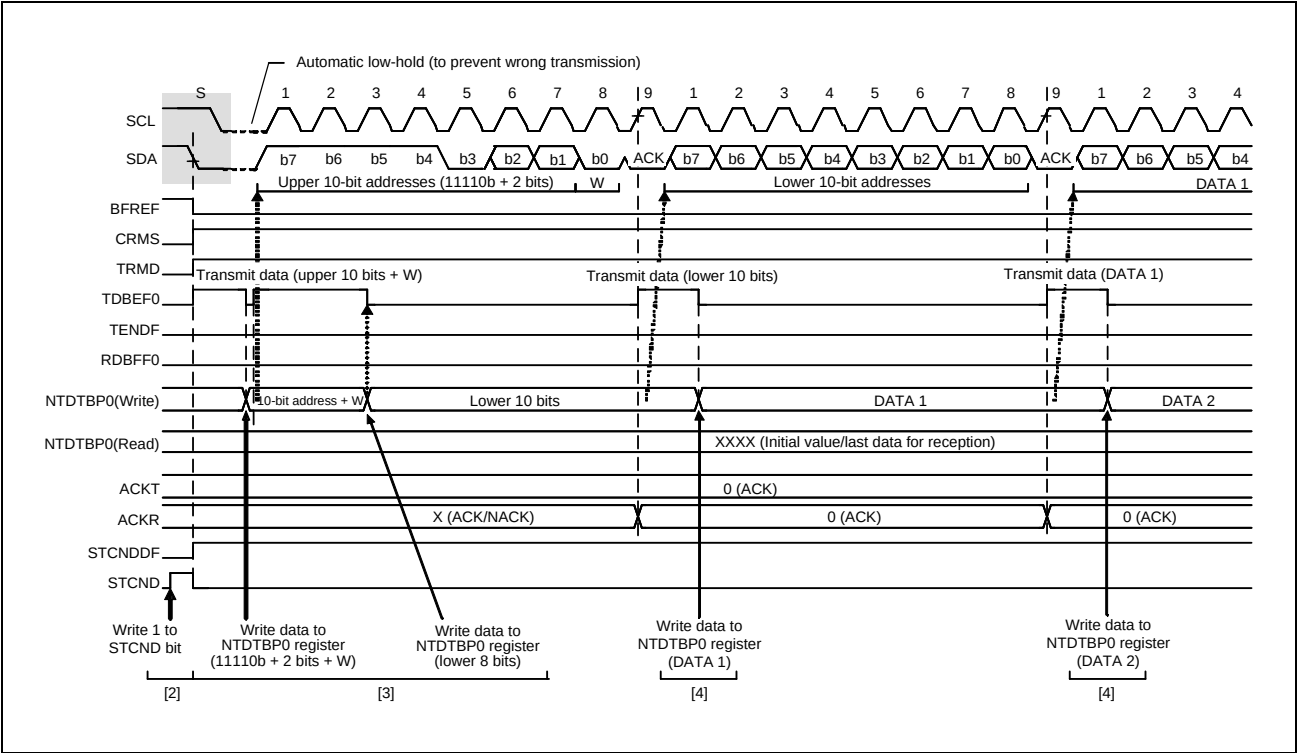


Figure 7.8-7 Master transmit operation timing (2) (10-bit address format)

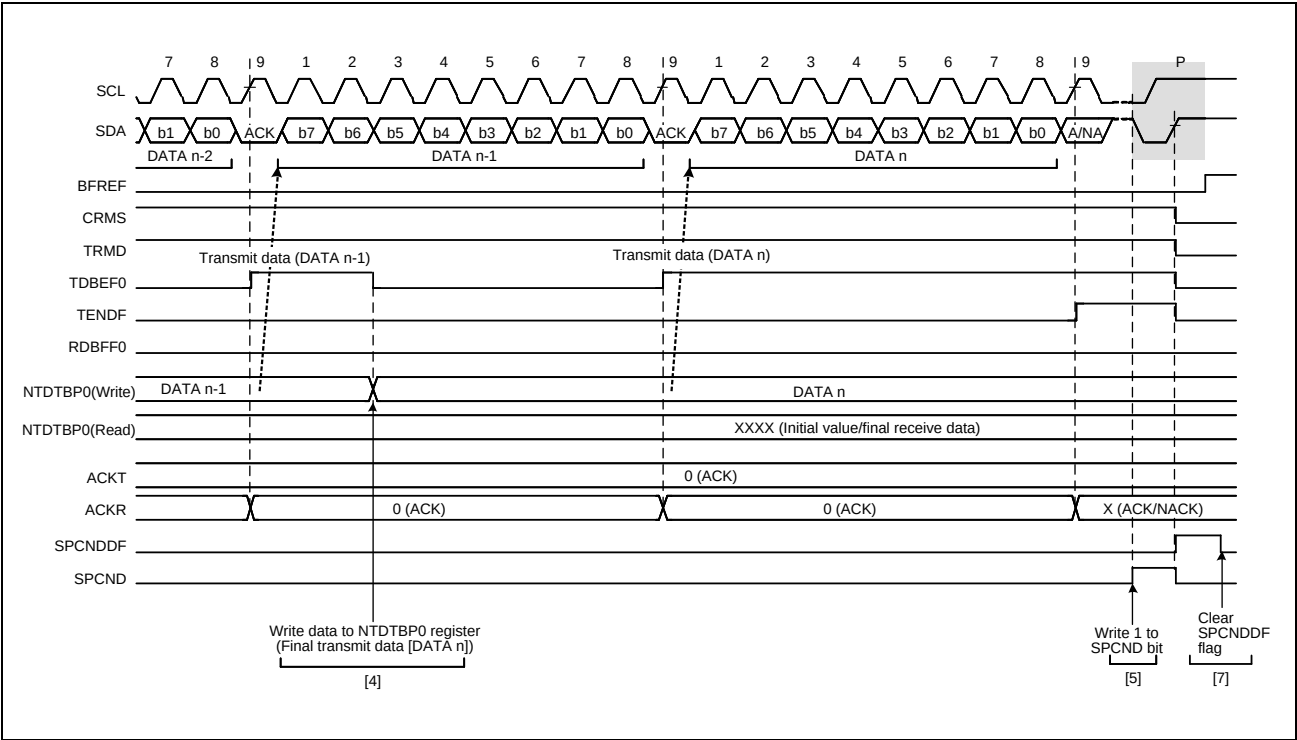


Figure 7.8-8 Master transmit operation timing (3)

(a-2) Data Read Transfer (Single Buffer transfer)

In master receive operation, I3C as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because I3C must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 7.8-115 and **Figure 7.8-116** show examples of usage of master reception (7-bit address format) and **Figure 7.8-9** to **Figure 7.8-11** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

1. Initial settings. For details, see **7.8.4.3.1 Initial Setting Flow**.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1b (START condition issuance request). Upon receiving the request, I3C issues a START condition. When I3C detects the START condition, the BFREF flag is automatically set to 0b and the BST.STCNDDF flag is automatically set to 1b and the STCND bit is automatically set to 0b. At this time, if the START condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the STCND bit = 1b, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1b, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1b in response to setting of the TRMD bit to 1b.
3. Check that the NTST.TDBEF0 flag = 1b, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0b, the data are transferred from the Normal Transmit Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1b. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the PRSST.TRMD bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRMD bit is set to 0b on the rising edge of the ninth cycle of SCL clock, placing I3C in master receive mode. At this time, the TDBEF0 flag is set to 0b. The NTST.RDBFF0 flag is automatically set to 1b when ACK response is received from the slave device. If the slave device is not recognized or a communication failure occurs, the BST.NACKDF flag will be set to 1b. At this time, set 1b to the CNDCTL.SPCND bit to issue a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a Repeated START condition. After that, transmitting 1111 0, the two higher-order bits of the slave address, and the R bit places I3C in master receive mode.
4. Dummy read the NTDTBP0 register after confirming that the NTST.RDBFF0 flag = 1b; this makes I3C start output of the SCL clock and start data reception.
5. After 1 byte of data has been received, the NTST.RDBFF0 flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the SCSTRCTL.ACKTWE bit. Reading the NTDTBP0 register at this time will produce the received data, and the RDBFF0 flag is automatically set to 0b at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ACKCTL.ACKT bit. Furthermore, if the next byte to be received is the next to last byte, set the SCSTRCTL.RWE bit to 1b (for wait insertion) before reading the NTDTBP0 register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ACKCTL.ACKT bit to 1b (NACK) in step 6, due to other interrupts, etc., this fixes the SCL line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a STOP condition is possible.
6. When the SCSTRCTL.ACKTWE bit = 0b and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ACKCTL.ACKT bit to 1b (NACK).

7. After reading the byte before last from the NTDTBP0 register, if the value of the NTST.RDBFF0 flag is confirmed to be 1b, write 1 to the CNDCTL.SPCND bit (STOP condition issuance request) and then read the last byte from the NTDTBP0 register. When 1b is written to the CNDCTL.SPCND bit, I3C is released from the wait state and issues the STOP condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low- hold state.
8. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00b and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the BST.SPCNDDF flag to 1b.
9. After checking that the BST.SPCNDDF flag = 1b, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

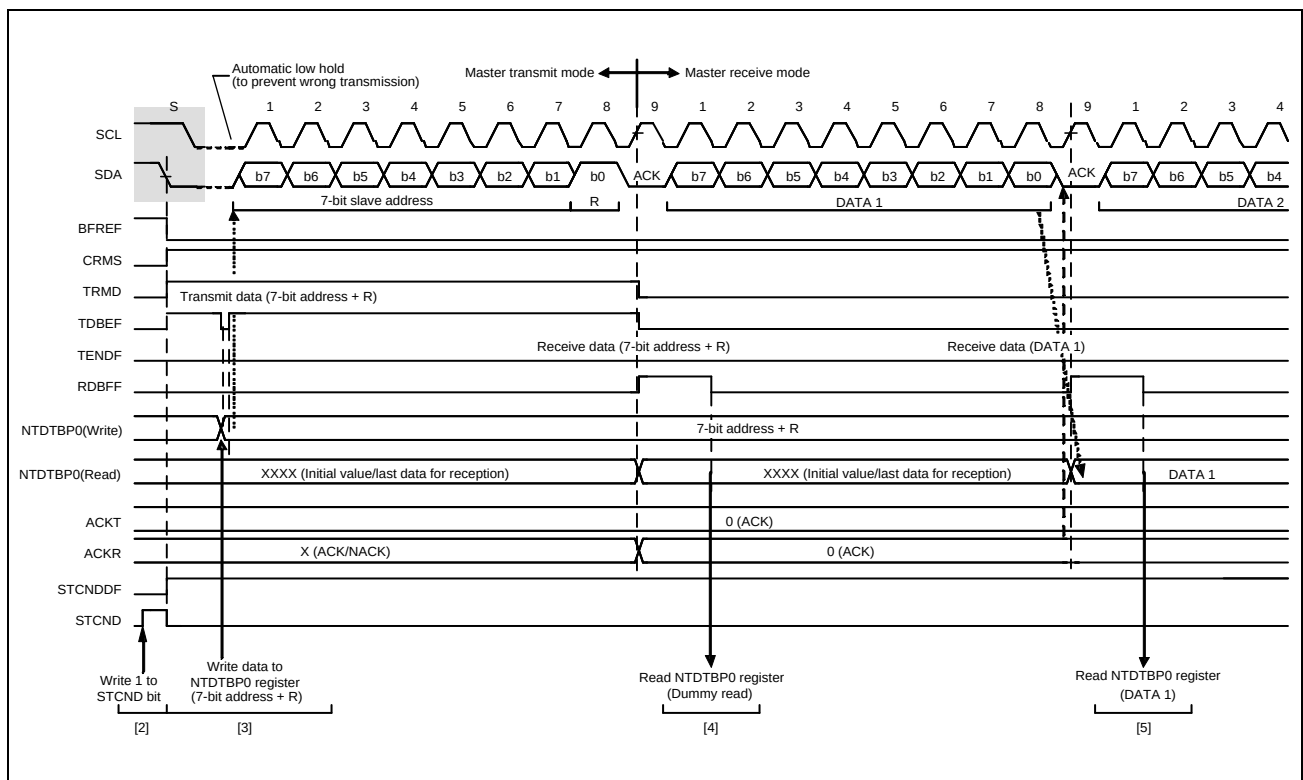


Figure 7.8-9 Master receive operation timing (1) (7-bit address format, when $ACKTWE = 0$)

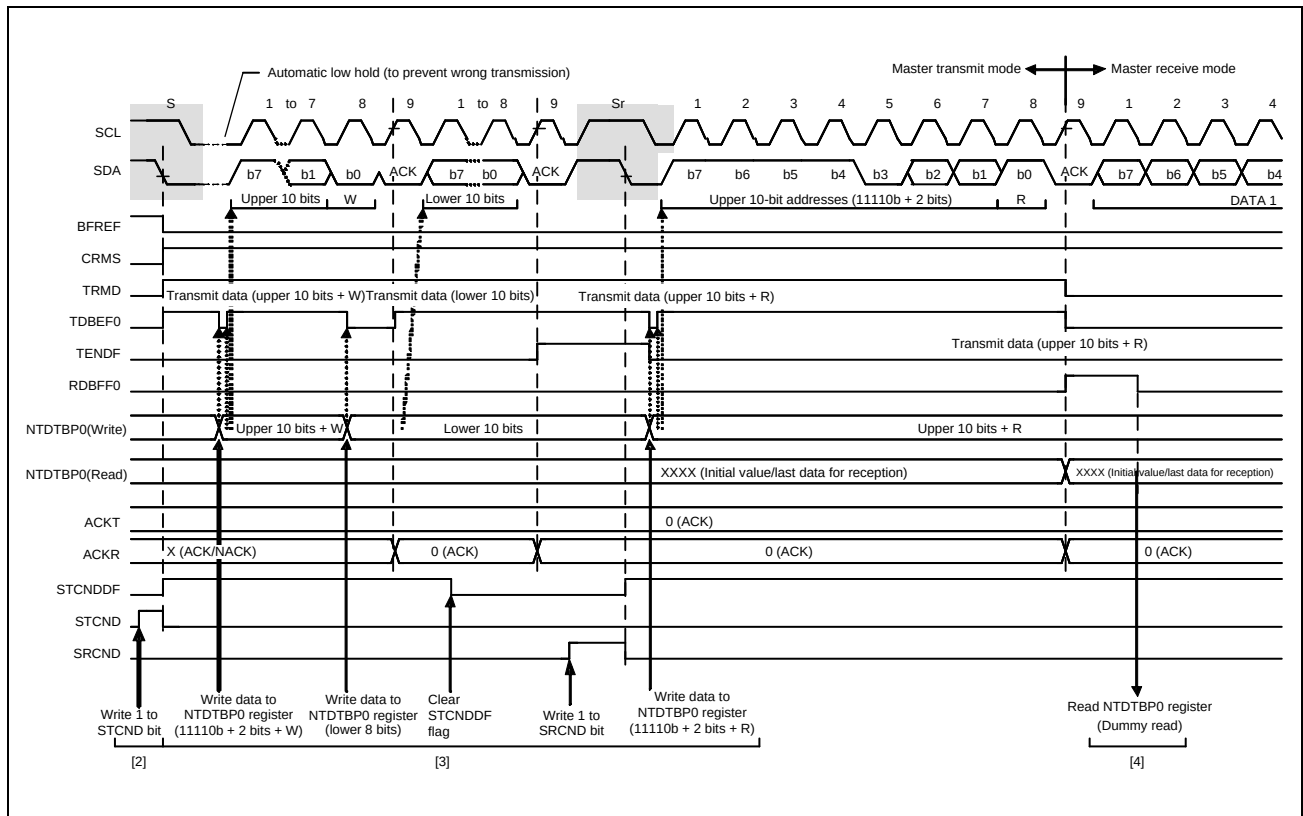


Figure 7.8-10 Master receive operation timing (2) (10-bit address format, when ACKTWE = 0)

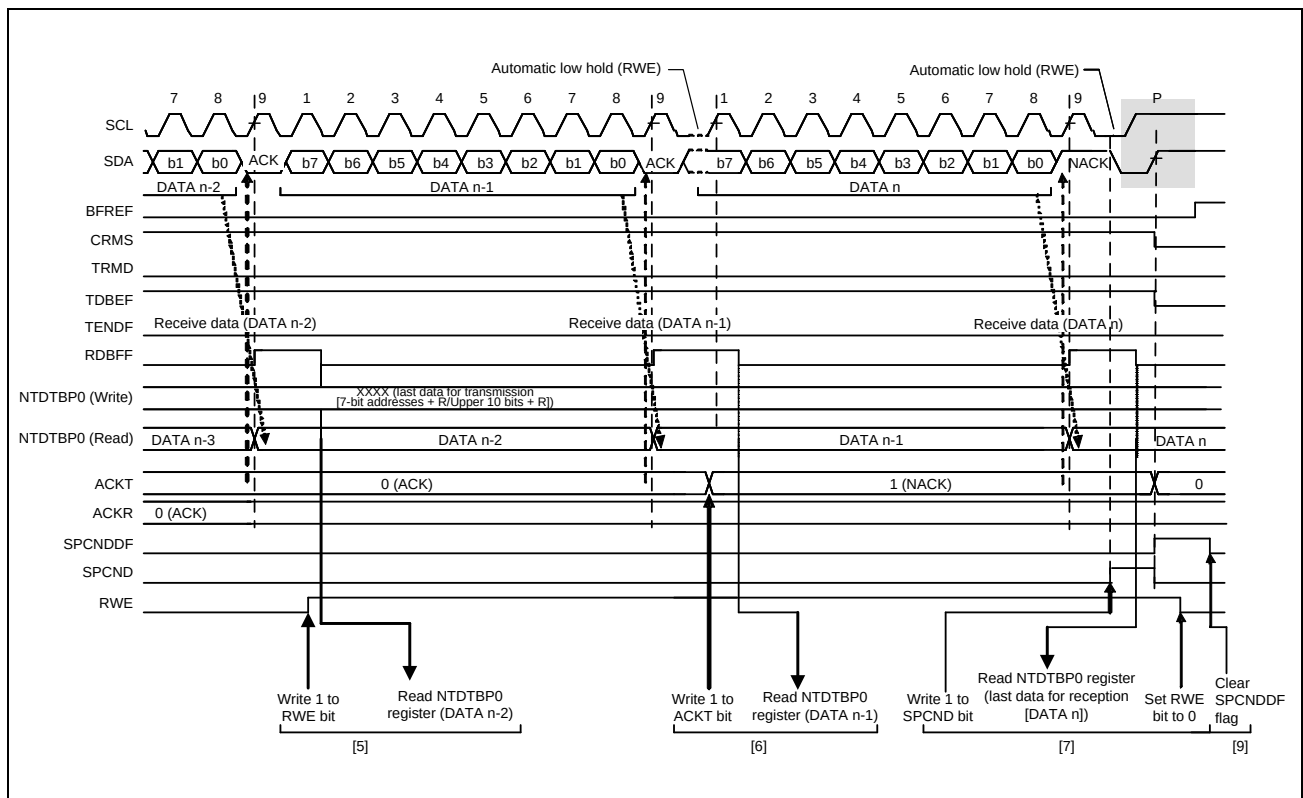


Figure 7.8-11 Master receive operation timing (3) (when ACKTWE = 0)

(b) I3C Master Operation

(b-1) Dynamic Address Assign Procedure

After initializing I3C, first execute Dynamic Address Assign Procedure for I3C Slave connected on the I3C Bus. The following describes the procedure.

1. Initial setting (see **(2) I3C Initial Setting Flow** for details)
2. Execute Dynamic Address Assign with ENTDAAs or SETDASA Common Command Code (CCC) for I3C Slave set in DAT (DATBASm register).
3. Write Command Descriptor (Address Assign Command) to Command Buffer via the NCMDQP register. When Command Descriptor is written to Command Buffer, Transaction is issued on I3C Bus.
4. When ENTDAAs is specified for CMD[7:0] of Address Assign Command:
Execute Dynamic Address Assign for I3C Slave for the number of DATs specified by DEV_COUNT[3:0] starting with DAT specified by DEV_INDEX[4:0] of Address Assign Command.
When SETDASA is specified for CMD[7:0] of Address Assign Command:
Execute Dynamic Address Assign for I3C Slave indicated by DAT specified by DEV_INDEX[4:0] of Address Assign Command.
5. In case of ENTDAAs, the Provisional ID, BCR, DCR transmitted from I3C Slave is stored in Receive Data Buffer (BCR is also automatically stored in the MSDCTm register).
Read the Provisional ID, BCR, and DCR from the Receive Data Buffer via the NTDTBP0 register with an interrupt by RDBFF0 = 1b.
6. When execution of Dynamic Address Assign is completed, issue STOP condition and store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor via the NRSPQP register and check the status.
8. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the value of DEV_COUNT[3:0] of the Address Assign Command.

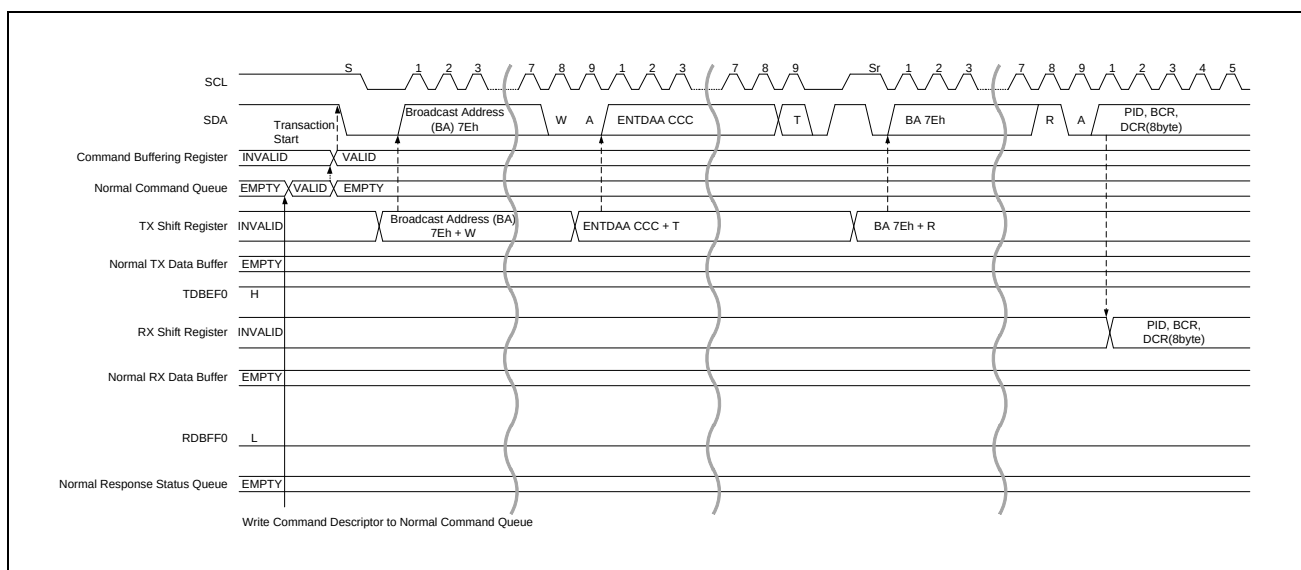


Figure 7.8-12 Dynamic address assign procedure (ENTDAA CCC) timing (1/3)

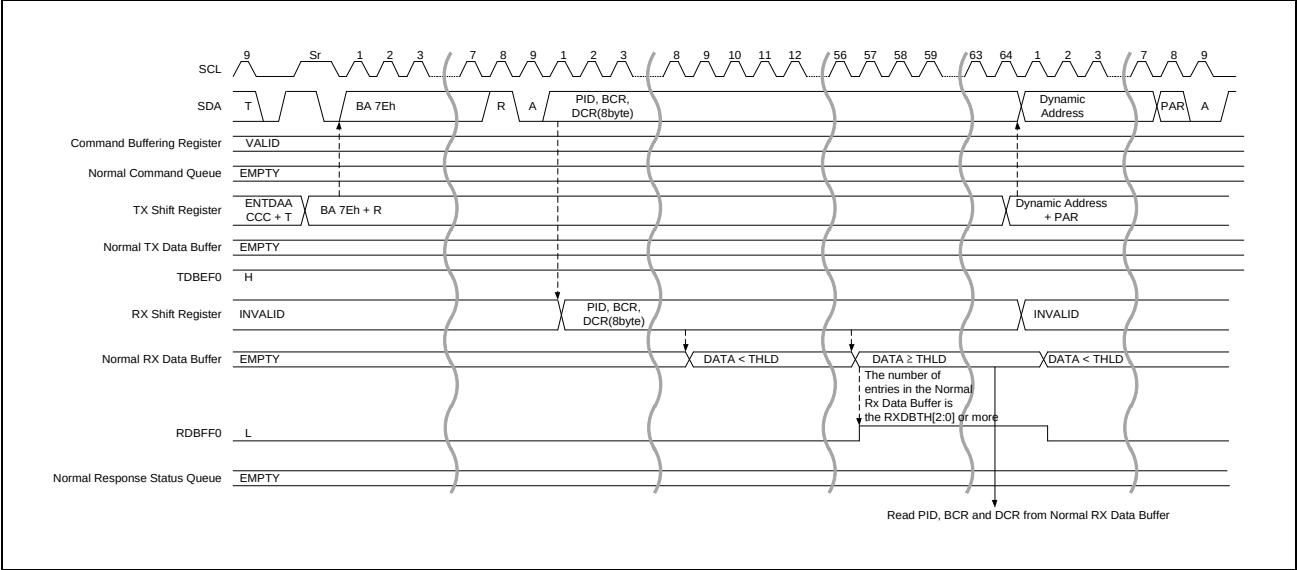


Figure 7.8-13 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

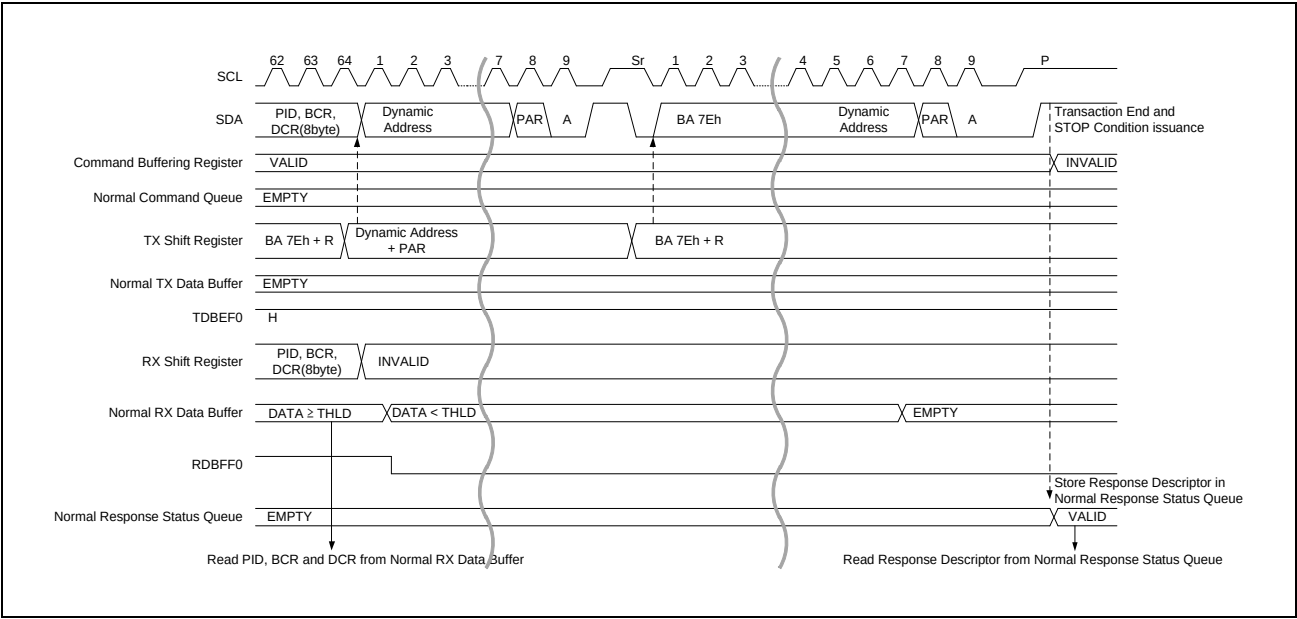


Figure 7.8-14 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

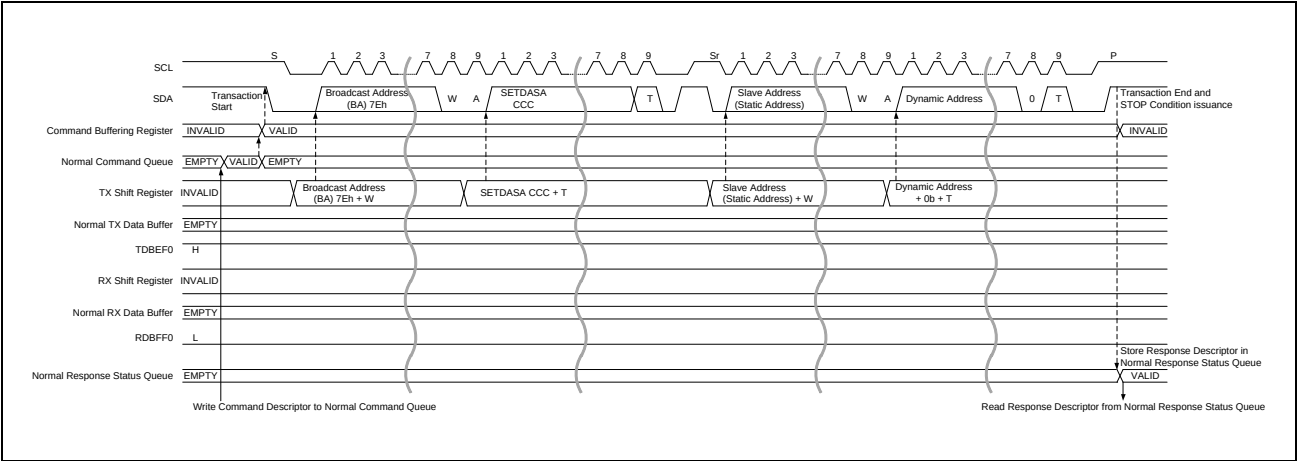


Figure 7.8-15 Dynamic address assign procedure (SETDASA CCC) timing

(b-2) SDR Data Write Transfer

1. Write data for transmission to the Transmit Data Buffer via the NTDTBP0 register.
2. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK).
4. If data for transmission still remain, write data for transmission by an interrupt with TDBEF0 = 1b to the Transmit Data Buffer via the NTDTBP0 register.
5. When data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, the Repeated START condition or STOP condition is issued and the Response Descriptor is stored in the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check that the value of the DATA_LENGTH[15:0] bits of the Response Descriptor is 0.

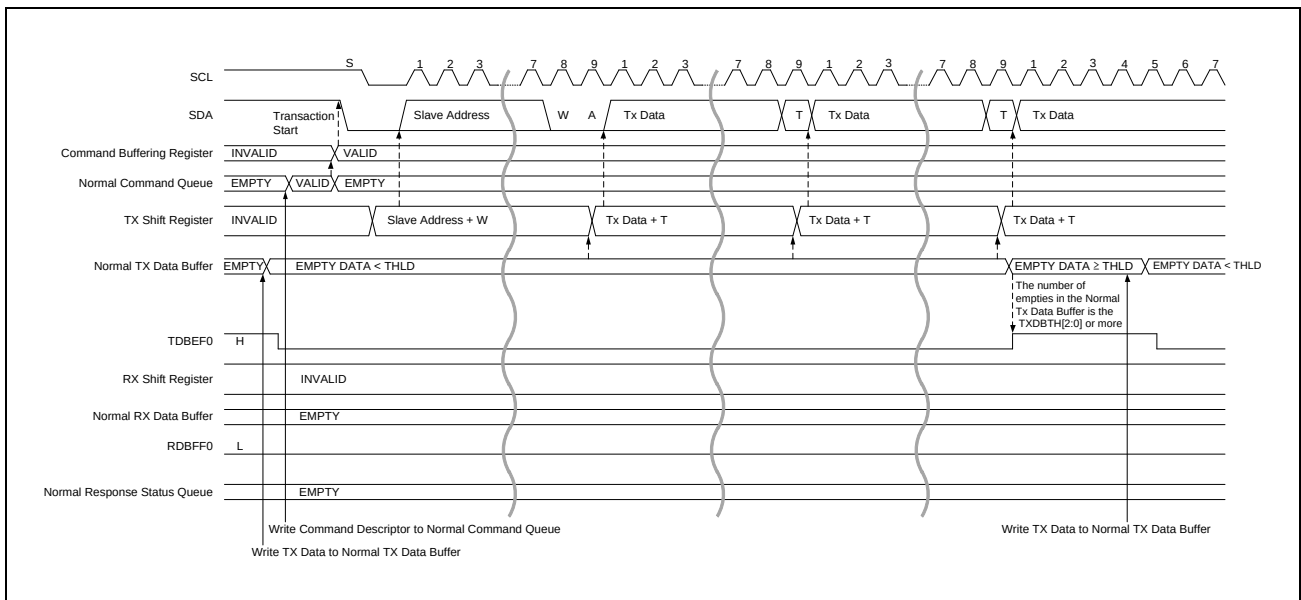


Figure 7.8-16 SDR data write transfer timing (1/2)

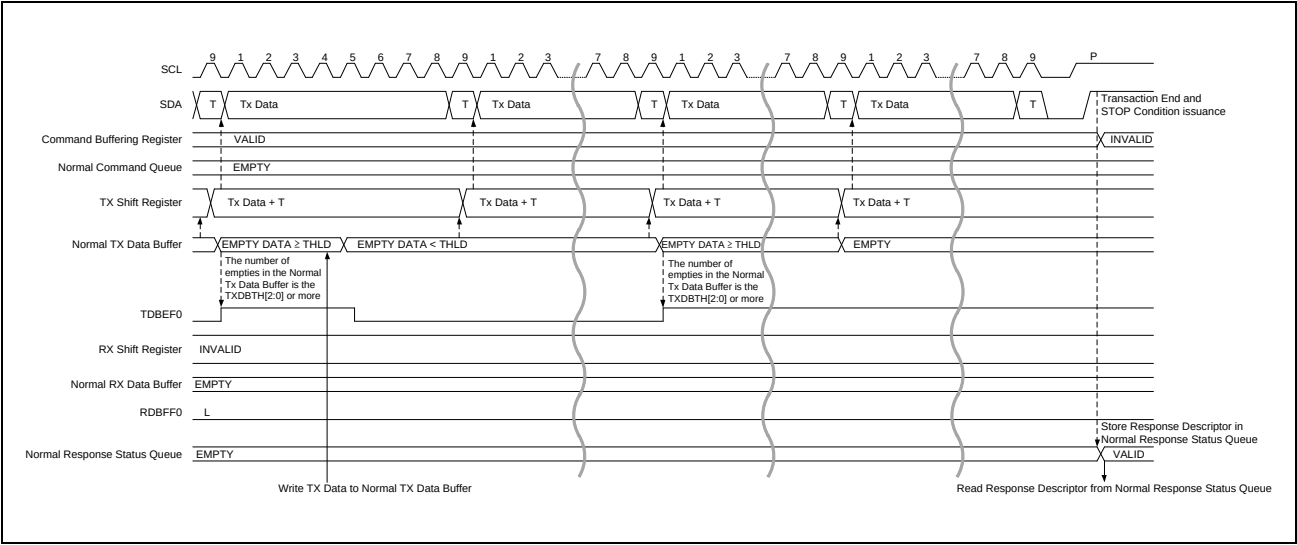


Figure 7.8-17 SDR data write transfer timing (2/2)

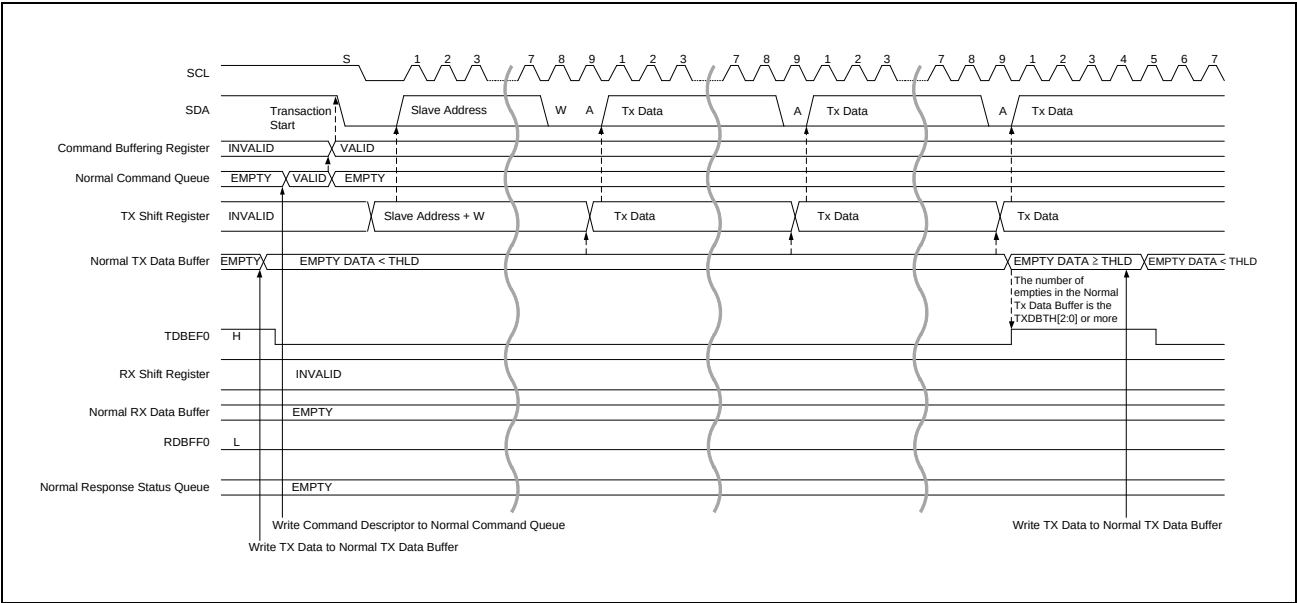


Figure 7.8-18 Legacy I²C message data write timing (1/2)

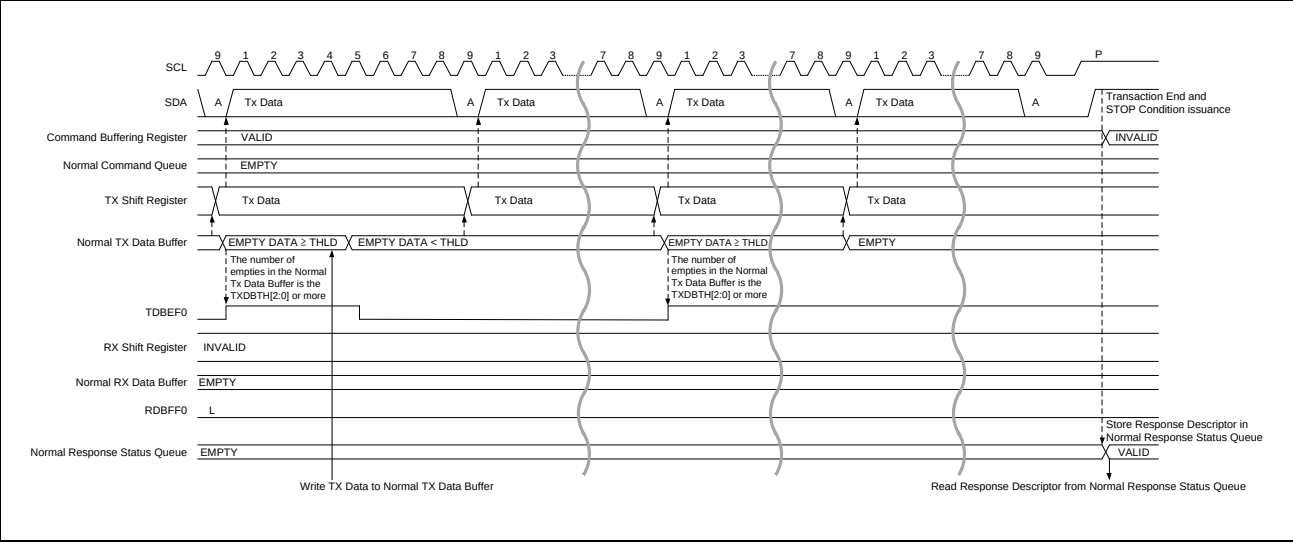


Figure 7.8-19 Legacy I²C message data write timing (2/2)

(b-3) SDR Data Read Transfer

1. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
2. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK).
3. Data received from the I3C Slave is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1b interrupt, the received data is read from the Receive Data Buffer via the NTDTBP0 register.
5. SDR:
Detecting Low in T-bit or receiving Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, issue Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
Legacy I²C Message:
When data reception for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, NACK is issued. After that, issue a Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the data length setting value of the Command Descriptor.

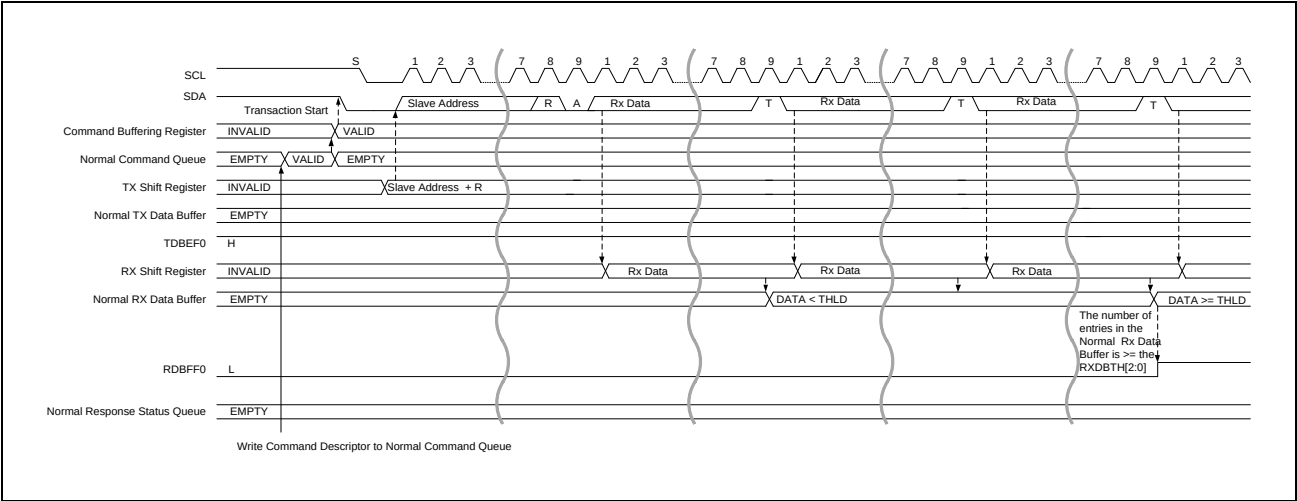


Figure 7.8-20 SDR data read transfer timing (1/2)

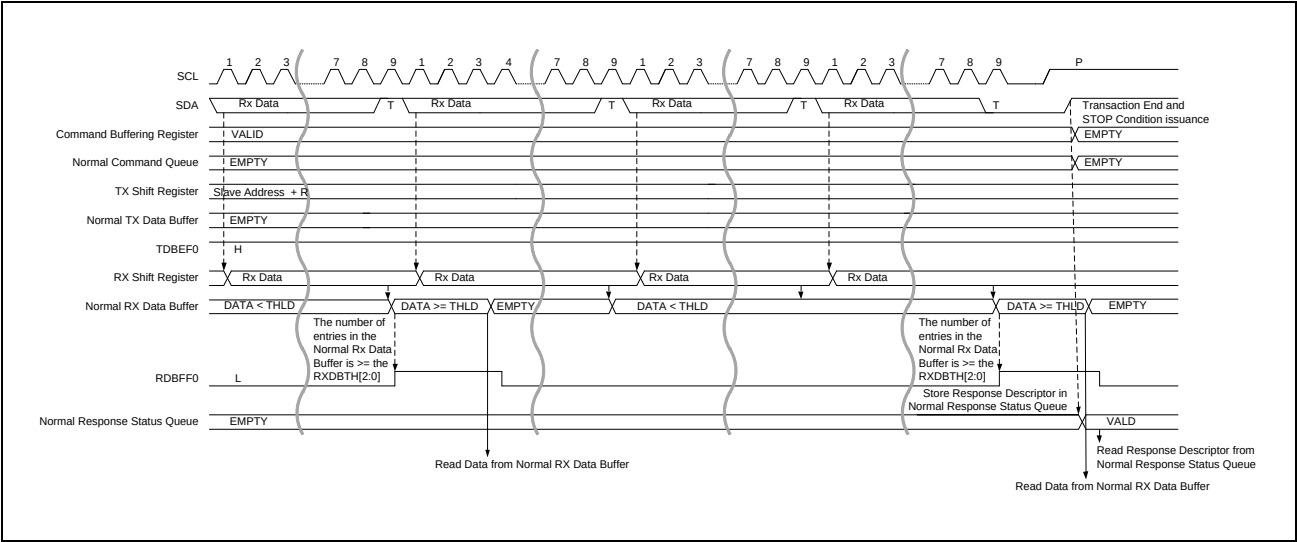


Figure 7.8-21 SDR data read transfer timing (2/2)

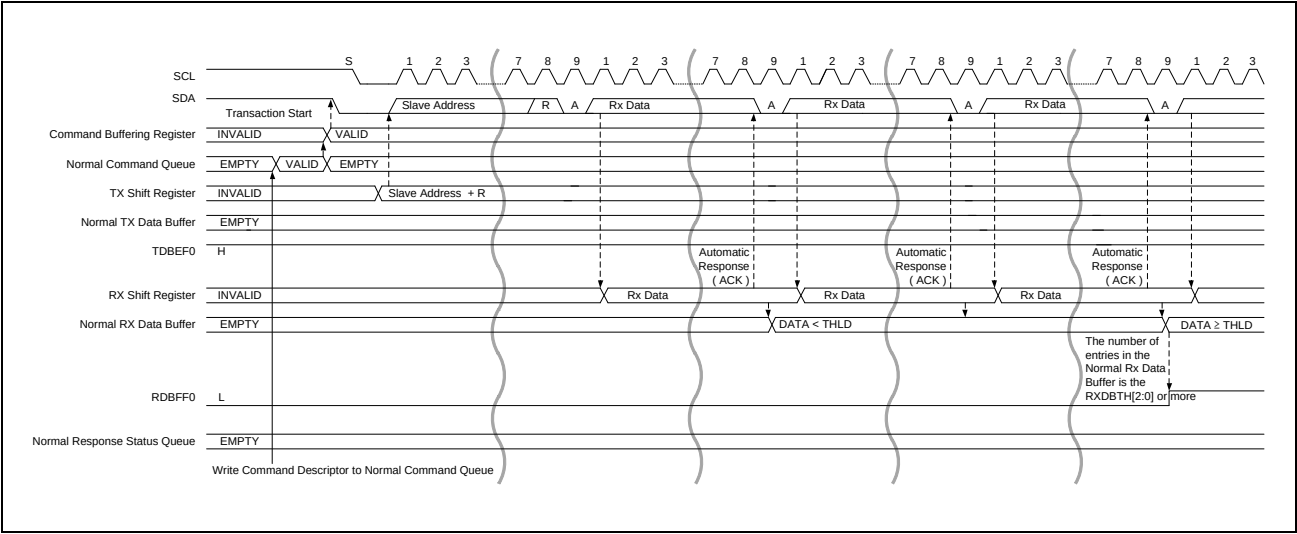


Figure 7.8-22 Legacy I2C message data read transfer timing (1/2)

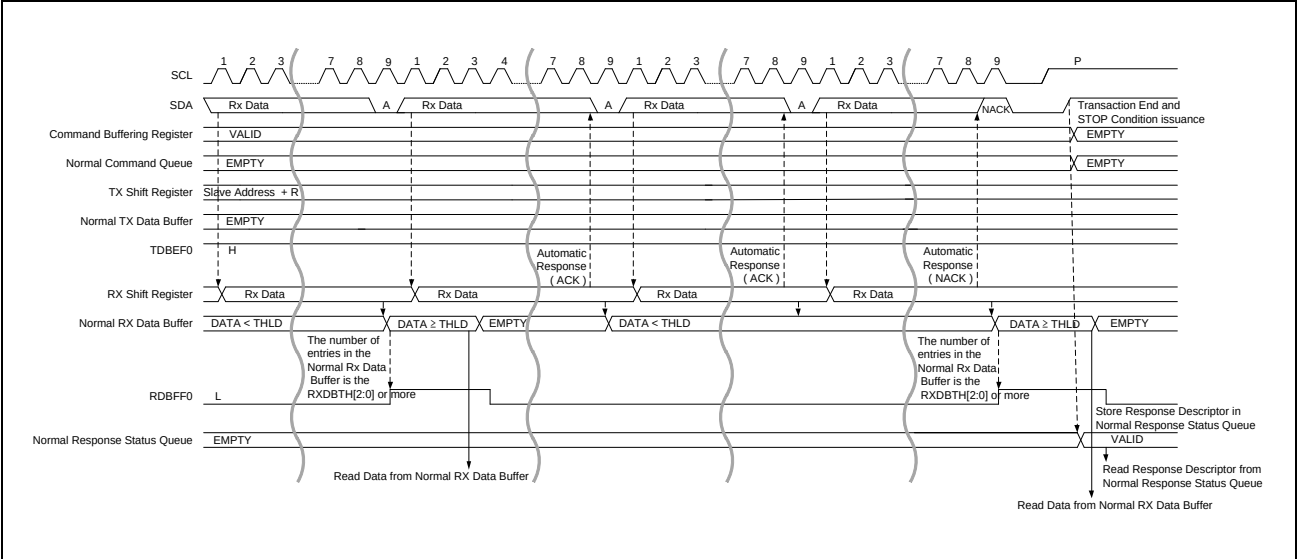


Figure 7.8-23 Legacy I2C message data read transfer timing (2/2)

(b-4) IBI Transfer

1. Write Command Descriptor to the Command Buffer and issue Transaction on I3C Bus.
If START Request (SDA Low Drive) is issued from the slave device, I3C drives SCL to Low and completes START condition.
Thereafter, the SCL is supplied and In-Band Interrupt Request is received.
2. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing In-Band Interrupt from I3C Slave, stop issuing Transaction.
3. According to **(8) In-Band Interrupt [I3C mode]**, detect In-Band Interrupt and process.
4. In the interrupt with IBIQEFF = 1b, read the IBI Status Descriptor from the IBI Status Buffer via the NIBIQP register and check the status.
When detected a Slave Interrupt Request and responded with ACK, Read the IBI Data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the IBI Status Descriptor from the IBI Data Buffer via the NIBIQP register.
5. Restart issuing Transaction of Command of Step1.

An example of the processing procedure after detection of In-Band Interrupt is shown below.

Processing procedure for detecting Mastership Request and transferring master right to Secondary Master

1. If the I3C Secondary Master wins the Arbitration, issue a DEFSLVS CCC and notify Slave information to Secondary Master.
2. Issue a GETACCMST CCC and complete CCC by a STOP condition.

Remark After transferring master right to Secondary Master, to get master right again, issue a Mastership Request according to (f) IBI Transfer of **(b) I3C Slave Operation**.
After Mastership Request is accepted by the Current Master, to get master right again at receiving the GETACCMST CCC and complete CCC by a STOP condition.

Processing procedure when Hot-Join Event is detected

1. Issue a Broadcast Command Code Enter Dynamic Address Assignment (ENTDAA) to start the Dynamic Address Assignment process.
2. Issue a DEFSLVS CCC and notify Slave information to Secondary Master.

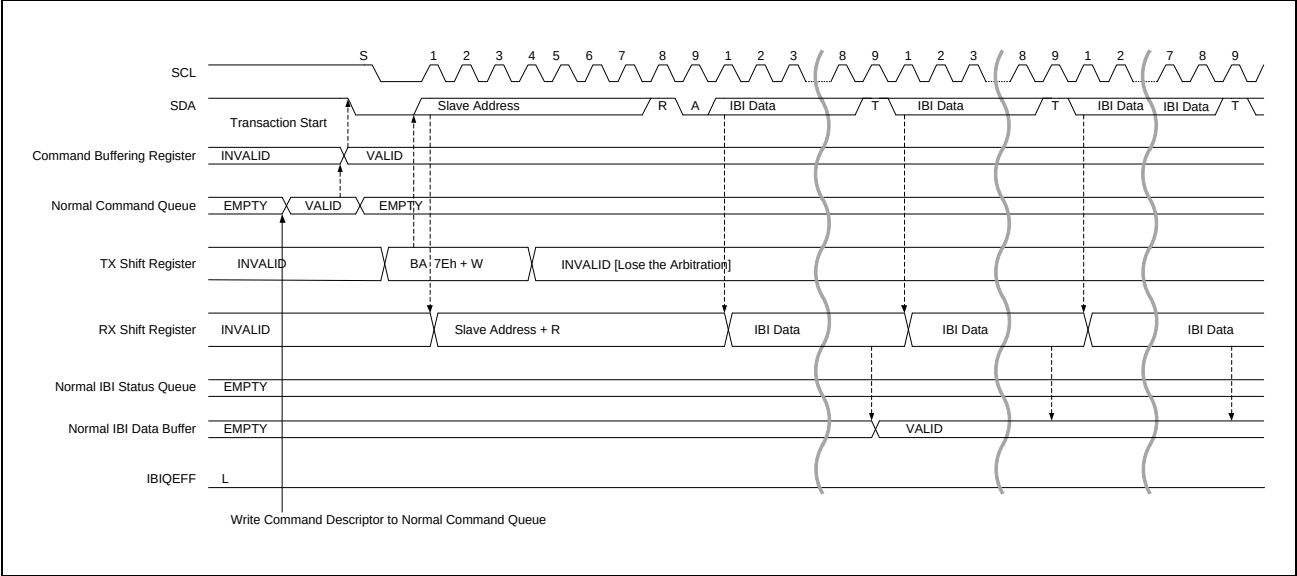


Figure 7.8-24 I3C master IBI transfer timing (1/2)

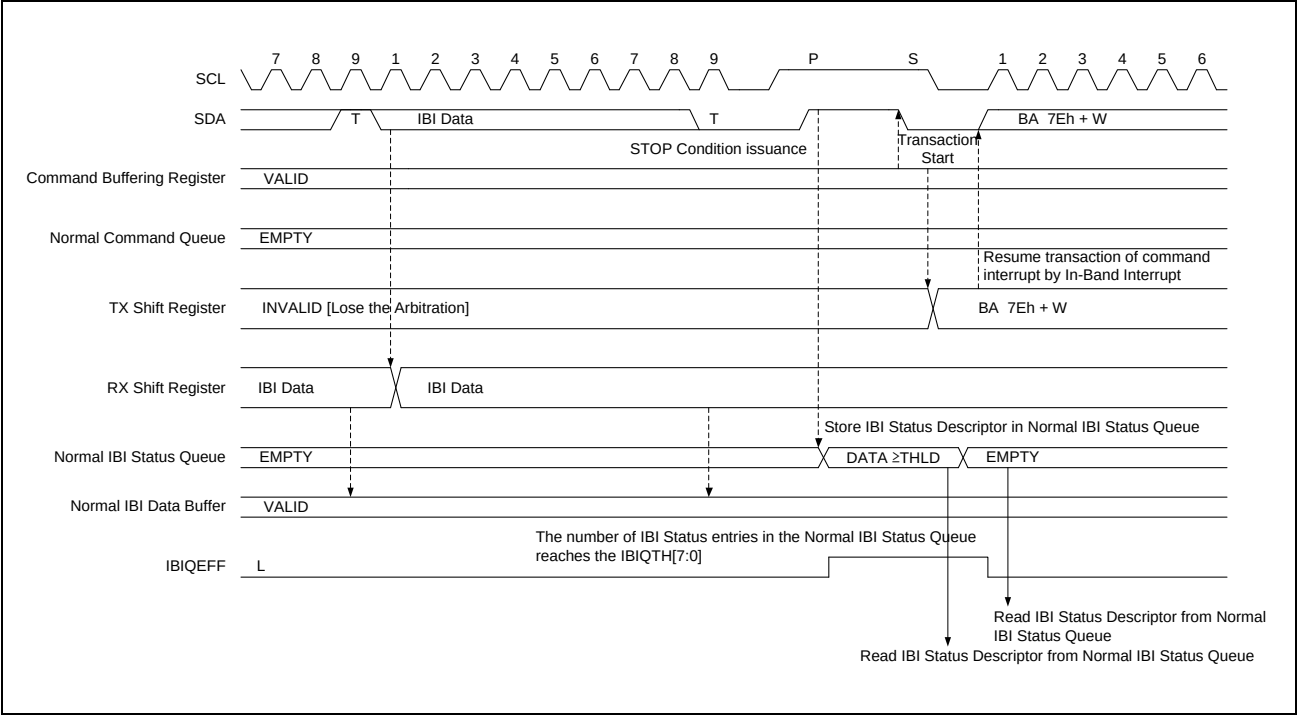


Figure 7.8-25 I3C master IBI transfer timing (2/2)

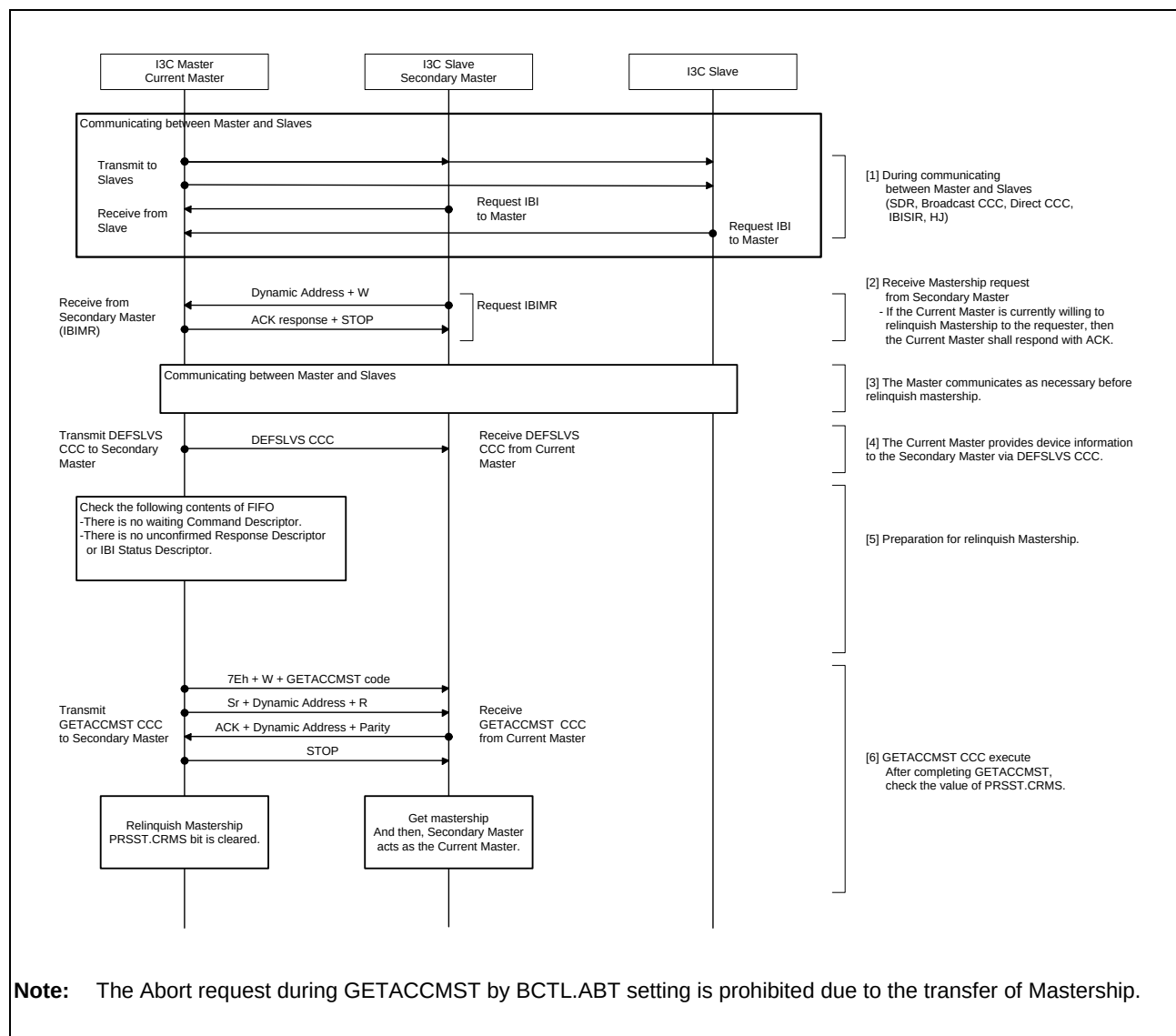


Figure 7.8-26 I3C Master Mastership processing flow

(2) Slave Mode Operation

(a) I²C Slave Operation

(a-1) Data Write Transfer (Single Buffer transfer)

In slave receive operation, the master device outputs the SCL clock and transmit data, and I3C returns acknowledgments as a slave device.

Figure 7.8-122 shows an example of usage of slave reception and **Figure 7.8-27** and **Figure 7.8-28** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

1. Initial settings. For details, see **7.8.4.3.1 Initial Setting Flow**. After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAFn (n = 0 to 2) to 1b on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0b, I3C continues to place itself in slave receive mode and sets the NTST.RDBFF0 flag to 1b.
3. After the BST.SPCNDDF flag is confirmed to be 0b and the NTST.RDBFF0 flag to be 1b, dummy read the NTDTBP0 register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).
4. When the NTDTBP0 register is read, I3C automatically sets the NTST.RDBFF0 flag to 0b. If reading of the NTDTBP0 register is delayed and a next byte is received while the RDBFF0 flag is still set to 1b, I3C holds the SCL line low from one SCL cycle before the timing with which RDBFF0 should be set. In this case, reading the NTDTBP0 register releases the SCL line from being held at the low level. When the BST.SPCNDDF flag = 1b and the NTST.RDBFF0 flag is also 1b, read the NTDTBP0 register until all the data is completely received.
5. Upon detecting the STOP condition, I3C automatically clears bits SVST.HOAF, GCAF, and SVAFn (n = 0 to 2) to 0b.
6. After checking that the BST.SPCNDDF flag = 1b, set the BST.SPCNDDF flag to 0b for the next transfer operation.

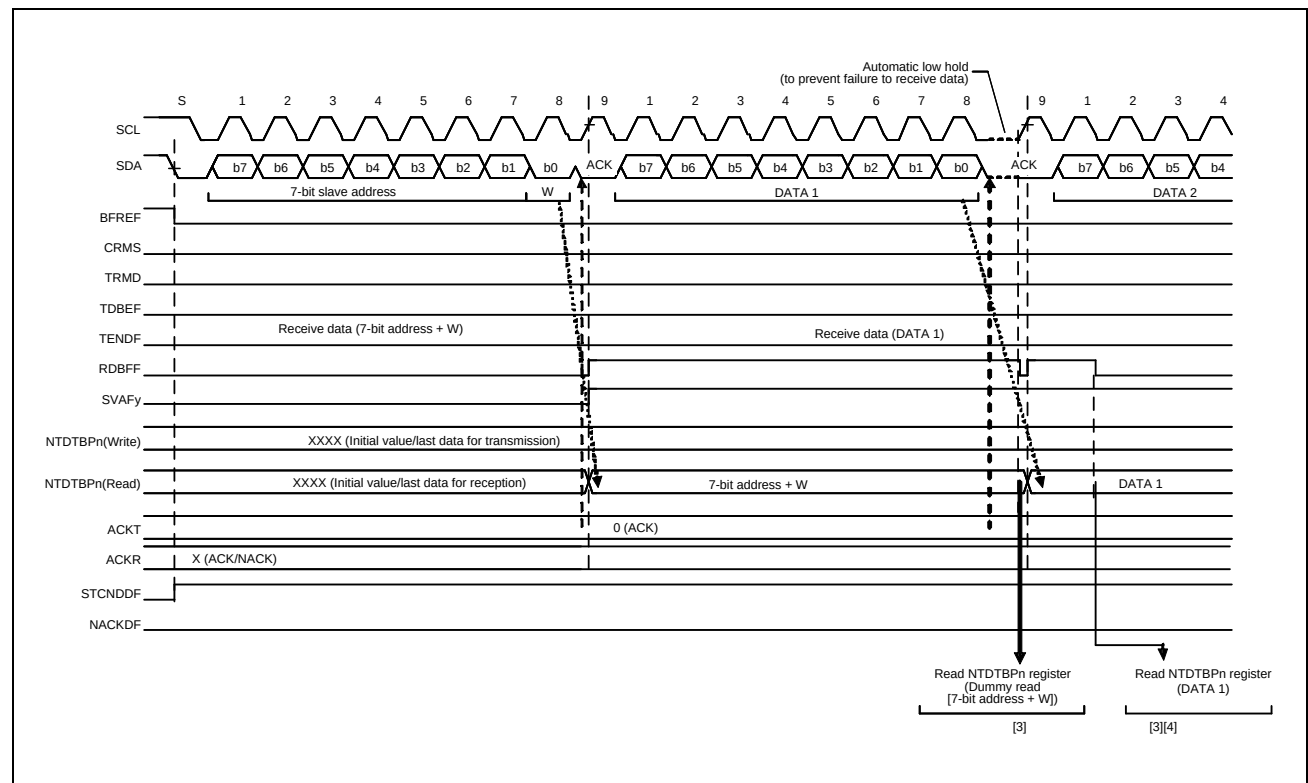


Figure 7.8-27 Slave receive operation timing (1) (7-bit address format, when ACKTWE = 0)

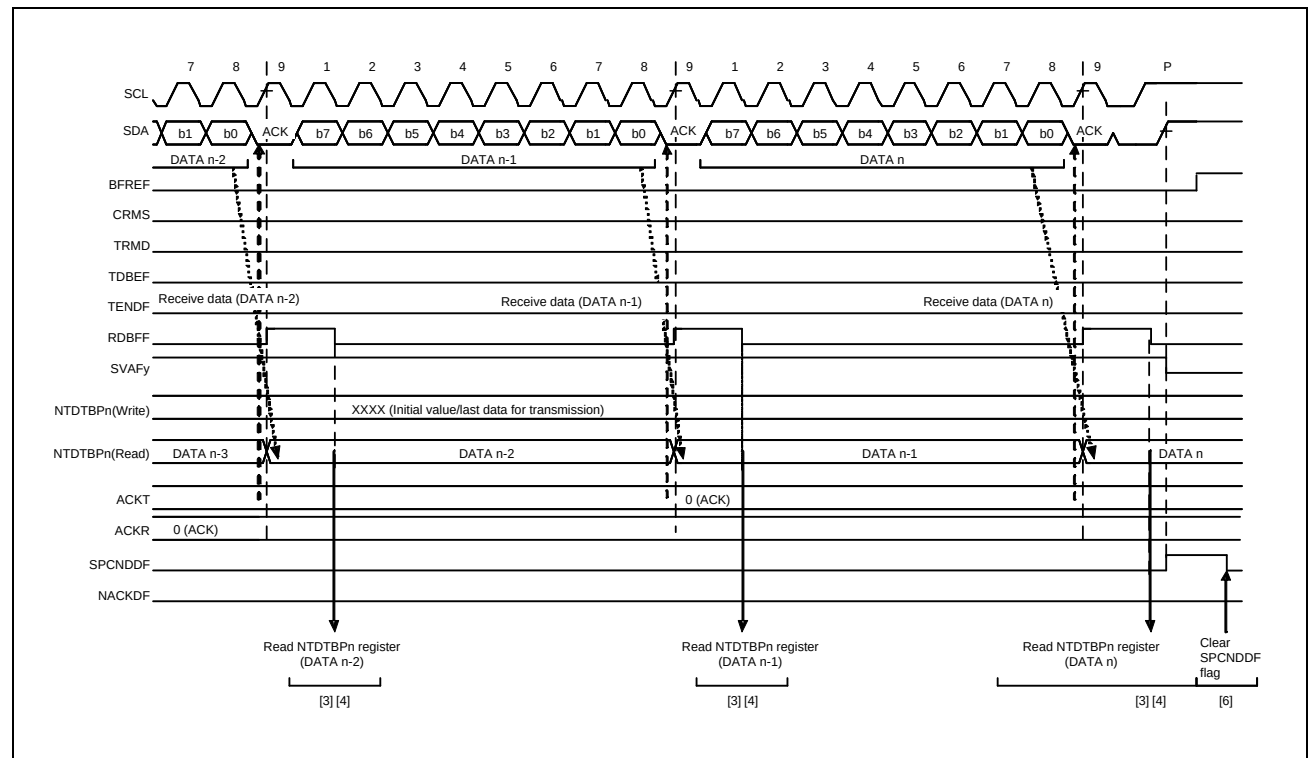


Figure 7.8-28 Slave receive operation timing (2) (when ACKTWE = 0)

(a-2) Data Read Transfer (Single Buffer transfer)

In slave transmit operation, the master device outputs the SCL clock, I3C transmits data as a slave device, and the master device returns acknowledgments.

Figure 7.8-121 shows an example of usage of slave transmission and **Figure 7.8-29** and **Figure 7.8-30** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

1. Initial settings. For details, see **7.8.4.3.1 Initial Setting Flow**.
After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAFn ($n = 0$ to 2) to 1b on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1b, I3C automatically places itself in slave transmit mode by setting both the PRSST.TRMD bit and the NTST.TDBEF0 flag to 1b.
3. After the NTST.TDBEF0 flag is confirmed to be 1b, write the data for transmission to the NTDTBP0 register. At this time, if I3C does not receive acknowledge from the master device (receives a NACK signal) while the BSTE.NACKDE bit = 1b, I3C aborts transfer of the next data.
4. Wait until the following (a) or (b) condition.
(a) The BST.NACKDF flag is set to 1b.
(b) The BST.TENDF flag is set to 1b while the NTST.TDBEF0 flag = 1b, after the last byte for transmission is written to the NTDTBP0 register.
5. When the BST.NACKDF flag or the BST.TENDF flag = 1b, dummy read the NTDTBP0 register to complete the processing.
6. Upon detecting the STOP condition, I3C automatically sets bits SVST.HOAF, GCAF, and SVAFn ($n = 0$ to 2), flags NTST.TDBEF0 and BST.TENDF, and the PRSST.TRMD bit to 0b, and enters slave receive mode.
7. After checking that the BST.SPCNDDF flag = 1b, set the BST.NACKDF and SPCNDDF flags to 0b for the next transfer operation.

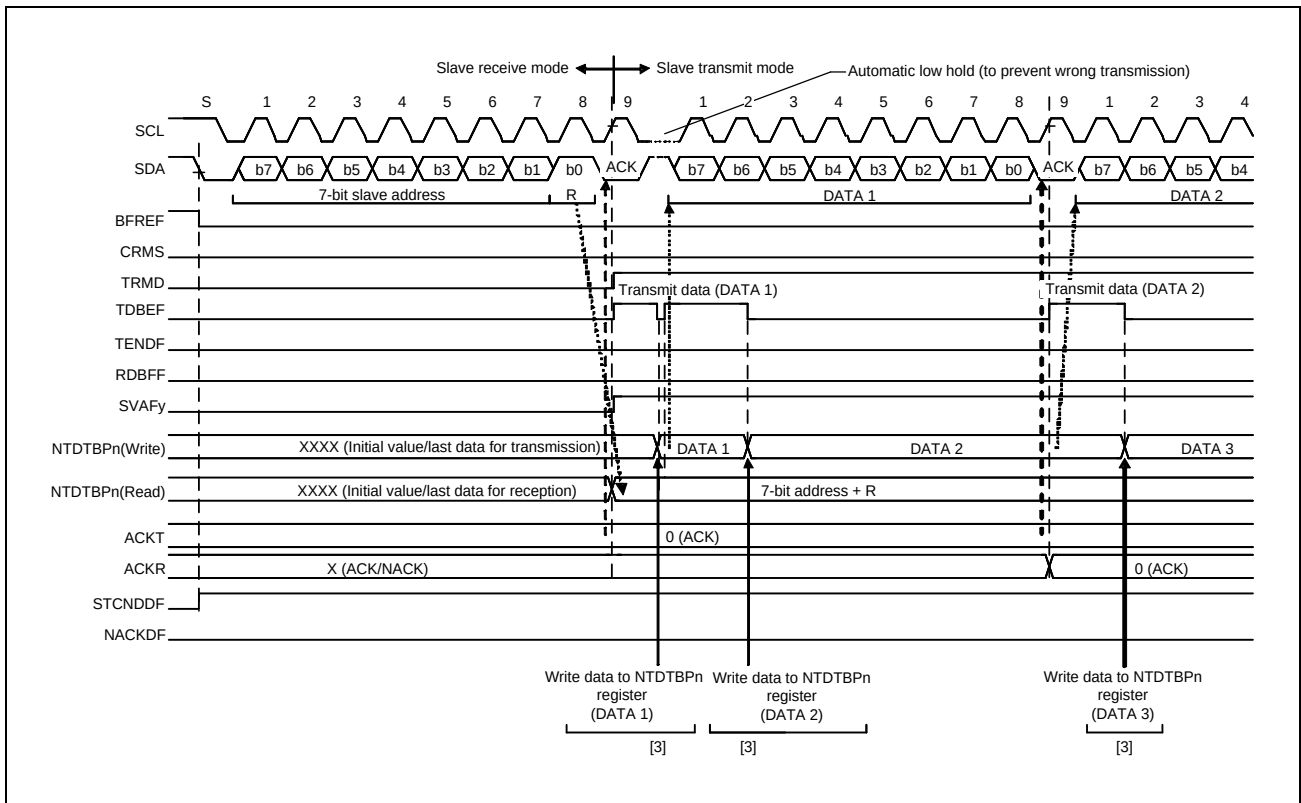


Figure 7.8-29 Slave transmit operation timing (1) (7-bit address format)

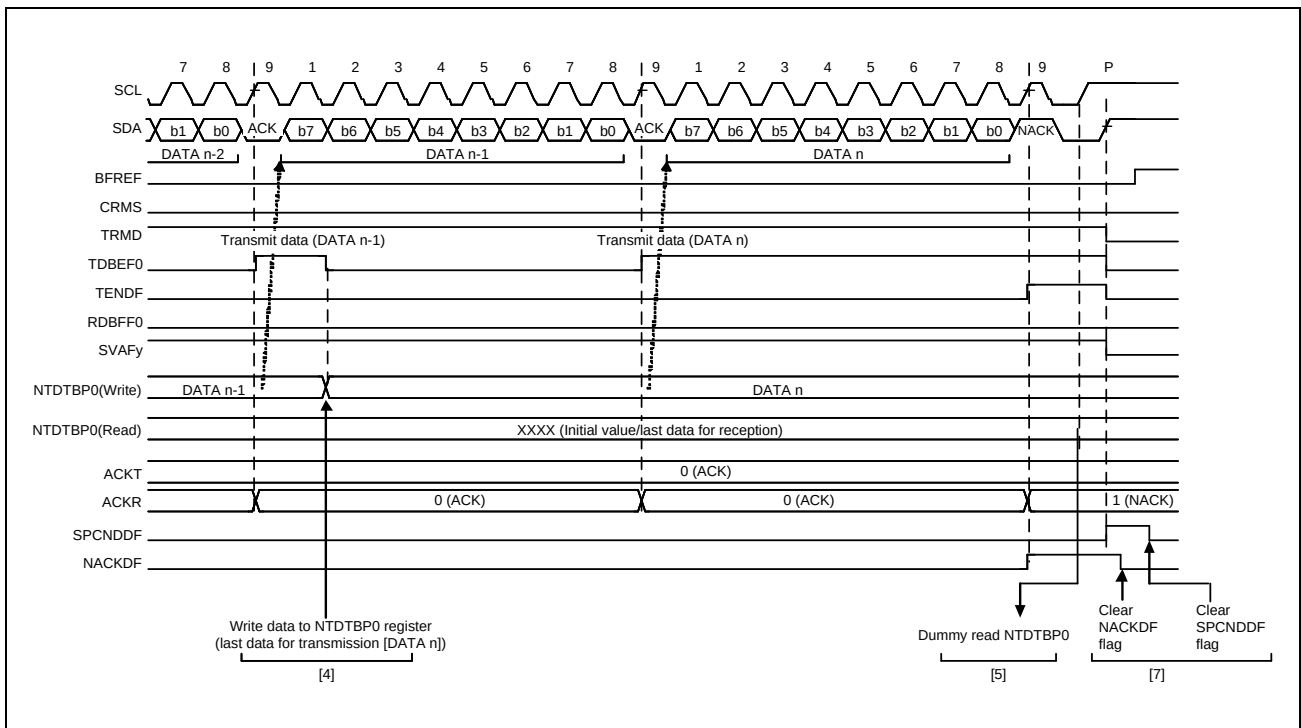


Figure 7.8-30 Slave transmit operation timing (2)

(b) I3C Slave Operation

(b-1) Dynamic Address Assign Procedure

After initializing I3C, the I3C master first performs Dynamic Address Assign Procedure.
The operation of R-I3 during the Dynamic Address Assign Procedure by ENTDAACCC is described below.

1. Initial setting (For details, see (2) I3C Initial Setting Flow)
2. When ENTDAACCC is received, I3C transmits Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCR[7:0]), DCR (SVDCT.TDCR[7:0]) until a dynamic address is assigned. (For details, see “In case of Broadcast CCC (ENTDAACCC)” of (f) CCC detection function [I3C mode].)
3. When ENTDAACCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Buffer.
4. Read Receive Status Descriptor via NRSQP register and check the status.
5. Read the data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the Receive Status Descriptor from the Receive Data Buffer via the NTDTBP0 register.

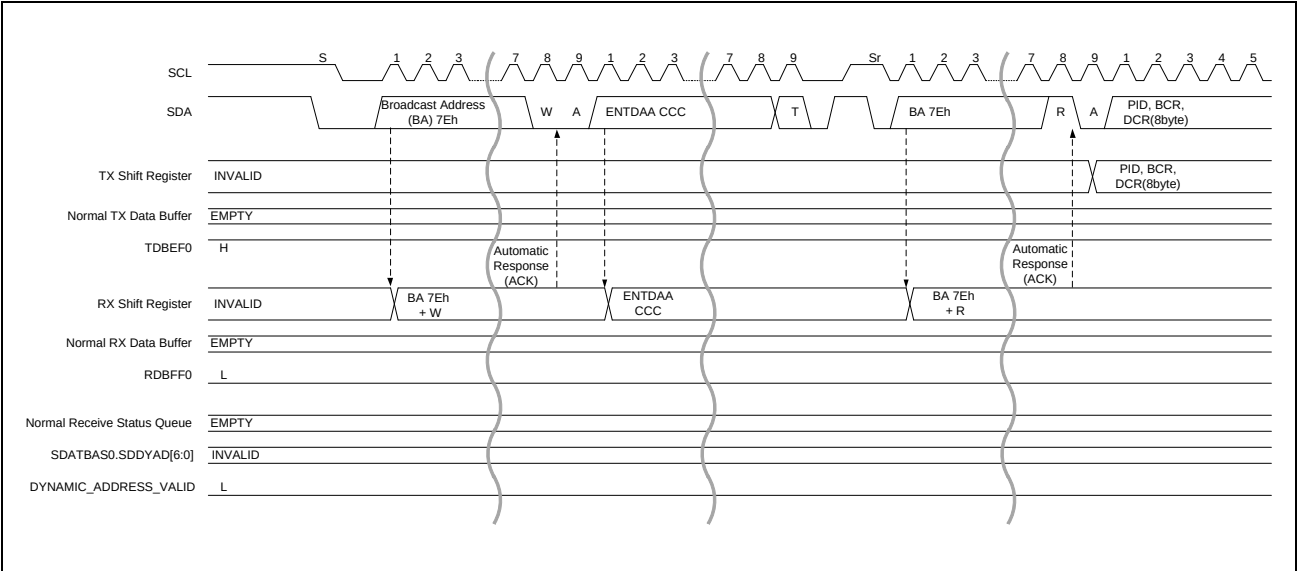


Figure 7.8-31 Dynamic address assign procedure (ENTDAACCC) timing (1/3)

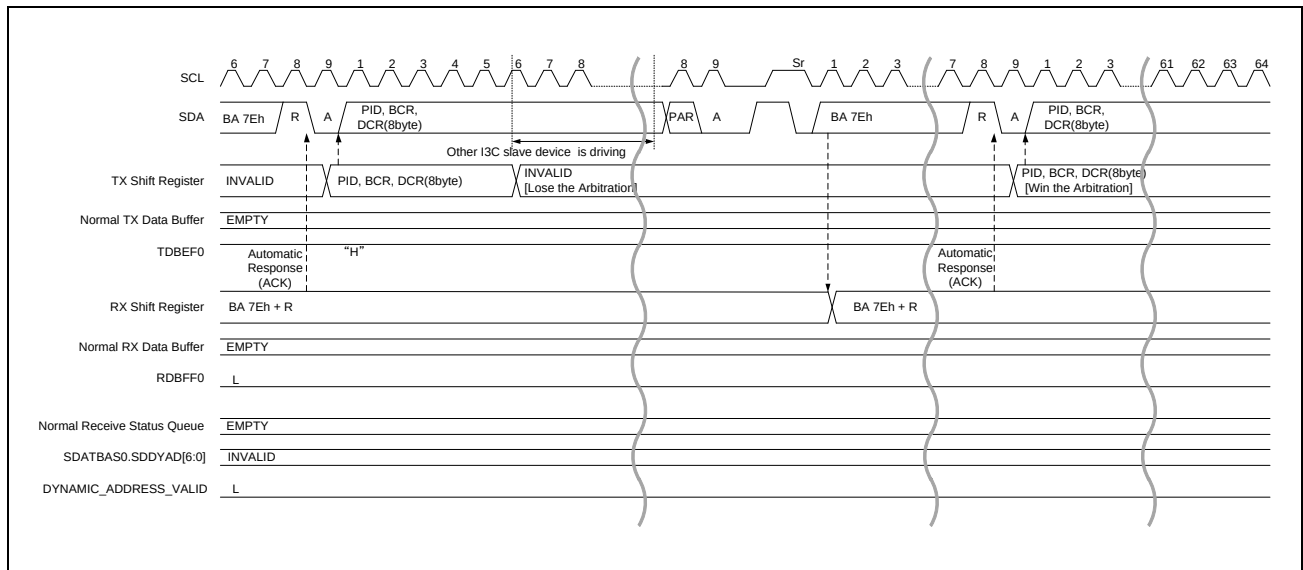


Figure 7.8-32 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

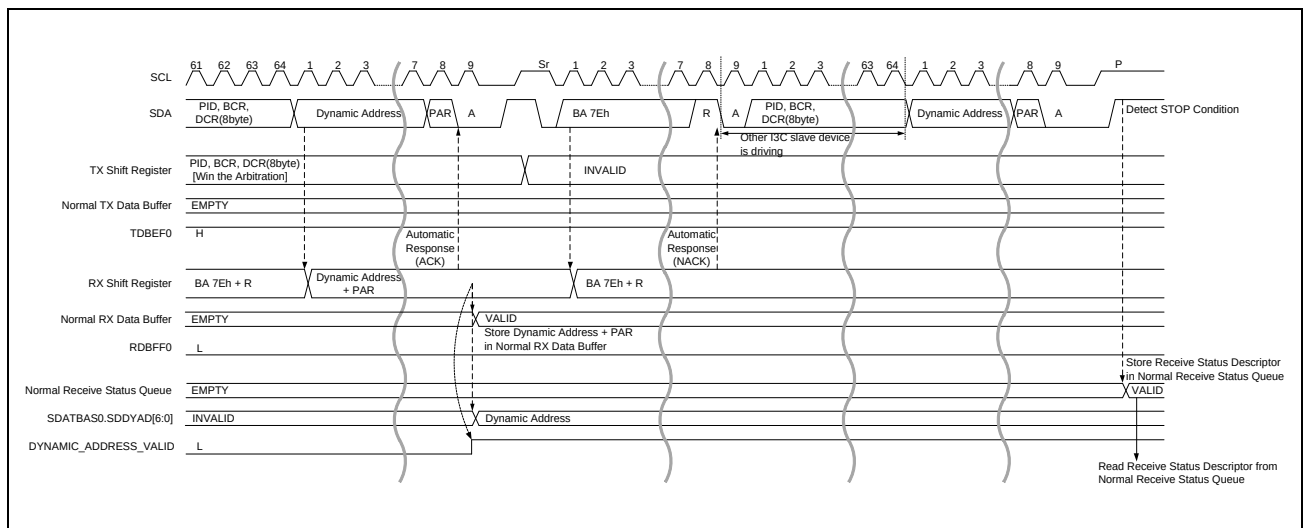


Figure 7.8-33 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

When communicating with a Static Address until the Dynamic Address is assigned from the I3C Master, by setting to the SDSTAD[6:0] bit of DAT (SDATBASn register), the SSTADV bit of the SVDVADn register is set to 1b and the Static Address Will be effective.

If the I3C Slave has a Static Address and the I3C Master executes the Dynamic Address Assign Procedure, it is possible to assign a Dynamic Address with SETDASA CCC.

The operation of I3C during SETDASA CCC Dynamic Address Assign Procedure is described below.

1. Initial setting (For details, see **(2) I3C Initial Setting Flow**)
2. When SETDASA CCC which agrees with its own Static Address is received, the SDDYAD [6:0] bit of DAT (SDATBAS0 register) is renewed and SDYADV bit of SVDVAD0 register is set in 1b. (For details, see “In case of Direct Write CCC” of **(f) CCC detection function [I3C mode].**)

- When SETDASA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Buffer.
- Read Receive Status Descriptor via NRSQP register and check the status.

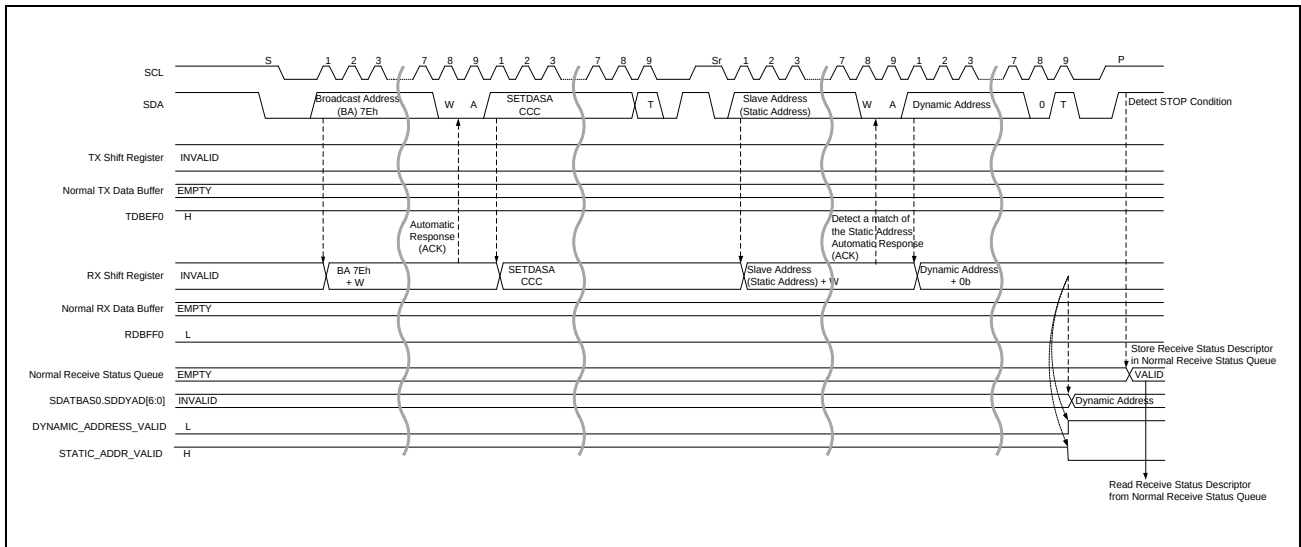


Figure 7.8-34 Dynamic address assign procedure (SETDASA CCC) timing

(b-2) SDR Data Write Transfer

- When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
When a Transaction is received, if the Receive Data Buffer is full, the I3C Slave will respond with NACK in the Address Header.
In preparation for retrying the I3C Master, read the data from the Receive Data Buffer via the NTDTBP0 register, and empty the Receive Data Buffer.
- Data received from I3C Master is stored in the Receive Data Buffer.
- With the RDBFF0 = 1b interrupt, the received data is read from the Receive Data Buffer via the NTDTBP0 register.
- When Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Buffer.
- Read Receive Status Descriptor via NRSQP register and check the status.

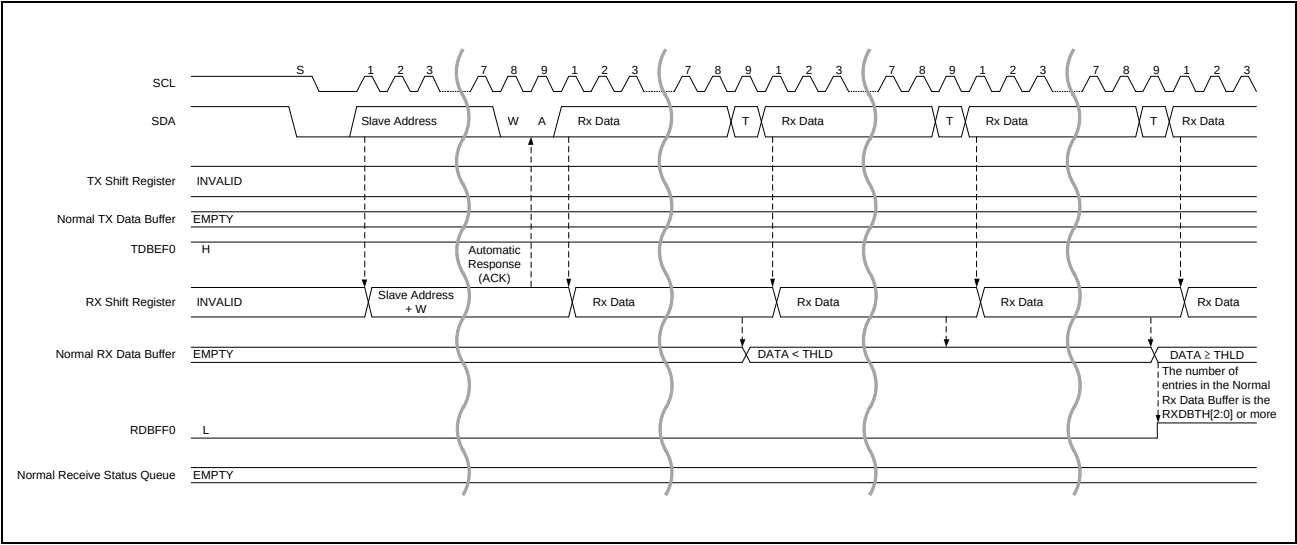


Figure 7.8-35 SDR data write transfer timing (1/2)

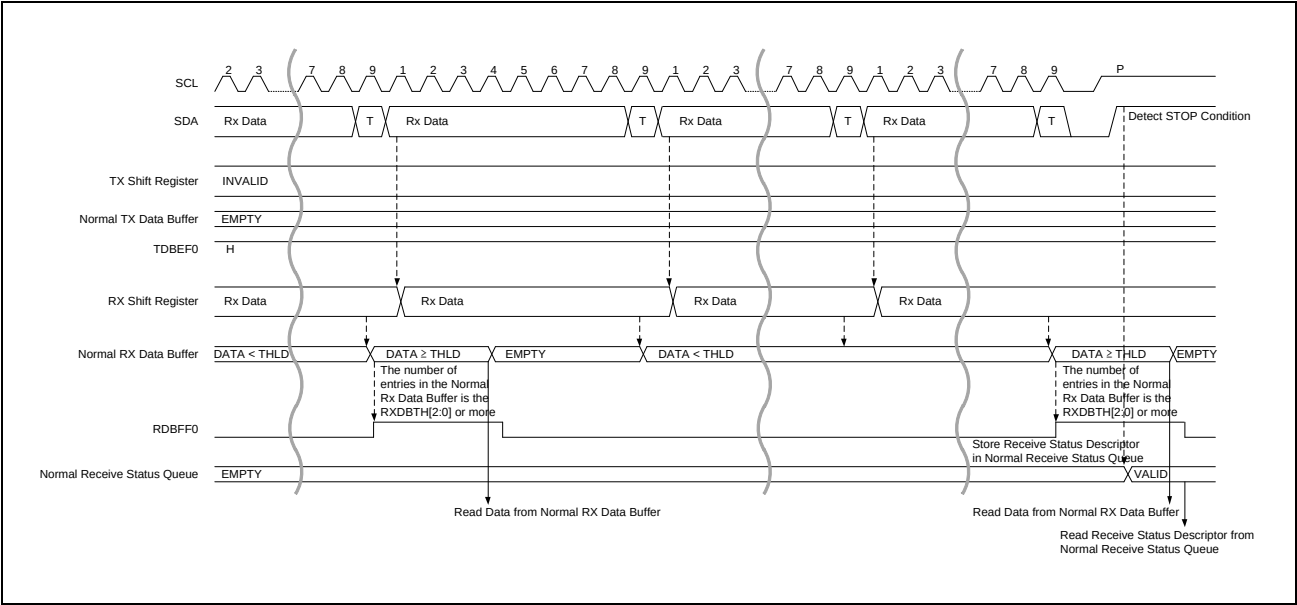
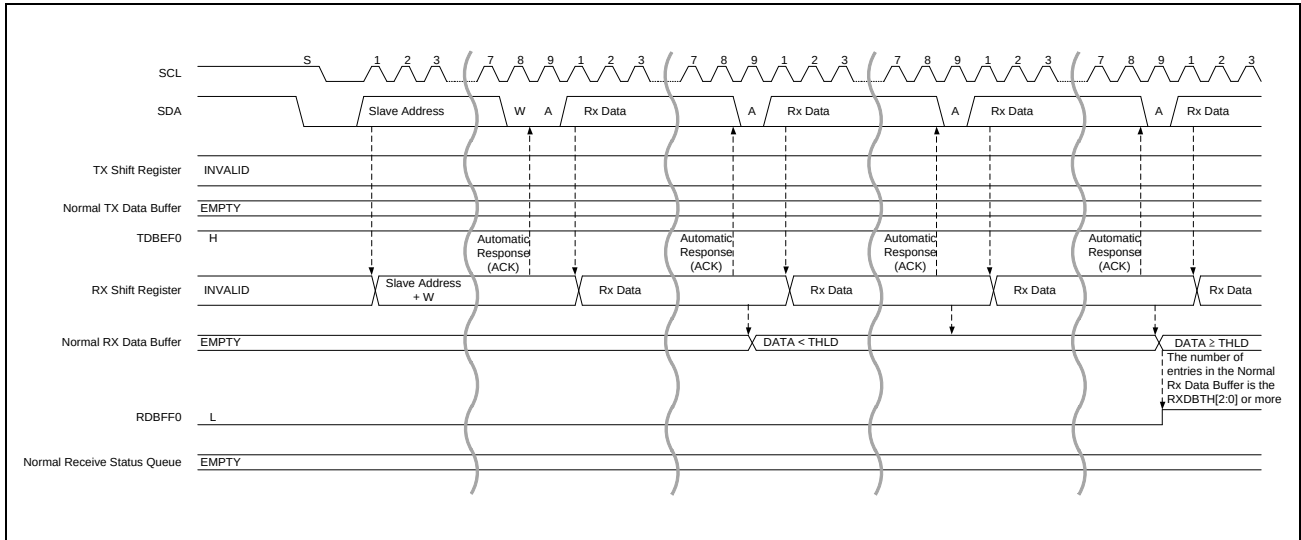
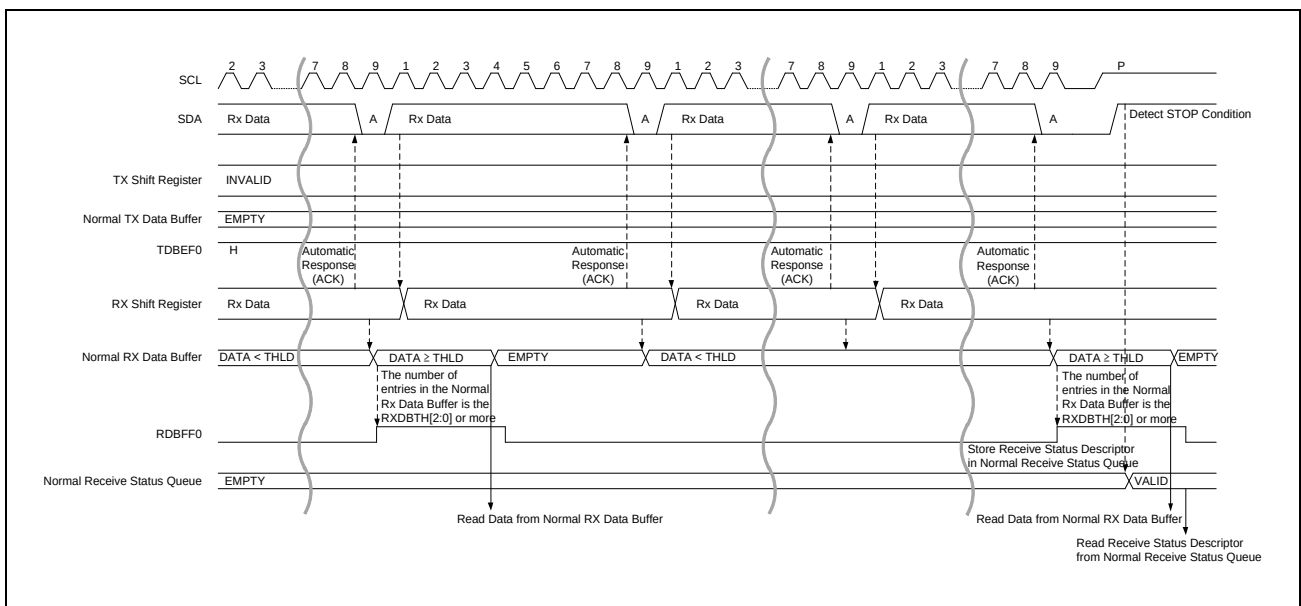


Figure 7.8-36 SDR data write transfer timing (2/2)

Figure 7.8-37 Legacy I²C message data write transfer timing (1/2)Figure 7.8-38 Legacy I²C message data write transfer timing (2/2)

(b-3) SDR Data Read Transfer

1. Write the data requested from the I3C Master to the Transmit Data Buffer via the NTDTBP0 register.
2. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
When a Transaction is received, if the Transmit Data Buffer is EMPTY, I3C Slave responds with NACK with the Address Header.
In preparation for retrying the I3C Master, write data to the Transmit Data Buffer via the NTDTBP0 register.
3. Transmit the data stored in the Transmit Data Buffer.
4. If data to be transmitted still remains, write the data to be transmitted with an interrupt by TDBEF0 = 1b to the Transmit Data Buffer via the NTDTBP0 register.
5. SDR:
When the transmission of the data stored in the Transmit Data Buffer is completed, Low is output to the T-bit following Data, and it is notified to the I3C Master that it is the final data.
Legacy I²C Message:
When NACK is detected, data transmission is terminated.
6. When a Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Buffer.
7. Read the Receive Status Descriptor via NRSQP and check the status.

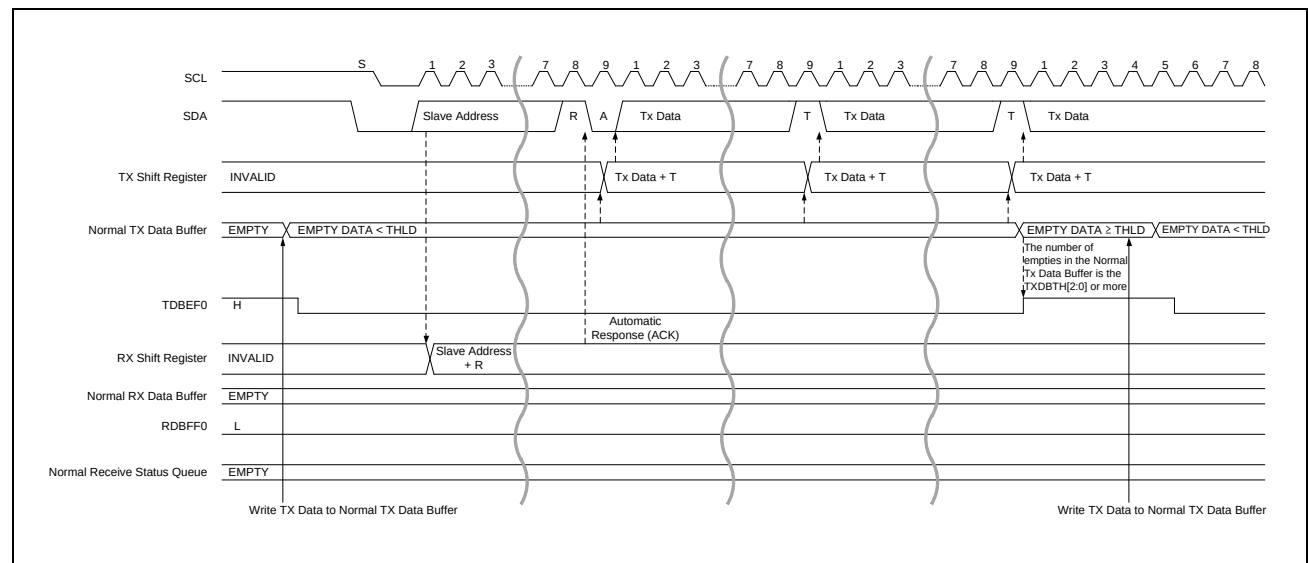


Figure 7.8-39 SDR data read transfer timing (1/2)

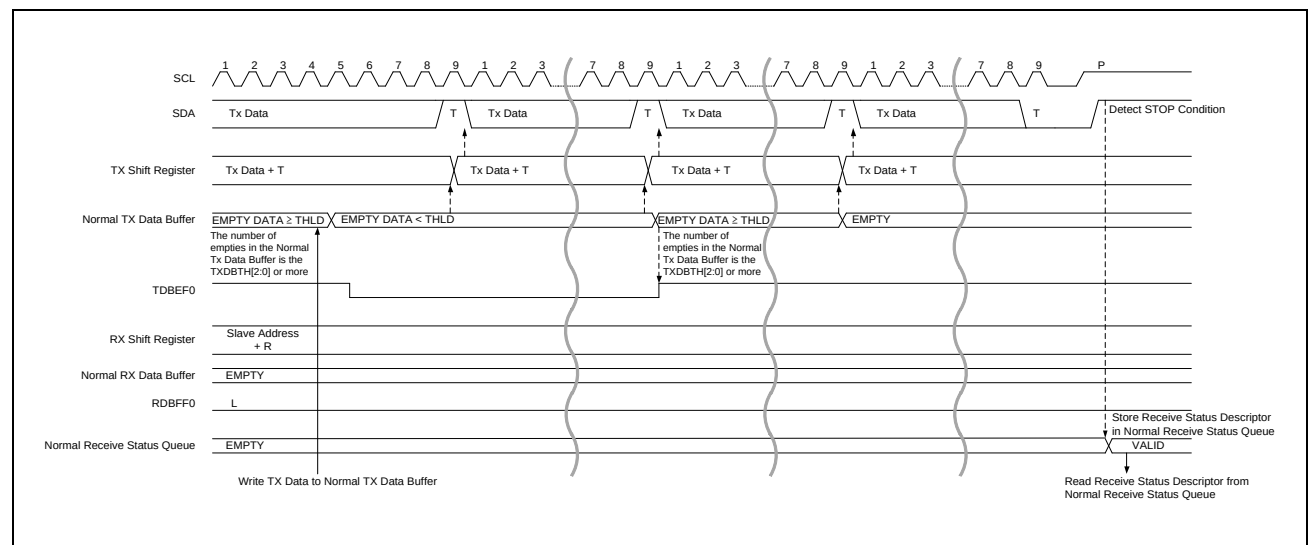


Figure 7.8-40 SDR data read transfer timing (2/2)

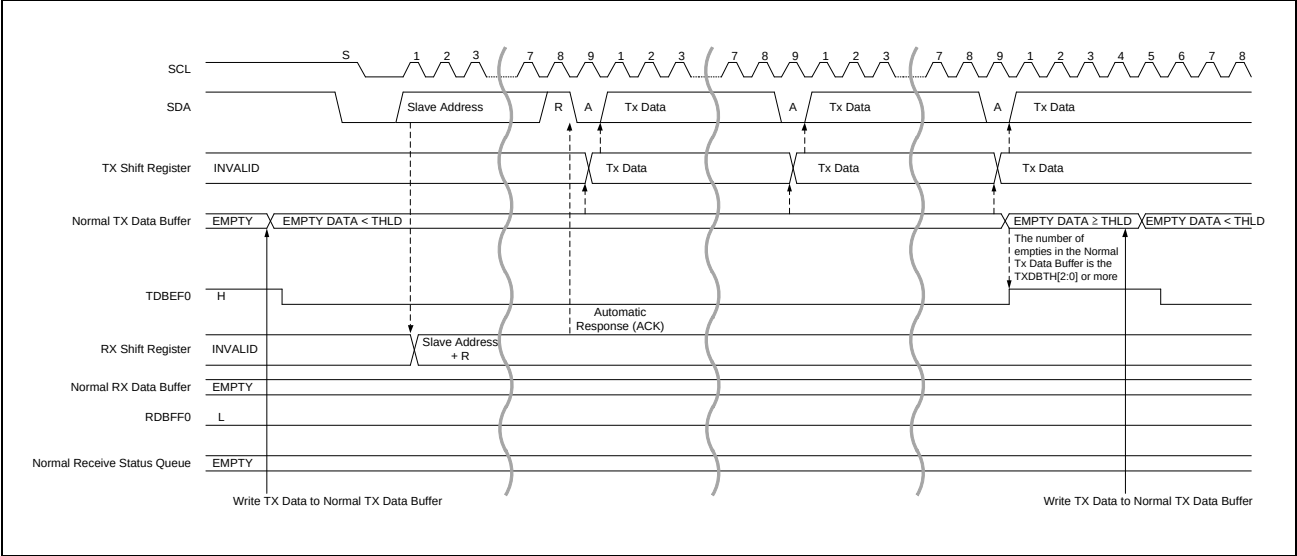


Figure 7.8-41 Legacy I²C message data read transfer timing (1/2)

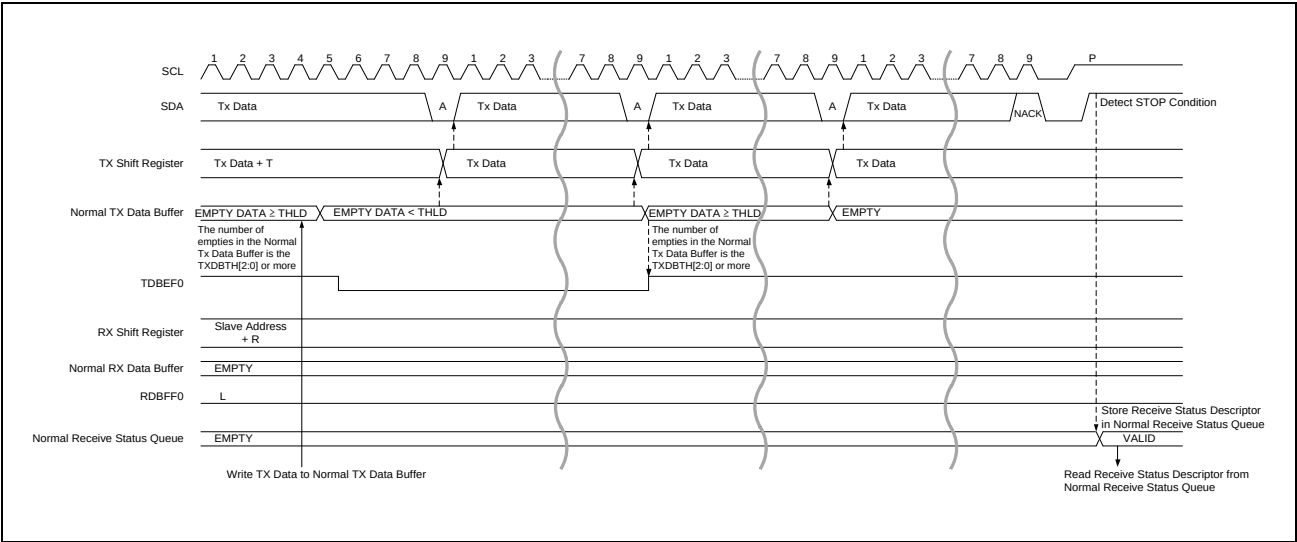


Figure 7.8-42 Legacy I²C message data read transfer timing (2/2)

(b-4) IBI Transfer

1. When sending Slave Interrupt Request.
When transmitting IBI Data, write IBI Data to the IBI Data Buffer via the NIBIQP register.
2. Write Command Descriptor (Immediate Transfer Command or Regular Transfer Command) to the Command Buffer for IBI Transfer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, IBI Transaction is issued under the following conditions.
 - When START condition is detected in Slave Interrupt Request or Mastership Request. (Does not apply a Repeated START condition)
 - If no START is forthcoming within the following Bus Condition, then this module issue a START Request by pulling the SDA line Low.
 - a) Slave Interrupt Request, Mastership Request: Bus Available
 - b) Hot-Join Event: Bus Idle
4. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing a Transaction from I3C Master, stop issuing Transaction.
When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
5. When sending Slave Interrupt Request:
 - When IBI data for transmission still remain, write IBI data with an interrupt by IBIQEFF = 1b to the IBI Data Buffer via the NIBIQP register.
 - When the transmission of IBI Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, output Low to the T-bit following IBI Data and notify the I3C Master that it is the final IBI Data.
6. When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor form the Response Buffer with the NRSPQP register and check the status. If NACK is responded, repeat steps 1 to 7.
8. When sending Slave Interrupt Request:
Check that the value of the DATA_LENGTH[15:0] bit of the Response Descriptor is 0b.

The Mastership processing flow is shown in **Figure 7.8-45**. When joining the I3C Bus by Hot-Join after the I3C Bus has already been configured, issue the Hot-Join according to the flow shown in **Figure 7.8-126**.

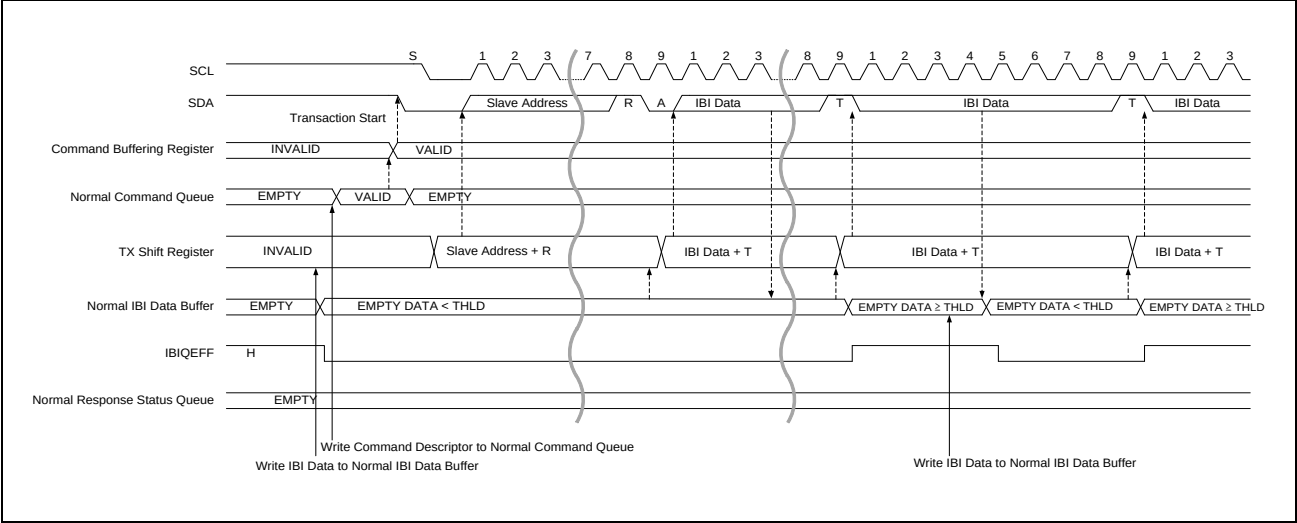


Figure 7.8-43 I3C slave IBI transfer timing (1/2)

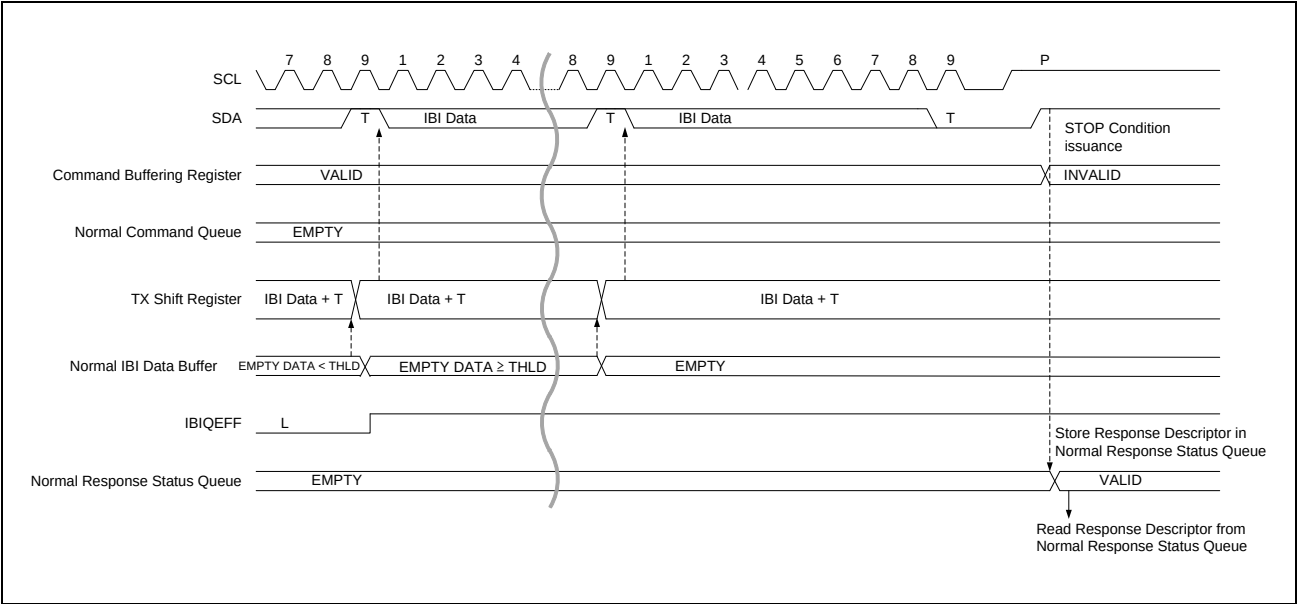


Figure 7.8-44 I3C slave IBI transfer timing (2/2)

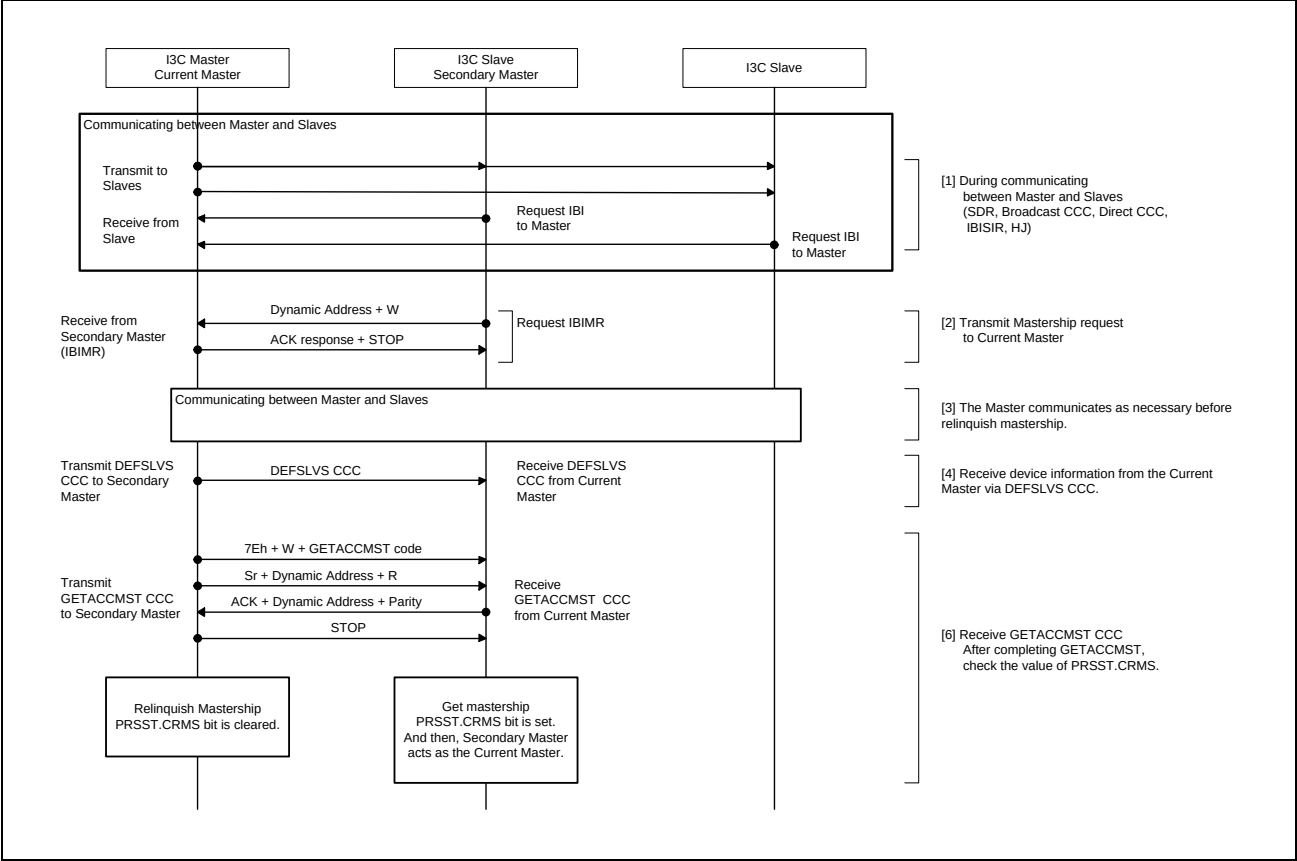


Figure 7.8-45 I3C Slave Mastership processing flow

7.8.4.2.2 Data Handler

The relationship between the transfer method and the queue is shown in **Table 7.8-11**.

Table 7.8-11 Transfer method and queue

Protocol	Transfer method	Queue/Buffer	size	Master	Slave	Secondary Master
I ² C Mode	Single buffer transfer	Normal Transmit Data	1 byte	✓	✓	—
		Normal Receive Data	1 byte	✓	✓	—
I3C Mode	Normal FIFO buffer transfer	Normal Command	4 QUEUES	✓	✓	✓
		Normal Response Status	4 QUEUES	✓	✓	✓
		Normal Transmit Data	16 DWORDs	✓	✓	✓
		Normal Receive Data	16 DWORDs	✓	✓	✓
		Normal Receive Status	2 QUEUES	—	✓	✓
		Normal IBI Status	2 QUEUES	✓	—	✓
		Normal IBI Data	8 DWORDs	✓	✓	✓

(1) Transfer Method in I²C Mode

(a) Single Buffer transfer

Each process (condition issue, data transfer, ACK/NACK response) is controlled by software.

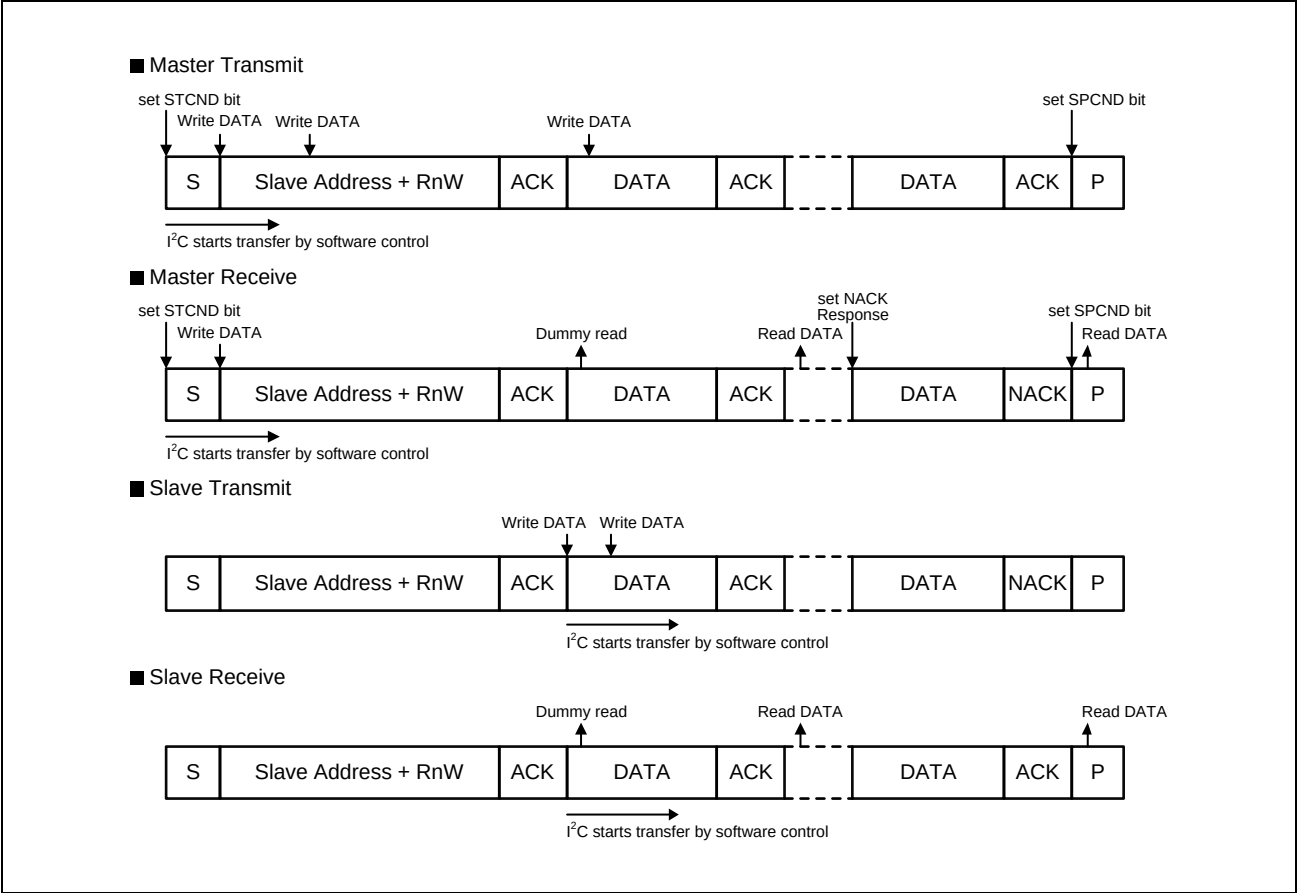


Figure 7.8-46 Data handler with single buffer transfer

(2) Transfer Method in I3C Mode

(a) Normal FIFO Buffer Transfer

I3C autonomously starts transfer when data and command are written.

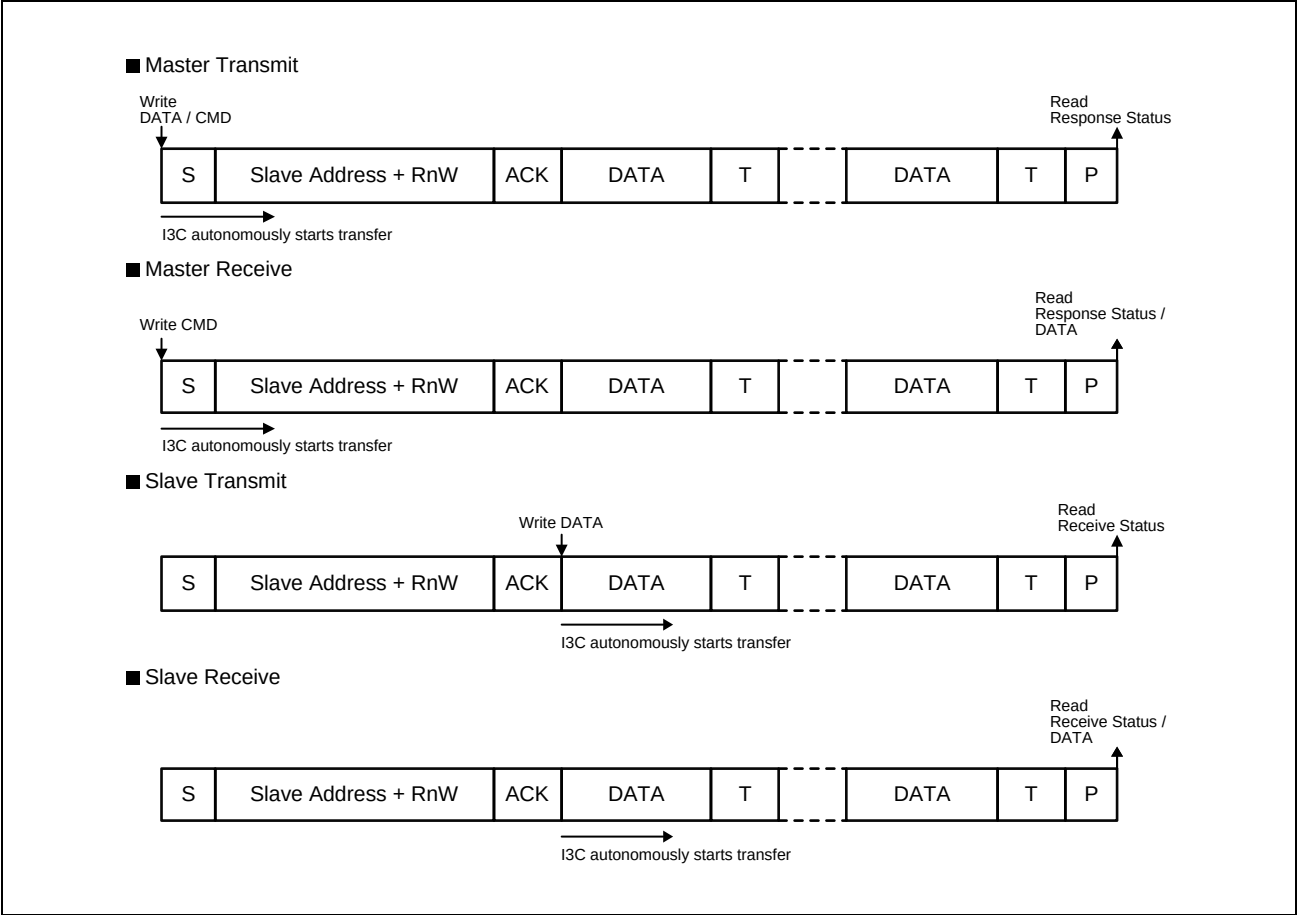


Figure 7.8-47 Data handler with normal FIFO buffer transfer

7.8.4.2.3 I²C/I3C Protocol

(1) Communication Protocol

(a) I²C Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a START condition or Repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is issued.

Figure 7.8-48 shows the I²C bus format, and **Figure 7.8-49** shows the I²C bus timing.

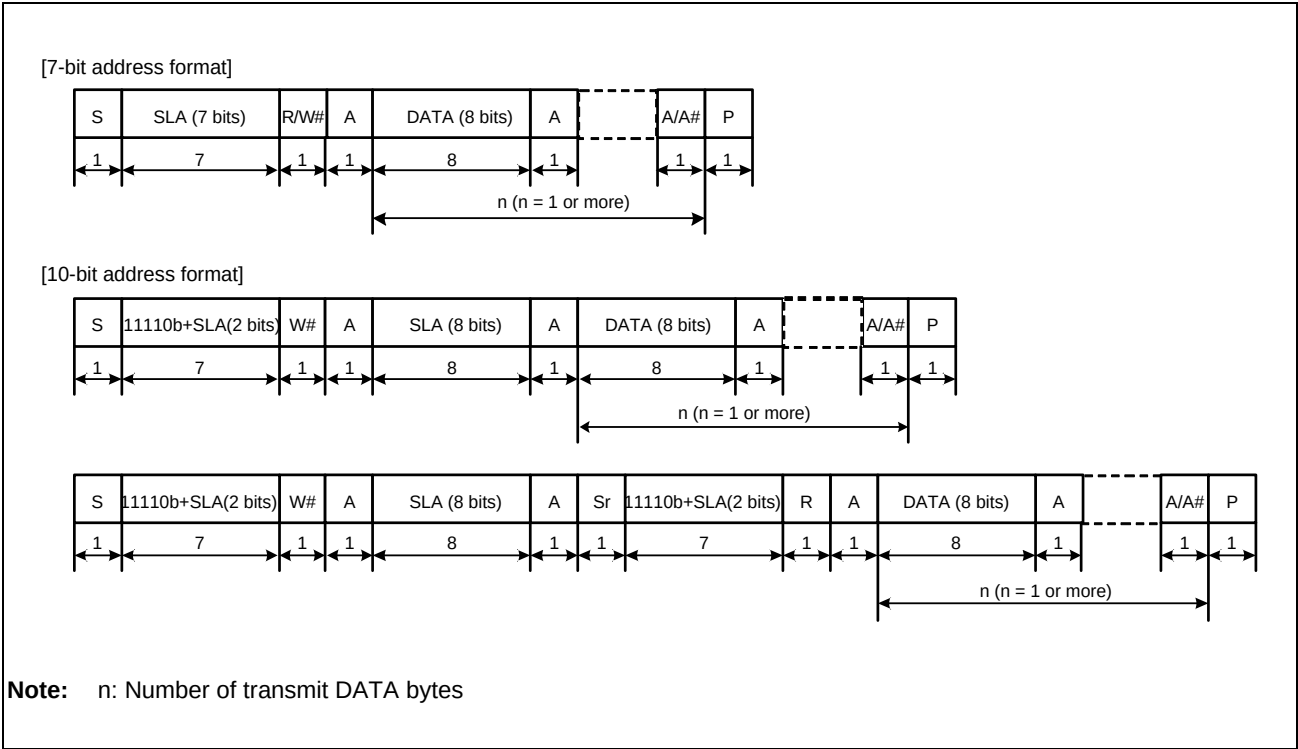
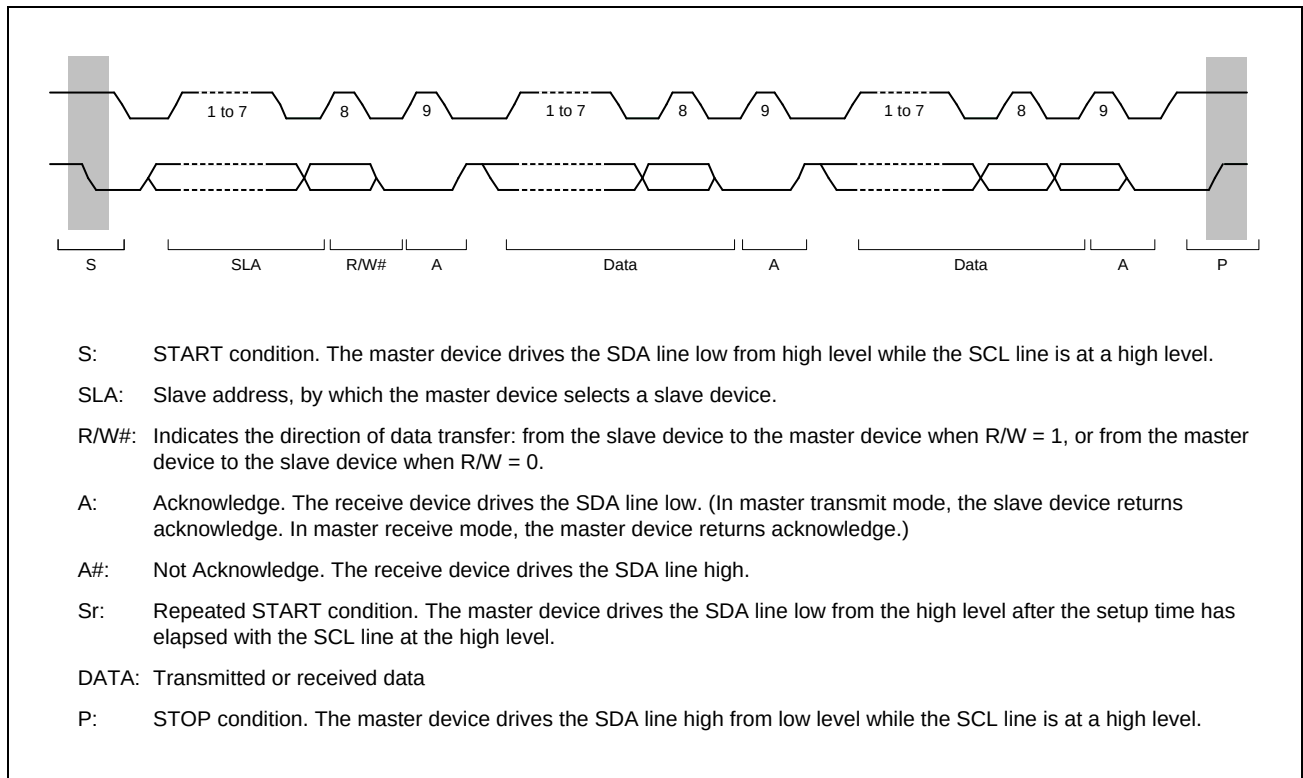


Figure 7.8-48 I²C bus format

Figure 7.8-49 I²C bus timing (SLA = 7 bits)

(b) I3C Communication Data Format

Figure 7.8-50 through **Figure 7.8-53** illustrate a typical communication for each of the four I3C Protocols. While these diagrams do not exhaustively illustrate all possible I3C communications, they do serve as useful introductions to the signaling and transmission formatting used in each I3C Protocol.

Figure 7.8-50 illustrates example communication using I3C Single Data Rate (SDR) coding with Broadcast (7Eh). It shows the Master reading a byte of data from the Slave at Address 2Bh in SDR Mode. From the Bus Free Condition, the Master issues a START by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain.

All Slaves ACK by pulling the SDA line Low (in the Figure, pink fill means the Slave is in control of the SDA line at this time). The Master then issues a Repeated START, then the Address of the Slave (2Bh) it wants to read followed by RnW (1 for Read). The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the SDA line Low. At this point, the Master continues to toggle the SCL line and release the SDA line, allowing the Slave to drive SDA to send one byte of data (4Ah) followed by T. T = 1b informs the Master that there is additional data, whereas T = 0b signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pullup, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I²C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I²C 50ns Spike Filter.

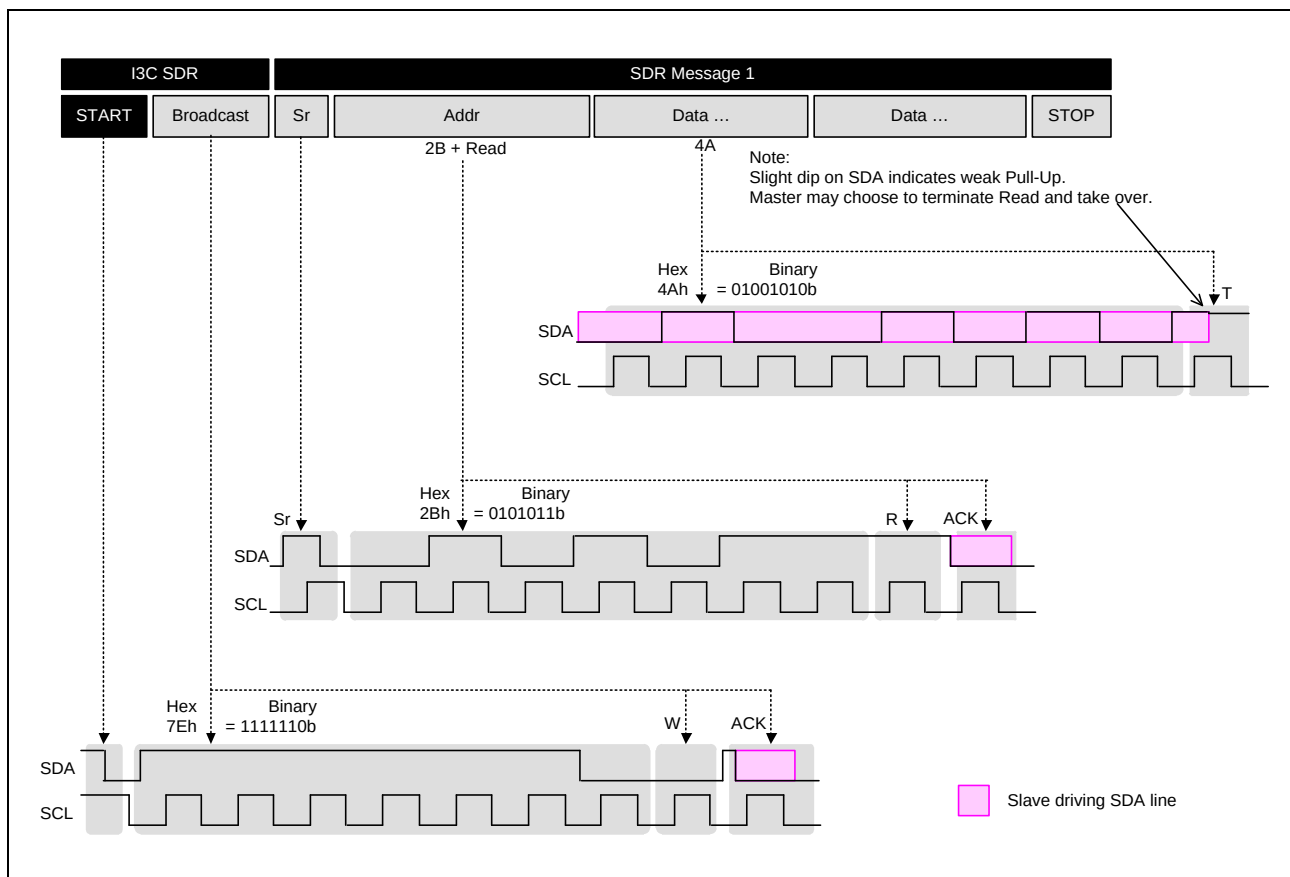


Figure 7.8-50 Example communication using I3C coding SDR with broadcast (7Eh)

Figure 7.8-51 illustrates example communication using I3C Single Data Rate (SDR) coding without Broadcast (7Eh). It shows the Master reading a byte of data from the Slave at Address 2Bh in SDR Mode. From the Bus Free Condition, The Master then issues a START, then the Address of the Slave (2Bh) it wants to read followed by RnW (1 for Read).

The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the SDA line Low. At this point, the Master continues to toggle the SCL line and release the SDA line, allowing the Slave to drive SDA to send one byte of data (4Ah) followed by T. T = 1b informs the Master that there is additional data, whereas T = 0b signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pull-up, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I²C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I²C 50ns Spike Filter.

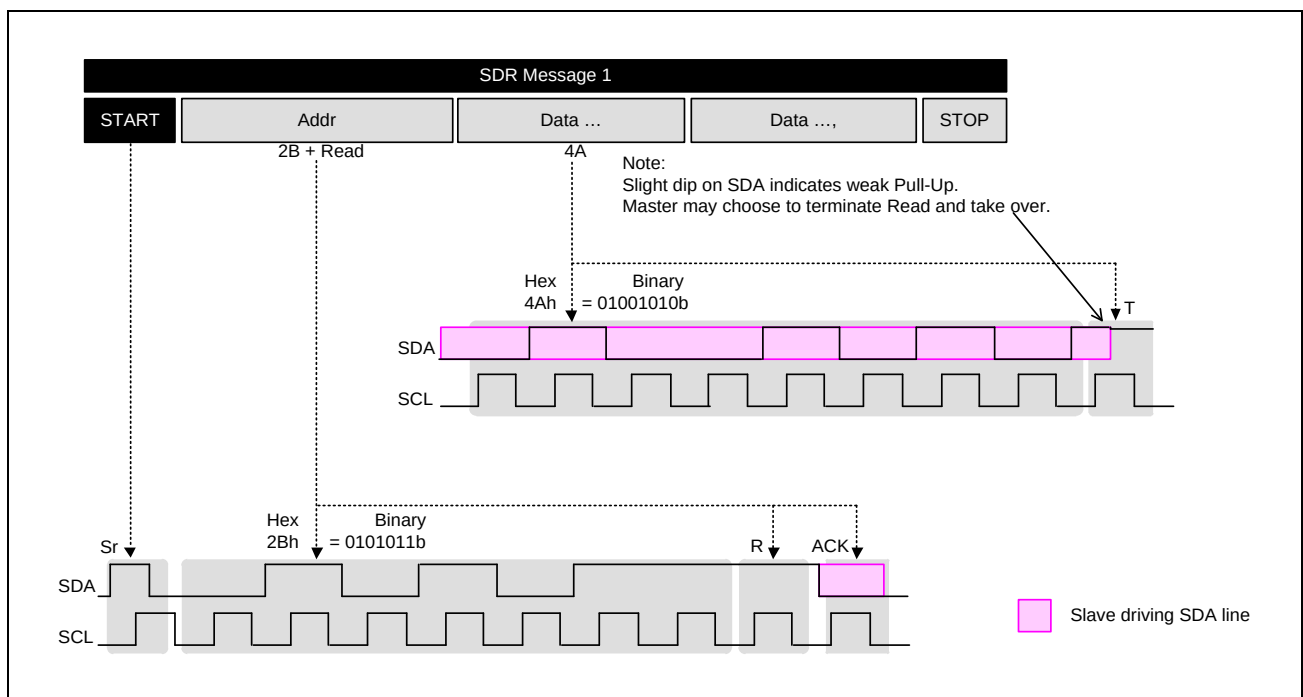


Figure 7.8-51 Example communication using I3C coding SDR without broadcast (7Eh)

Figure 7.8-52 shows the Master issuing a CCC Direct Command to a single Slave. This particular command (GETPID) reads the Provisional ID of a Slave.

From the Bus Free Condition, the Master issues a START by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0h for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Direct Common Command Code for GETPID (8Ch) followed by parity bit T (odd parity = 0 for 8Ch) then the 7-bit Dynamic Address of the Slave (chosen arbitrarily here to be 2Bh) followed by a RnW bit (1 for Read). Then the Master turns on a pull-up resistor and goes to Open Drain, allowing the Slave at Address 2Bh to ACK by pulling SDA Low, which tells the Master that the Slave Acknowledges the command and will comply. (Alternatively, the Slave may NACK by not pulling SDA Low, which would inform the Master that the Slave will not comply – in this case, that an error occurred.) Following the ACK the Slave outputs its 48-bit PID one byte at a time, and then the Master issues a Repeated START (this part of the waveform sequence is not shown in the Figure).

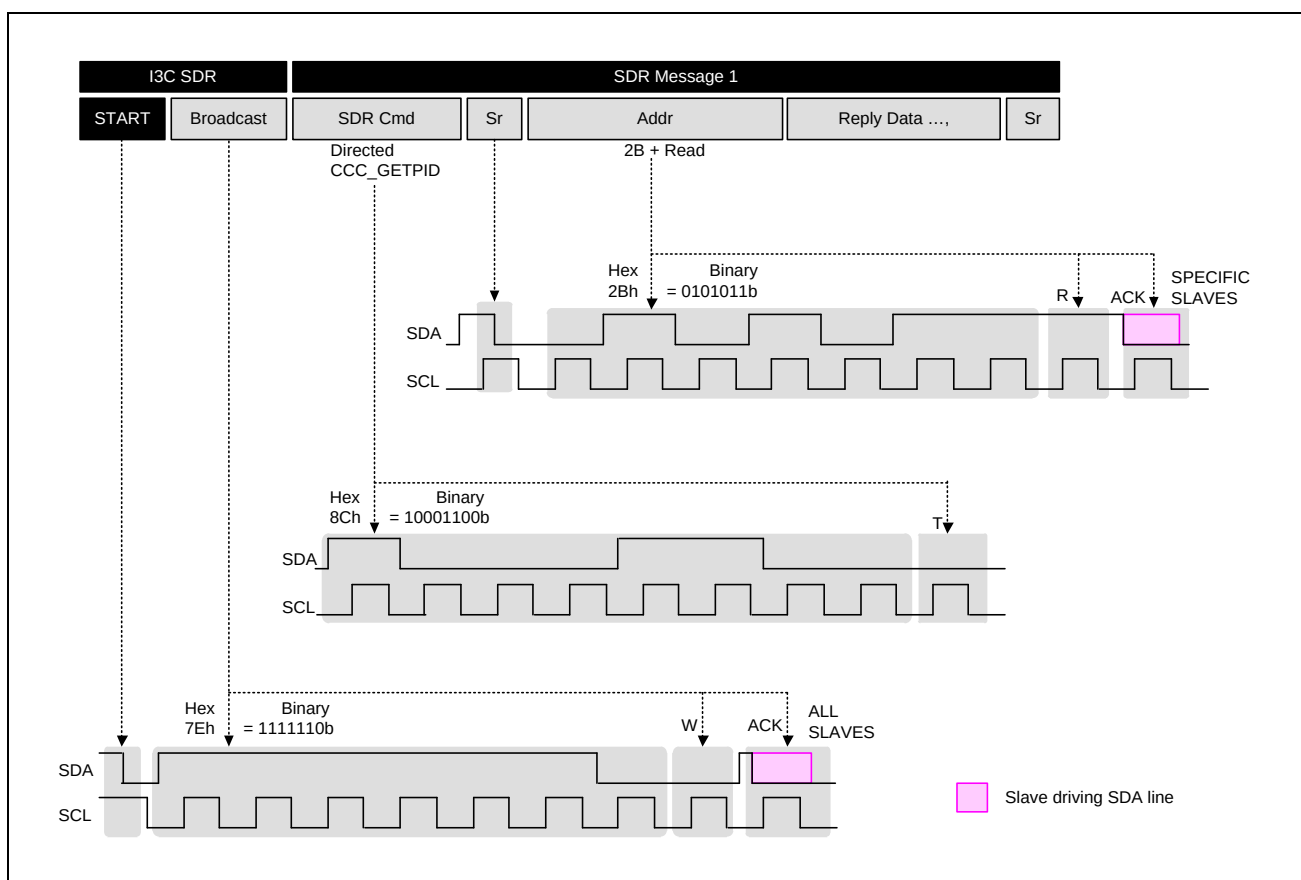


Figure 7.8-52 Example communication using I3C coding SDR with CCC direct addressing

Figure 7.8-53 illustrates example SDR communication with a CCC Broadcast command. The command used in this example sets the Maximum Read Length of all Slaves to 43 bytes (002Bh).

From the Bus Free Condition, the Master issues a START by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0h for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for SETMRL (09h) followed by parity bit T (odd parity = 1 for 09h), and then 2 data bytes (MSB first) to define the maximum number of bytes which can be read from a Slave in a single read operation. Each data byte is followed by a T bit (parity bit – odd parity). After this the Master issues a Repeated START.

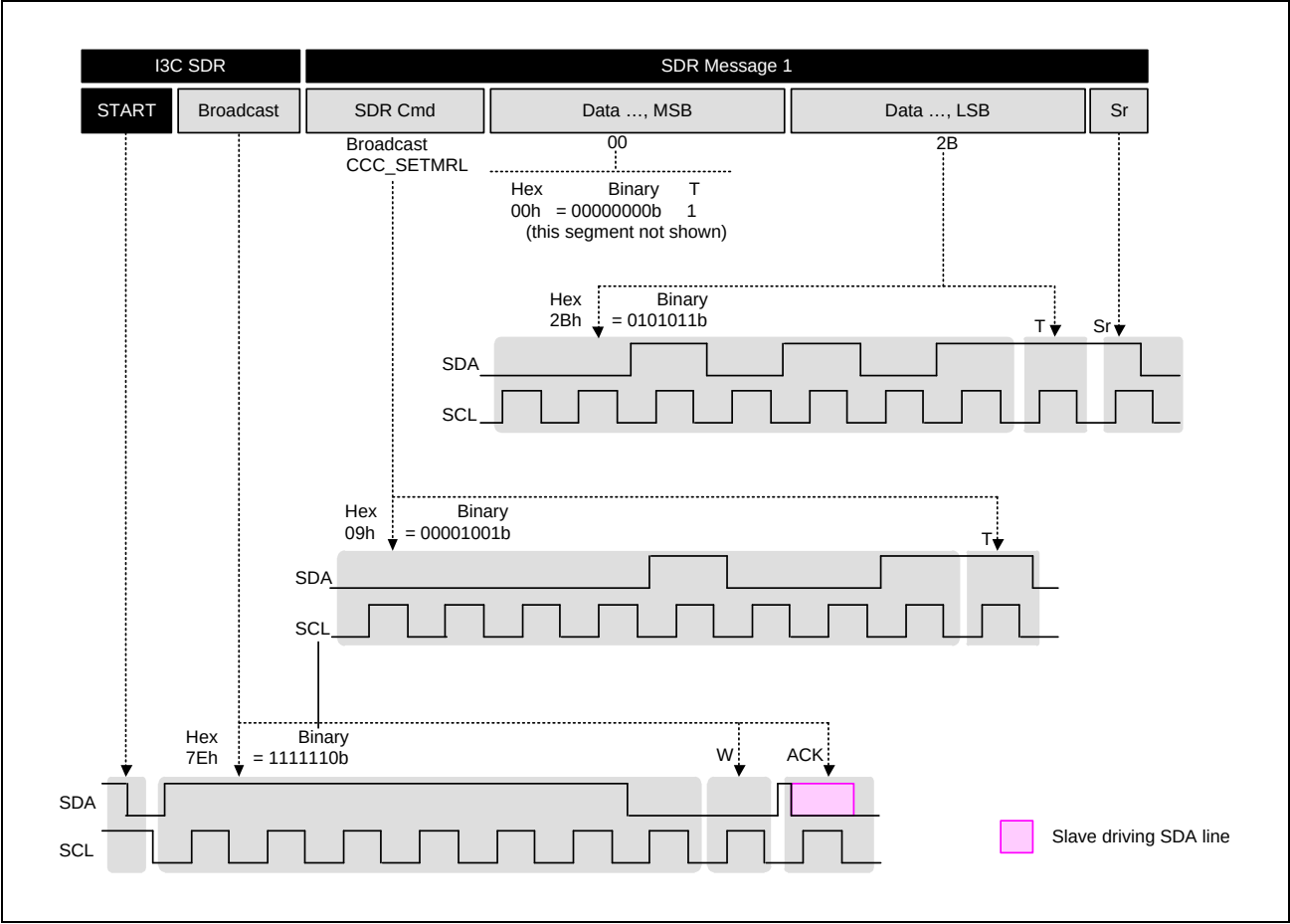


Figure 7.8-53 Example communication using I3C coding SDR with CCC broadcast

(2) Bus Conditions

I3C defines three distinct conditions in which the I3C Bus shall be considered inactive: Bus Free, Bus Available, and Bus Idle (see **Figure 7.8-54**).

(a) Bus Free Condition

State on the I3C Bus where both the SCL line and the SDA line are High for at least the period set by BFRECDT.FRECYC[8:0] bit.

(b) Bus Available Condition [I3C mode]

State on the I3C Bus where both the SCL line and the SDA line are High for at least the period set by BAVLCDT.AVLCYC[8:0] bit.

A Slave may only issue a START Request (For example, for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

(c) Bus Idle Condition [I3C mode]

State on the I3C Bus where both the SCL line and the SDA line are High for at least the period set by BIDLCDT.IDLCYC[17:0] bit.

A Slave may only issue a START Request (For example, for a Hot-Join) after a Bus Idle Condition. Specifications are as follows. IDLE needs to be the largest.

$\text{BFRECDT.FRECYC}[8:0] < \text{BAVLCDT.AVLCYC}[8:0] < \text{BIDLCDT.IDLCYC}[17:0]$

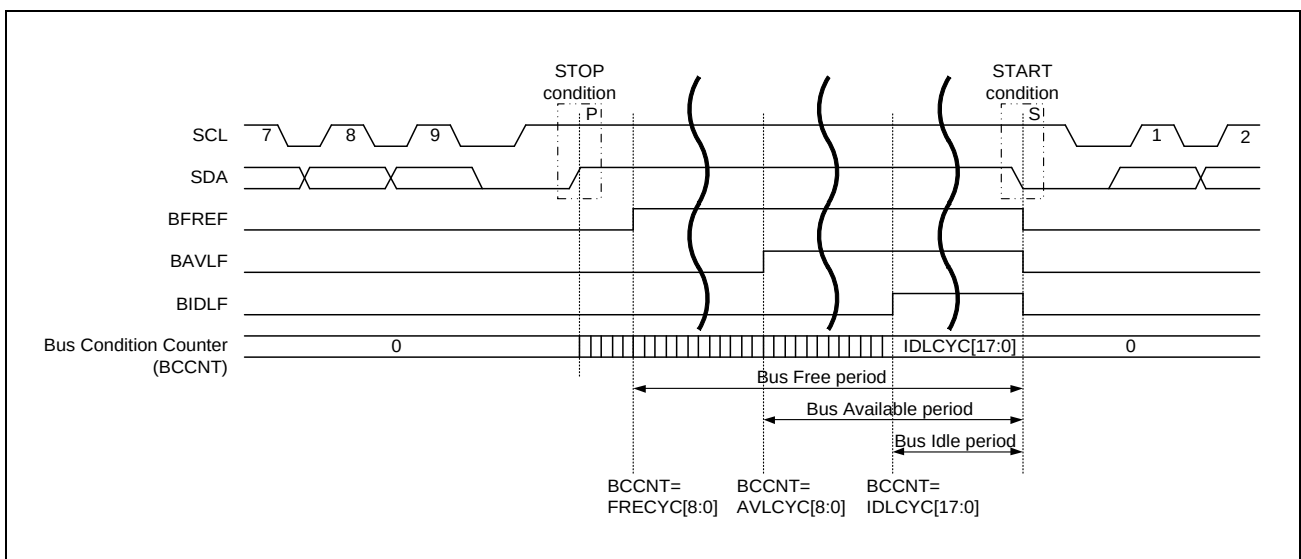


Figure 7.8-54 Bus conditions

(3) START Condition / Repeated START Condition / STOP Condition Issuing Function

(a) Issuing a START Condition

I3C issues a START condition when the CNDCTL.STCND bit is set to 1.

Set the STCND bit to 1b (START condition issuance request) when the BCST.BFREF flag is set to 1b (bus free state).

I3C issues a START condition.

When a START condition is issued normally, I3C automatically shifts to the master transmit mode. A START condition is issued in the following sequence.

[START condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] and the START condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in STDBR.SBRLO[7:0].

(b) Issuing a Repeated START Condition

I3C issues a Repeated START condition when the CNDCTL.SRCND bit is set to 1.

When the SRCND bit is set to 1b, a Repeated START condition issuance request is made and I3C issues a Repeated START condition when the BCST.BFREF flag = 0b (bus busy state) and the PRSST.CRMS bit = 1b (master mode).

A Repeated START condition is issued in the following sequence. [Repeated START condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] and the Repeated START condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] and the Repeated START condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].

NOTE

When issuing Repeated START conditions request, write the slave address to NTDTBP0 after confirming CNDCTL.SRCND = 0. Data written in the period of CNDCTL.SRCND = 1 is not forwarded because retransmission condition before the occurrence.

To issue a Repeated START condition in Hs-mode, follow the steps below.

- 1) Wait until PRSTDBG.SCOLV = 0b.
- 2) Set EXTBR.EBRHO[7:0] to satisfy the hold time of the Repeated START condition.
- 3) Set the CNDCTL.SRCND bit to 1b.
- 4) After confirming CNDCTL.SRCND = 0b, wait until PRSTDBG.SCOLV = 0b.
- 5) Set EXTBR.EBRHO[7:0] according to the High period of the SCL clock in Hs-mode.
- 6) Write the slave address to NTDTBP0.

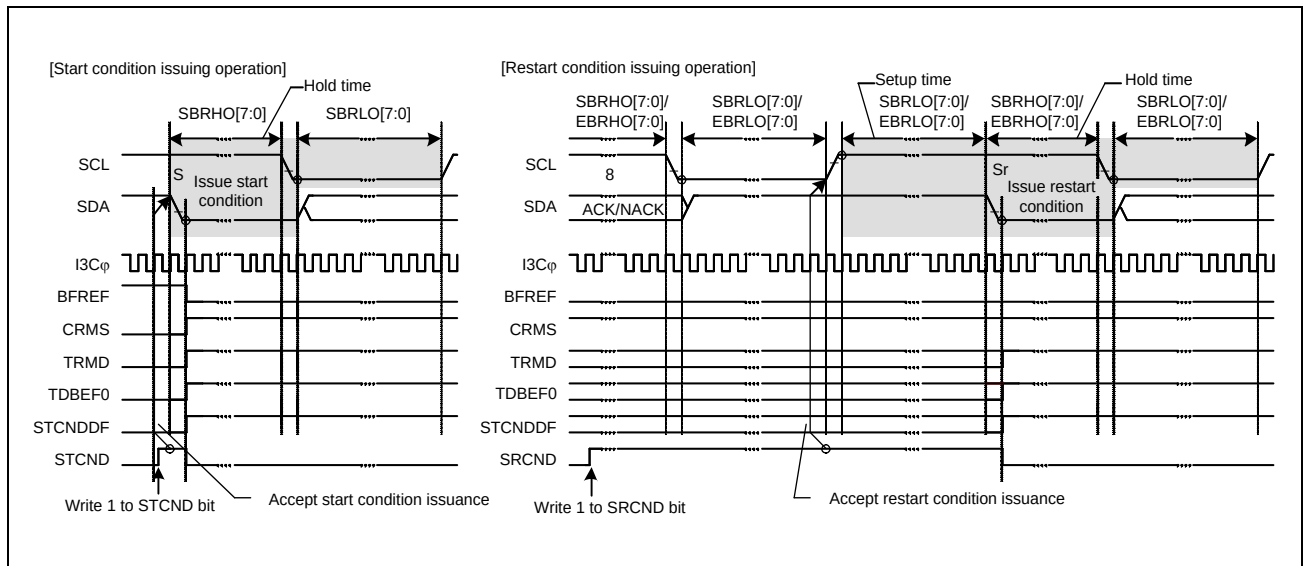


Figure 7.8-55 START condition / repeated START condition issue timing (STCND and SRCND bits)

Figure 7.8-56 shows the operation to issue a Repeated START condition after the master transmission. [Repeated START condition issuance after the master transmission]

- Initial setting. For details, see **7.8.4.3.1 Initial Setting Flow**.
- Read the BFREF flag in BCST to check that the bus is open, and then set the STCND bit in CNDCTL to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag is automatically set to 0b and the STCNDDF flag in BST is automatically set to 1b and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the STCND bit = 1b, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and CRMS and TRMD bits in PRSST is automatically set to 1b, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1b in response to setting of the TRMD bit to 1b.
- Check that the NTST.TDBEF0 flag = 1b, and then write the value for transmission (the slave address and the R/W# bit) to NTDTBP0. Once the data for transmission are written to NTDTBP0, the TDBEF0 flag is automatically set to 0b, the data are transferred from NTDTBP0, and the TDBEF0 flag is again set to 1b. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Since the BST.NACKDF flag being 1b at this time indicates that no slave device recognized the address or there was an error in communications, write 1b to CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to NTDTBP0 as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to NTDTBP0.
- After confirming that the NTST.TDBEF0 flag = 1b, write the data for transmission to the NTDTBP0 register. I3C automatically holds the SCL line low until the data for transmission are ready, a Repeated START condition is issued or a STOP condition is issued.
- After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1b, and then, after check that the BST.STCNDDF flag = 1b, set the BST.STCNDDF flag to 0b.
- Set the SRCND bit in CNDCTL to 1b (Repeated START condition issuance request). Upon receiving the request, I3C issues a Repeated START condition.

- After check that the BST.STCNDDF flag = 1, write the value for transmission (the slave address and the R/W# bit) to NTDTBP0.

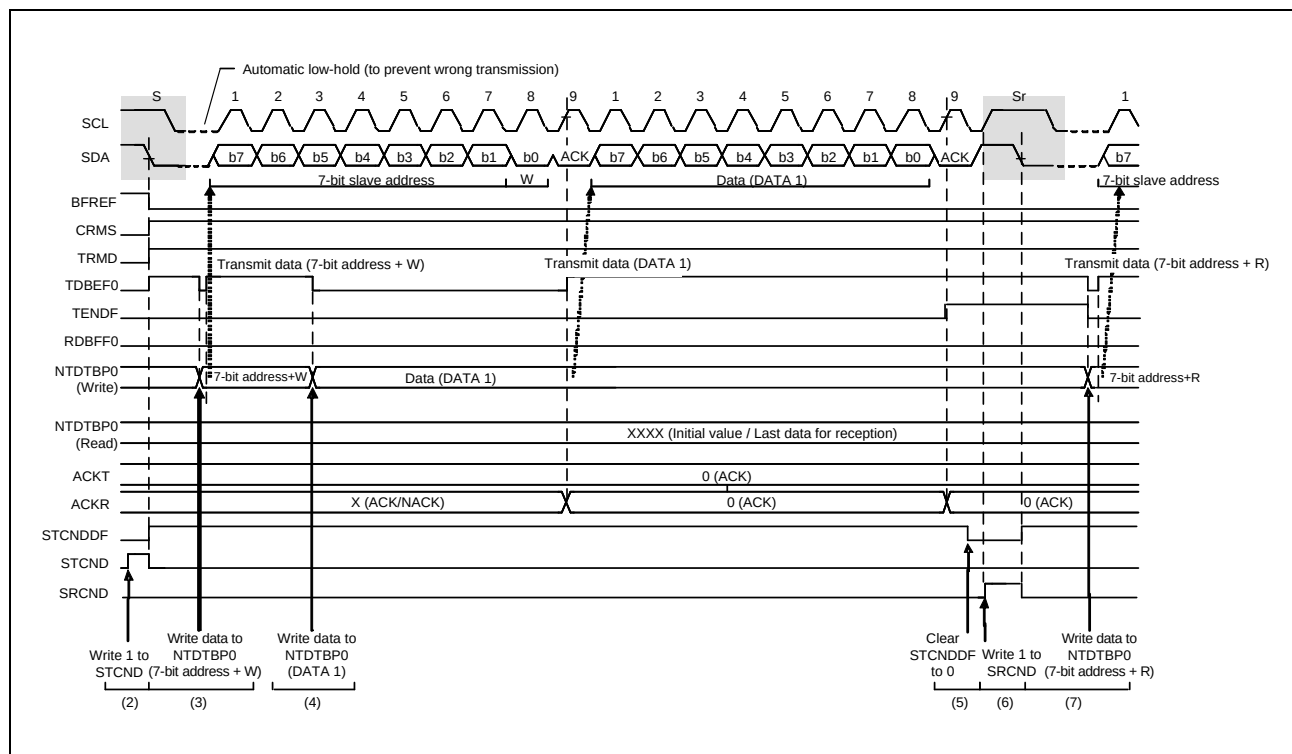


Figure 7.8-56 Repeated START condition issuance after the master transmission timing

(c) Issuing a STOP Condition

I3C issues a STOP condition when the SPCND bit in CNDCTL is set to 1.

When the SPCND bit is set to 1b, a STOP condition issuance request is made and I3C issues a STOP condition when the BCST.BFREF flag = 0b (bus busy state) and the PRSST.MST bit = 1b (master mode).

A STOP condition is issued in the following sequence.

[STOP condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] and the STOP condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in BFRECDT.FRECYC[8:0] and the bus free time.
- Set the BFREF flag to 1b (to release the bus mastership).

NOTE

To issue a STOP condition in Hs-mode, follow the steps below.

- 1) Wait until PRSTDBG.SCOLV=0b.
- 2) Set EXTBR.EBRHO[7:0] to satisfy the setup time for the STOP condition.
- 3) Set the CNDCTL.SPCND bit to 1b.
- 4) Wait until CNDCTL.SPCND=0b.
- 5) Set EXTBR.EBRHO [7: 0] according to the High period of the SCL clock in Hs-mode.

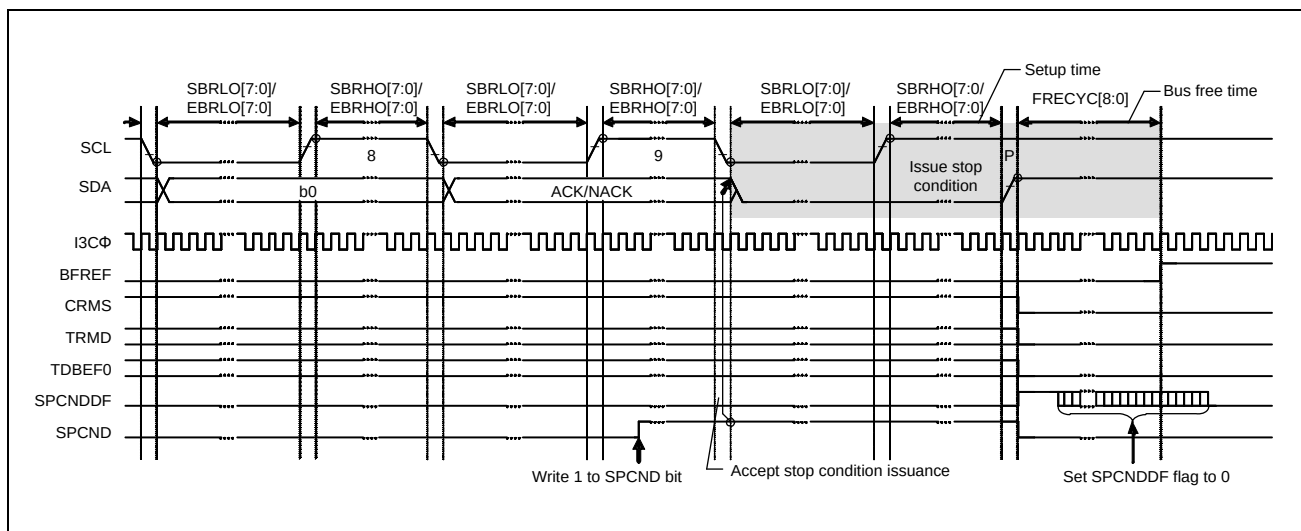


Figure 7.8-57 STOP condition issue timing (SPCND bit)

(4) Address Match Detection

I3C can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

(a) Slave-Address Match Detection [I²C mode]

I3C can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the SVCTL.SVAEn bit (n = 0 to 2) is set to 1b, the slave addresses set in the SVDVADn register (n = 0 to 2) can be detected.

When I3C detects a match of the set slave address, the corresponding SVST.SVAFn flag (n = 0 to 2) is set to 1b at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1b by the following R/W# bit. This causes a receive data full interrupt (INT_ri3c_rx_n) or transmit data empty interrupt (INT_ri3c_tx_n) to be generated. The SVAFn flag is used to identify which slave address has been specified.

Figure 7.8-58 to Figure 7.8-60 show the SVAFn flag set timing in three cases.

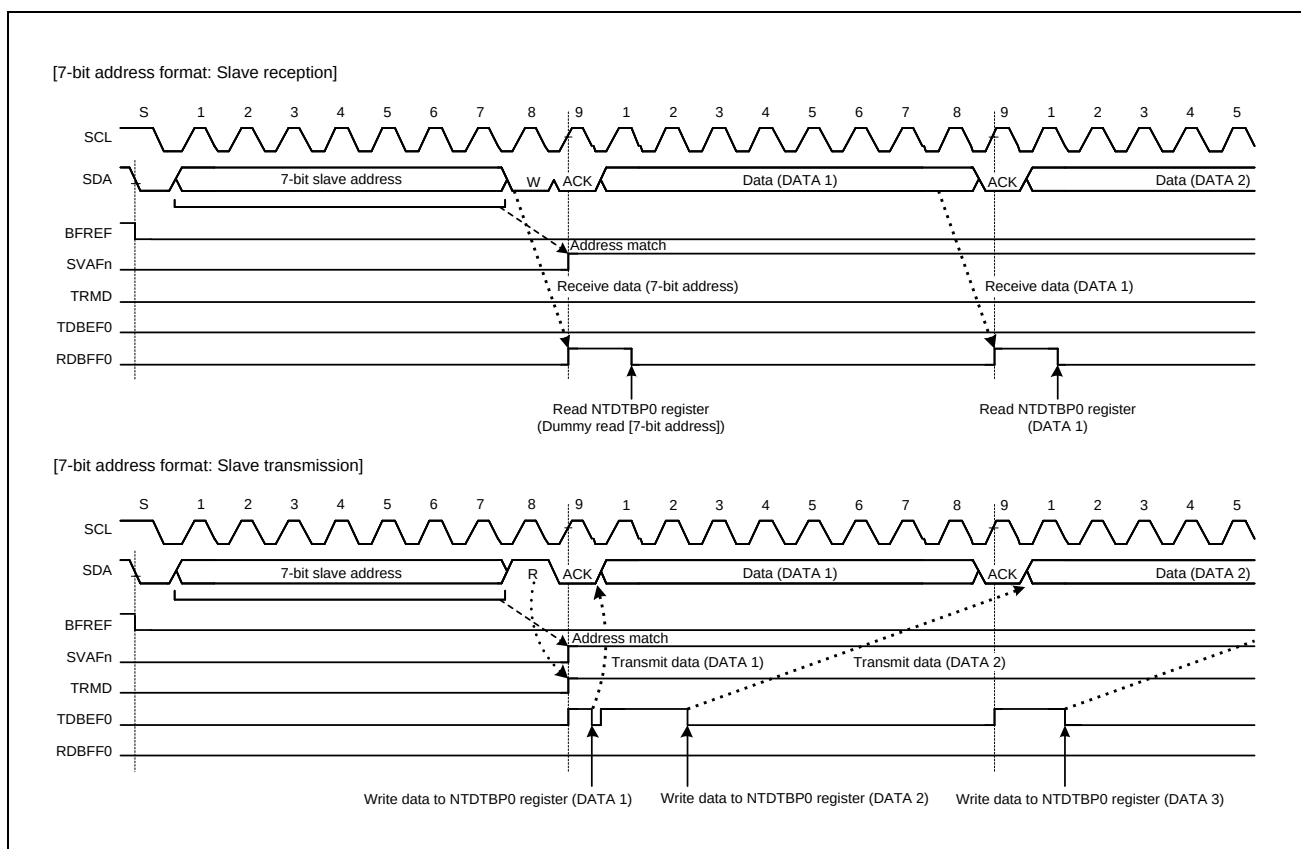


Figure 7.8-58 SVAFn flag set timing with 7-bit address format selected

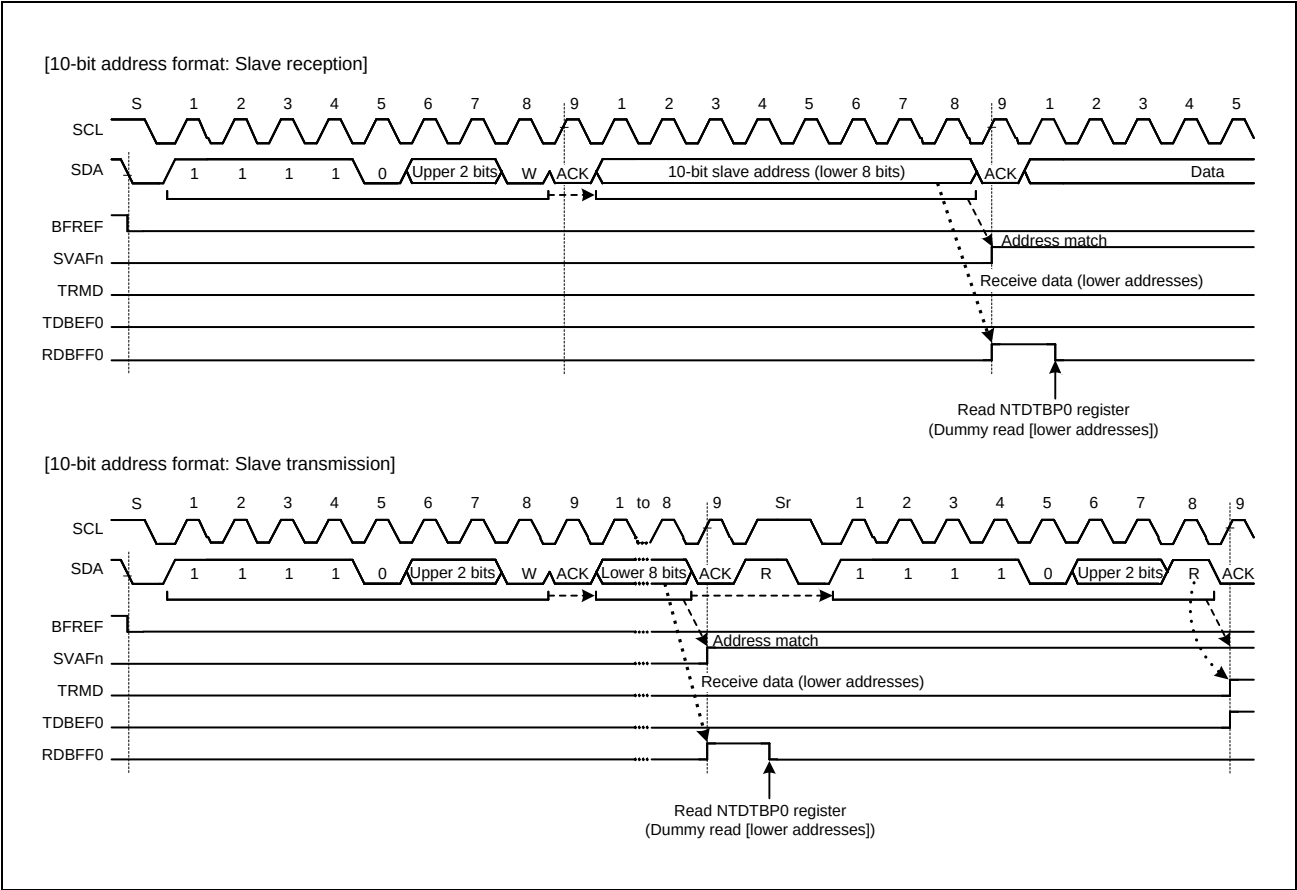


Figure 7.8-59 SVAFn flag set timing with 10-bit address format selected

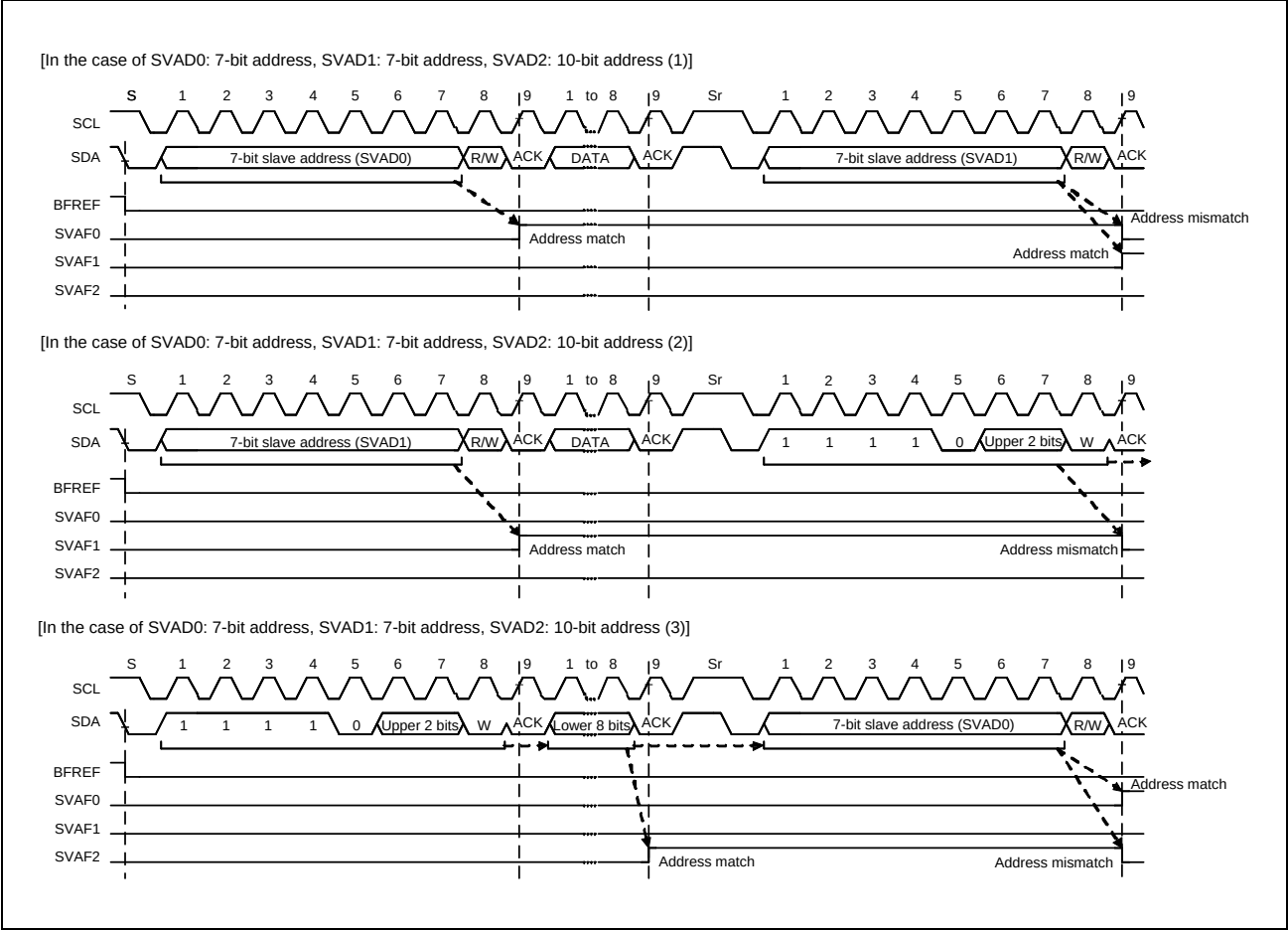


Figure 7.8-60 SVAFn flag set/clear timing with 7-bit/10-bit address formats mixed

(b) Detection of the General Call Address [I²C mode]

I3C has a facility for detecting the general call address (0000 000 + 0 (write)). This is enabled by setting the SVCTL.GCAE bit to 1b.

If the address received after a START or Repeated START condition is issued is 0000 000 + 1 (read) (start byte), I3C recognizes this as the address of a slave device with an all-zero address but not as the general call address.

When I3C detects the general call address, both the SVST.GCAF flag and the NTST.RDBFF0 flag are set to 1b on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (INT_ri3c_rx_n). The value of the GCAF flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

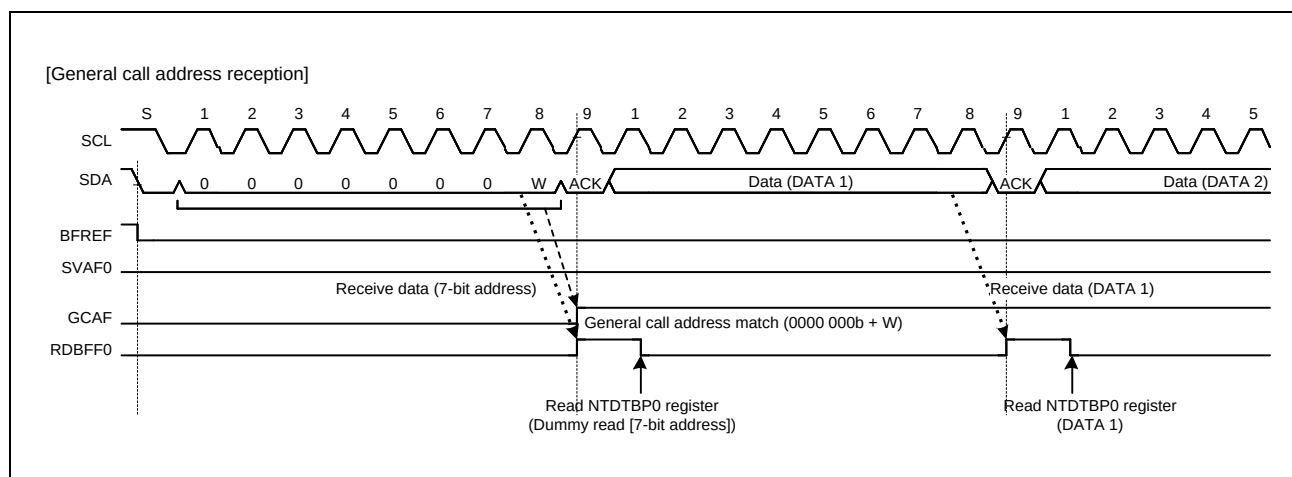


Figure 7.8-61 Timing of GCAF flag setting during reception of general call address

(c) Device-ID Address Detection [I²C mode]

I3C module has a facility for detecting device-ID addresses conformant with the I²C-bus specification (Rev.03). When I3C receives 1111 100b as the first byte after a START condition or Repeated START condition was issued with the SVCTL.DVIDE bit set to 1b, I3C recognizes the address as a device ID, sets the SVST.DVIDF flag to 1b on the rising edge of the ninth SCL clock cycle when the following R/W# bit = 0b, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, I3C sets the corresponding SVST.SVAFn flag (n = 0 to 2) to 1b.

After that, when the first byte received after a START or Repeated START condition is issued matches the device ID address (1111 100b) again and the following R/W# bit = 1, I3C does not compare the second and subsequent bytes and sets the NTST.TDBEF0 flag to 1b.

In the device-ID address detection function, I3C sets the DVIDF flag to 0 if a match with I3C's own slave address is not obtained or a match with the device ID address is not obtained after a match with I3C's own slave address and the detection of a Repeated START condition. If the first byte after detection of a START or Repeated START condition matches the device ID address (1111 100b) and the R/W# bit = 0b, I3C sets the DVIDF flag to 1b and compares the second and subsequent bytes with I3C's slave address. If the R/W# bit = 1b, the DVIDF flag holds the previous value and I3C does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DVIDF flag after confirming that TDBEF0 flag = 1b.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

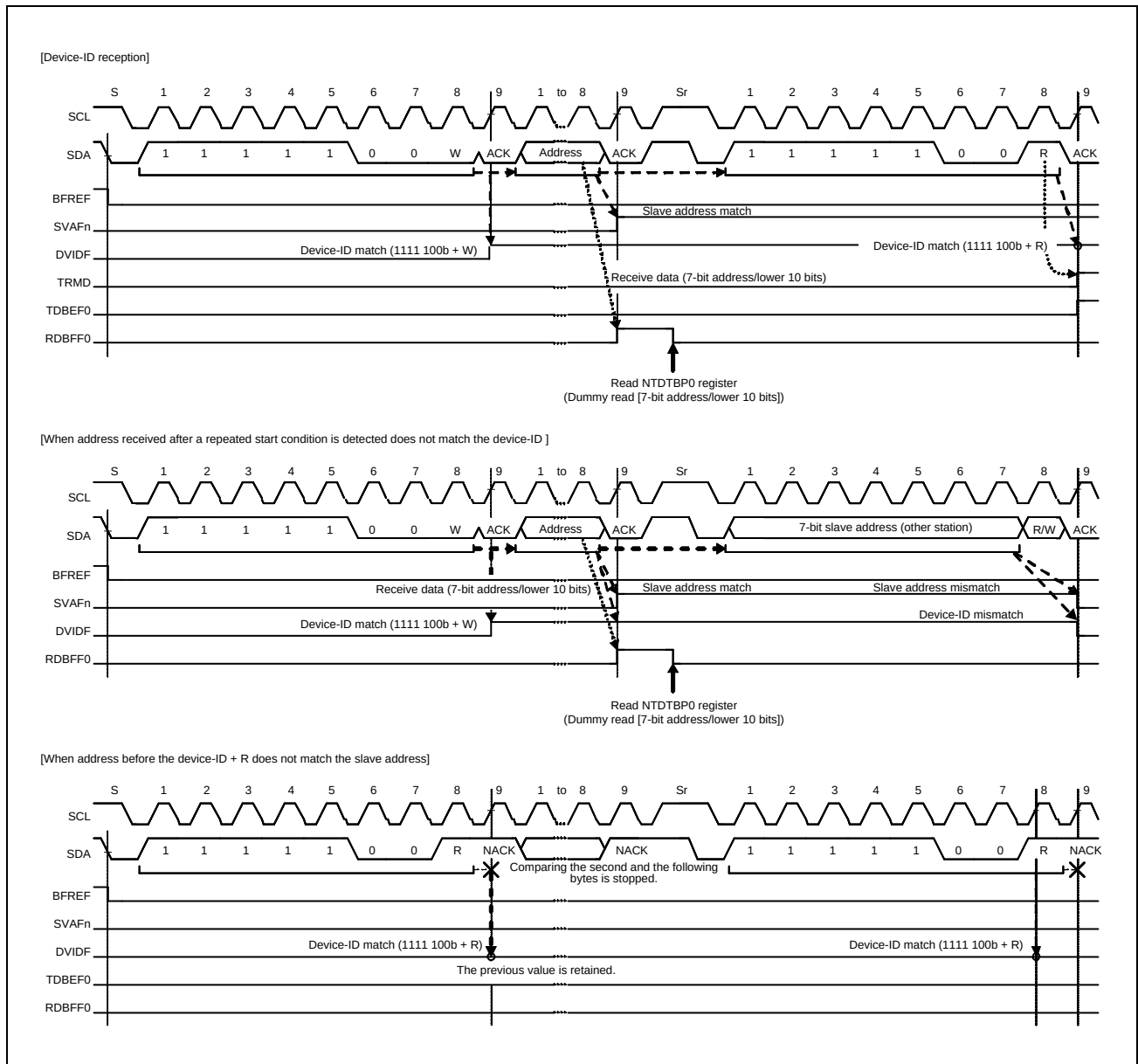


Figure 7.8-62 SVAFn/DVIDF flag set/clear timing during reception of device-ID

(d) Host Address Detection [I²C mode]

I3C has a function to detect the host address while the SMBus is operating. When the SVCTL.HOAE bit is set to 1b while the BFCTL.SMBS bit = 1b, I3C can detect the host address (0001 000) in slave receive mode (bits CRMS and TRMD in the PRSST register = 00b).

When I3C detects the host address, the SVST.HOAF flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the NTST.RDBFF0 flag is set to 1b when the R/W# bit = 0b (Wr bit). This causes a receive data full interrupt (INT_ri3c_rx_n) to be generated. The HOAF flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1b), I3C can also detect the host address. After the host address is detected, I3C operates in the same manner as normal slave operation.

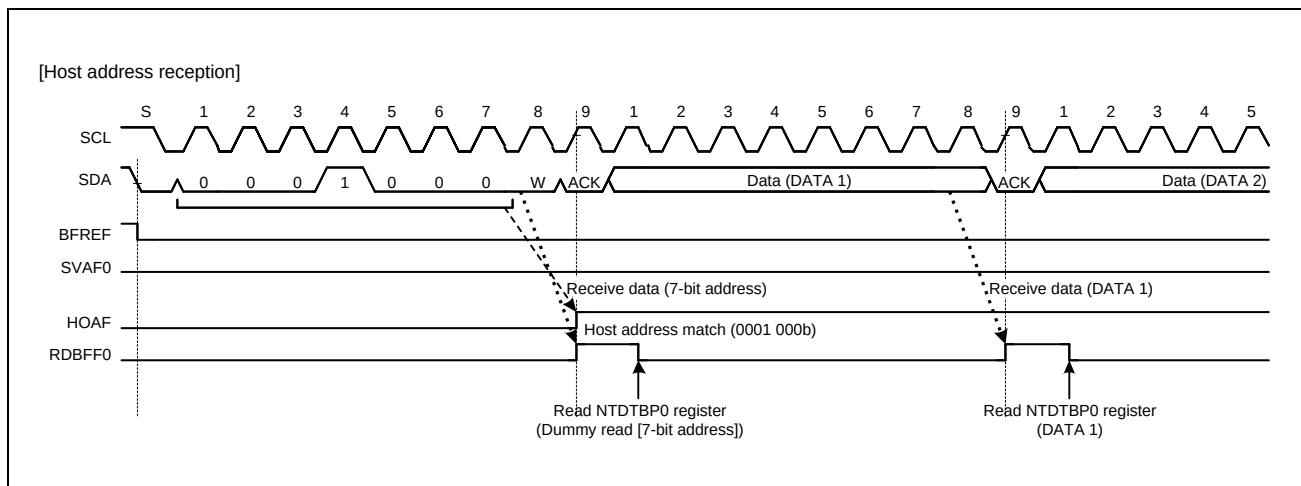


Figure 7.8-63 HOAF flag set timing during reception of host address

(e) Hs-mode master code Detection [I²C mode]

IC has a facility for detecting the Hs-mode master code (0000 1XXXb). When IIC receives the Hs-mode master code (0000 1XXXb) as the first byte after a START condition was issued with the SVCTL.HSMCE bit set to 1b, this module recognizes the address as the Hs-mode master code, sets the SVST.HSMCF flag to 1b on the rising edge of the ninth SCL clock cycle. The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADn.SVAD[9:0] (n = 0 to 2). When IIC detects a match of the set slave address, the corresponding SVST.SVAFn flag (n = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (INT_ri3c_rx_n) or transmit data empty interrupt (INT_ri3c_tx_n) to be generated. The SVAFy flag is used to identify which slave address has been specified. The SVST.HSMCF flag is cleared to 0b when the STOP condition is detected.

NOTE

If the Hs-mode master code (0000 1XXXb) is received with the SVCTL.HSMCE bit set to 0b, other patterns are ignored until the STOP condition is detected.

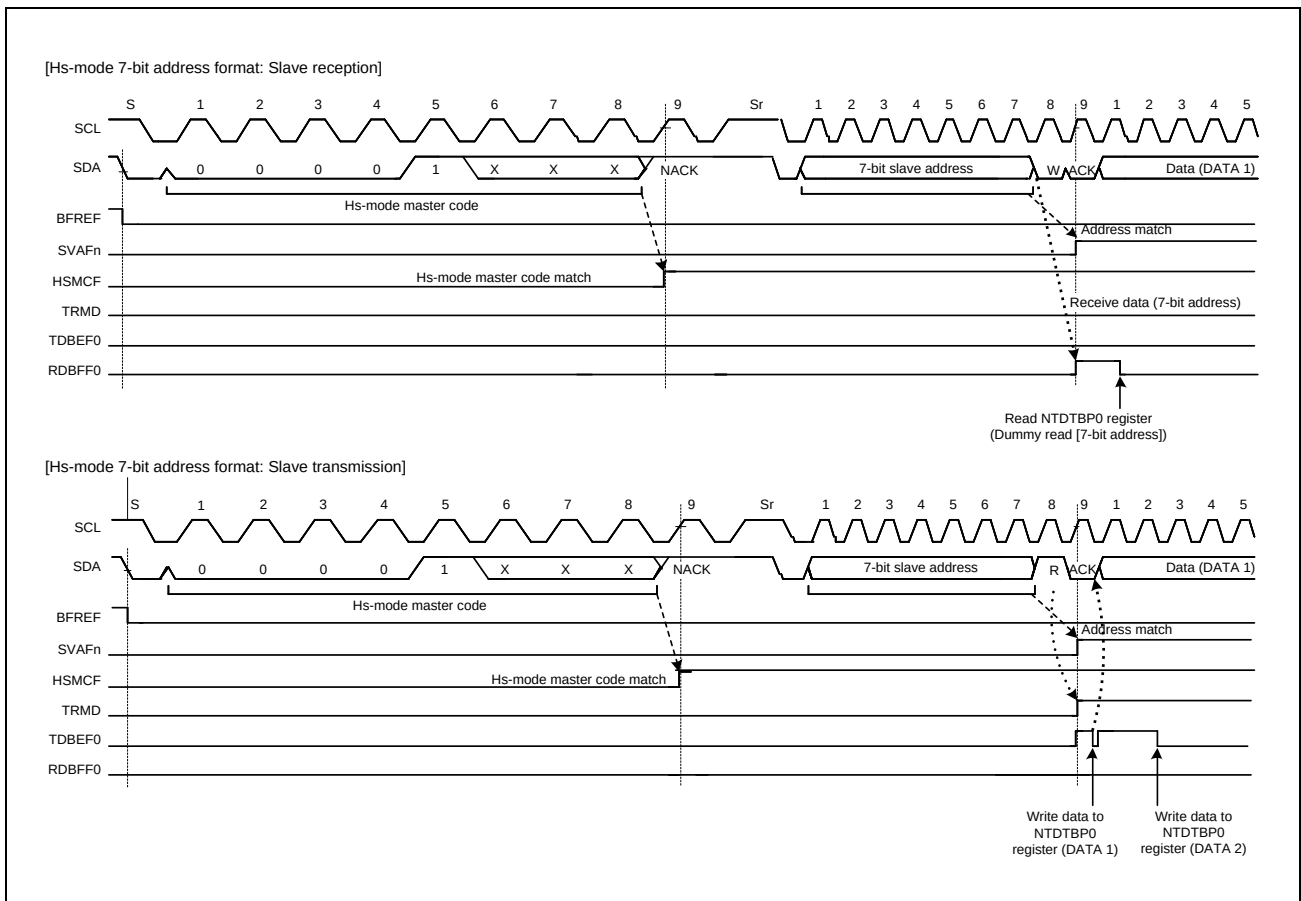


Figure 7.8-64 SVAFn/HSMCF Flag Set Timing during Reception of Hs-mode master code

(f) CCC detection function [I3C mode]**◆ In case of Broadcast CCC**

1. It receives Broadcast Address (7Eh) and R/W# = 0 after START condition or Repeated START.
2. Respond to ACK.
3. Receive Common Command Code (CCC).
4. In accordance with the CCC, the following data is stored. (Storage destination: see **Table 7.8-12**)
5. Store the Receive Status Descriptor into the Receive Status Queue.

◆ In case of Broadcast CCC (ENTDAA)

1. It receives Broadcast Address (7Eh) and R/W# = 0b after START condition.
2. Respond to ACK.
3. Receive ENTDAA.
4. If receives Broadcast Address (7Eh) and R/W# = 1b after Repeated START.
5. When the Dynamic Address is not assigned, ACK response is done.
6. This Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCR[7:0]) and DCR (SVDCT.TDCR[7:0]) are transmitted.
7. When winning the arbitration in a transmission of the above Step 6, the dynamic address following that is received. When losing arbitration in a transmission of the above Step 6, processing of Step 6 is repeated from Step 4.
8. When parity of the Dynamic Address is valid, ACK response is done.
9. When parity of the Dynamic Address is invalid, NACK replies, and repeat the process from Steps 4 to 7.
10. SDATBAS0.SDDYAD[6:0] is renewed and the SVDVAD0.SDYADV bit is set to 1b.
11. Upon detecting the STOP condition, Store the Receive Status Descriptor into the Receive Status Queue.

◆ In case of Direct Write CCC

1. It receives Broadcast Address (7Eh) and R/W# = 0b after START condition or Repeated START.
2. Respond to ACK.
3. Receive Common Command Code (CCC).
4. Receive Dynamic Address and R/W# = 0b after Repeated START.
5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
If they do not match, it responds with NACK and waits for Repeated START or STOP.
6. In accordance with the CCC, the following data is stored. (Storage destination: see **Table 7.8-12**)
7. Store the Receive Status Descriptor into the Receive Status Queue.

◆ In case of Direct Read CCC

1. It receives Broadcast Address (7Eh) and R/W# = 1b after START condition or Repeated START.
2. Respond to ACK.
3. Receive Common Command Code (CCC).

4. Receive Dynamic Address and R/W# = 1b after Repeated START.
5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
If they do not match, it responds with NACK and waits for Repeated START or STOP.
6. Respond from SFR according to CCC. (Responding CCC: see **Table 7.8-12**)
7. Store the Receive Status Descriptor into the Receive Status Queue.

Table 7.8-12 Common command code operation

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
00h	Broadcast	ENEC	Yes	—	SFR
01h	Broadcast	DISEC	Yes	—	SFR
02h	Broadcast	ENTAS0	No	—	SFR
03h	Broadcast	ENTAS1	No	—	SFR
04h	Broadcast	ENTAS2	No	—	SFR
05h	Broadcast	ENTAS3	No	—	SFR
06h	Broadcast	RSTDAA	No	—	SFR
07h	Broadcast	ENTDAA	Yes	Yes	SFR
08h	Broadcast	DEFSLVS	Yes	—	FIFO
09h	Broadcast	SETMWL	Yes	—	SFR
0Ah	Broadcast	SETMRL	Yes	—	SFR
0Bh	Broadcast	ENTTM	Yes	—	SFR
28h	Broadcast	SETXTIME	Yes	—	FIFO
29h	Broadcast	SETAASA	No	—	SFR
80h	Direct Write	ENEC	Yes	—	SFR
81h	Direct Write	DISEC	Yes	—	SFR
82h	Direct Write	ENTAS0	No	—	SFR
83h	Direct Write	ENTAS1	No	—	SFR
84h	Direct Write	ENTAS2	No	—	SFR
85h	Direct Write	ENTAS3	No	—	SFR
86h	Direct Write	RSTDAA	No	—	SFR
87h	Direct Write	SETDASA	Yes	—	SFR
88h	Direct Write	SETNEWDA	Yes	—	SFR
89h	Direct Write	SETMWL	Yes	—	SFR
8Ah	Direct Write	SETMRL	Yes	—	SFR
8Bh	Direct Read	GETMWL	—	Yes	SFR
8Ch	Direct Read	GETMRL	—	Yes	SFR
8Dh	Direct Read	GETPID	—	Yes	SFR
8Eh	Direct Read	GETBCR	—	Yes	SFR
8Fh	Direct Read	GETDCR	—	Yes	SFR
90h	Direct Read	GETSTATUS	—	Yes	SFR
91h	Direct Read	GETACCMST	—	Yes	SFR
94h	Direct Read	GETMXDS	—	Yes	SFR
98h	Direct Write	SETXTIME	Yes	—	FIFO
99h	Direct Read	GETXTIME	—	Yes	SFR

(5) Arbitration-Lost Detection [I²C mode]

In addition to the normal arbitration-lost detection function defined by the I²C-bus specification, the I3C has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

(a) Master Arbitration-Lost Detection (MALE Bit)

The I3C drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a START condition, this module causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the CNDCTL.STCND bit is set to 1 while the BCST.BFREF flag is 0 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.

When a start condition is issued successfully, if the data for transmission including the address bits (the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output, that is, the SDA30 pin is in the high-impedance state) and the low level is detected on the SDA line, the I3C loses in arbitration.

I3C detects master arbitration-lost when the following conditions are met while the BSTE.ALE bit = 1b and the BFCTL.MALE bit = 1b (master arbitration-lost detection enabled).

If arbitration of mastership is lost, I3C immediately enters slave receive mode.

If a slave address (including the general call address) matches its own address at this time, I3C continues in slave operation.

[Conditions for master arbitration-lost]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a START condition was issued by setting the CNDCTL.STCND bit to 1b while the BCST.BFREF flag was set to 1b (erroneous issuing of a START condition)
- Setting of the CNDCTL.STCND bit to 1b (START condition double-issue error) while the BFREF flag is set to 0

Remark I3C does not issue a START condition.

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (bits CRMS and TRMD in the PRSST register = 11)

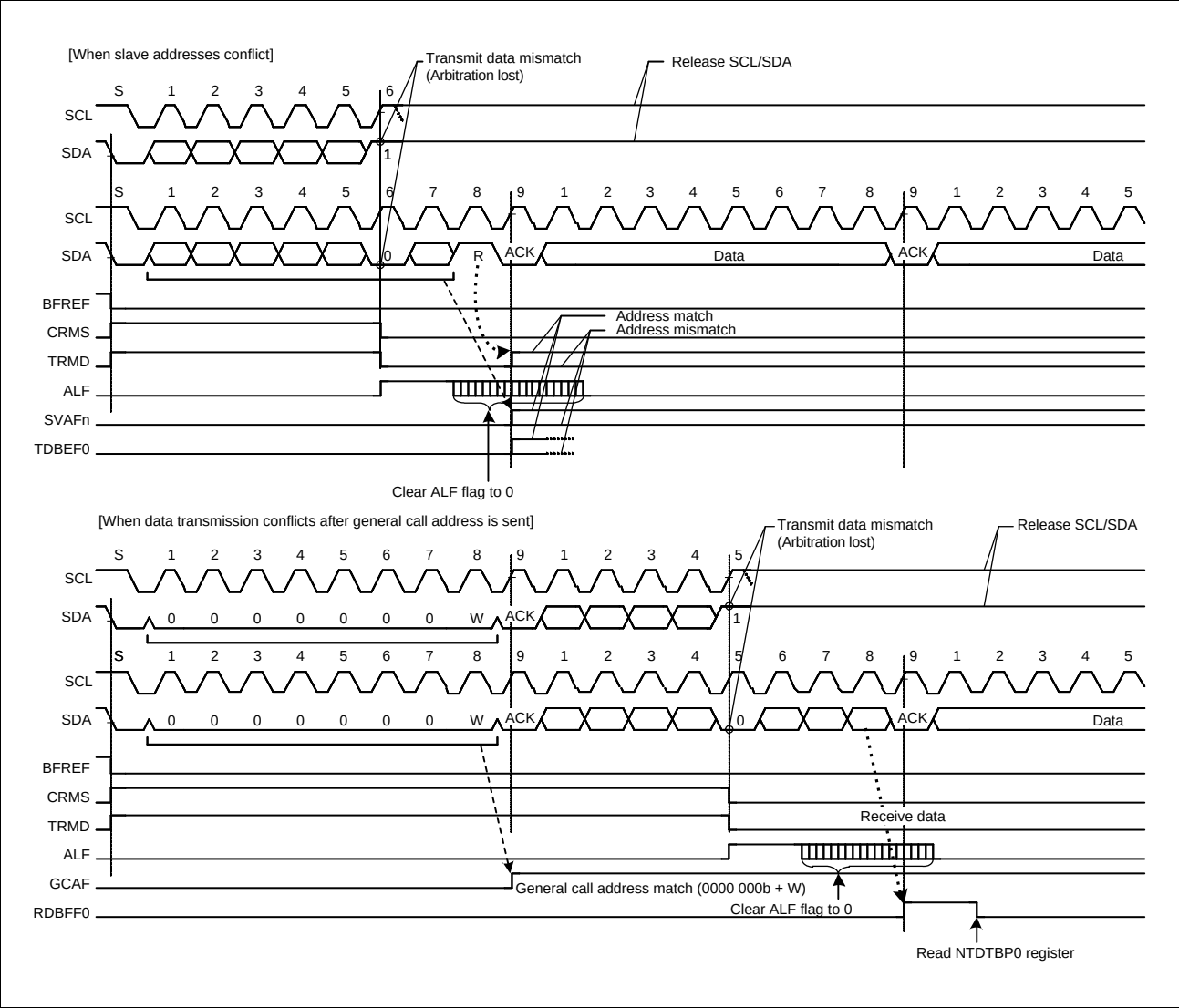


Figure 7.8-65 Examples of master arbitration-lost detection (MALE = 1b)

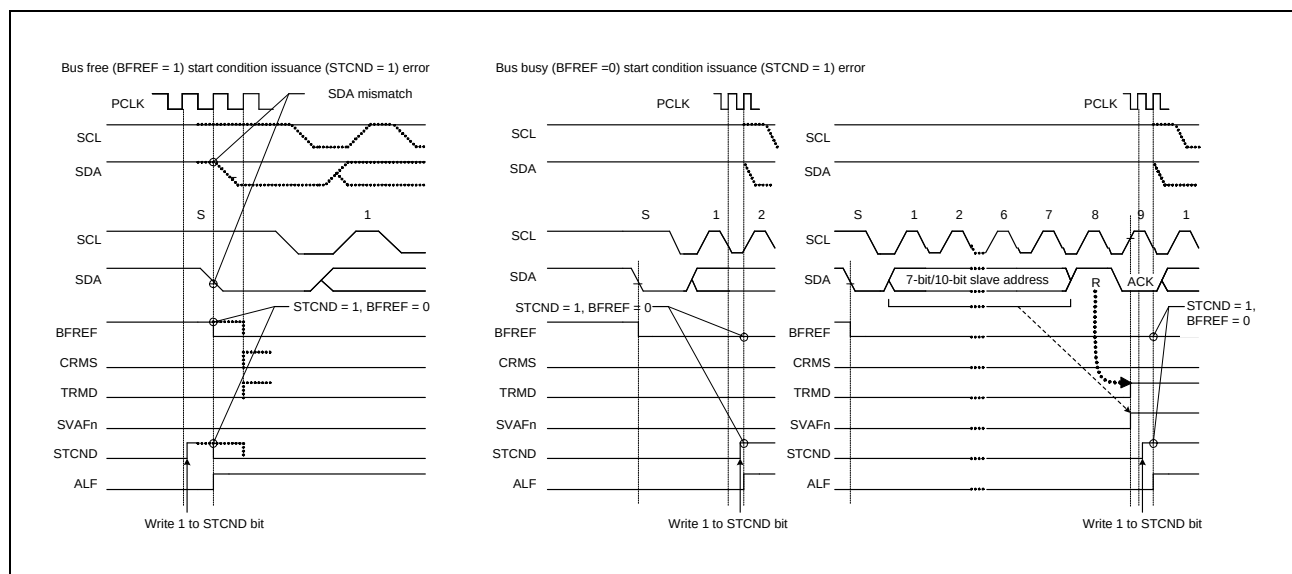


Figure 7.8-66 Arbitration-lost detection when a START condition is issued (MALE = 1b)

(b) Arbitration-Lost Detection during NACK Transmission (NALE Bit)

The I3C has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device.

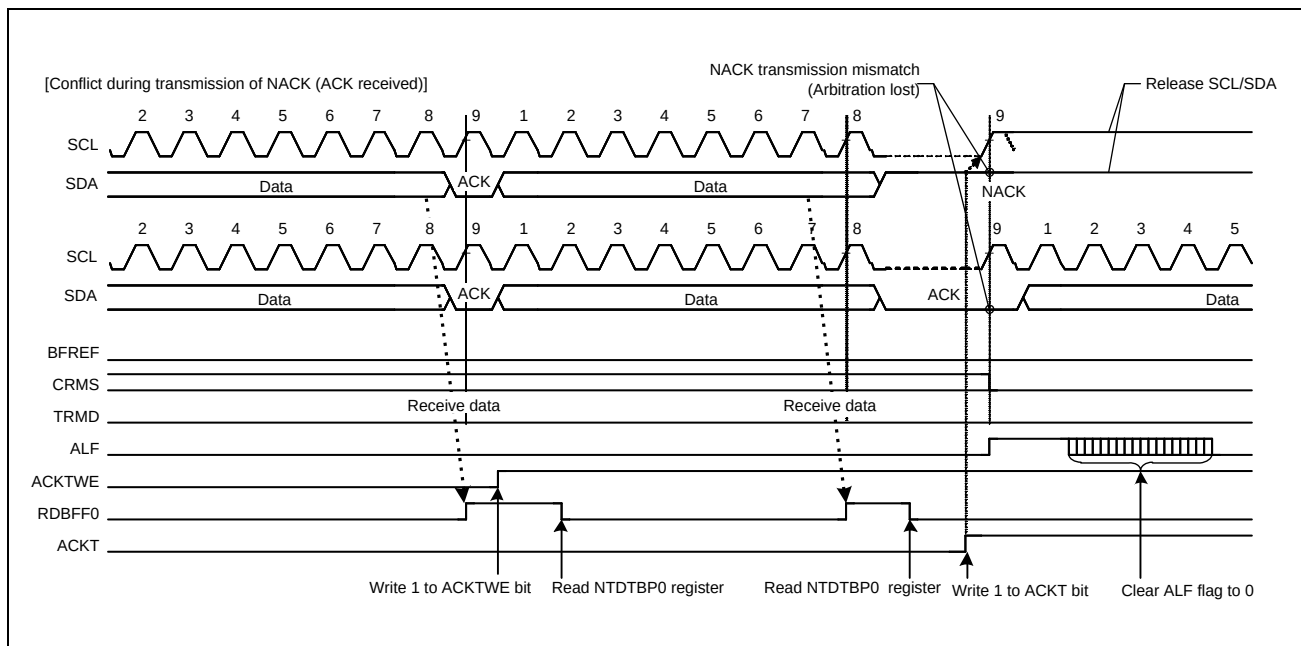


Figure 7.8-67 Example of arbitration-lost detection during transmission of NACK (NALE = 1b)

The following section explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. In this example, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When this module receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, this module is immediately released from the slave-matched state and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing)

necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign Address command.

The I3C detects arbitration-lost during transmission of NACK when the following condition is met while the BSTE.ALE bit = 1b and the BFCTL.NALE bit = 1b (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (ACKCTL.ACKT bit = 1b)

(c) Slave Arbitration-Lost Detection (SALE Bit)

The I3C has a function to cause arbitration to be lost if the data for transmission (the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output, that is, the SDA pin is in the high-impedance state and the low level is detected on the SDA line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

If arbitration is lost during transmission of DATA, this module is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of FFh).

The I3C detects slave arbitration-lost when the following condition is met while the BSTE.ALE bit = 1b and the BFCTL.SALE bit = 1b (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in slave transmit mode (bits CRMS and TRMD in the PRSST register = 01b).

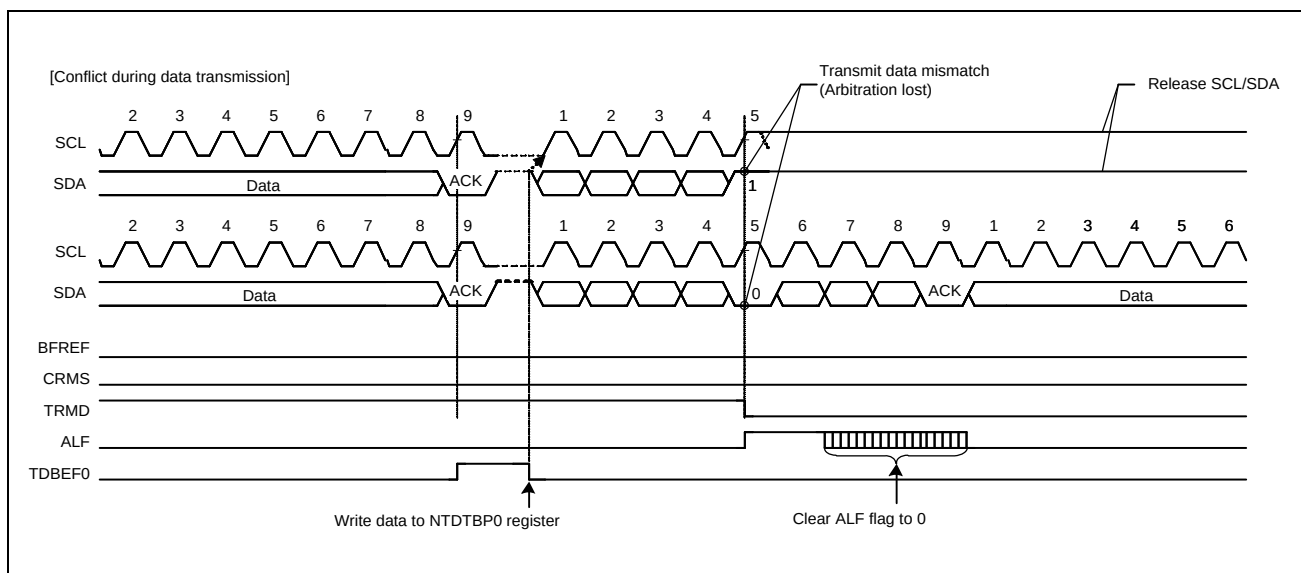


Figure 7.8-68 Example of slave arbitration-lost detection (SALE = 1)

(6) Clock Stretching [I²C mode]

(a) Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the I²C bus transmit data register (NTDTBP0) with I3C in transmission mode (PRST.TRMD = 1b), the SCL line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a START condition or Repeated START condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

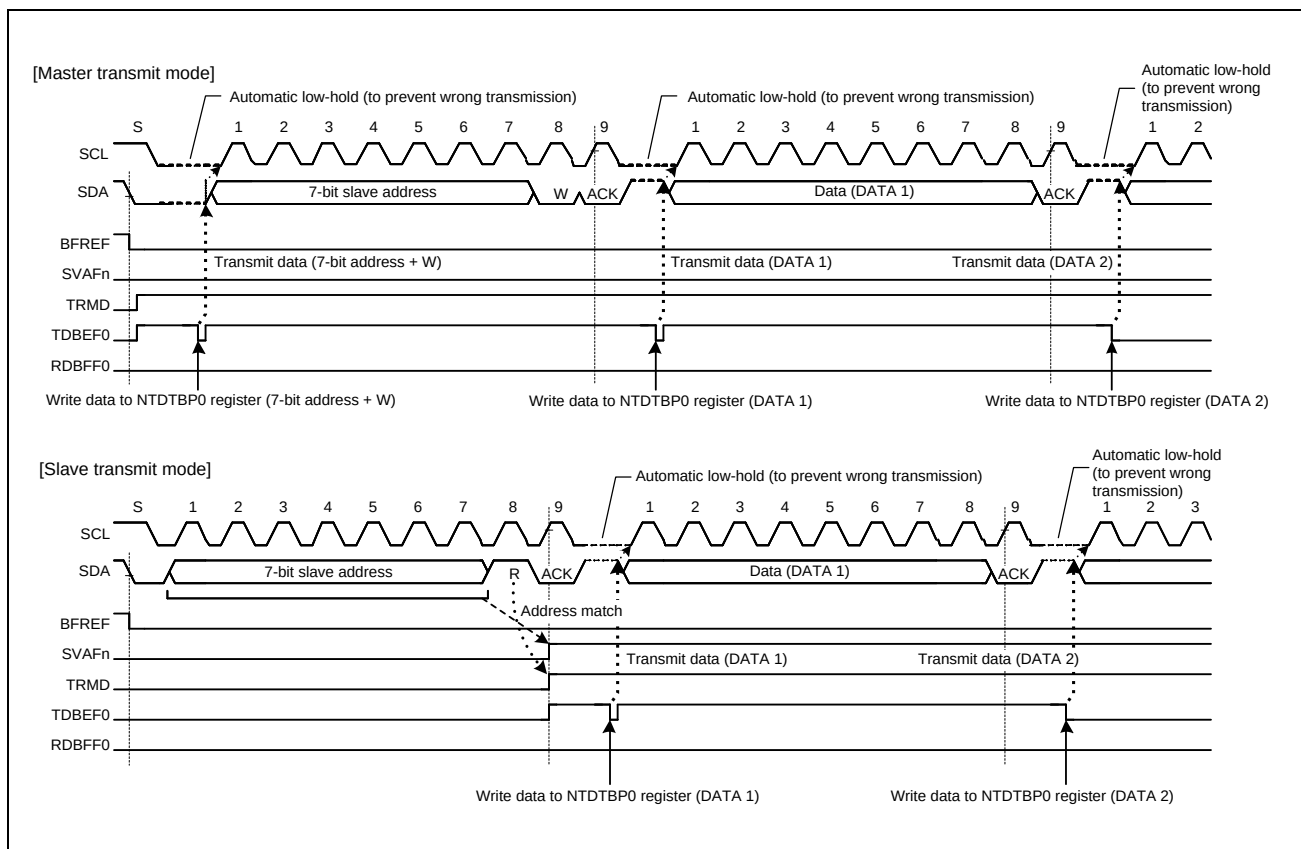


Figure 7.8-69 Automatic low-hold operation in transmit mode

(b) NACK Reception Transfer Abort Function

I3C has a function to abort transfer operation when NACK is received in transmit mode (PRST.TRMD = 1b). This function is enabled when the BSTE.NACKDE bit is set to 1b (transfer abort enabled). If the next transmit data has already been written (NTST.TDBEF0 = 0b) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically aborted. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0b.

If the transfer operation is aborted by this function (BST.NACKDF = 1b), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKDF flag to 0b. In master transmit mode, restore operation using either of the methods below:

- After issuing a Repeated START condition, set the NACKDF flag to 0b
- After issuing a STOP condition, set the NACKDF flag to 0b and then issue a START condition

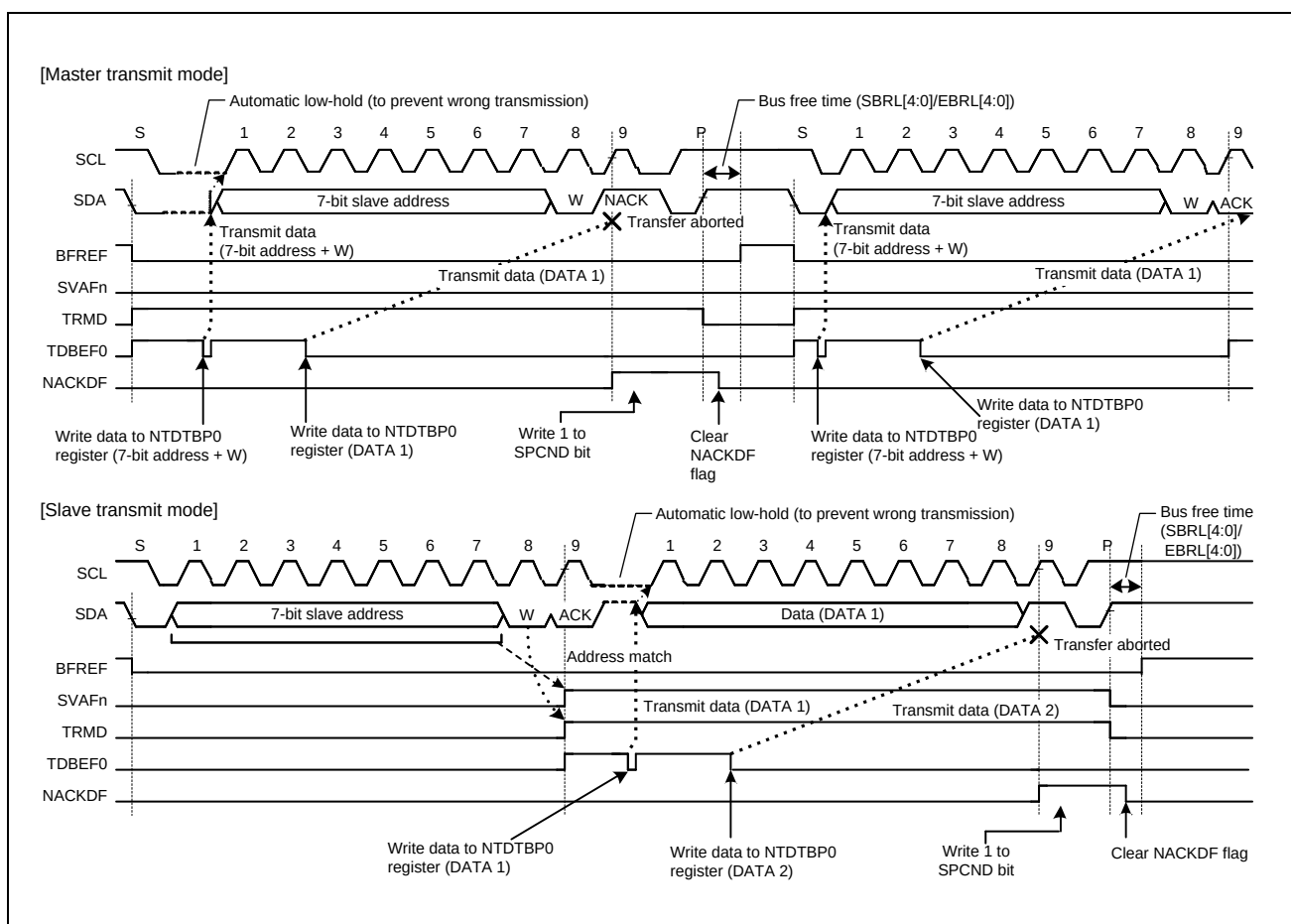


Figure 7.8-70 Abort of data transfer when NACK is received (NACKDE = 1b)

(b-1) Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (NTDTBP0) read is delayed for a period of one transfer frame or more with receive data full (NTST.RDBFF0 = 1b) in receive mode (PRSS.TRMD = 0b), I3C holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, I3C's own slave address or another slave address is received after a STOP condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RWE and ACKTWE bits in SCSTRCTL.

(b-2) 1-Byte Receive Operation and Automatic Low-Hold Function Using the RWE Bit

When the SCSTRCTL.RWE bit is set to 1b, I3C performs 1-byte receive operation using the RWE bit function.

Furthermore, when the SCSTRCTL.ACKTWE bit = 0b, I3C automatically sends the ACKCTL.ACKT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the RWE bit function. This low-hold is released by reading data from NTDTBP0, which enables bitwise receive operation.

The RWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(b-3) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the ACKTWE Bit

When the SCSTRCTL.ACKTWE bit is set to 1b, I3C performs 1-byte receive operation using the ACKTWE bit function.

When the ACKTWE bit is set to 1b, the NTST.RDBFF0 flag (receive data full) is set to 1b at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKCTL.ACKT bit, but cannot be released by reading data from NTDTBP0, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The ACKTWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

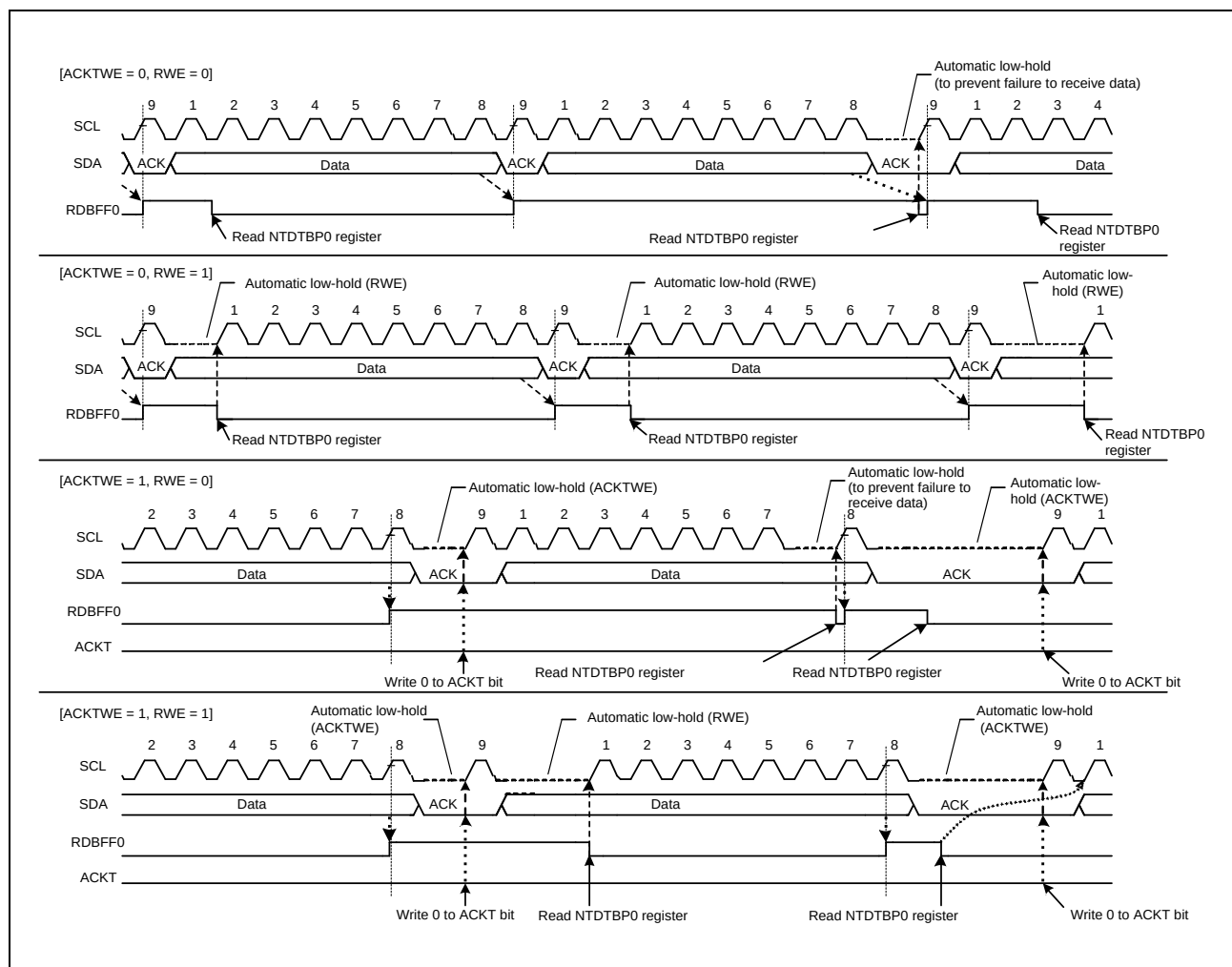


Figure 7.8-71 Automatic low-hold operation in receive mode (using ACKTWE and RWE bits)

(7) Clock Stalling [I3C mode]

I3C has the function of stalling the SCL during the SCL Low period.

The SCL stall control is described in the table below.

Table 7.8-13 I3C clock stalling

Clock stalling condition	Clock stalling control	Clock stalling period
I3C Transfer, ACK/NACK Phase	SCSTLCTL.ACKPE bit setting	During the count period of SCSTLCTL.STLCYC [15: 0] value
	Transmit Data FIFO Empty	Until data is written to the Transmit FIFO
	Receive Data FIFO Full	Until data is read from the Receive FIFO
I3C Write Data Transfer, Parity Bit	SCSTLCTL.PARPE bit setting	During the count period of SCSTLCTL.STLCYC [15: 0] value
	Transmit Data FIFO Empty	Until data is written to the Transmit FIFO
I3C Read Transfer, Transition Bit	Receive Data FIFO Full	Until data is read from the Receive FIFO
Assigned Address Phase	SCSTLCTL.AAPE bit setting	During the count period of SCSTLCTL.STLCYC [15: 0] value

The following figure shows the stalling timing of each Condition.

(a) I3C Transfer, ACK/NACK Phase

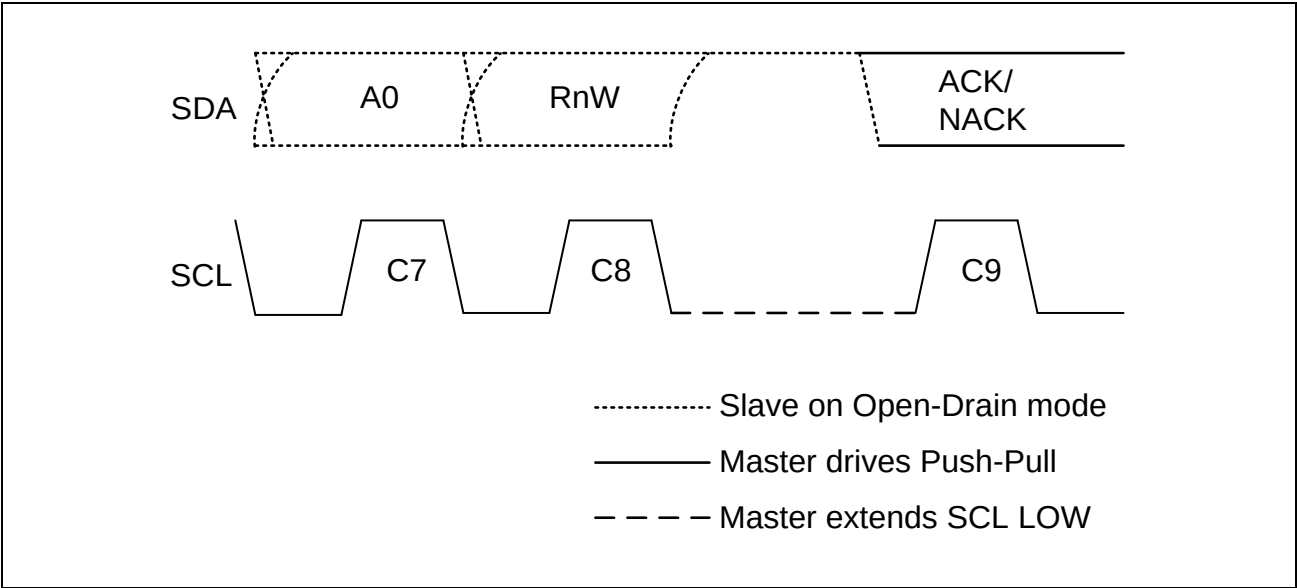


Figure 7.8-72 Master clock stalling in ACK phase

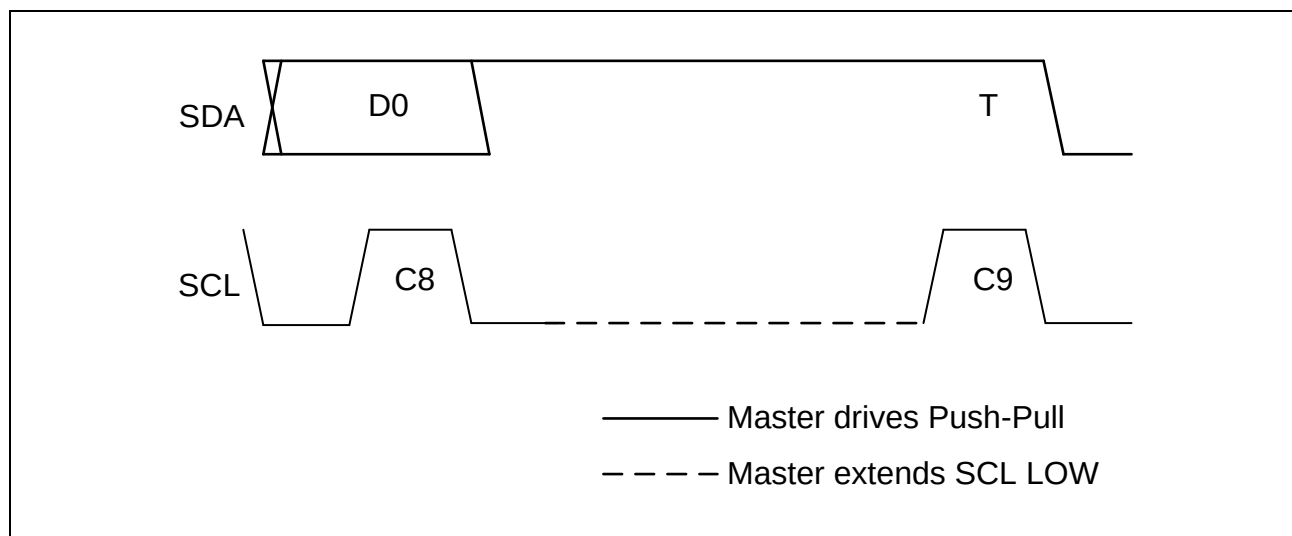
(b) I3C Write Data Transfer, Parity Bit

Figure 7.8-73 Master clock stalling in write parity bit

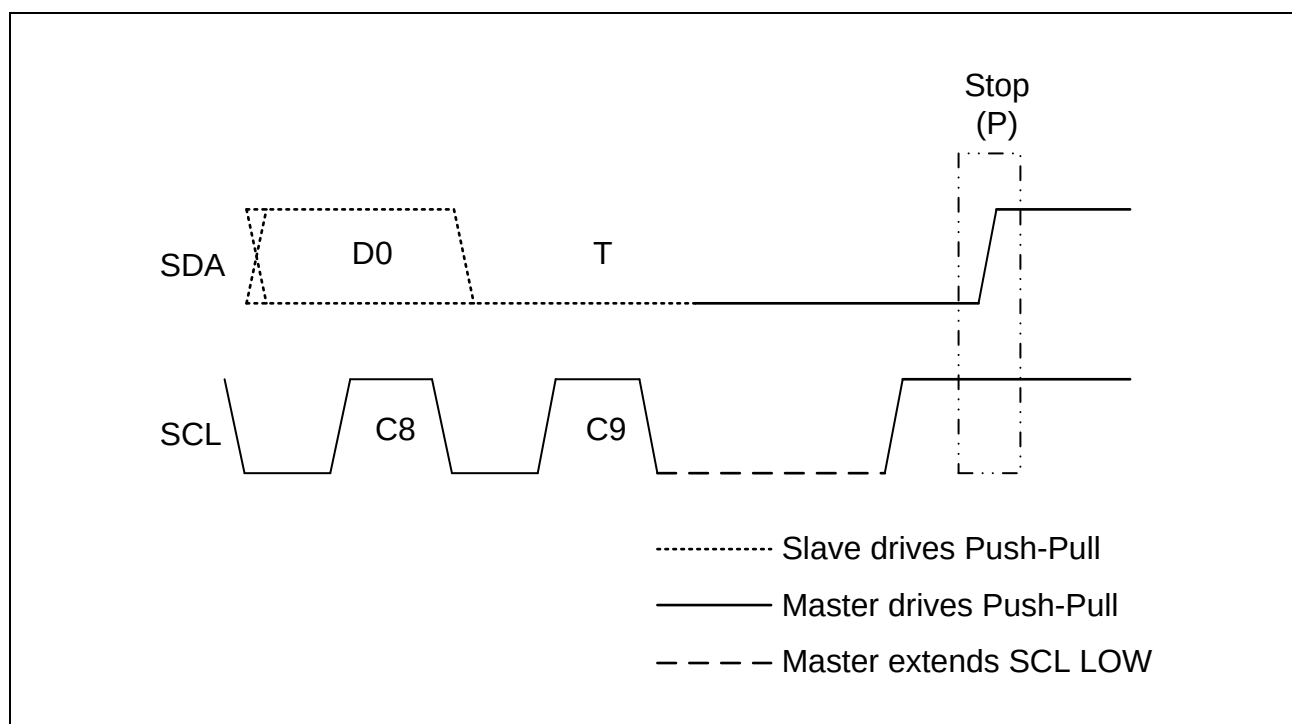
(c) I3C Read Transfer, Transition Bit

Figure 7.8-74 Master clock stalling in T-bit before next read data

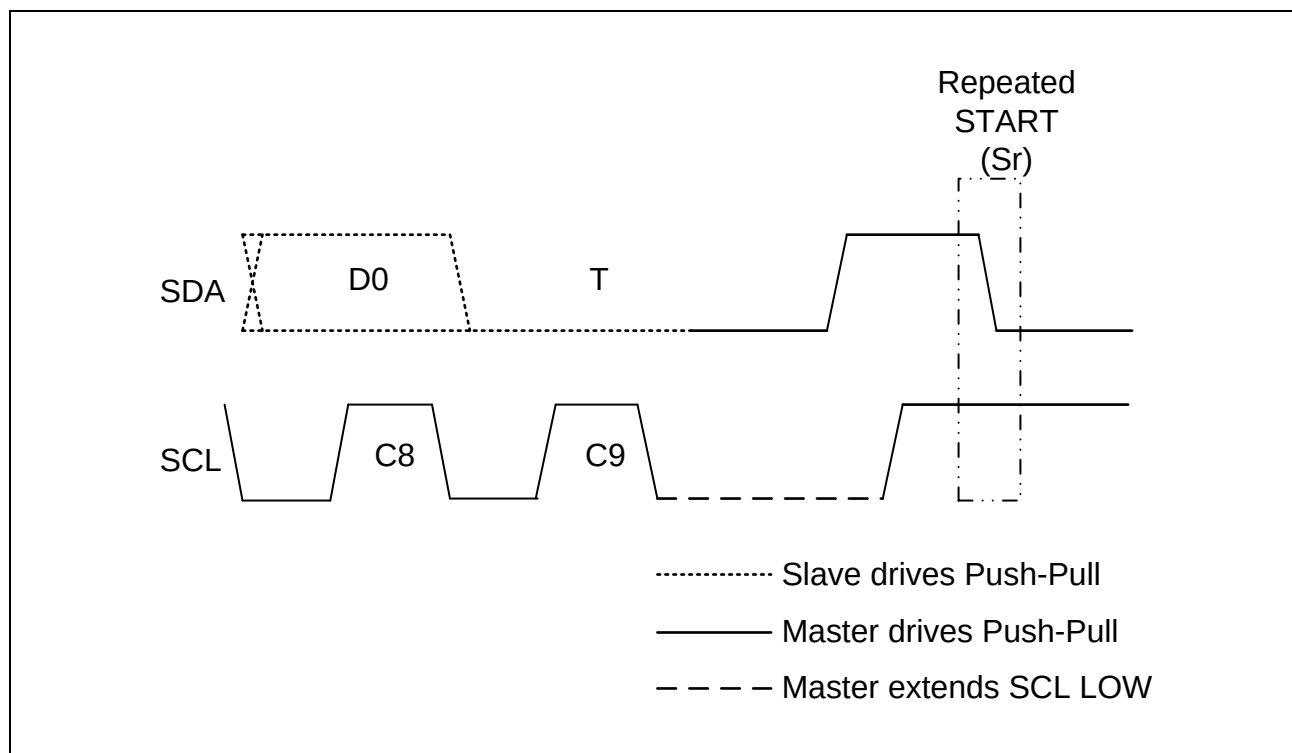


Figure 7.8-75 Master clock stalling in T-bit before STOP

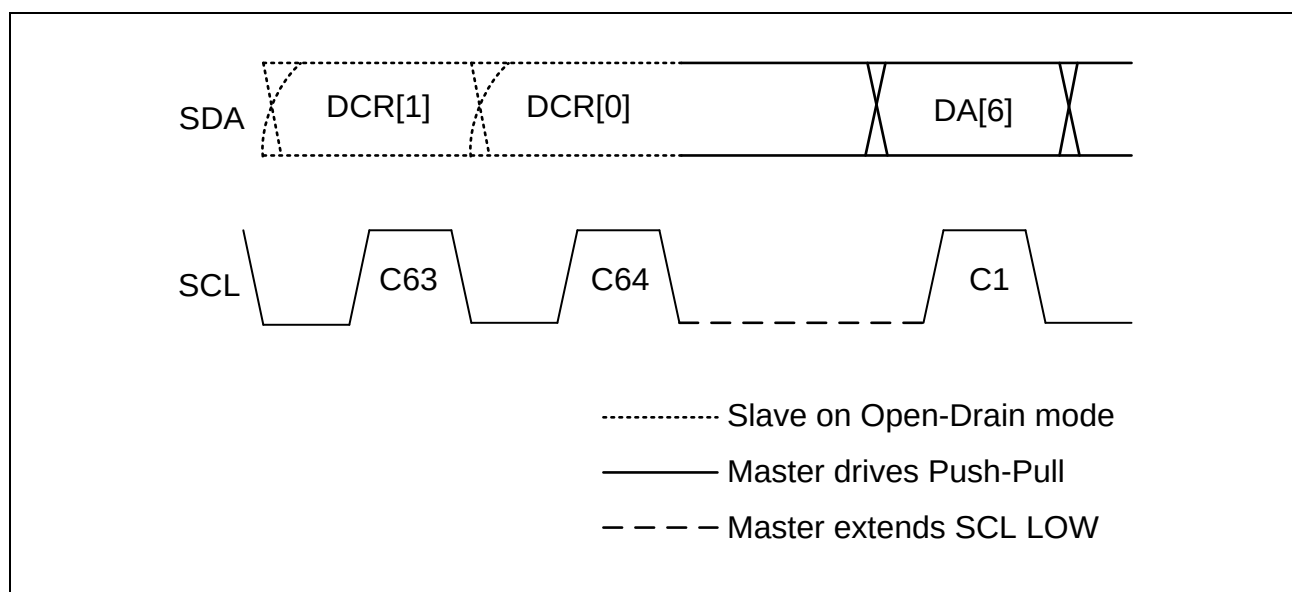


Figure 7.8-76 Master clock stalling in low T-bit before repeated START

(d) Dynamic Address Assignment, First Bit of Assigned Address

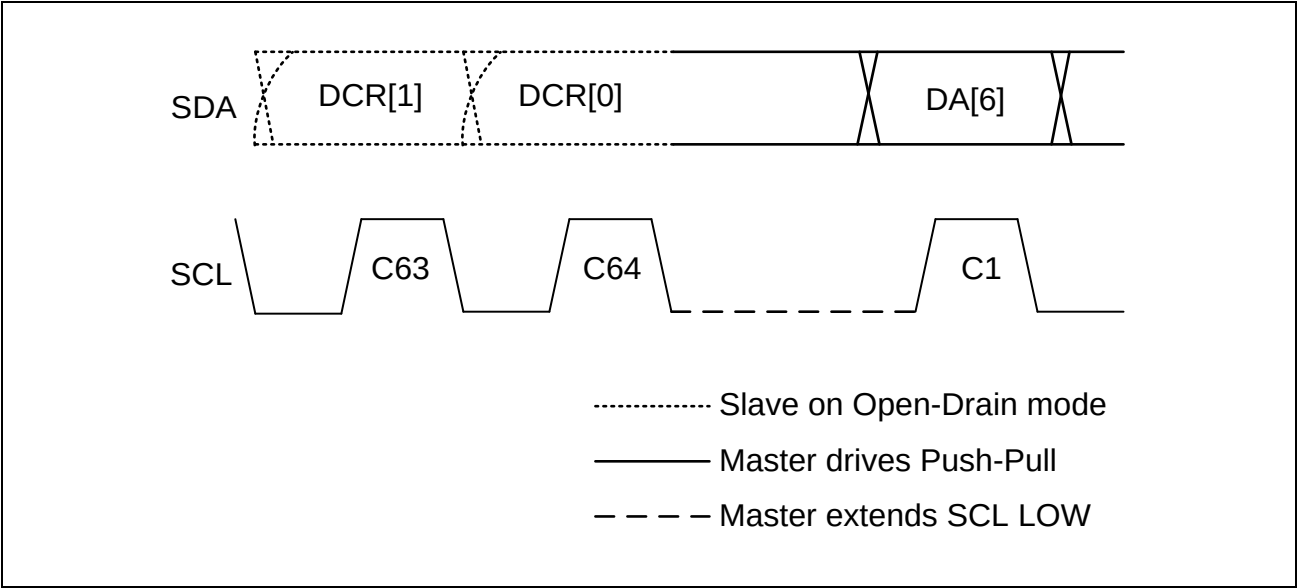


Figure 7.8-77 Master clock stalling in dynamic address first bit

(8) In-Band Interrupt [I3C mode]

I3C detects In-Band Interrupt in the arbitrated Address Header following a START condition (but not following a Repeated START). If START Request (SDA Low Drive) is issued from Slave Device, I3C drives SCL low and completes START condition. After that, it supplies SCL and receives In-Band Interrupt Request.

The In-Band Interrupt to be detected is classified into the following three types.

- Slave Interrupt Request
- Mastership Request
- Hot-Join Event

The operation when detecting each In-Band Interrupt is described below.

(a) Slave Interrupt Request

1. Detect Slave Address with RnW bit High in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBASm register).
3. When it does not match DAT.DVDYAD[7:0]:
Responds NACK, then issues the STOP condition. When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 1b:
It operates in the following order:
 - (a) Responds NACK.
 - (b) Issues Repeated START condition, then automatically issues Direct DISEC CCC to the detected Slave.
 - (c) Issues the STOP condition.When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 0b:
Responds ACK.
4. When DAT.DVIBIPL = 0b:
Issues the STOP condition.
When DAT.DVIBIPL = 1b:
Drives the SCL to receive the IBI Data from the Slave following the ACK response and receives IBI Data.
It stores the received IBI Data into the IBI Data Queue.
Each time IBI Data of the size set by the NQTHCTL.IBIDSSZ[7:0] bits is received, the IBI Status Descriptor is stored in the IBI Status Queue.
5. After detection of Low of T-bit following IBI Data, issues STOP condition.
6. After issues of STOP condition NACK response:
 - If IBINCTL.NRSIRCTL = 0b, the IBI Status Descriptor is not stored into the IBI Status Queue.
 - If IBINCTL.NRSIRCTL = 1b, the IBI Status Descriptor is stored into the IBI Status Queue.ACK response:
Stores the IBI Status Descriptor into the IBI Status Queue.

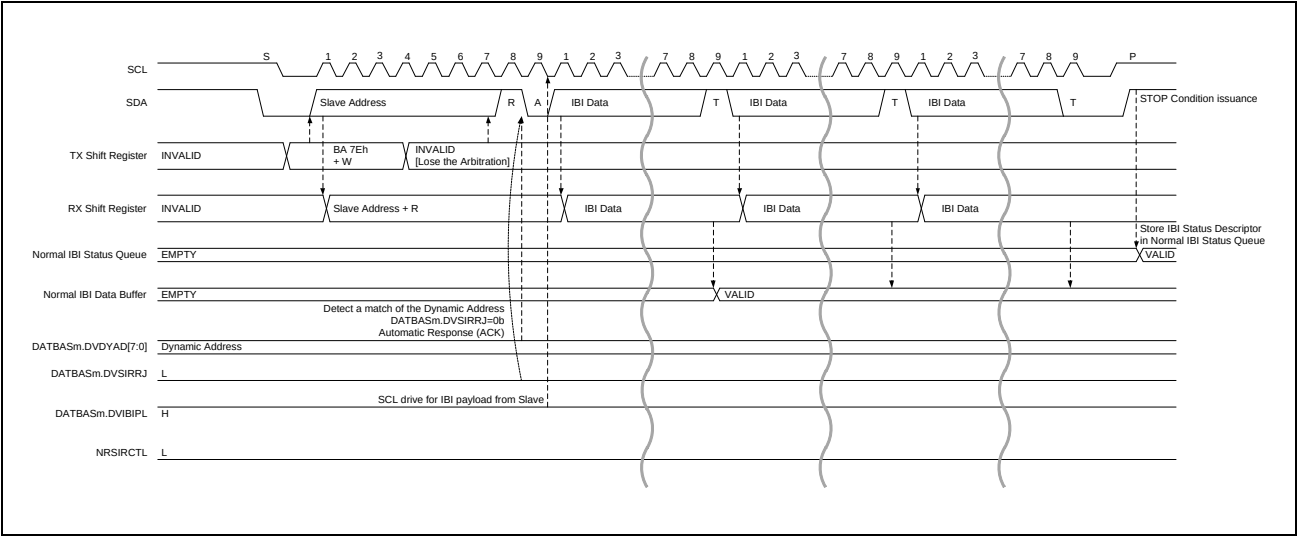


Figure 7.8-78 Slave interrupt request: ACK and DVIBIPL = 1

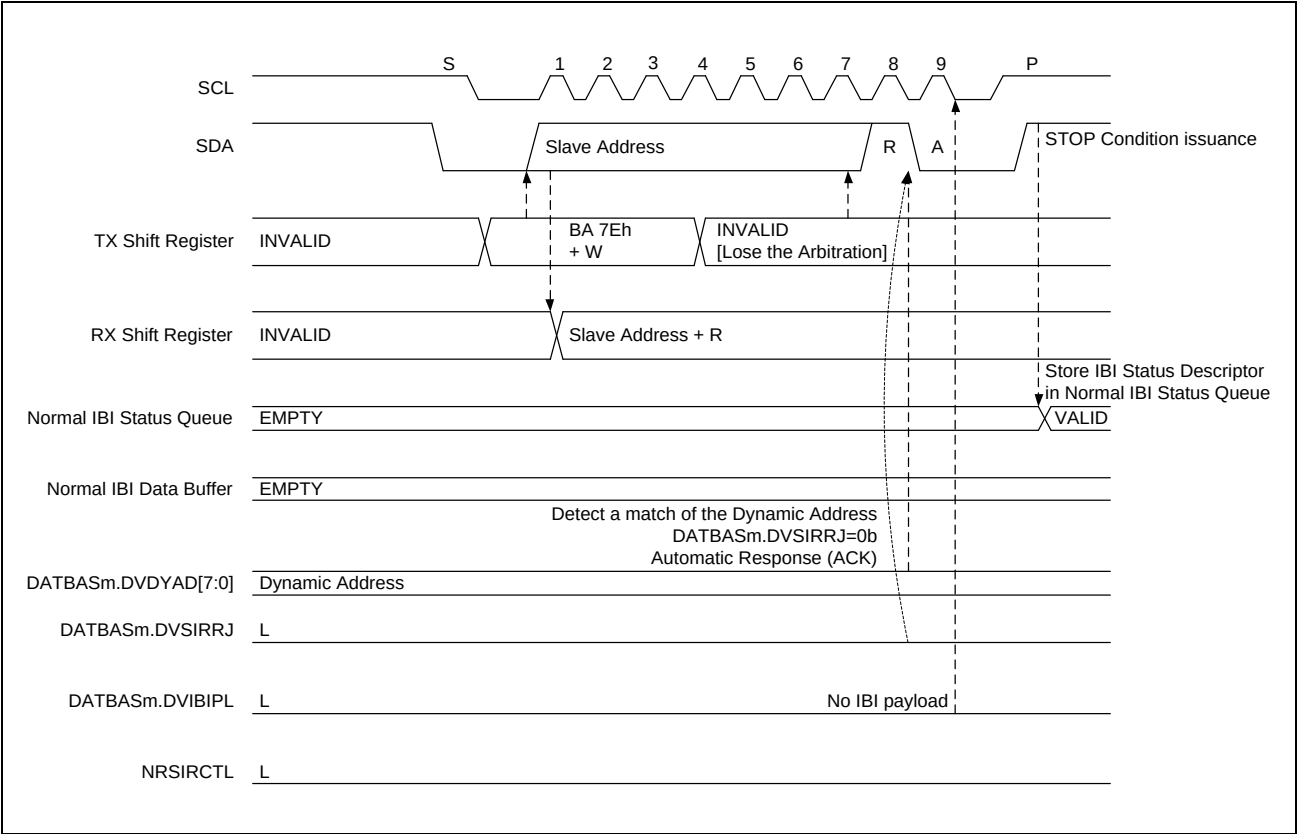


Figure 7.8-79 Slave interrupt request: ACK and DVIBIPL = 0

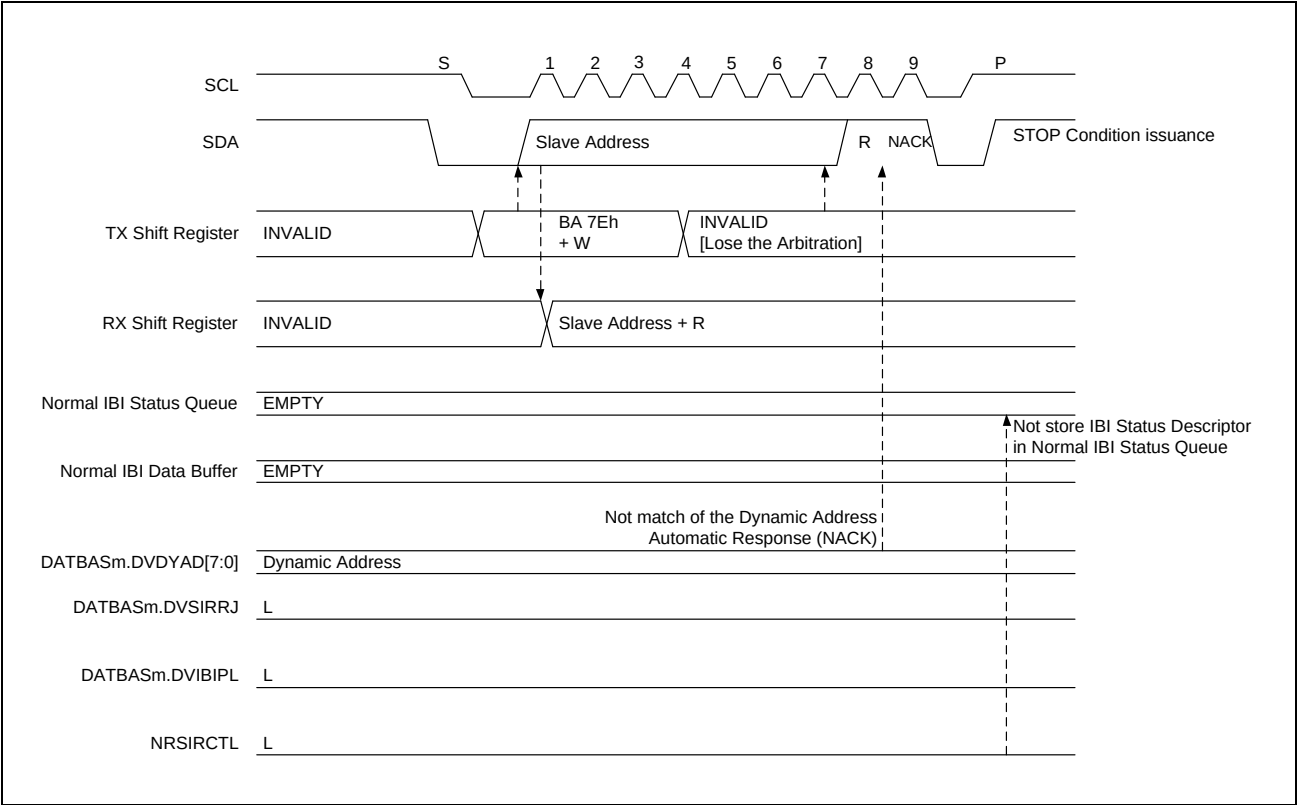


Figure 7.8-80 Slave interrupt request: NACK (not match the SDATBAS0.SDDYAD[6:0]) and NRSIRCTL = 0

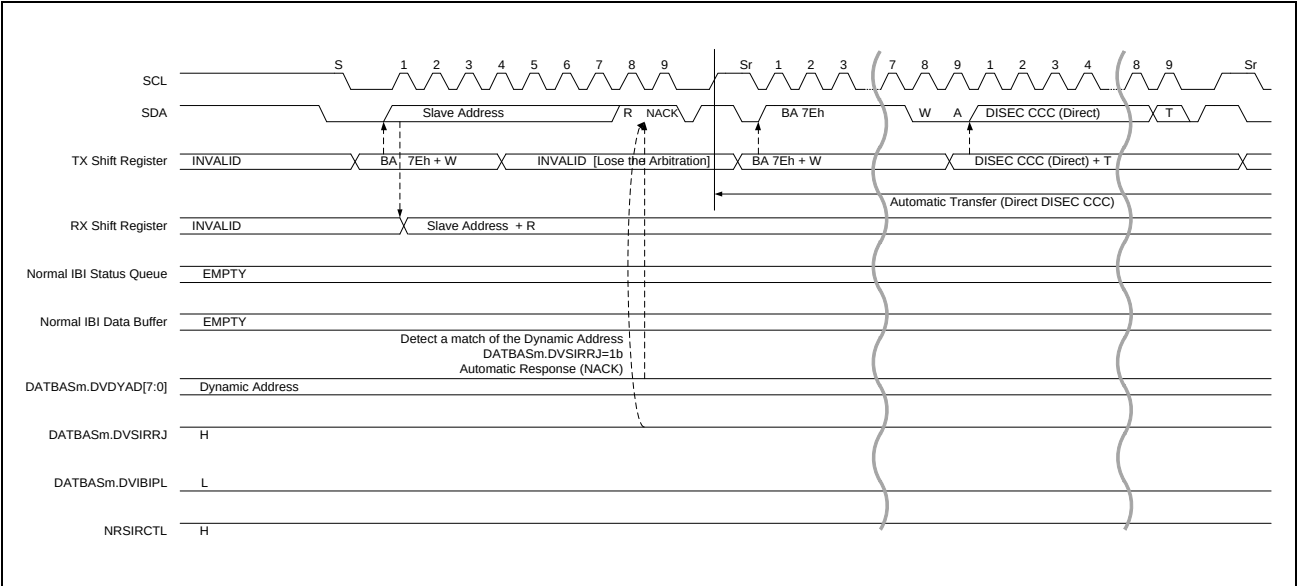


Figure 7.8-81 Slave interrupt request: NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (1/2)

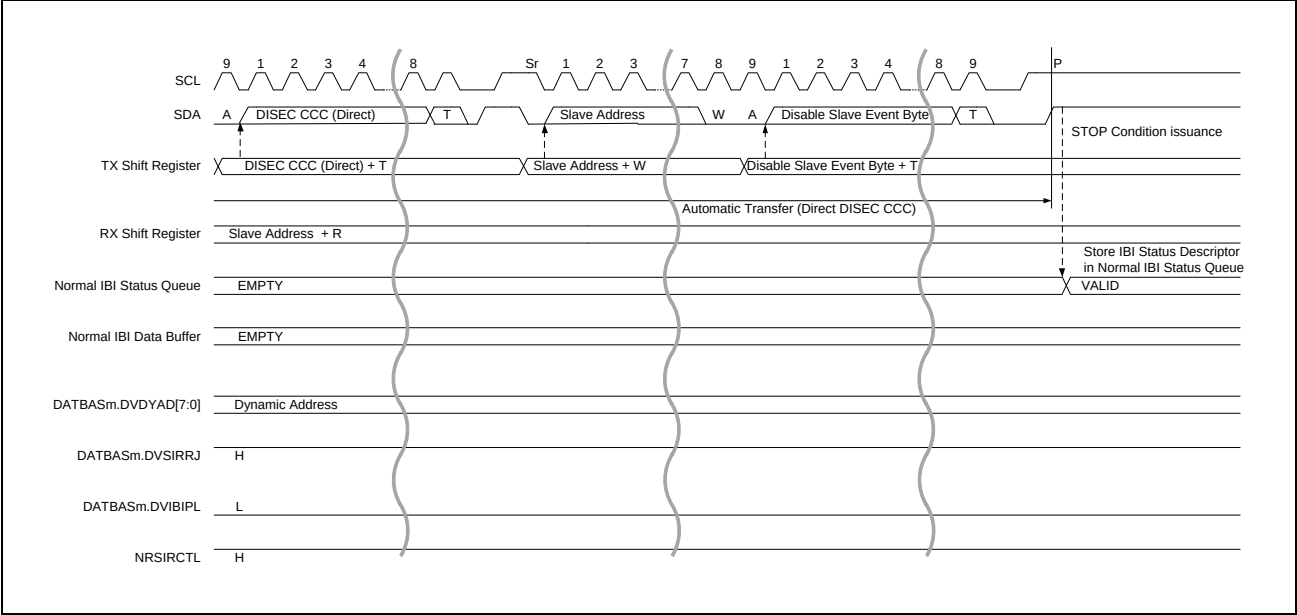


Figure 7.8-82 Slave interrupt request: NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (2/2)

(b) Mastership Request

1. Detect Slave Address with RnW bit Low in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBASm register).
3. When it does not match DAT.DVDYAD[7:0]:
Responds NACK, then issues the STOP condition.
When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is other than I3C Master (01):
Responds NACK, then issues the STOP condition.
When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is I3C Master (01):
 - When DAT.DVMRRJ = 1b
It operates in the following order.
 - (a) Responds NACK.
 - (b) Issued Repeated START condition and automatically issues Direct DISEC CCC to the detected Slave.
 - (c) Issues the STOP condition.
 - When DAT.DVMRRJ = 0b
Responds ACK, then issues STOP condition.
4. After issues of STOP condition,
NACK response:
 - If IBINCTL.NRMRCTL = 0b, the IBI Status Descriptor is not stored into the IBI Status Queue.
 - If IBINCTL.NRMRCTL = 1b, the IBI Status Descriptor is stored into the IBI Status Queue.
 ACK response:
Stores the IBI Status Descriptor into the IBI Status Queue.

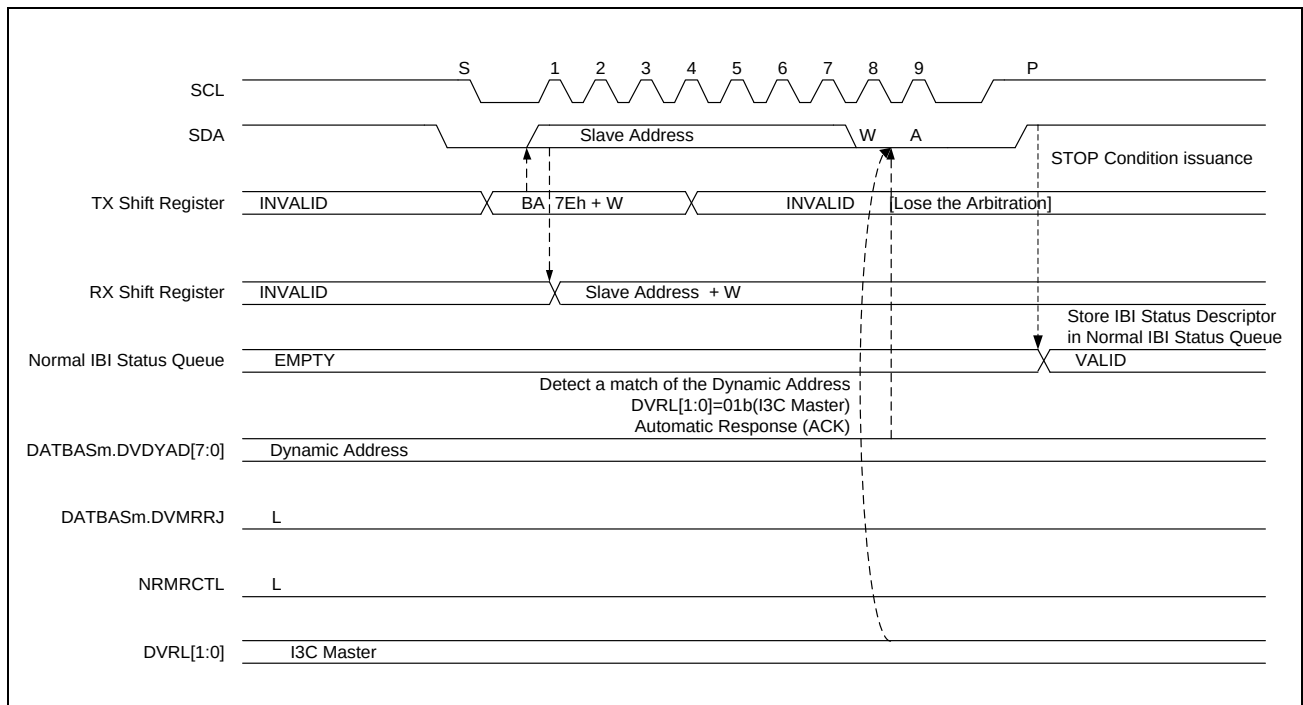


Figure 7.8-83 Mastership request: ACK

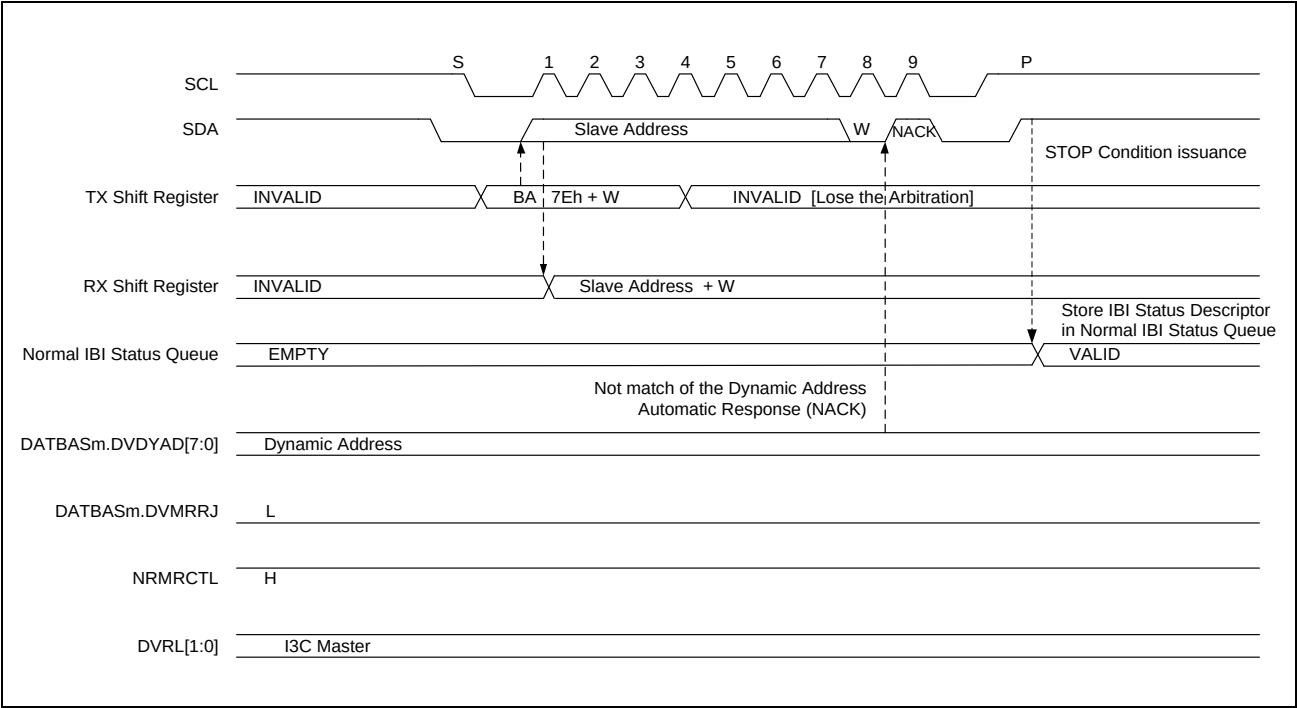


Figure 7.8-84 Mastership request: NACK (not match the DATBASm.DVDYAD[7:0]) and NRMCTL = 1

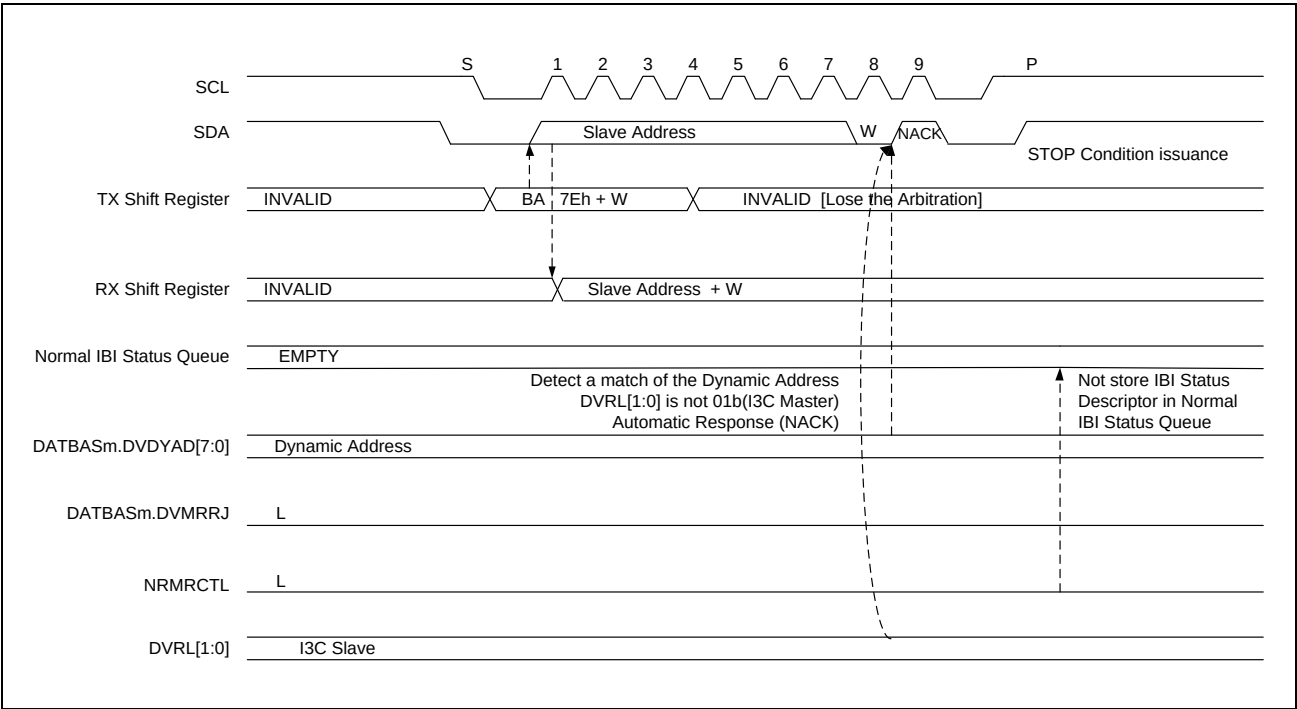


Figure 7.8-85 Mastership request: NACK (Device Role[1:0] is not 01 (I3C master)) and NRMCTL = 0

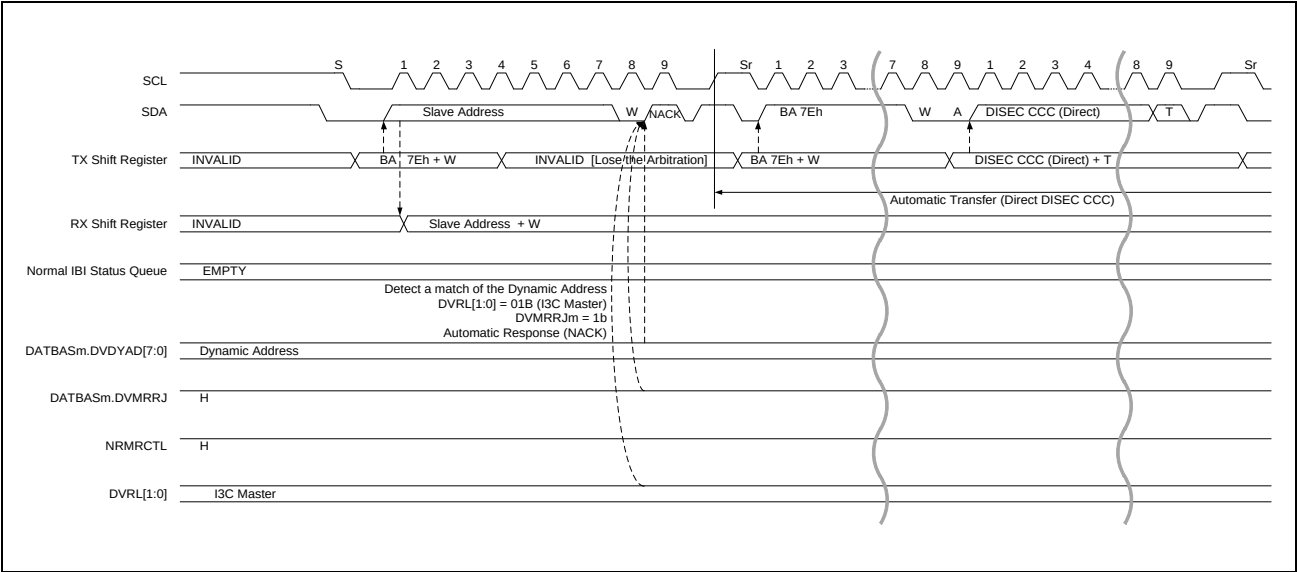


Figure 7.8-86 Mastership request: NACK (DVMRRJ = 1) and NRMRCTL = 1 (1/2)

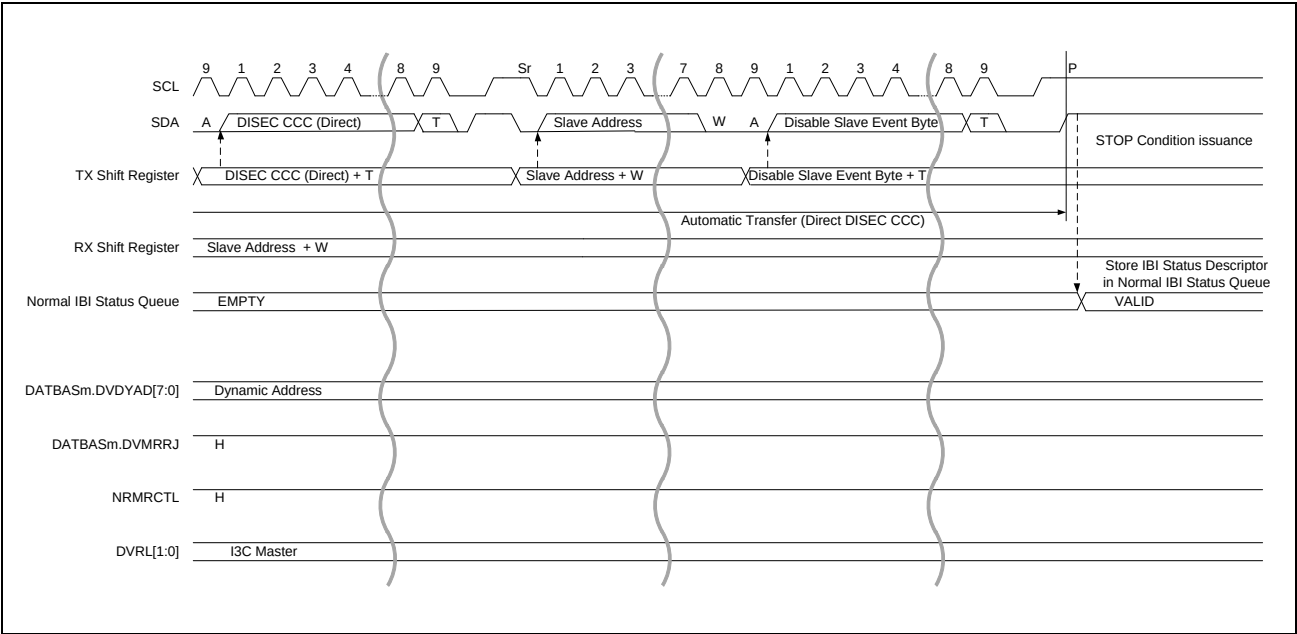


Figure 7.8-87 Mastership request: NACK (DVMRRJ = 1) and NRMRCTL = 1 (2/2)

(c) Hot-Join Event

1. Detect the Hot-Join Address (02h) with RnW bit Low in the Address Header.
2. When BCTL.HJACKCTL = 1b,
It operates in the following order.
 - (a) Responds NACK.
 - (b) Issues Repeated START condition and automatically issues Broadcast DISEC CCC.
 - (c) Issues the STOP condition.
 When BCTL.HJACKCTL = 0b,
Responds ACK, then issues STOP condition.
3. After issues of STOP condition, NACK response:
 - If IBINCTL.NRHJCTL = 0b, the IBI Status Descriptor is not stored into the IBI Status Queue.
 - If IBINCTL.NRHJCTL = 1b, store the IBI Status Descriptor into the IBI Status Queue.
 ACK response:
Stores the IBI Status Descriptor into the IBI Status Queue.

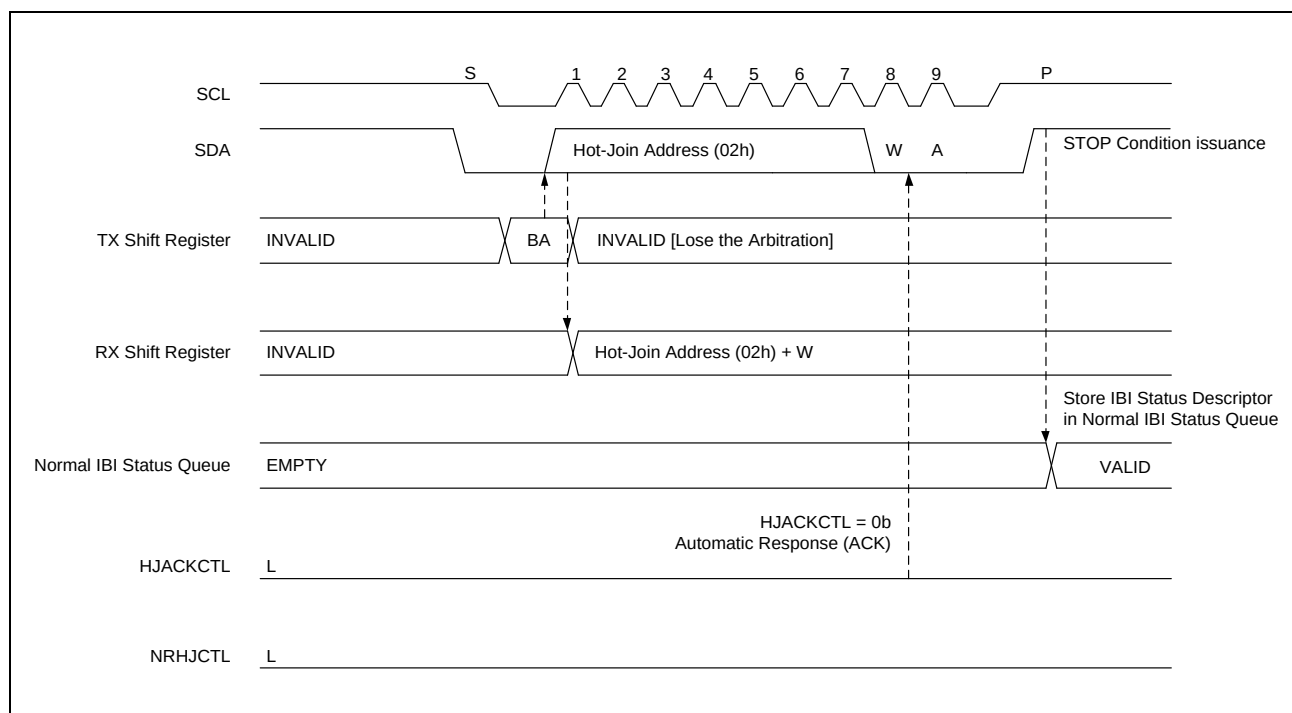


Figure 7.8-88 Hot-join event: ACK

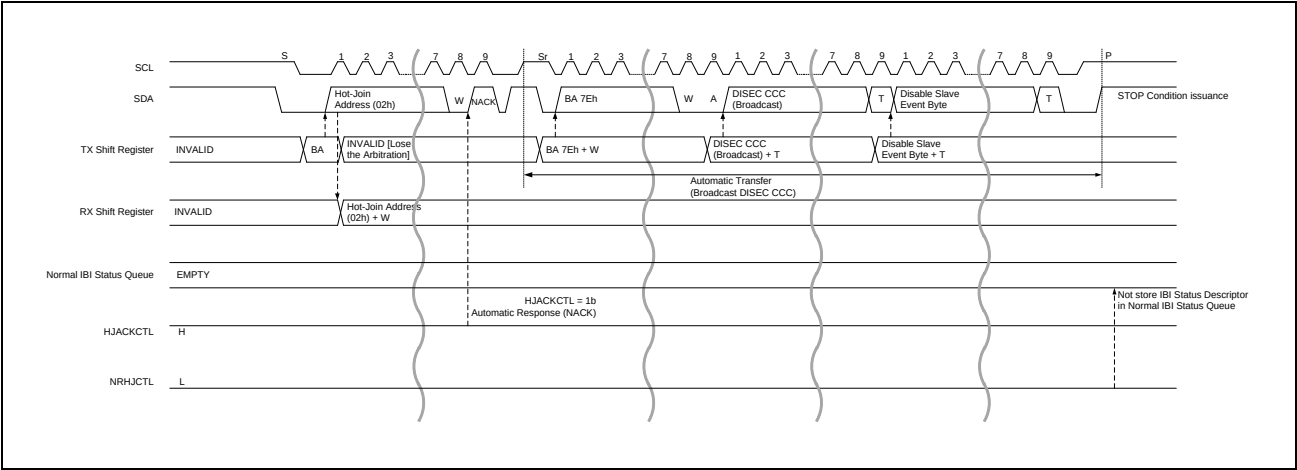


Figure 7.8-89 Hot-join event: NACK (HJACKCTL = 1) and NRHJCTL = 0

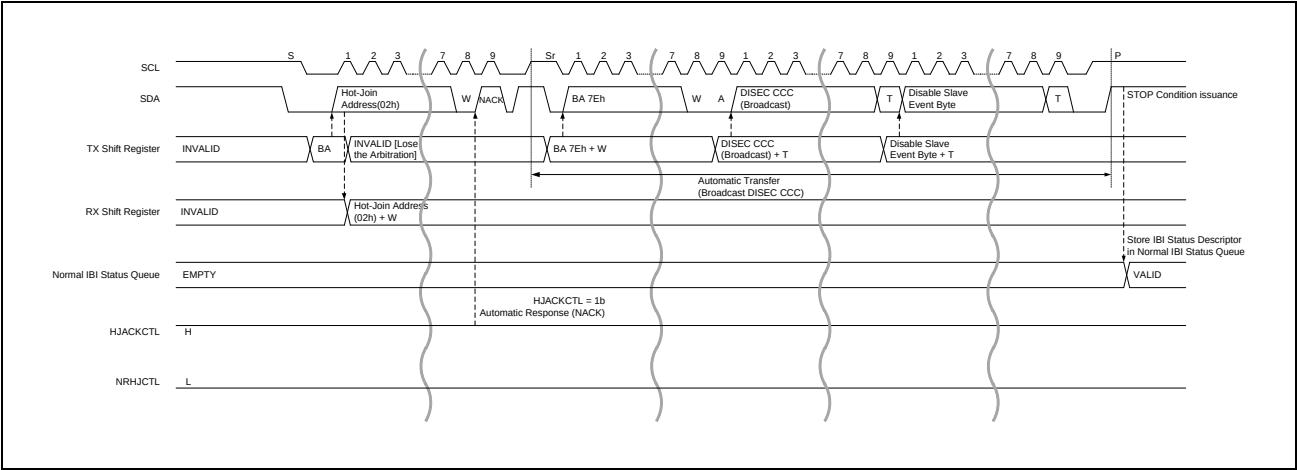


Figure 7.8-90 Hot-join event: NACK (HJACKCTL = 1) and NRHJCTL = 1

(9) Timing Control [I3C mode]

Timing Control is a function that enables Master to efficiently read data from Slave by controlling and grasping the timing at which the Slave Device samples the sensor value.

For details, refer to 5.1.8 *Timing Control of MIPI I3C Specification v1.0*.

In I3C, Timing Control supports the following three modes.

- Sync Mode
- Async Mode 0 (Asynchronous Basic Mode)
- Async Mode 1 (Asynchronous Advanced Mode)

The resources for realizing Timing Control in each mode are described below.

(a) Sync Mode

1. <I3C Master>

When STCTL.STOE is set to 1, when the master sends an ST message (SETXTIME CCC with ST subcommand), there is a function to issue an event (ri3c_elcstev) under the START condition of the ST message.

While measuring the T_{ph} period with an external timer, the start of T_{ph} and the Delay Time [DT] of the ST message can be measured by capturing the count value with the ri3c_elcstev event.

The measured value of Delay Time is sent as a DT message (SETXTIME CCC with DT subcommand) following the ST message.

2. <I3C Slave>

When STCTL.STOE is set to 1, there is a function to issue an event (ri3c_elcstev) for each START condition. STCTL.STOE is cleared when an ST message is received (SETXTIME CCC using the ST subcommand).

Check the reception of the ST message with the Receive Status Descriptor, and use an external timer to correct the T_{ph} period based on the count value captured in the ri3c_elcstev event and the Delay Time obtained from the DT message.

While measuring the T_{ph} period with an external timer, the start of T_{ph} and the Delay Time [DT] of the ST message can be measured by capturing the count value with the ri3c_elcstev event.

The sampling timing is recalculated by the corrected T_{ph}.

(b) Async Mode 0 (Asynchronous Basic Mode)

For timing control in Async Mode 0, set the ATCTL register if necessary.

1. <I3C Master>

I3C has counters of MREF (32bit) and MC2 (16bit) for Async Mode 0.

<MREF Counter>

When ATCCNTE.ATCE is enabled, the counter starts counting.

Its value is captured as MREF on the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.

<MC2 Counter>

After enabling ATCCNTE.ATCE, the counter counts up from the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave to the SCL rise edge next to the Tbit after Mandatory Byte, and its value is captured as MC2.

The MREF and MC2 capture values are stored next to the IBI Status Descriptor when IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1.

The MREF counter implemented in I3C is a 32-bit counter.

However, if the 32-bit counter is insufficient due to system requirements, I3C has events (ri3c_mrefovf, ri3c_mrefcpt) for expansion. These events are enabled by setting ATCTL.MREFOE to 1b. ri3c_mrefovf is output when the internal MREF counter overflows.

The MREF counter can be extended by using it as a count event for an external timer. ri3c_mrefcpt is output at the same timing as the capture timing of the internal MREF counter. By using it as the capture timing of the external timer, it can be used as an MREF counter concatenated with the value stored in the IBI Data buffer.

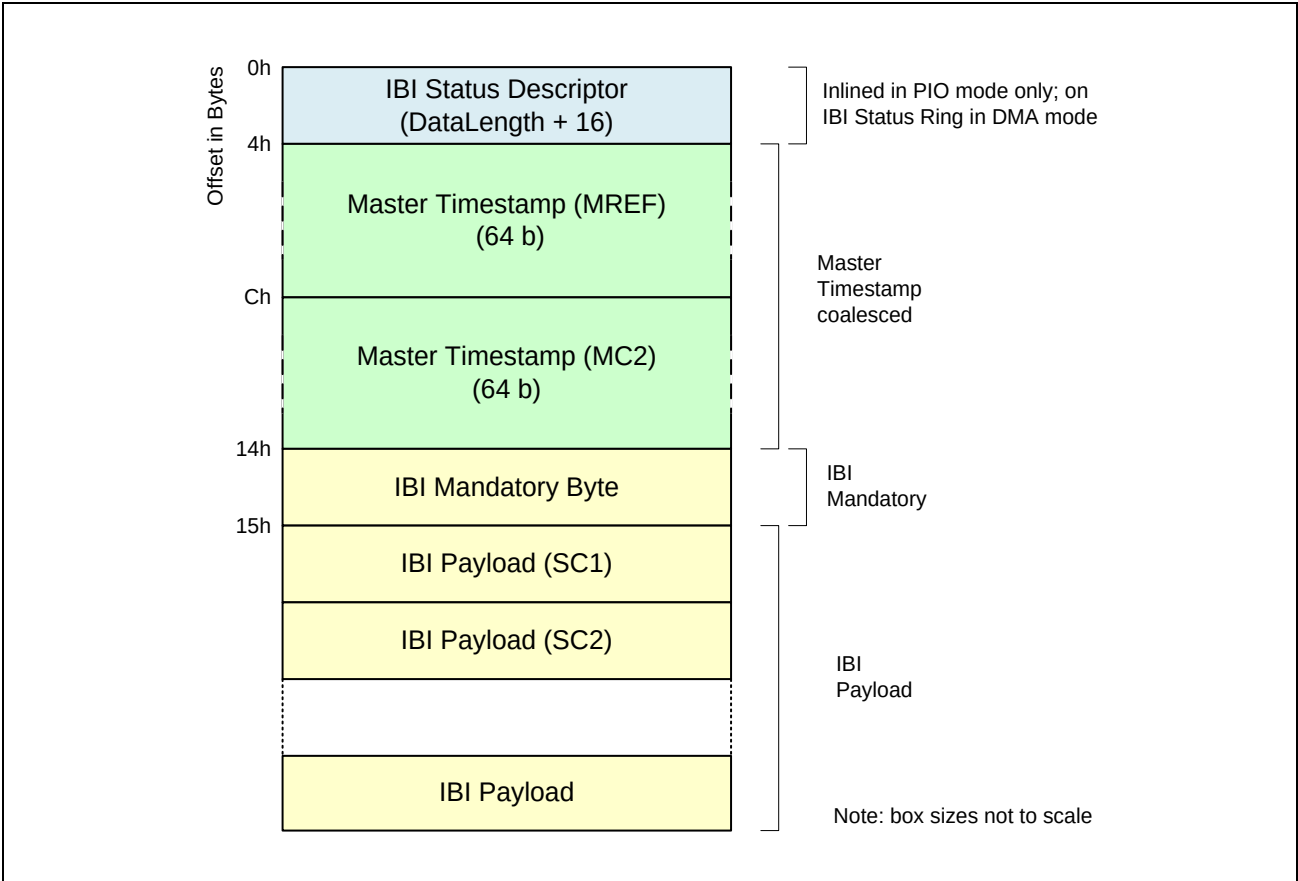


Figure 7.8-91 Master Timestamp Counters for IBI Event

Remark Please evaluate the Sensor Event time of I3C Slave according to the calculation formula of the MIPI I3C specification v1.0 document.

2. **<I3C Slave>**
I3C has counters of SC1 (16bit) and SC2 (8bit) for Async Mode 0.

<SC1 Counter>
After enabling ATCCNTE.ATCE, the counter counts up from SC1 count trigger (ATTRG.ATSTRG bit) to SCL rise edge next to ACK for the IBI, and its value is captured as SC1.

<SC2 Counter>
After enabling ATCCNTE.ATCE, the counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and its value is captured as SC2.
When the CETSS.ASYNE [0] bit = 1b and the ITS bit in Command Descriptor for issuing IBI is 1b, the SC1 and SC2 capture values are transmitted following the IBI Mandatory Byte as shown in the following figure.

If the SC1 and SC2 counters overflow, FFFFh and FFh are captured and CETSS.ICOVF is set to 1b.
The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1 and SC2 to the number of transmission data.

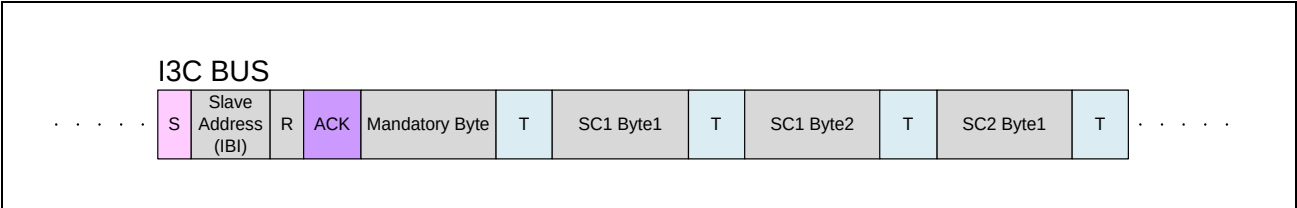


Figure 7.8-92 Example of Asynchronous Mode 0 Timestamp Data Transfer

(c) Async Mode 1 (Asynchronous Advanced Mode)
For timing control in Async Mode 1, set the ATCTL register if necessary.

1. **<I3C Master>**
I3C has counters of MREF (32bit), MSyncCNT (32bit) and MC2 (16bit) for Async Mode 1.

<MREF Counter>
When ATCCNTE.ATCE is enabled, the counter starts counting.
Its value is captured as MREF at the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.
The MREF extension method described in “1. I3C Master” of “(b) Async Mode 0” is also possible in Async Mode 1.

<MSyncCNT Counter>
When ATCCNTE.ATCE is enabled, the counter starts counting.
Its value is captured as MSyncCNT for each aME (SDA falling edge of START condition), and stored in the capture register.

<MC2 Counter>
After enabling ATCCNTE.ATCE, the counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and its value is captured as MC2.
The MREF and MC2 capture values are stored next to the IBI Status Descriptor when the IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1b. (Same as Async Mode 0)
When ATCTL.AMEOE is enabled, an aME Event (ri3c_elcamev) is issued for each aME. Use that event as a trigger to read the MSyncCNT capture value from the MRCCPT register and hold it in an external memory.

2. **<I3C Slave>**
I3C has counters of SC1 (16bit), SC2 (8bit) and aME_TICK (8bit) for Async Mode 1.

<SC1 Counter>
After enabling ATCCNTE.ATCE, the counter counts up from SC1 count trigger*1 to the first aME, and its value is captured as SC1.

<SC2 Counter>
After enabling ATCCNTE.ATCE, the counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and its value is captured as SC2.

<aME_TICK Counter>

After enabling ATCCNTE.ATCE, the counter counts every aME, and its value is captured as aME_TICK at the SCL rise edge next to ACK for the IBI.

The aME_TICK counter is cleared on the first aME after the SC1 count trigger.

When the CETSS.ASYNE[1] bit = 1b and the ITS bit in Command Descriptor for issuing IBI is 1b, the SC1, SC2 and aME_TICK capture values are transmitted following the IBI Mandatory Byte as shown in the following figure. If the SC1 and SC2 counters overflow, FFFFh and FFh are captured and CETSS.ICOVF is set to 1b.

The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1, SC2 and aME_TICK to the number of transmission data.

Note 1. SW or external trigger can be selected by selection bits.

The SC1 count trigger select method described in “2. I3C Slave” of “(b) Async Mode 0” is also possible in Async Mode 1

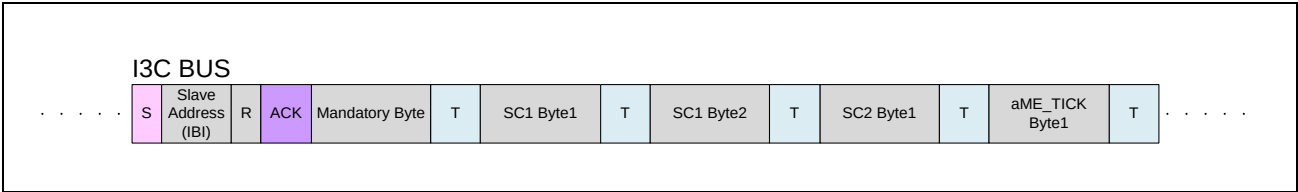


Figure 7.8-93 Example of Asynchronous Mode 1 Timestamp Data Transfer

(10) Port Control

(a) Extra SCL Clock Cycle Output Function

In master mode, I3C module has a facility for the output of extra SCL clock cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from I3C with single cycles of the SCL clock as the unit in the case of a bus error where I3C cannot issue a Repeated START condition or a STOP condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the OUTCTL.EXCYC bit is set to 1b, an additional clock pulse at the frequency set by the REFCKCTL.IREFCKS[2:0] bits and the STDBR.SBRHO[7:0] and STDBR.SBRLO[7:0] registers is output from the SCL pin. After output of this clock pulse, the EXCYC bit automatically becomes 0b. After confirming that the EXCYC bit is 0, wait for the setup time of the Repeated START condition or STOP condition, and then confirm the detection of the Repeated START condition or STOP condition. If the Repeated START condition or STOP condition is not detected, consecutive additional clock pulses can be output by writing 1b to the EXCYC bit again.

When I3C module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a Repeated START condition or a STOP condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the SDILV bit in PRSTDBG. After the SDA line has been released by the slave device, the preset of a Repeated START condition or a STOP condition is issued.

Use this function with the BFCTL.MALE bit set to 0b (master arbitration-lost detection is disabled).

[Output conditions for using the EXCYC bit in OUTCTL]

- When the bus is free (BFREF flag in BCST = 1b) or in master mode (CRMS bit = 1b in PRSST and BFREF flag = 0b in BCST)
- When the communication device does not hold the SCL line low

Figure 7.8-94 shows the operation timing of the extra SCL clock cycle output function (EXCYC bit).

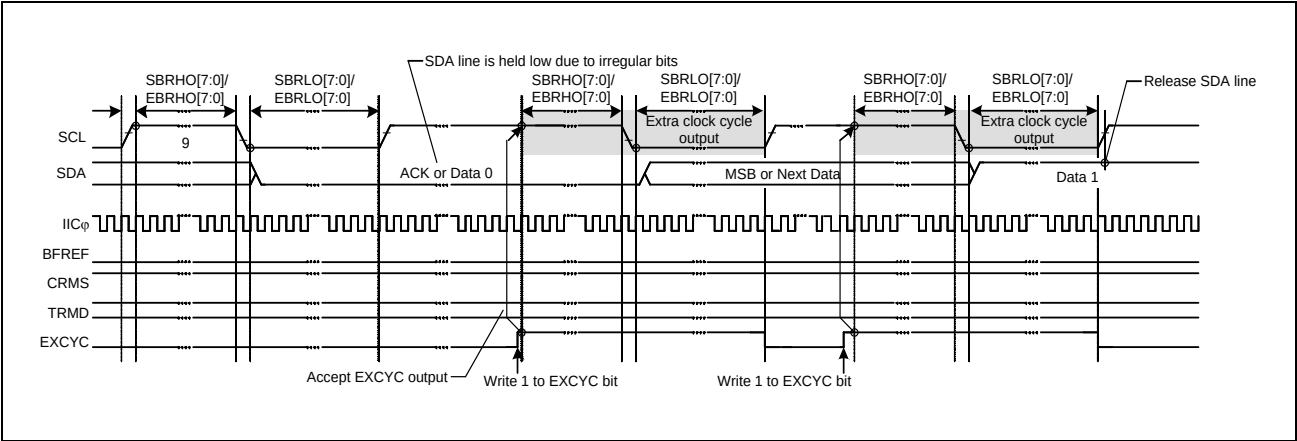


Figure 7.8-94 Extra SCL clock cycle output function (EXCYC bit)

(11) SMBus Operation [I²C mode]

I3C is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the BFCTL.SMBS bit to 1b. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the REFCKCTL.IREFCKS[2:0] bits, the STDBR.SBRHO[7:0] bits, and the STDBR.SBRLO[7:0] bits. In addition, determine the values of the OUTCTL.SDODCS bit and the OUTCTL.SDOD[2:0] bits to meet the data hold time specification of 300 ns or more. If I3C is used only as an I²C slave device, the transfer rate setting is not necessary, whereas the STDBR.SBRLO[7:0] bits needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave device address table basic registers 0 to 2 (SDATBASn.SDSTAD[6:0] bits (n = 0 to 2)), and set the corresponding SDATBASn.SDADLS bit (7-bit/10-bit address format select) (n = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the BFCTL.SALE bit to 1b to enable the slave arbitration-lost detection function.

(a) SMBus Timeout Measurement

(a-1) Measuring timeout of slave device

The following period (timeout interval: $T_{\text{LOW:SEXT}}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with the GPT timer using a START condition detection interrupt (INT_ri3c_st_n) and STOP condition detection interrupt (INT_ri3c_sp_n) of I3C. The measured timeout period must be within the total clock low-level period [slave device] $T_{\text{LOW:SEXT}}$: 25 ms (max.) of the SMBus specification.

If the time measured with the GPT exceeds the clock low-level detection timeout T_{TIMEOUT} : 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1b to the RSTCTL.INTLRST bit to issue an internal reset of I3C. When an internal reset is issued, I3C stops driving the bus for the SCL pin and SDA pin and make the SCL/SDA pin outputs high-impedance, which releases the bus.

(a-2) Measuring timeout of master device

The following periods (timeout interval: $T_{\text{LOW:MEXT}}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to STOP condition

To measure timeout for master devices, measure these periods with the GPT timer using a START condition detection interrupt (INT_ri3c_st_n), STOP condition detection interrupt (INT_ri3c_sp_n), and transmit end interrupt (I3Cn_TEND) or receive data buffer full interrupt (INT_ri3c_rx_n) of I3C. The measured timeout period must be within the total clock low-level extended period (master device) $T_{\text{LOW:MEXT}}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{\text{LOW:MEXT}}$ from START condition to STOP condition must be within $T_{\text{LOW:SEXT}}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the BST.TENDF flag in master transmit mode (master transmitter) and the NTST.RDBFF0 flag in master receive mode (master receiver). For this reason, perform bitwise transmit operation in master transmit mode, and hold the SCSTRCTL.ACKTWE bit 0 until the byte just before reception of the final byte in master receive mode. While the ACKTWE bit = 0b, the RDBFF0 flag is set to 1b at the rising edge of the ninth SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{\text{LOW:MEXT}}$: 10 ms (max.) of the SMBus specification or the total of measured periods exceeds the clock low-level detection timeout

TTIMEOUT: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to NTDTBP0).

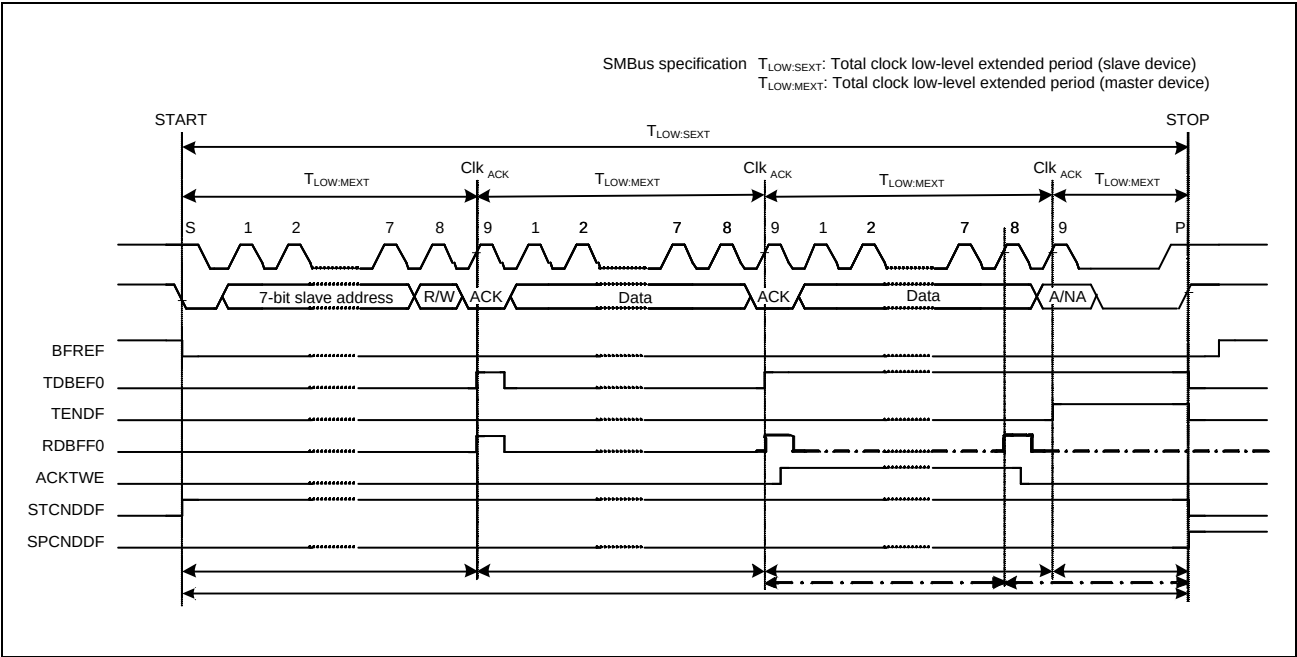


Figure 7.8-95 SMBus timeout measurement

(b) Packet Error Code (PEC)

This product incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of I3C. For the CRC generating polynomials of the CRC calculator, see **7.6 CRC Operation Unit (CRC)**.

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the SCSTRCTL.ACKTWE bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCL line low at the falling edge of the eighth clock cycle.

(c) SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For this product to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so I3C has a function for detecting the host address. To detect the host address as a slave address, set the BFCTL.SMBS bit and the SVCTL.HOAE bit to 1b. Operation after the host address has been detected is the same as normal slave operation.

(12) Common Command Codes (CCC) [I3C mode]

For the common command code (CCC), refer to 5.1.9 *Common Command Codes (CCC)* in *MIPI I3C Specification v1.0*. This I3C is based on *Table 15 I3C Common Command Codes* in 5.1.9.3 *Common Command Definitions* of *MIPI I3C Specification v1.0*.

The MIPI Reserved area and Vendor Extension area of Command Code are described below.

- I3C Master mode:
When sending CCCs in the MIPI Reserved area and Vendor Extension area from the I3C Master, only Broadcast/Direct SET CCCs using the Immediate Transfer Command can be sent.
Sending Direct GET CCC is not supported.
- I3C Slave mode:
Only Broadcast/Direct SET CCC can be received for CCC in MIPI Reserved area and Vendor Extension area.
Receiving Direct GET CCC is not supported.

7.8.4.2.4 Error Detection

(1) SDR Error Detection and Recovery Methods for I3C Slave Devices [I3C mode]

The seven error types summarized in **Table 7.8-14** are supported for all I3C slave devices. Each error type is further explained below the table.

Table 7.8-14 SDR slave error types

Error type	Description	Error detection method
S0	Broadcast address/W (= 7Eh/W) or Dynamic address/RW	Detect any of the following: 3Eh/W 5Eh/W 6Eh/W 76h/W 7Ah/W 7Ch/W 7Fh/W 7Eh/R
S1	CCC code	Parity check, using T-Bit
S2	Write data	Parity check, using T-Bit
S3	Assigned address during Dynamic address arbitration	Parity check, using PAR Bit
S4	7Eh/R after Sr during Dynamic address arbitration	Detect any value other than 7Eh/R after Sr during Dynamic Address Arbitration
S5	Transaction after detecting CCC	Detect illegally formatted CCC
S6 (optional)	Monitoring error	Slave detects (through monitoring) that transmitted Data differs from what it intended to transmit (Does not apply during Dynamic address arbitration)

(2) SDR Error Detection and Recovery Methods for I3C Master Devices [I3C mode]

The two error types summarized in **Table 7.8-15** are supported for all I3C master devices. Each error type is further explained below the table.

Table 7.8-15 SDR master error types

Error type	Description	Error detection method
M0	Transaction after sending CCC	Detect illegally formatted CCC
M1 (optional)	Monitoring error	Master detects (through monitoring) transmitted data different from what it intended to transmit (Does not apply during Dynamic address arbitration)
M2	No response to Broadcast address (7Eh)	Master detects NACK after Broadcast address (7Eh) transmission

(3) Timeout Error Detection

I3C includes a timeout function for detecting when the SCL line has been stuck longer than the predetermined time. I3C can detect an abnormal bus state by monitoring that the SCL line is stuck low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, I3C can detect the timeout and report the bus hung state.

This timeout function is enabled when $\text{BSTE.TODE} = 1\text{b}$. It detects a hung state that the SCL line is stuck low or high during the following conditions: (When $\text{TMOCTL.TOMDS}[1:0] = 00\text{b}$)

- The bus is busy ($\text{BCST.BFREF} = 0\text{b}$) in master mode ($\text{PRST.CRMS} = 1\text{b}$).
- I3C's own slave address is detected (SVST register is not 0000h) and the bus is busy ($\text{BCST.BFREF} = 0\text{b}$) in slave mode ($\text{PRST.CRMS} = 0\text{b}$).
- The bus is free ($\text{BCST.BFREF} = 1\text{b}$) while generation of a START condition is requested ($\text{CNDCTL.STCND} = 1\text{b}$).

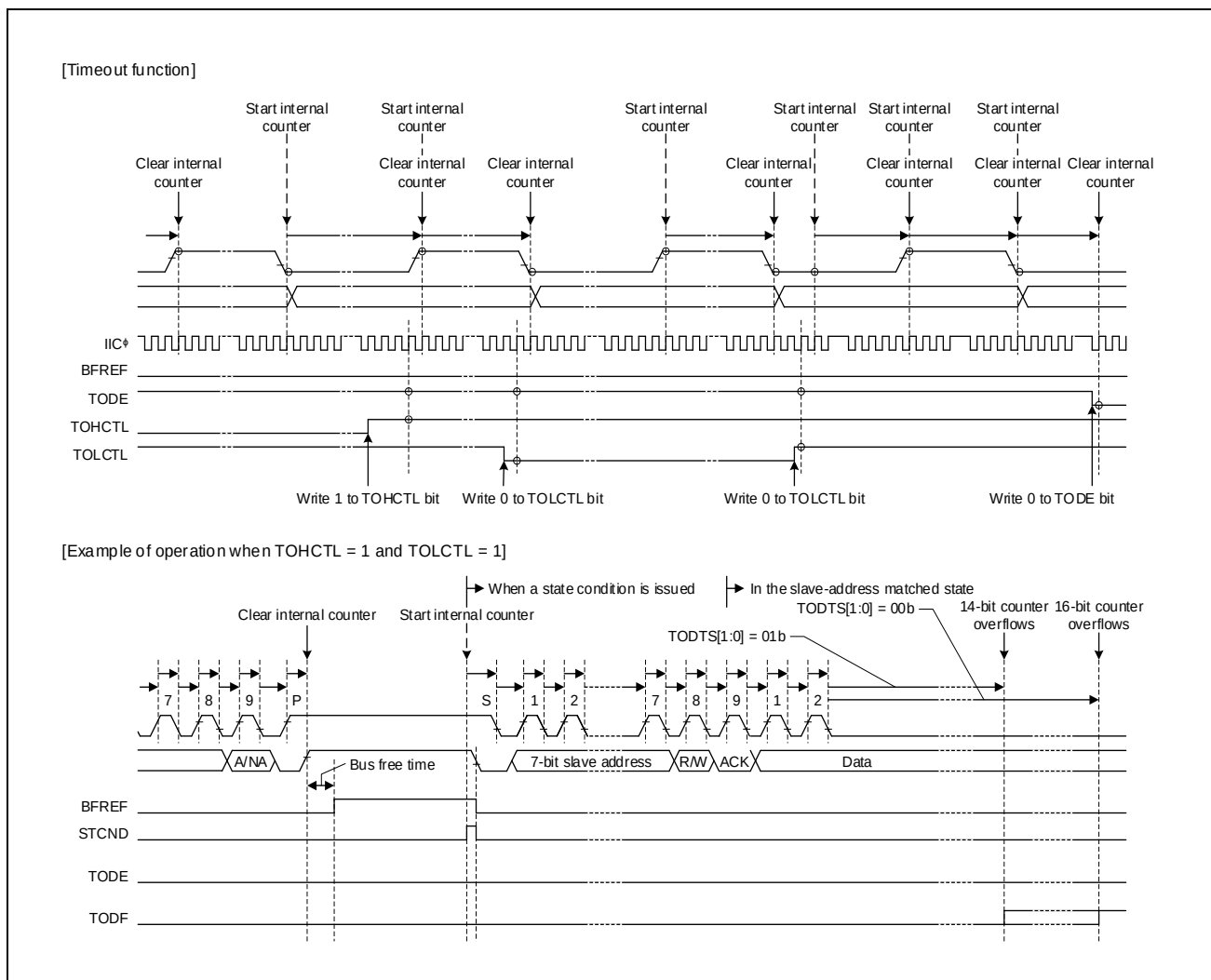


Figure 7.8-96 Timeout error detection (TODE, TODTS[1:0], TOHCTL, and TOLCTL bits)

(4) Resume Operation [I3C mode]

I3C enters the Halt state as a result of any type of error occurring in a transfer.

The error type is indicated by the field ERR_STATUS in Response Descriptor or Receive Status Descriptor. After I3C has entered the Halt state, the user must write the value 1 to the RSM bit to resume operation. I3C shall auto-clear the RSM bit once it has initiated the next Command transfer or detected the START condition.

(5) Abort Operation [I3C mode]

When the BCTL.ABT bit is set to 1b, I3C relinquish control of the bus before completing the currently issued transfer. In response to an abort request, I3C issues the STOP condition on the bus after the complete data byte is transferred or received. After I3C has aborted, the user shall clear the BCTL.ABT bit to allow operation on the bus.

Remark For Read transaction, when BCTL.ABT is set to 1b, that receive data is stored in Receive data buffer.

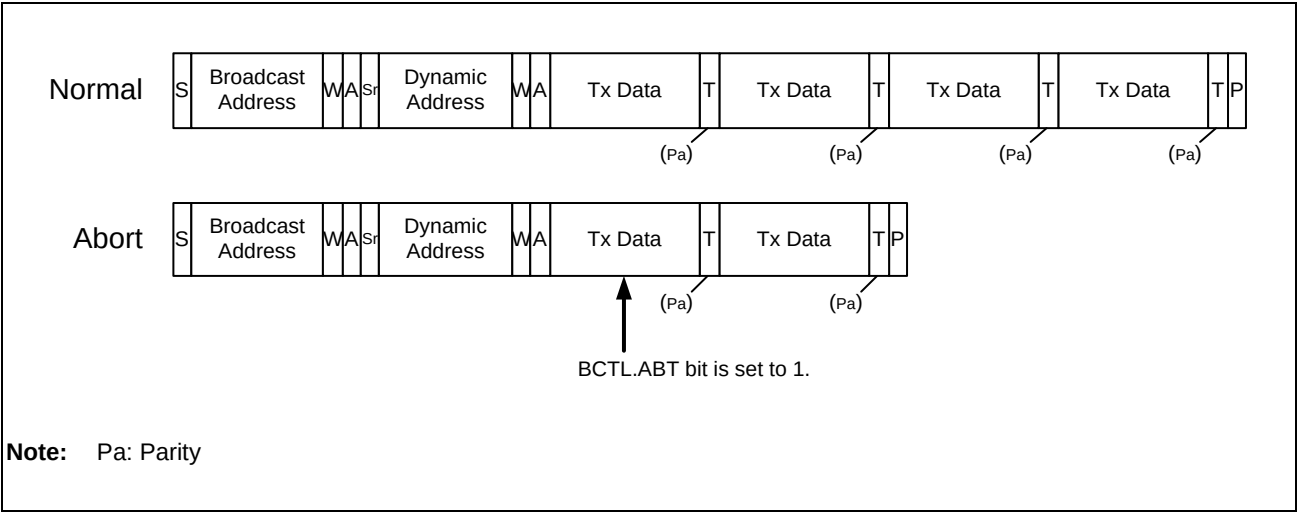


Figure 7.8-97 Abort operation of SDR write transfer

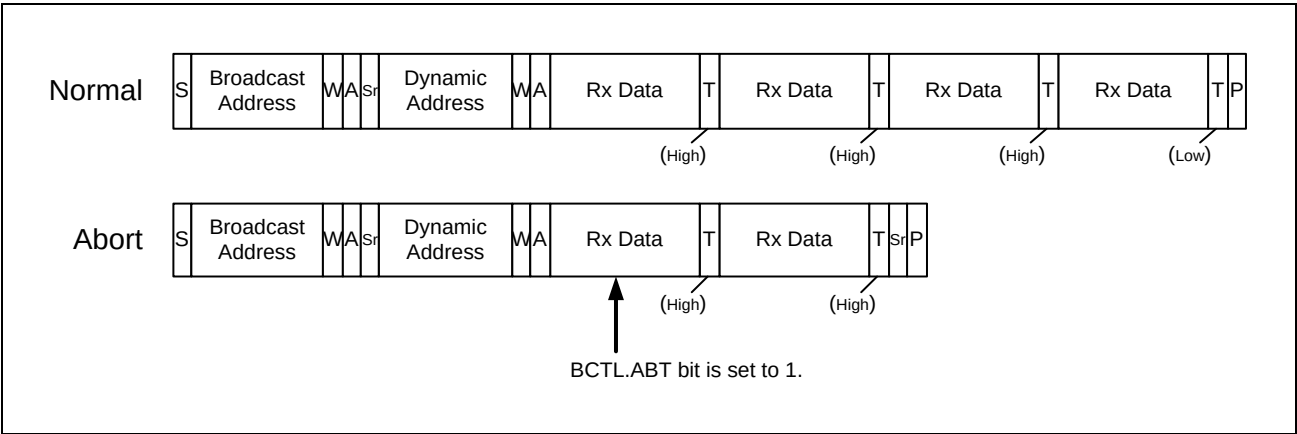


Figure 7.8-98 Abort operation of SDR read transfer

(6) Error Recovery Operation [I3C mode]

(a) Error Recovery Operation

When an error occurs, the flags of INST.INEF, NTST.TEF, NTST.TABTF, HTST.TEF and HTST.TABTF are set to 1b according to the cause of the error.

Or the interrupts INT_ri3c_ierr_n, INT_ri3c_terr_n and INT_ri3c_abort_n associated with each flags are asserted. (When detection and interrupts are enabled.)

There is a possibility of communication error or I3C internal error.

If an error occurs, I3C will be suspended. (BCTL.RSM becomes 1b.) After I3C is suspended, the application must write the value 1 to the BCTL.RSM bit to resume I3C operation and recover from the suspended state.

The error recovery flow is shown in **Figure 7.8-99** and **Figure 7.8-100**.

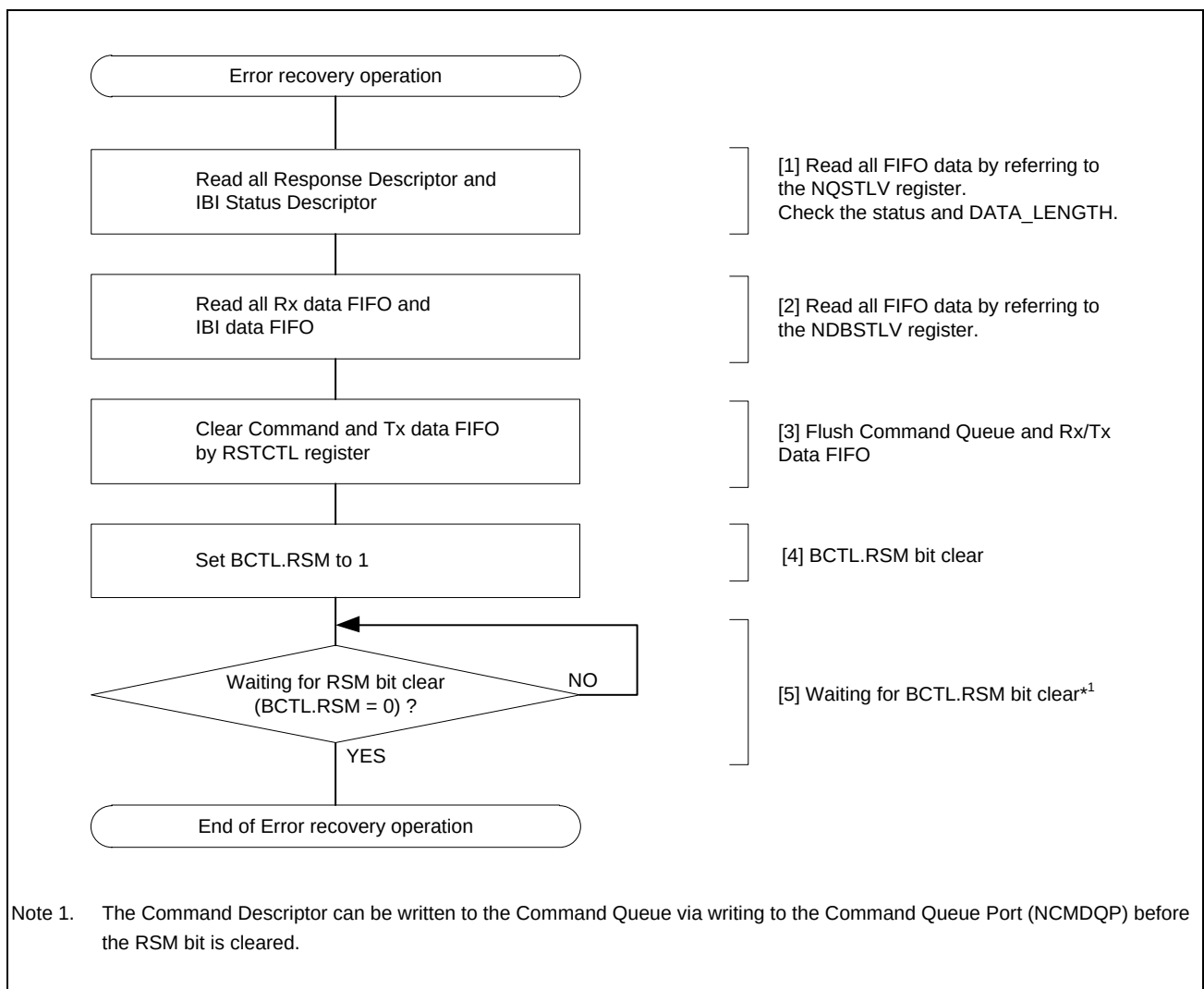


Figure 7.8-99 Example of Error Recovery Operation Flowchart for I3C Master

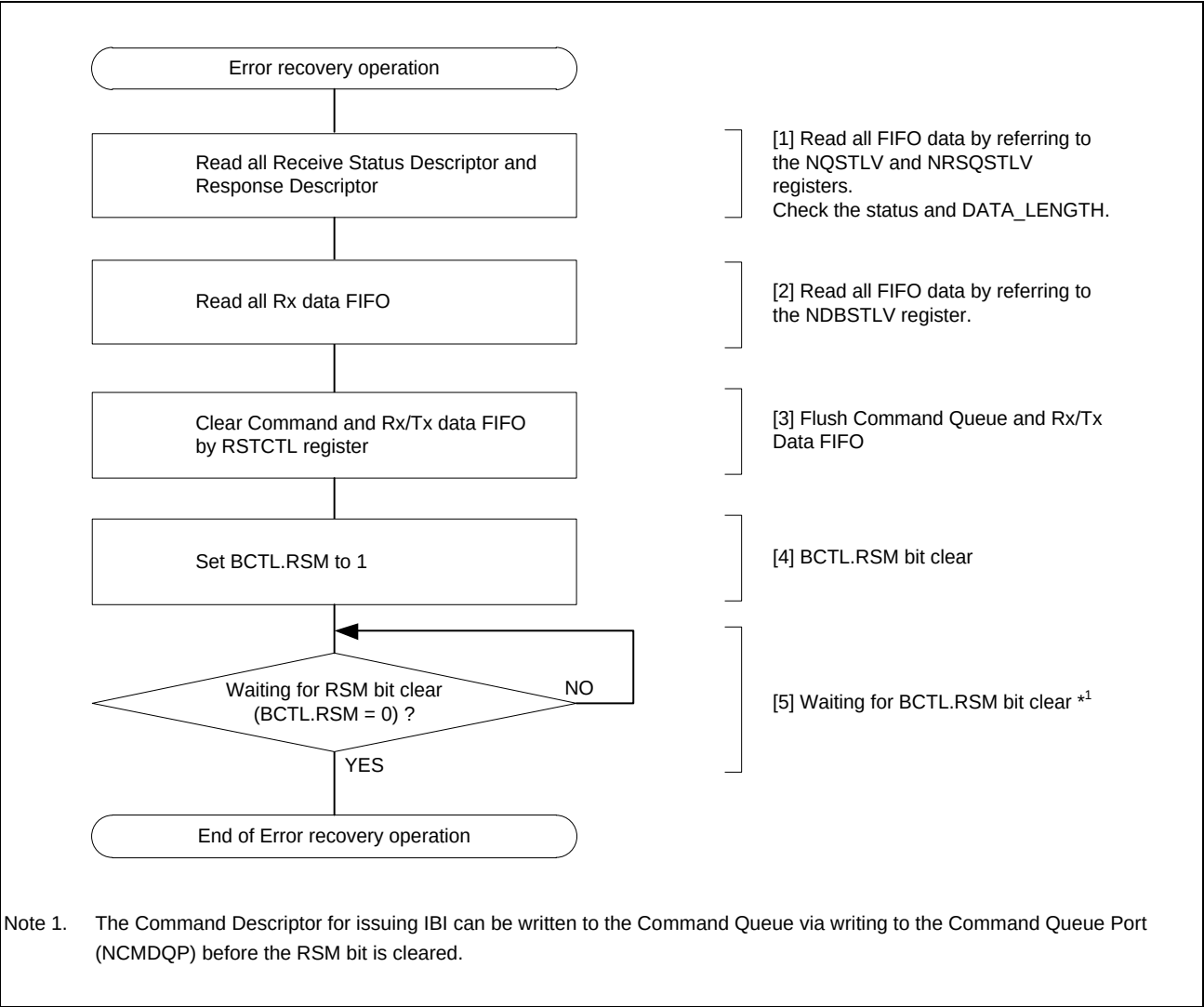


Figure 7.8-100 Example of Error Recovery Operation Flowchart for I3C Slave

When I3C Slave recovers from an error according to the error recovery flow, after setting BCTL.RSM to 1b, BCTL.RSM becomes 0b after detecting a state in which Bus Available period communication is not performed on I3C Bus.

If communication occurs on the I3C bus within the Bus Available period, BCTL.RSM will not be set to 0b and error recovery will not be completed, and a NACK response will be transmitted.

(b) Master Error Detection and Escalation Handling

If the Master does not receive an ACK of a transmitted private Message to a Slave and Steps 1 and 2 described in Chapter 5.1.10.2.4 of MIPI I3C Spec v1.0 fail, the processing flow of Step 3 shown in **Figure 7.8-101** and **Figure 7.8-102** is followed.

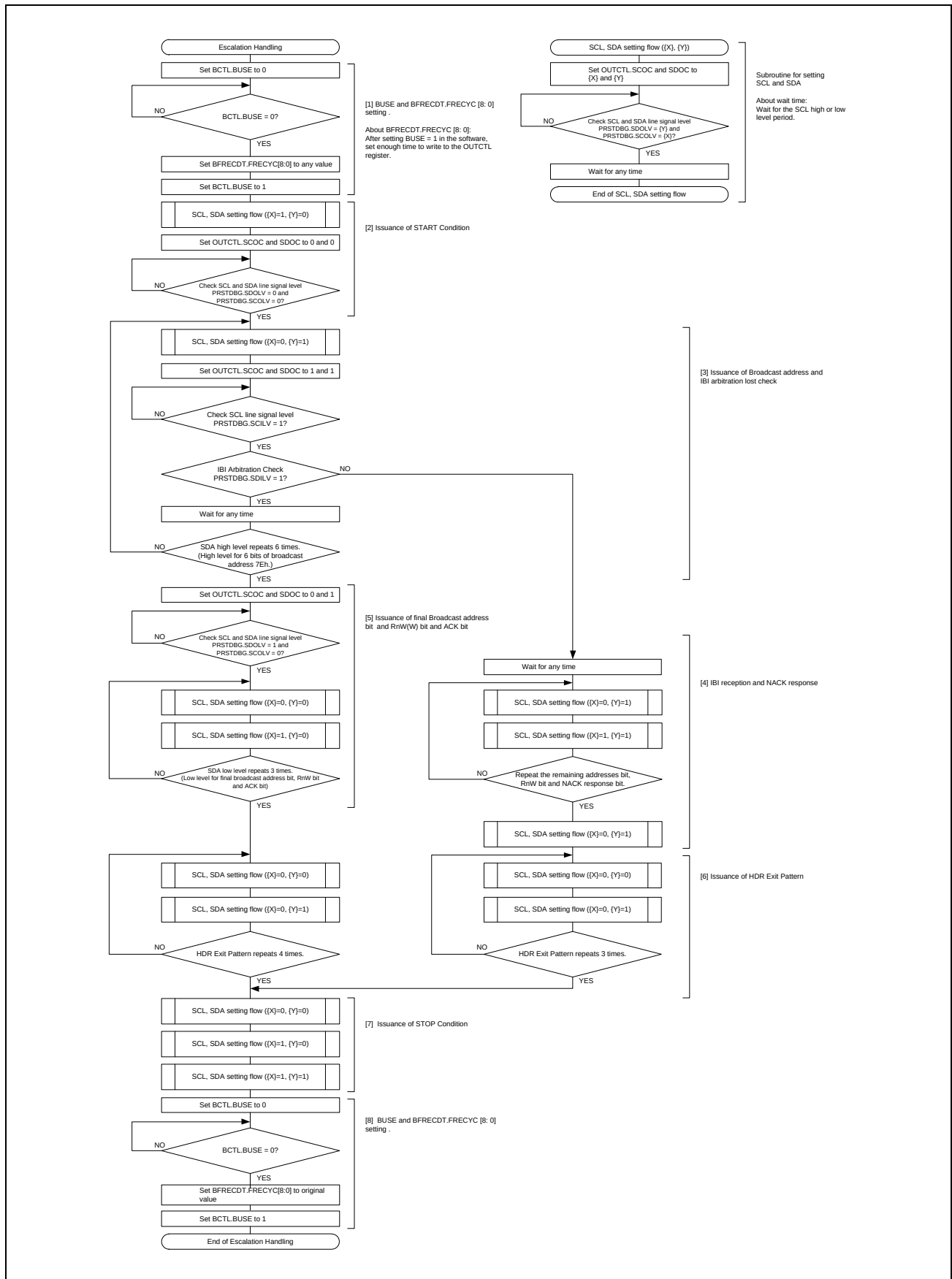


Figure 7.8-101 Escalation Handling Flowchart for I3C Master

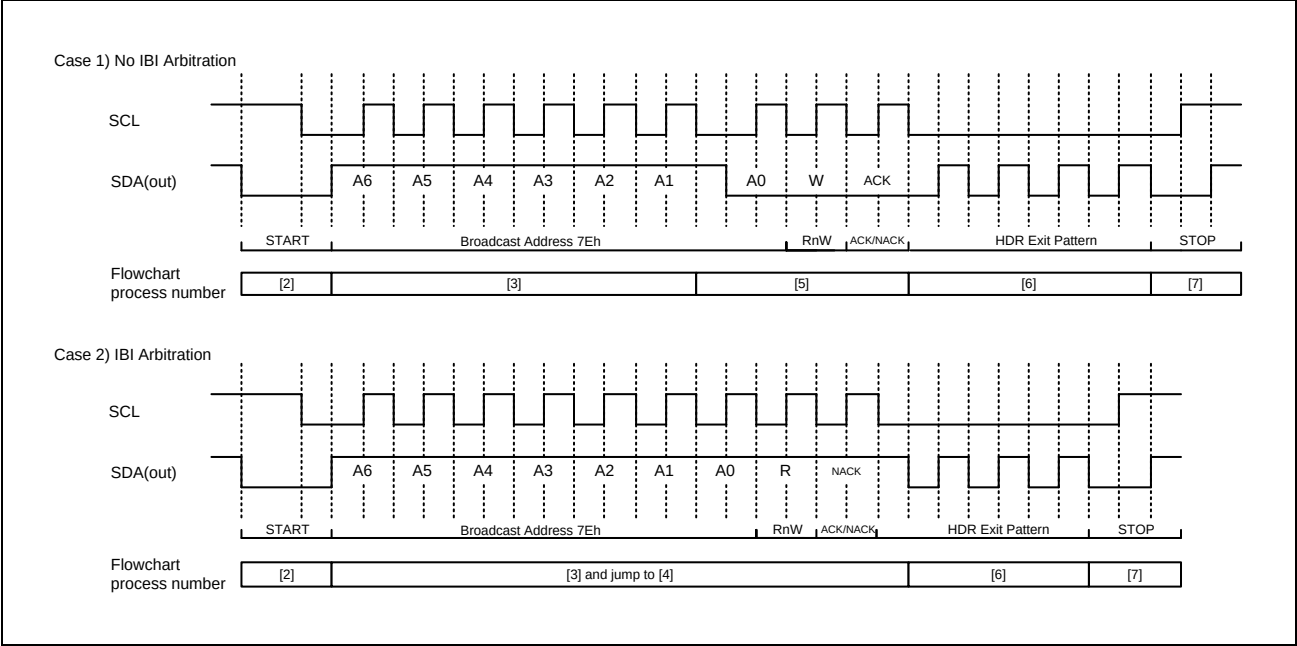


Figure 7.8-102 Escalation Handling timing chart for I3C Master

7.8.4.2.5 Low Power Function [I²C/I3C common]

(1) Wake-Up Function [I²C mode]

This module is equipped with the Wake-up function that causes the microcomputer to transition from low power consumption mode with system clock is stopped (software standby mode, snooze, etc.) to the normal operation. The Wake-up function is used to generate a Wake-Up interrupt signal when the received data matches the address set to Wake-Up interrupt factor also receives data in a state where the operating clock (I3C_0_PCLK/I3C_0_TCLK) is stopped (I3C_0_PCLK/I3C_0_TCLK asynchronous operation). This wake-up interrupt signal causes the microcomputer to transition to the normal operation. After Wake-Up interrupt occurs, switch I3C to I3C_0_PCLK/I3C_0_TCLK synchronous operation, it will be able to continue the communication operation.

The Wake-up function has four wake-up operation modes (normal WU mode 1, normal WU mode 2, command recovery mode, and EEP response mode). The table below describes the behavior in these four wake-up operation modes.

Table 7.8-16 Wake-Up Operation Mode

Mode	ACK Response Timing	ACK Type Responded before Recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation	SCL State before Recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation
Normal WU mode 1	Before recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation* ¹	ACK	Fixed to "L"
Normal WU mode 2	After recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation* ²	Before recovery: no response (NACK level retained) After recovery: ACK response	Fixed to "L"
Command recovery mode	Before recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation* ¹	ACK	Open
EEP response mode	Before recovery to I3C_0_PCLK/I3C_0_TCLK synchronous operation* ¹	NACK	Open

Note 1. Switching timing from I3C_0_PCLK/I3C_0_TCLK asynchronous operation to I3C_0_PCLK/I3C_0_TCLK Synchronous operation is the fall of the 9th clock of SCL.

Note 2. Switching timing from I3C_0_PCLK/I3C_0_TCLK asynchronous operation to I3C_0_PCLK/I3C_0_TCLK Synchronous operation is the fall of the 8th clock of SCL.

The following can be selected as Wake-Up interrupt factor.

- Host address detection (valid when SVCTL.HOAE=1b)
- General call address detection (valid when SVCTL.GCAE=1b)
- Slave address 0*¹ detection (valid when SVCTL.SVAE0=1b)
- Slave address 1*¹ detection (valid when SVCTL.SVAE1=1b)
- Slave address 2*¹ detection (valid when SVCTL.SVAE2=1b)

Note 1. Only 7-bit address can be set. Please set SDADLS bit to "0b" in SDATBASn.

(a) Normal Wake-Up mode 1

This section describes the behavior, the timing, and a use case of normal WU mode 1.

1. A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

<Before Wake-Up recovery>

ACK is sent in response to the data received with its own slave address.

<During Wake-Up recovery>

ACK response is made at the 9th clock cycle of SCL, and the SCL is held low afterwards.*¹

<After Wake-Up recovery>

Normal operation continues.

If the slave address does not match, the SCL line is not held low after the fall of the 9th clock cycle of SCL, and the slave operation continues. See **Figure 7.8-103** below for a use case.

2. A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to **Figure 7.8-104**.

Note 1. Between ninth clock cycle and first clock cycle during Wake-Up recovery, SCSTRCTL.RWE=1 does not work.

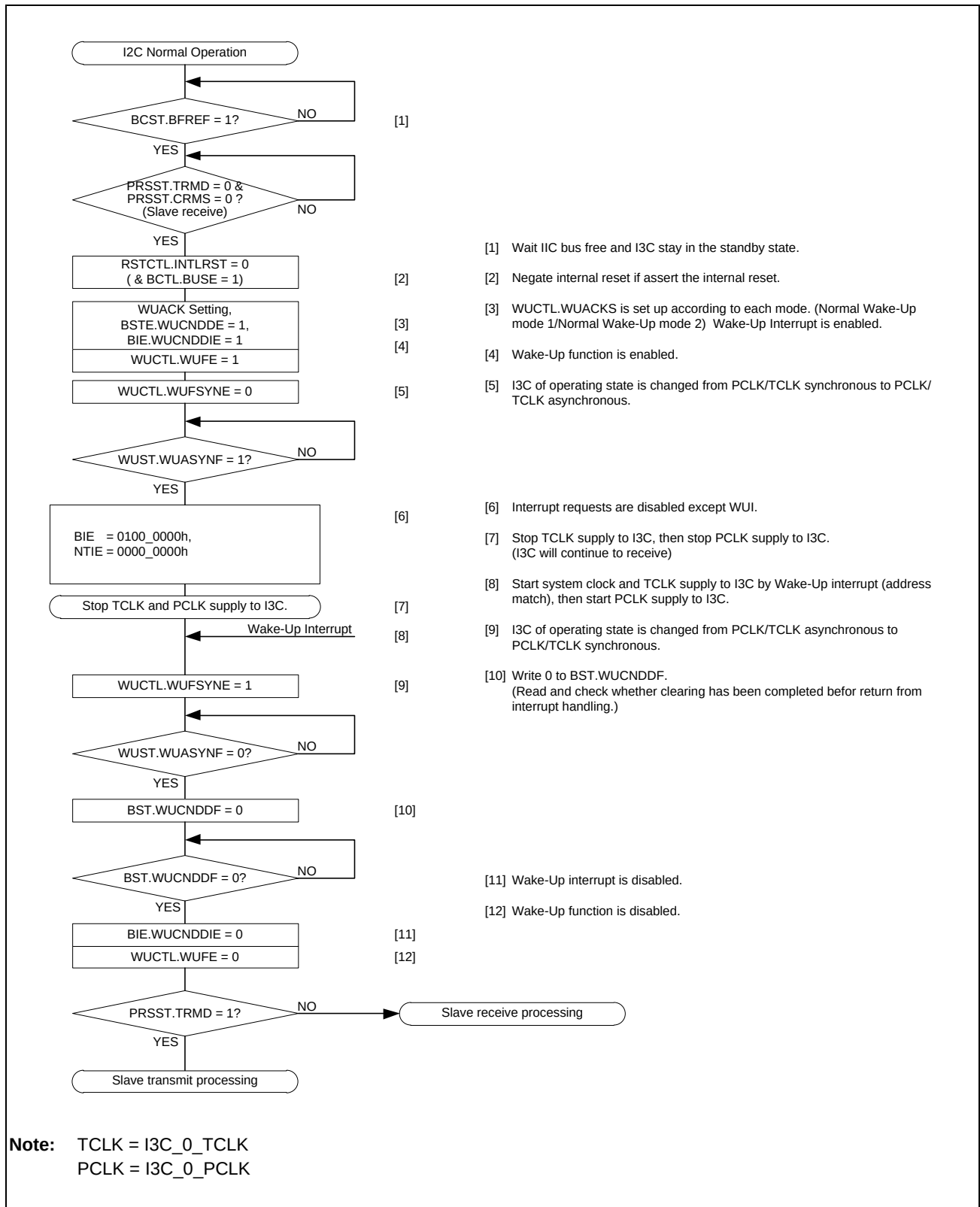


Figure 7.8-103 Use Case of Normal WU Mode 1 (Wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

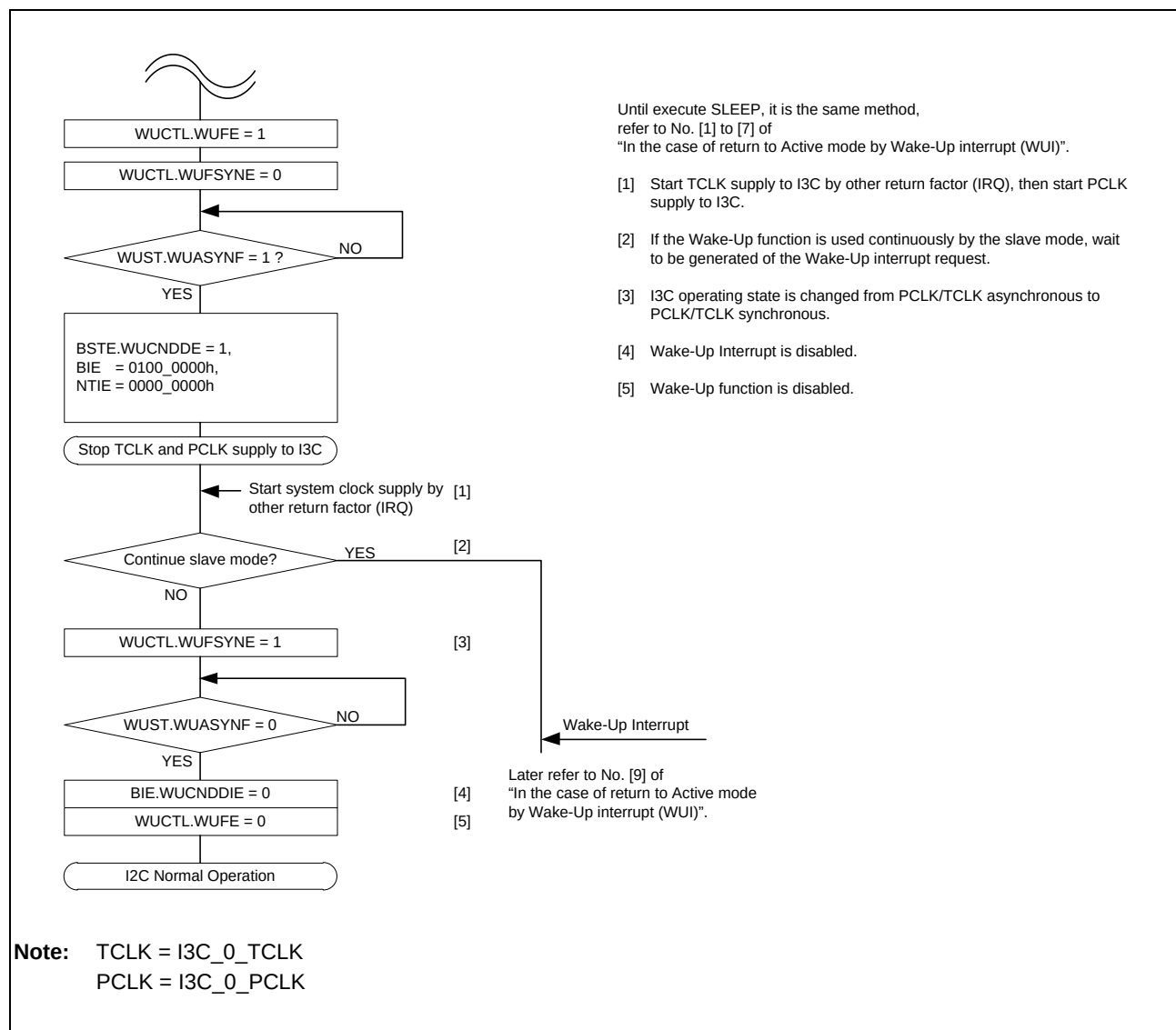


Figure 7.8-104 Use Case of Normal WU Modes 1 and 2 (Wake-up recovery by other recovery causes (IRQ))

(b) Normal Wake Up Mode2

This section describes the behavior, the timing, and a use case of normal WU mode 2.

1. A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

<Before Wake-Up recovery>

No response to the data received with its own slave address (until 8th SCL cycle end)

<During Wake-Up recovery>

Holding the SCL line low during the 8th and 9th clock cycles

<After Wake-Up recovery>

Returning ACK at the 9th clock cycle of SCL, and continuing the normal operation

If the slave address does not match, the SCL line is not held low after the fall of the 8th SCL clock cycle. The slave operation continues. See **Figure 7.8-105** below for a use case.

2. A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to **Figure 7.8-104**.

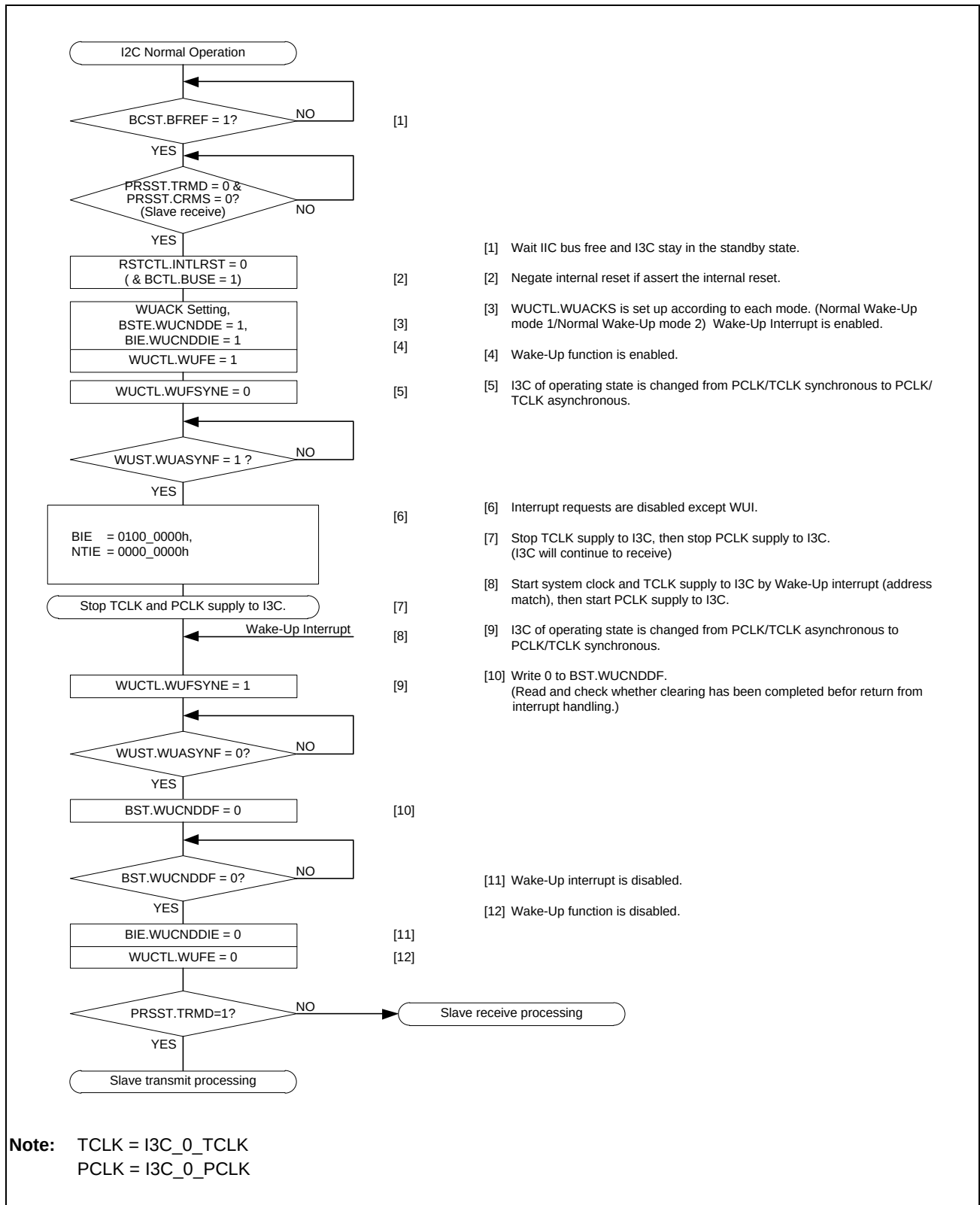


Figure 7.8-105 Use Case of Normal WU Mode 2 (Wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

(c) Command recovery mode/ EEP response mode (Special Wake Up mode)

In the command recovery mode and EEP response mode, the SCL line is not held low during the wake-up recovery period (after the rise of the 9th clock cycle of SCL), so other I²C/I3C devices can use the I²C bus during this period. This section describes the behavior, the timing, and use cases of the command recovery mode and the EEP response mode.

1. A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

<Before Wake-Up recovery>

In response to the data received with its own slave address, ACK (command recovery mode) or NACK (EEP response mode) is returned.

<During Wake-Up recovery>

The SCL line is not held low.

<After Wake-Up recovery>

Normal operation continues after I3C initial setting. If the slave address does not match, the slave operation continues. See **Figure 7.8-106** below for a use case.

Remark 1. Because the SCL line is not held low during wake-up recovery, the transmission/reception of the data that follows the slave address is not possible.

Remark 2. The command recovery mode and the EEP response mode are internal reset (RSTCTL.INTLRST= 1b) states. Therefore, the match of the slave address does not set the SVST flags (HOAF, GCAF, and SVAF2, SVAF1, SVAF0).

2. A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to **Figure 7.8-107**.

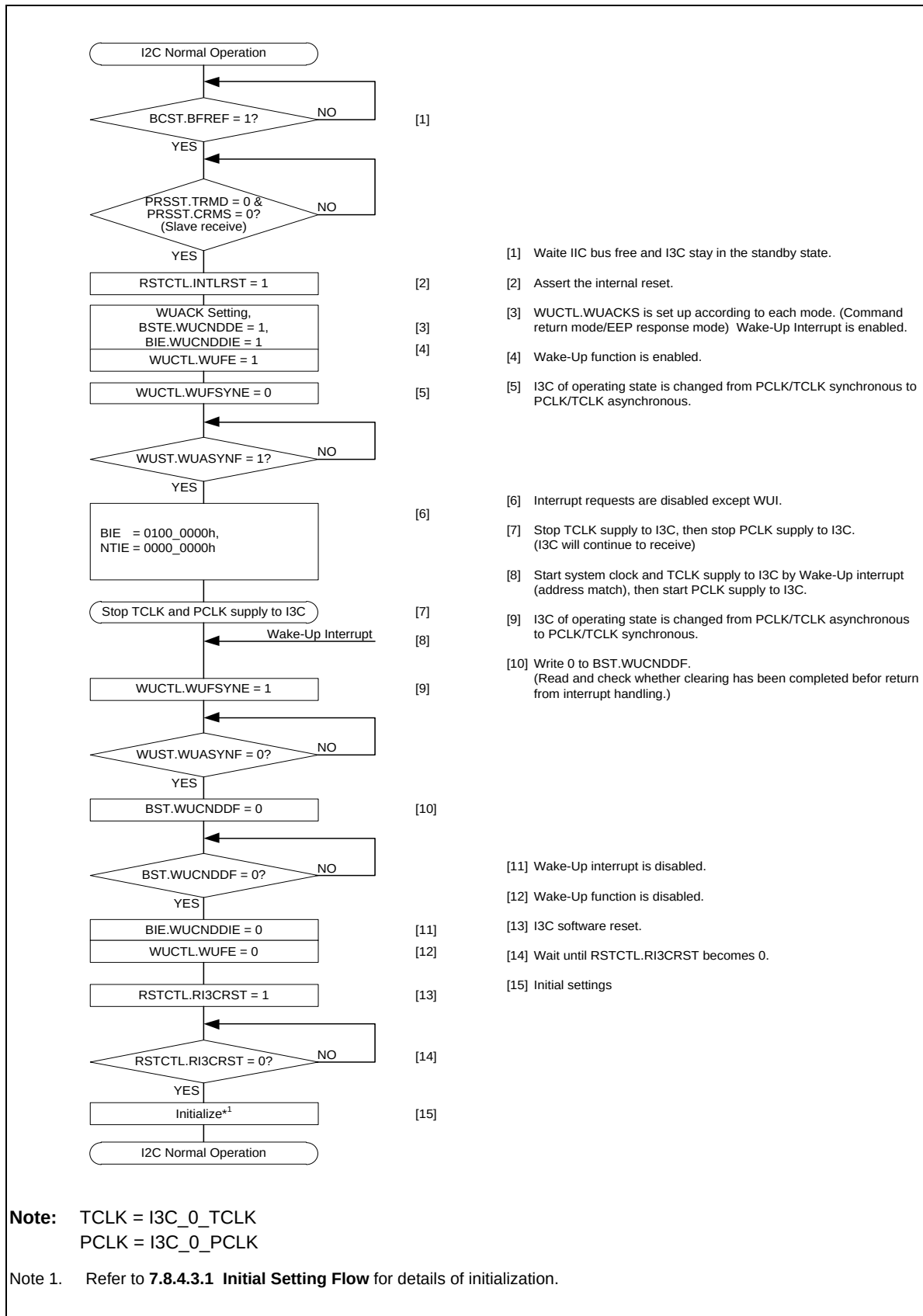


Figure 7.8-106 Use Case of Command Recover Mode and EEP Response Mode (Wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

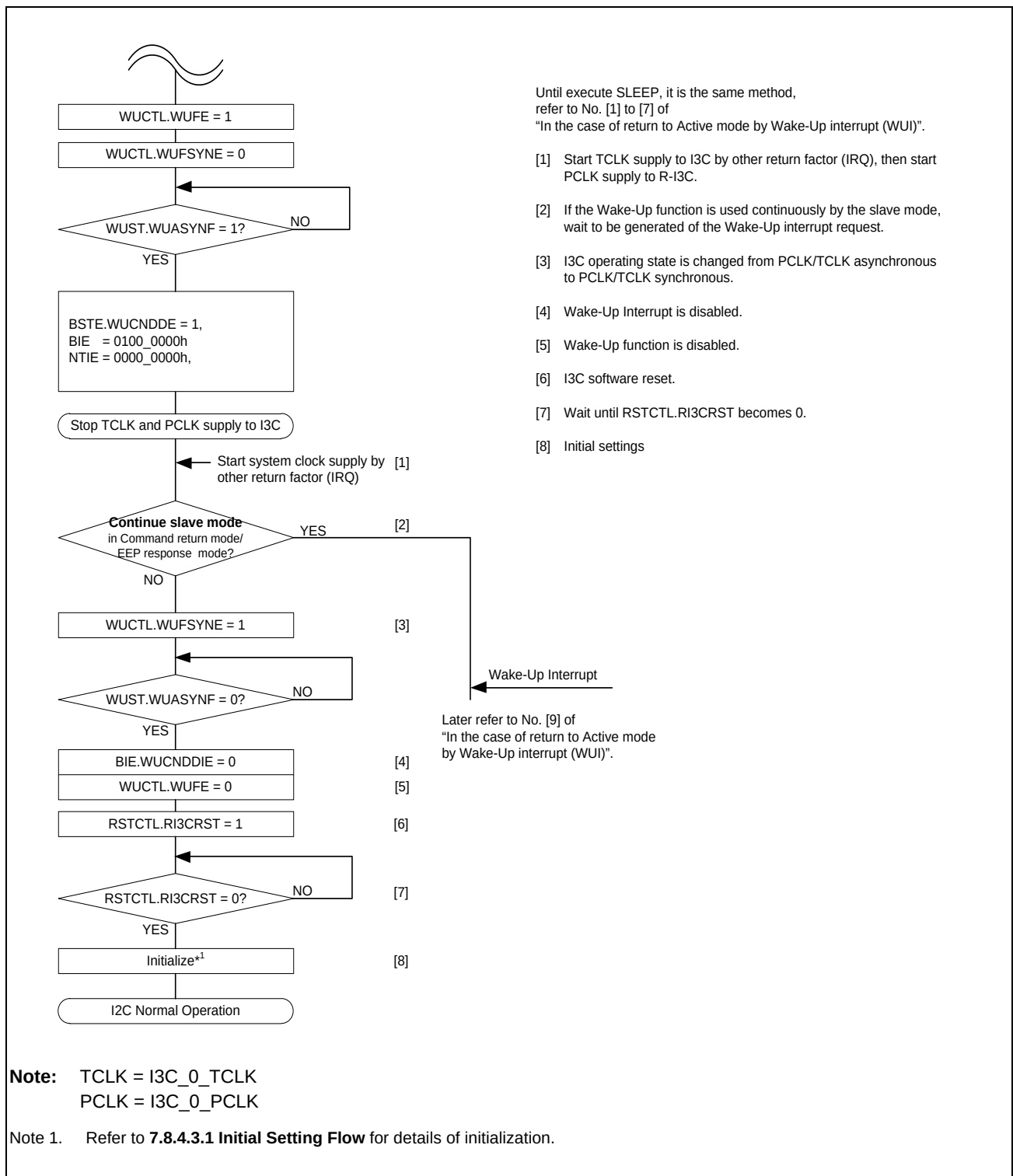


Figure 7.8-107 Use Case of Command Recover Mode and EEP Response Mode (Wake-up recovery by other recovery causes (IRQ))

(d) Precautions on the use of the Wake-Up function.

Precautions on the use of the Wake-Up function are shown below.

1. Do not change the registers in the I3C except the WUCTL.WUFSYNE bit while the WUST.WUASYNF flag = 1b (during I3C_0_PCLK/I3C_0_TCLK asynchronous operation).
2. Set WUCTL.WUFE = BSTE.WUCNDDE = BIE.WUCNDDIE = 1b and PRSST.CRMS = PRSST.TRMD = 0b (slave reception mode) before switching I3C_0_PCLK/I3C_0_TCLK asynchronous mode.
3. You cannot select the device ID and the 10-bit slave address for Wake-Up interrupt factor. Set the DVIDE bit in SVCTL and SDADLS bit in SDATBASn to 0b.
4. Set all bits in BIE(TENDIE, NACKDIE, SPCNDDIE, STCNDDIE, ALIE, TODIE) and TDBEIE and RDBFIE bits in HTIE to 0b (Interrupt disabled) before switching the asynchronous operation.
5. Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE=1b).
6. Wake-up interrupt is generated during I3C_0_PCLK/I3C_0_TCLK asynchronous operation (when WUST.WUASYNF=1b). In case of detecting slave address matching in I3C_0_PCLK/I3C_0_TCLK synchronous mode (WUST.WUASYNF=0), Wake-Up interrupt does not occur, and BST.WUCNDDF flag will not be set.
7. In I3C_0_PCLK/I3C_0_TCLK synchronous operation mode, the I3C may initiate the next reception upon detecting a conflict between the timing of writing "0b" to the WUCTL.WUFSYNE bit and the START condition. In this case, when data communication ends and a STOP condition is detected, the WUST.WUASYNF flag becomes 1b (I3C_0_PCLK/I3C_0_TCLK asynchronous mode transition) and Wake-Up event detection starts.
8. If you want to switch from I3C_0_PCLK/I3C_0_TCLK asynchronous operation to I3C_0_PCLK/I3C_0_TCLK synchronous operation without address match detection, it will switch on the STOP condition detection. When the WUCTL.WUFSYNE bit is set to 1 in a bus free state, I3C_0_PCLK/I3C_0_TCLK asynchronous operation continues (Reception operation: waiting communication frame). WUST.WUASYNF flag becomes to 0b when I3C detects the STOP condition of the next communication frame, and I3C switches to I3C_0_PCLK/I3C_0_TCLK synchronous operation.
9. After writing 0 to WUFSYNE bit in WUCTL, do not change I3C operation mode setting register (BFCTL, SCSTRCTL, ACKCTL, INCTL, SVCTL, SDATBASn) until switched to the I3C_0_PCLK/I3C_0_TCLK asynchronous operation from I3C_0_PCLK/I3C_0_TCLK synchronous operation (while WUST.WUASYNF flag = 1b). If register value changes by the interrupt processing etc. in this period, the I3C might malfunction without succeeding to the setting to the asynchronous operation.
10. During I3C_0_PCLK/I3C_0_TCLK asynchronous operation (WUST.WUASYNF = 1b), do not refer to each flag of SVST, BST, HTST register and BCST.BFREF flag.
11. Do not set ACKCTL.ACKT = 1b in order to make an ACK response in the synchronization unit when Wake-Up is performed by slave address match in Normal Wake-Up Mode 2.

(2) Wake-Up Function [I3C mode]

(a) I3C Master Wake-Up

Wake-Up interrupt causes of I3C Master are shown below.

- SDA Low detection (IBI request from I3C Slave)

See below for details on the master Wake-Up operation.

1. <Before Wake-Up recovery>

SDA Low Drive is detected and the INTWU interrupt is asserted.

2. <During Wake-Up recovery>

Keep SCL Line High.

3. <After Wake-Up recovery>

Drive SCL Low and complete START condition. SCL is supplied on the I3C Bus and IBI from I3C Slave is received.

Remark 1. If active mode (normal operation) is entered due to other factors, disable the Wake-Up function as necessary.

Remark 2. After confirming PRSTDBG.SDILV = 1b, set WUCTL.WUFE = 0b.

Remark 3. Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE=1b).

(b) I3C Slave Wake-Up

Wake-Up interrupt causes of I3C Slave are shown below.

- Broadcast Address (7Eh) and detection of its own Slave Address match

See below for details on the slave Wake-Up operation.

1. <Before Wake-Up recovery>

STEP 1:

If I3C detects BA (7Eh/W) following a START (or Repeated START) Condition, then I3C shall generate ACK (after 7Eh/W).

STEP 2:

If I3C detects its own Dynamic Address after a Repeated START condition following Step1, then I3C shall generate NACK (after its own Dynamic Address) and then issues a INTWU interrupt.

<During Wake-Up recovery>

I3C always generates NACK.

<After Wake-Up recovery>

Normal operation continues.

Remark 1. If active mode (normal operation) is entered due to other factors, disable the Wake-Up function as necessary.

Remark 2. Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE=1b).

7.8.4.2.6 Other

(1) SCL Synchronization Circuit [I²C mode]

This function is enabled while the PRTS.PRTMD bit is set to 1.

In generation of the SCL clock, I3C starts counting out the value for width at high level specified in STDBR.SBRHO[7:0] when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete.

When I3C detects the falling edge of the SCL line, it starts counting out the width at low level period specified in STDBR.SBRLO[7:0], and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I²C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, I3C is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line while in master mode.

When I3C has detected a rising edge on the SCL line and thus started counting out the width at high level specified in STDBR.SBRHO[7:0], and the level on the SCL line falls because an SCL signal is being generated by another master device, I3C stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in STDBR.SBRLO[7:0]. When I3C finishes counting out the width at low level, it stops driving the SCL line to the low level (releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in this module, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. When I3C finishes outputting the low-level period of the SCL clock, the SCL line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCSYNE bit in BFCTL is set to 1b.

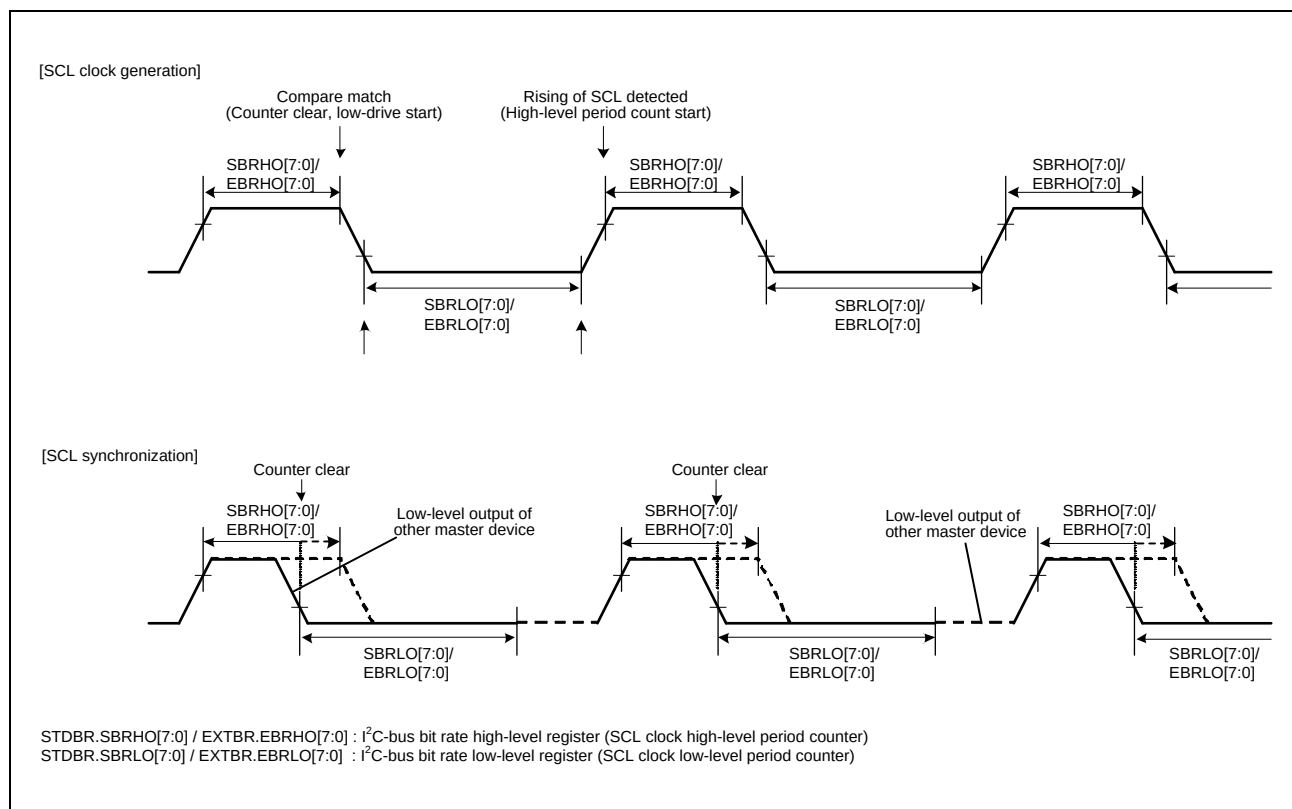


Figure 7.8-108 Generation and synchronization of the SCL signal

(2) Facility for Delaying SDA Output [I²C mode]

I3C module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the START, Repeated START, and STOP conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300 ns (minimum) data- hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDOD[2:0] bits in OUTCTL to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (while the SDOD[2:0] bits in OUTCTL are set to any value other than 000b), the SDODCS bit in OUTCTL selects the clock source for counting by the SDA output delay counter as the internal base clock (I3C ϕ) for I3C module or as a clock signal derived by dividing the frequency of the internal base clock by two (I3C ϕ /2). The counter counts the number of cycles set in the SDOD[2:0] bits in OUTCTL. After counting of the set number of cycles of delay is completed, I3C module places the required output (START, Repeated START, or STOP condition, data, or an ACK or NACK signal) on the SDA line.

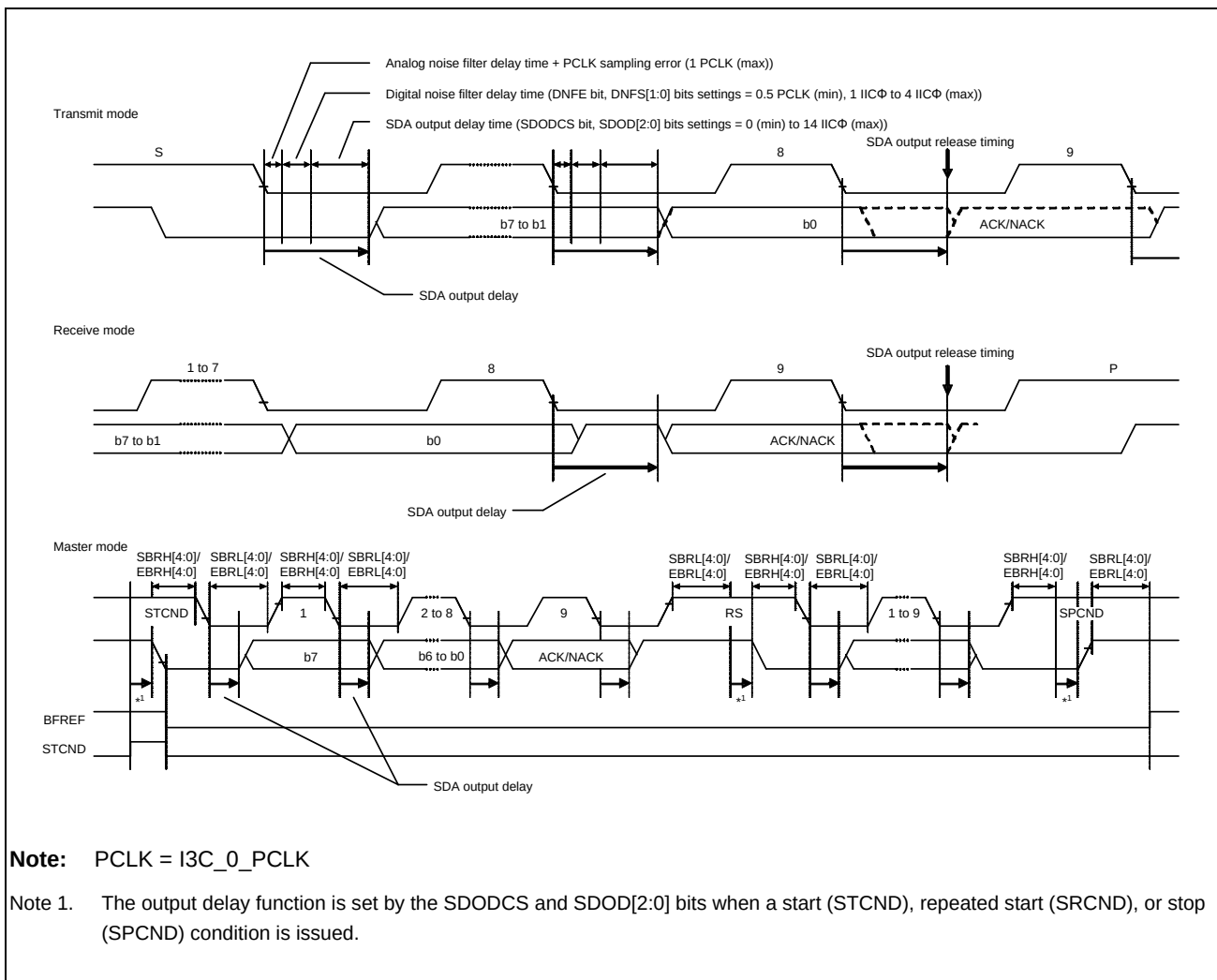


Figure 7.8-109 SDA output delay facility

(3) Digital Noise-Filter Circuits [I²C mode]

The states of the SCL and SDA pins are conveyed to the internal circuitry through digital noise-filter circuits. **Figure 7.8-110** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of I3C consists of 16 flip-flop circuit stages connected in series and a match detection circuit. During the Hs-mode transfer, only the first four flip-flop circuit stages are enabled.

The number of effective stages in the digital noise filter is selected by the INCTL.DNFS[3:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to sixteen I3C ϕ cycles.

The input signal to the SCL pin (or SDA pin) is sampled on rising edges of the I3C ϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the INCTL.DNFS[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (I3C_0_TCLK) and the transfer rate is small (For example, data transfer at 400 kbps with I3C_0_TCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.

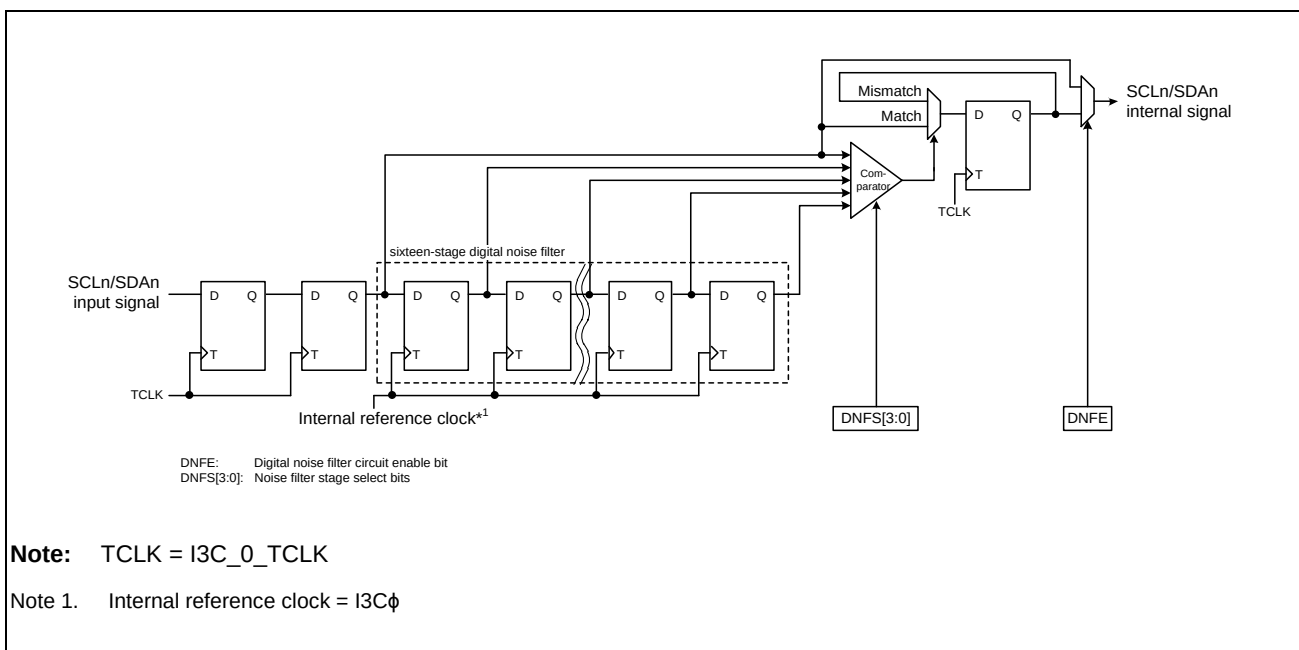


Figure 7.8-110 Block diagram of digital noise filter circuit

7.8.4.3 Operation

7.8.4.3.1 Initial Setting Flow

(1) I²C Initial Setting Flow (Single Buffer Transfer)

Before starting data transmission and reception, initialize I3C according to the procedure in **Figure 7.8-111**.

First, set the BCTL.BUSE bit to 0b (SCL, SDA pins not driven).

Next, set the RSTCTL.RI3CRST bit to 1b (I3C reset). This initializes all registers and the internal state. Then, waits for RI3CRST to become 0b.

This initializes the various flags and some registers. See **7.8.7 Reset Descriptions**.

After that, set registers SDATBASn.SDADLS, SDATBASn.SDSTAD[9:0], STDBR, INCTL, OUTCTL, TMOCTL, SCSTRCTL, ACKCTL, and BFCTL, then set the other registers as necessary (for initial settings of I3C, see **Figure 7.8-111**).

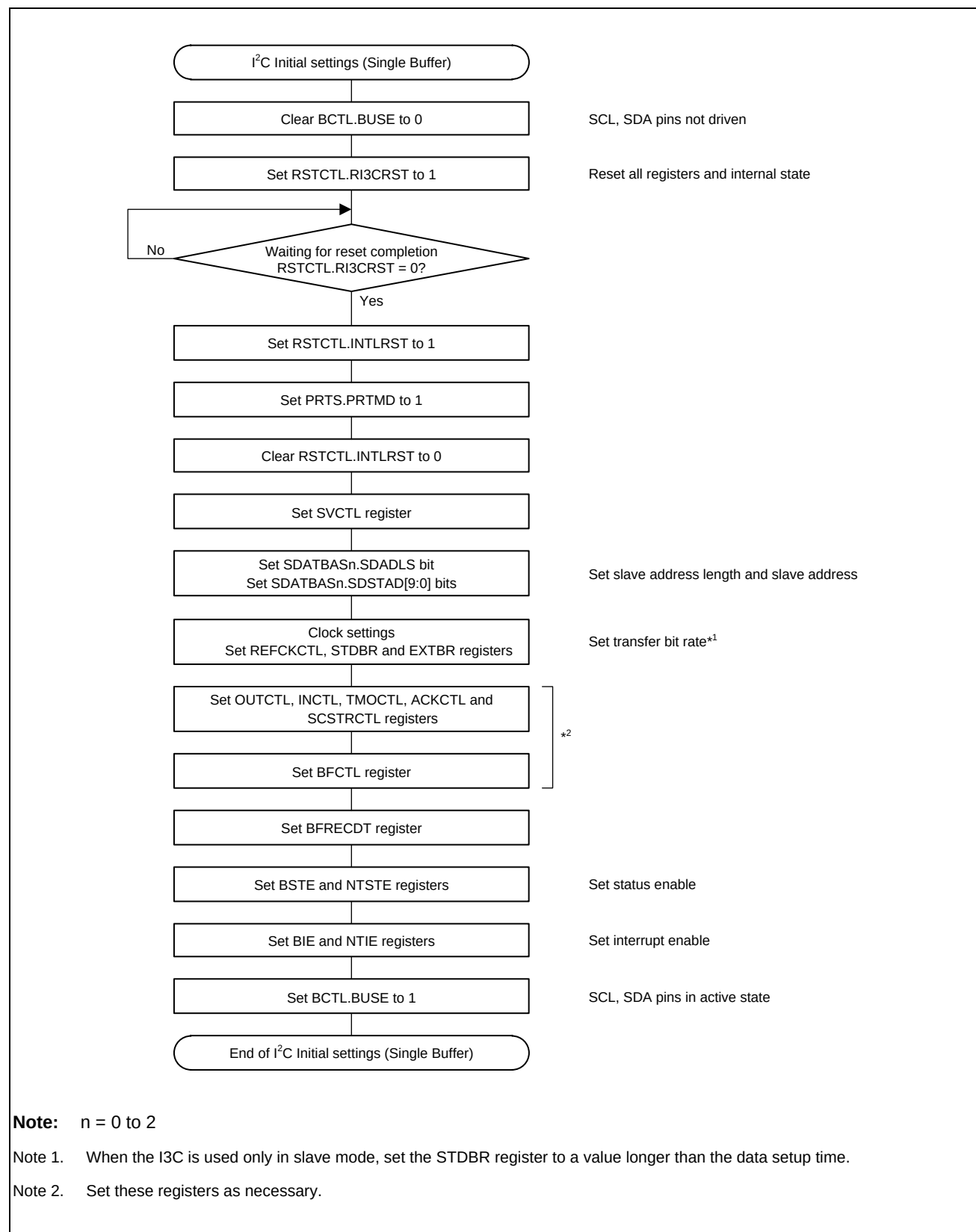


Figure 7.8-111 Example of I²C Initialization Flowchart (Single Buffer Transfer)

(2) I3C Initial Setting Flow

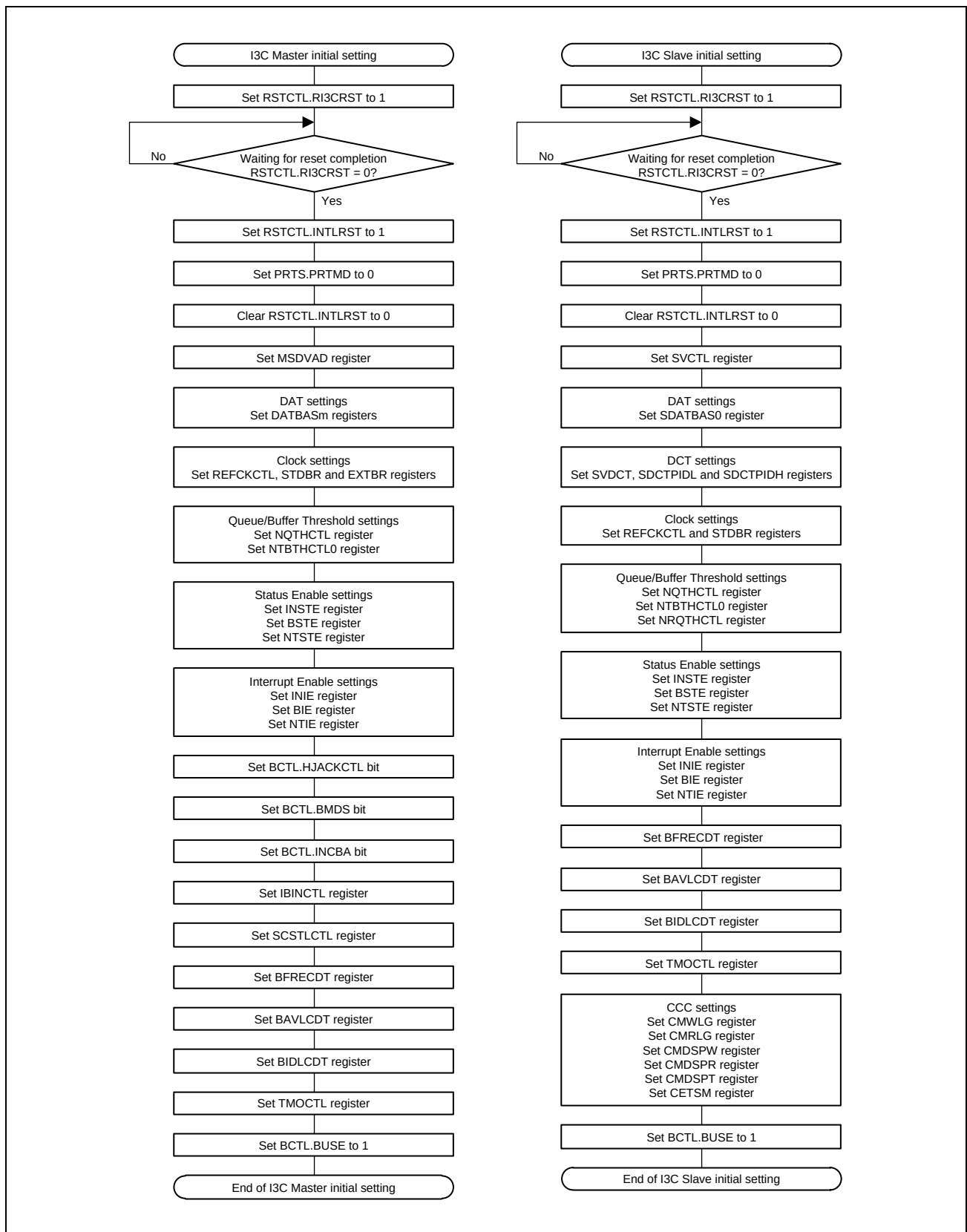


Figure 7.8-112 Example of I3C initialization flowchart

7.8.4.3.2 I3C Communication Flow

Figure 7.8-113 illustrates how I3C communication is initiated:

- All I3C communication occurs within a frame. The frame begins with a START, followed by one or more transfers, and a STOP.

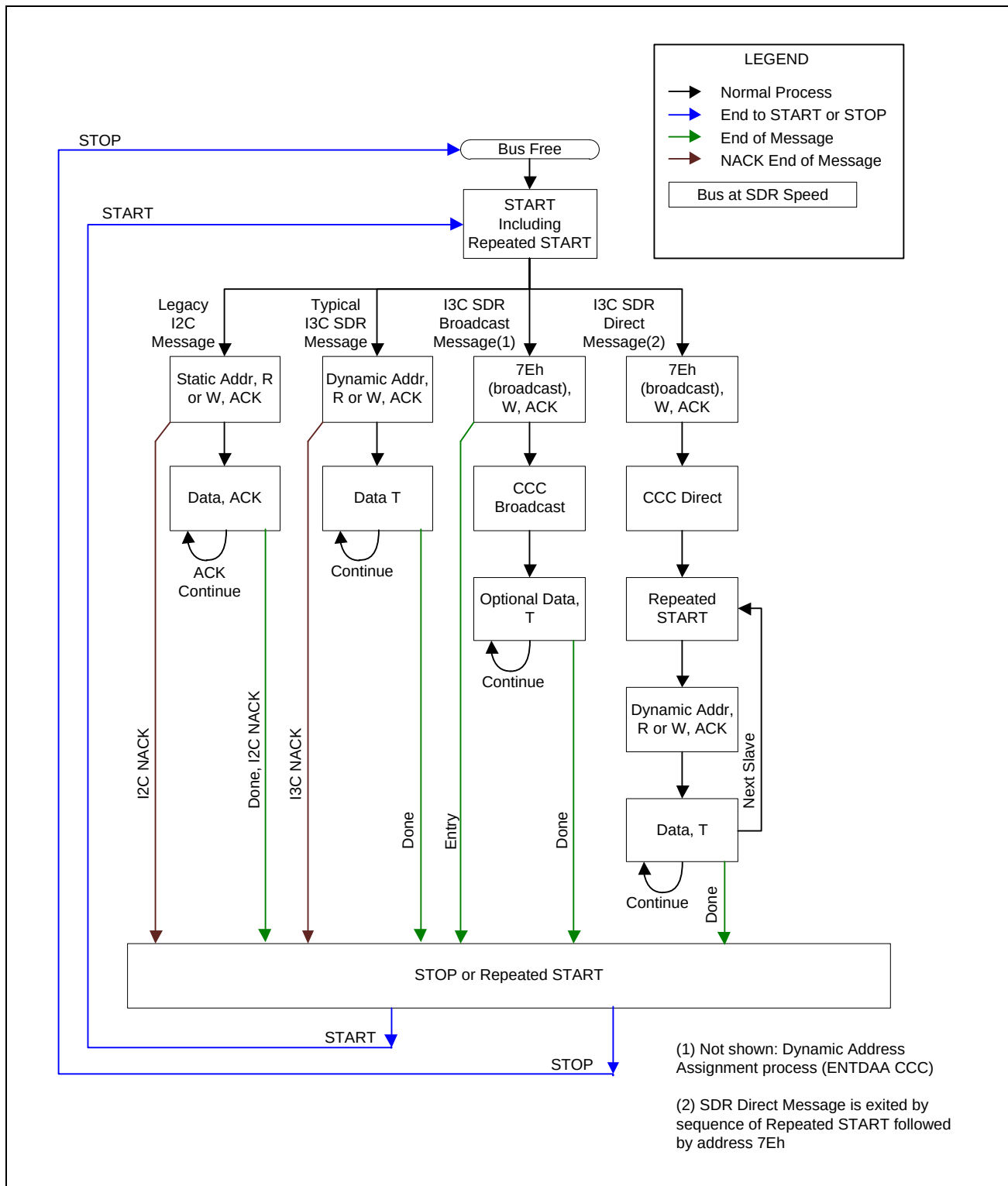


Figure 7.8-113 I3C communication flow

I3C is based on a frame encapsulation approach. A frame includes a data payload. The transfer protocol for the data payload is either SDR. Frames are bordered by I²C-like bus management.

The I3C frame always includes at least the START, the Header, the Data, and the STOP. The Header following a START allows for Bus Arbitration. The Master uses the Header to address Slave device (s). Slave devices (s) may use the Header Arbitration for multiple purposes: for In-Band Interrupt, for Hot-Join, and for Secondary Master functionality.

I3C allows only one Master to have control of the I3C bus at a time. Mechanisms for handoff of the Master role from one device to another device are provided.

7.8.4.3.3 Master Mode Communication Flow

(1) I²C Master Transmission Flow (Single Buffer Transfer)

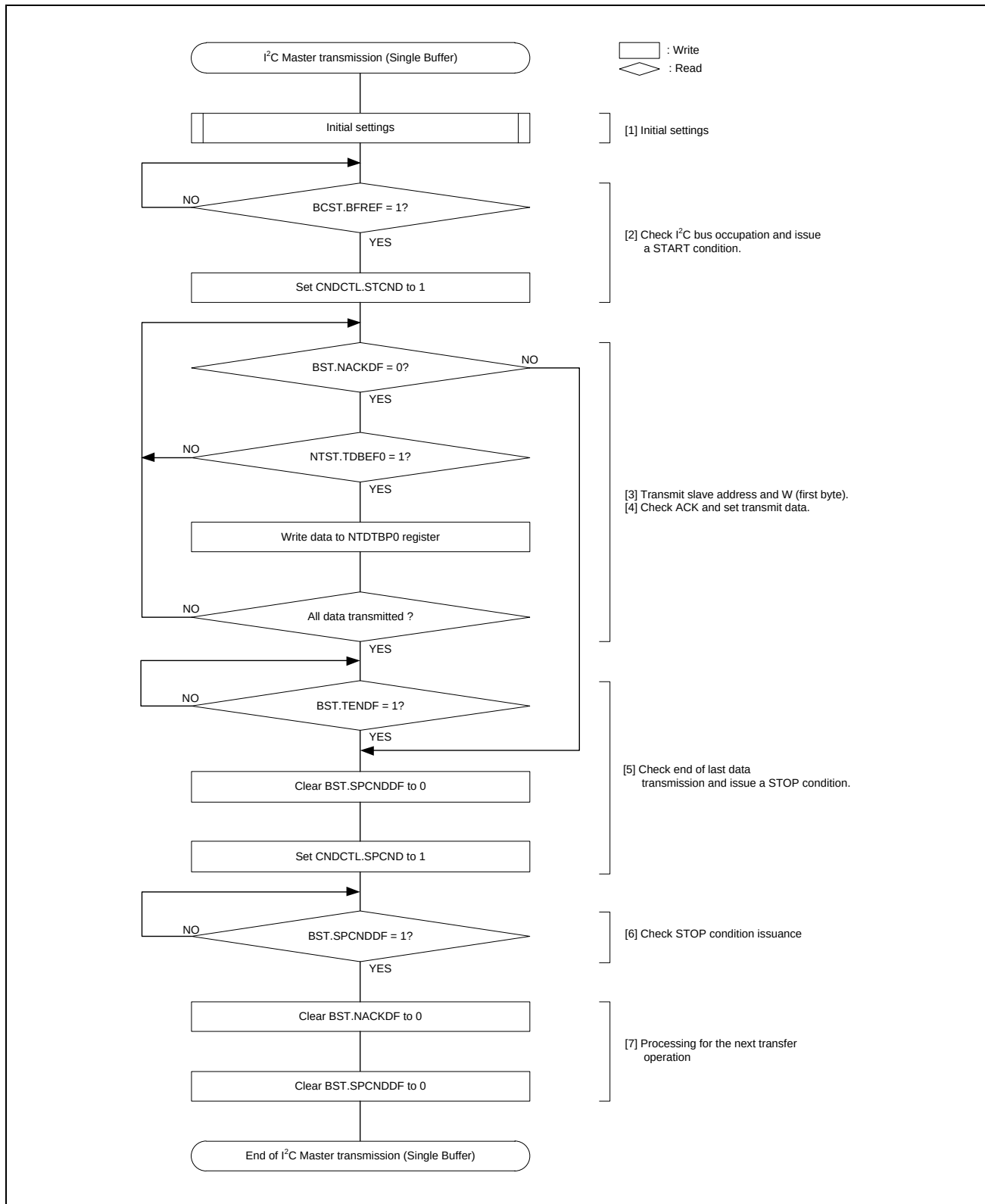
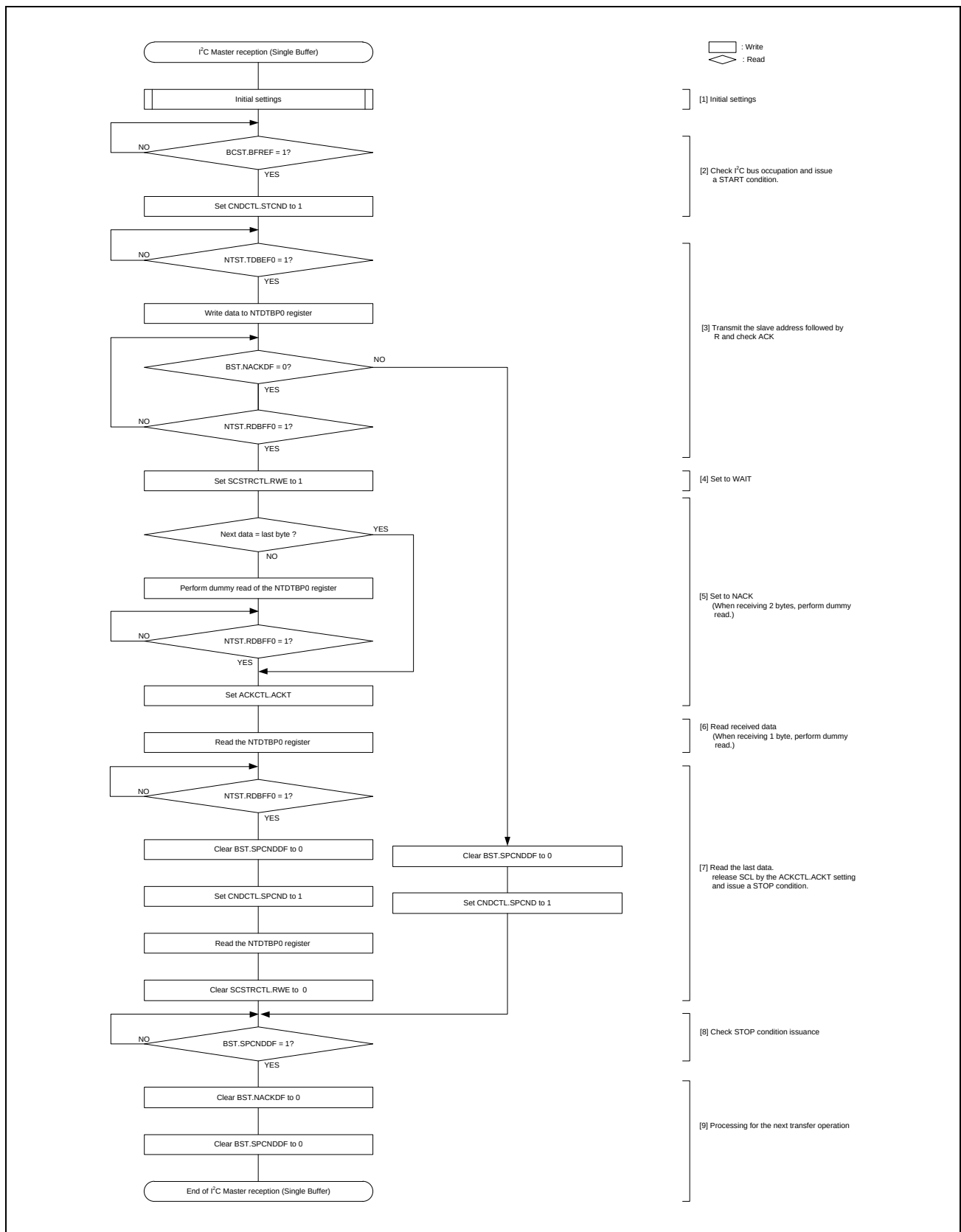
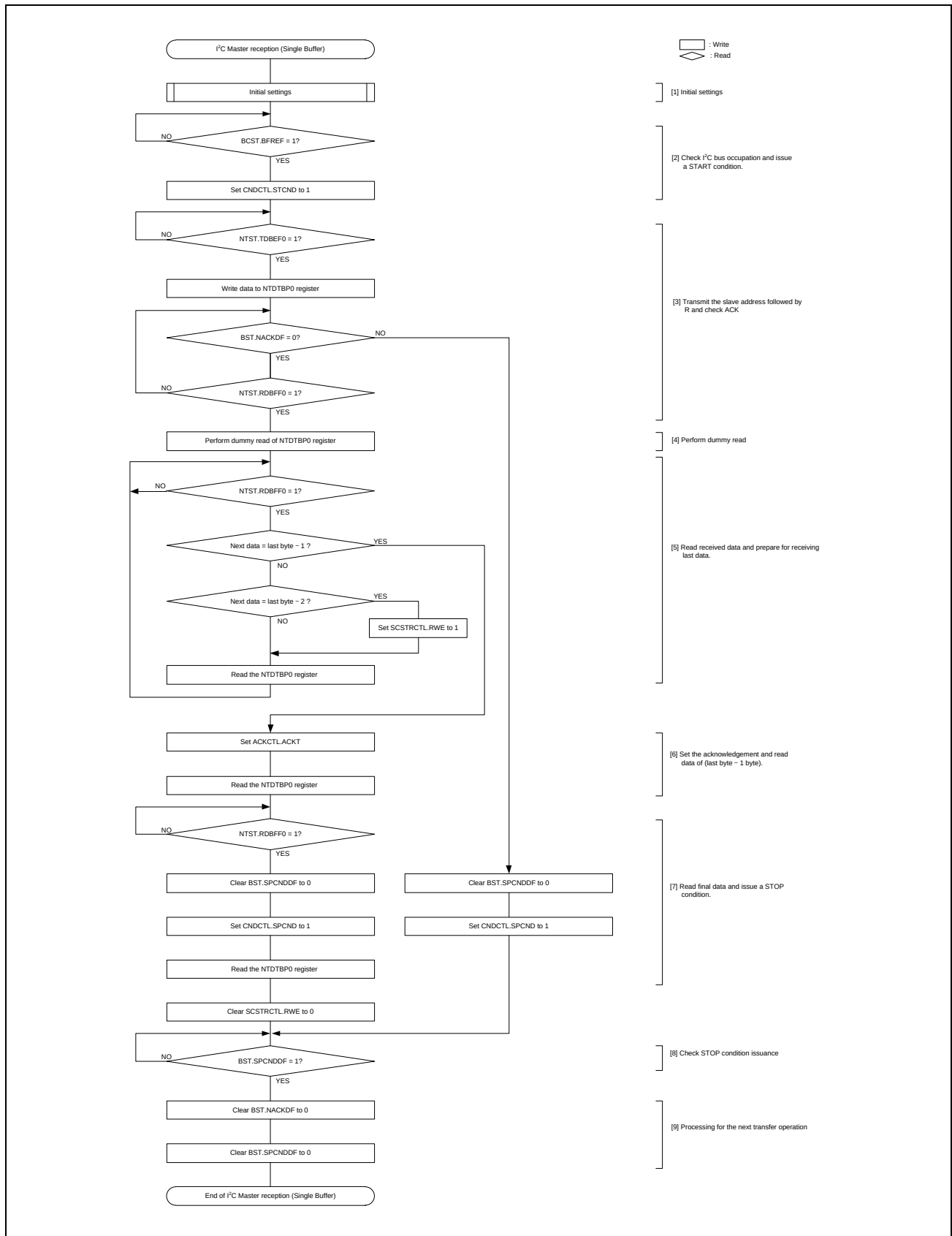


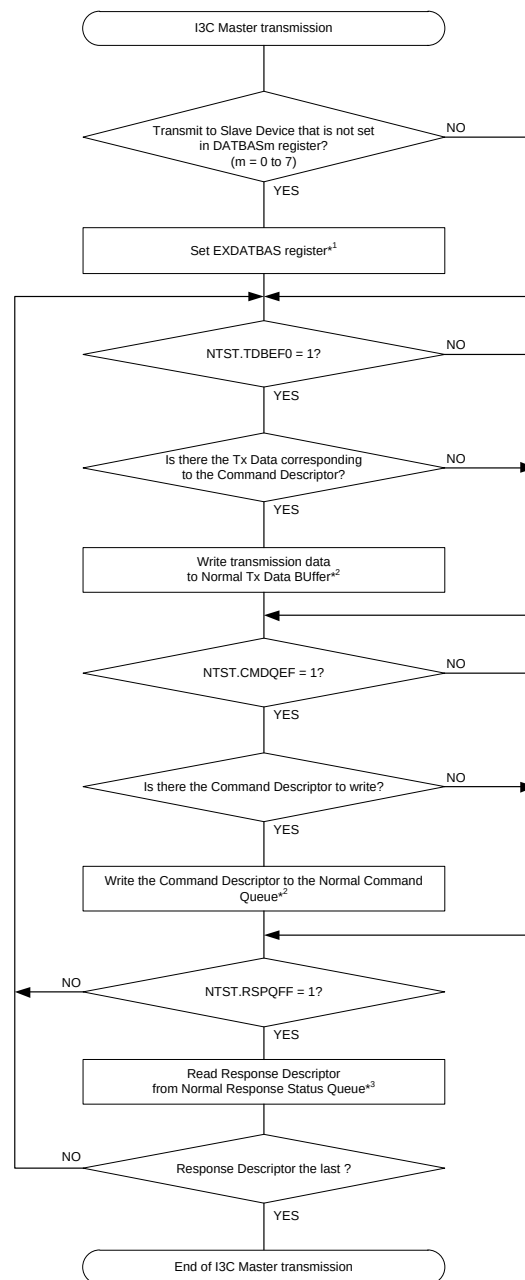
Figure 7.8-114 Example of I²C master transmission flowchart (single buffer transfer)

(2) I²C Master Reception Flow (Single Buffer Transfer)Figure 7.8-115 Example of I²C master reception flowchart (7-bit address format, 1 or 2 bytes)

Figure 7.8-116 Example of I²C master reception flowchart (7-bit address format, 3 bytes or more)

(3) I3C Master Transmission Flow (Normal FIFO Buffer Transfer)

Master transmission flow in I3C normal FIFO buffer transfer is common to Legacy I²C and SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.

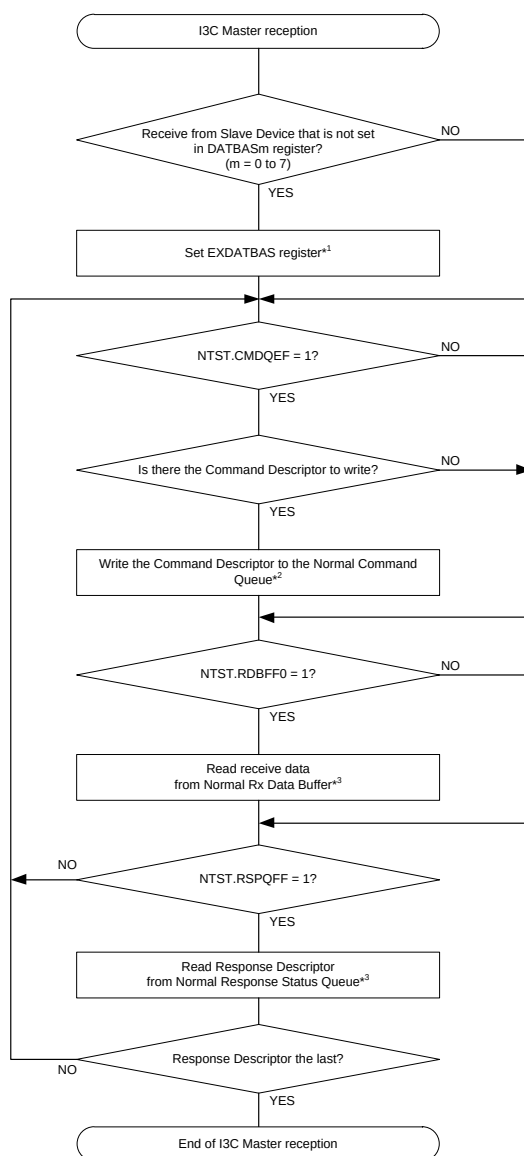
Note 2. In the case the register is written by a way other than DMAC, clear corresponding flags after write access.

Note 3. In the case the register is read by a way other than DMAC, clear corresponding flags after read access.

Figure 7.8-117 Example of I3C master transmission flowchart (normal FIFO buffer transfer)

(4) I3C Master Reception Flow (Normal FIFO Buffer Transfer)

Master reception flow in I3C normal FIFO buffer transfer is common to Legacy I²C and SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.

Note 2. In the case the register is written by a way other than DMAC, clear corresponding flags after write access.

Note 3. In the case the register is read by a way other than DMAC, clear corresponding flags after read access.

Figure 7.8-118 Example of I3C master reception flowchart (normal FIFO buffer transfer)

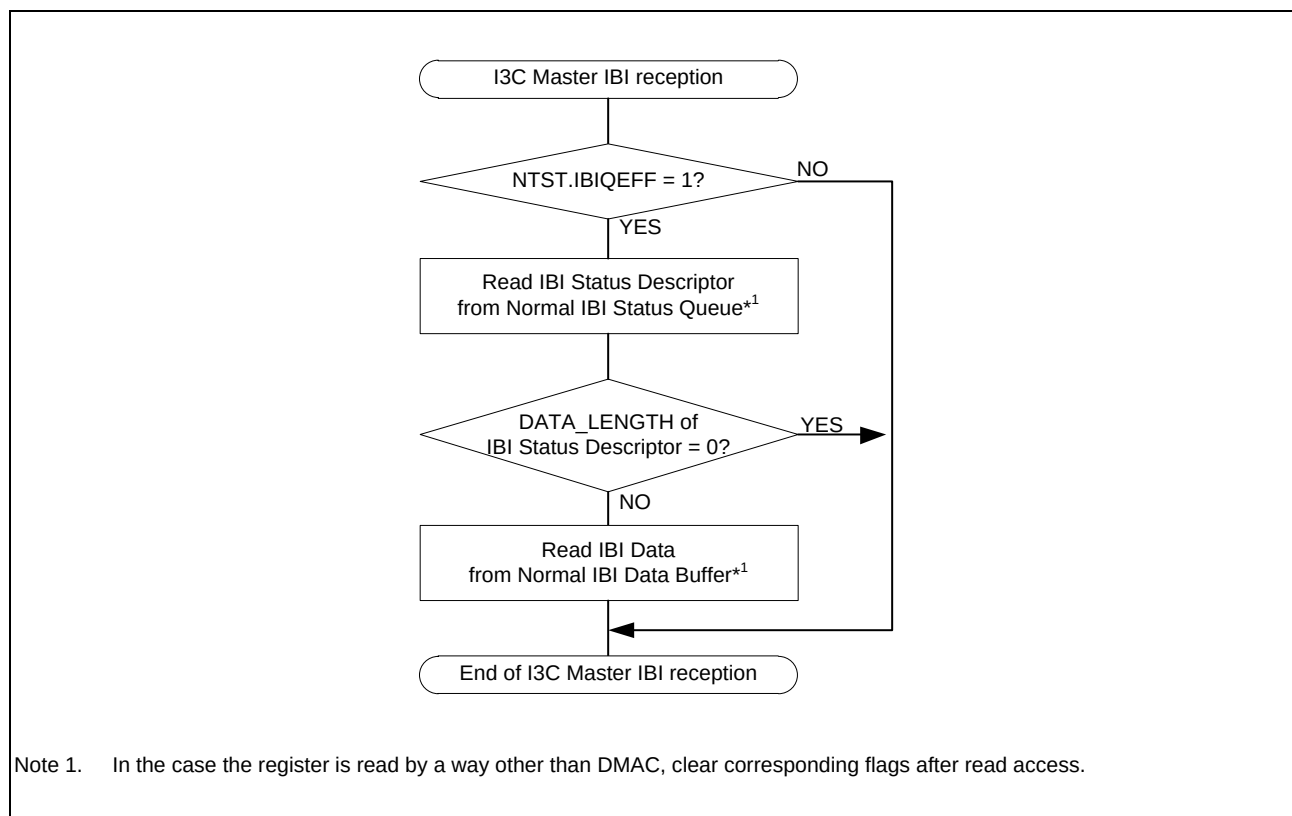
(5) I3C Master IBI Reception Flow

Figure 7.8-119 Example of I3C master IBI reception flowchart

(6) I3C Master Wake-Up Flow

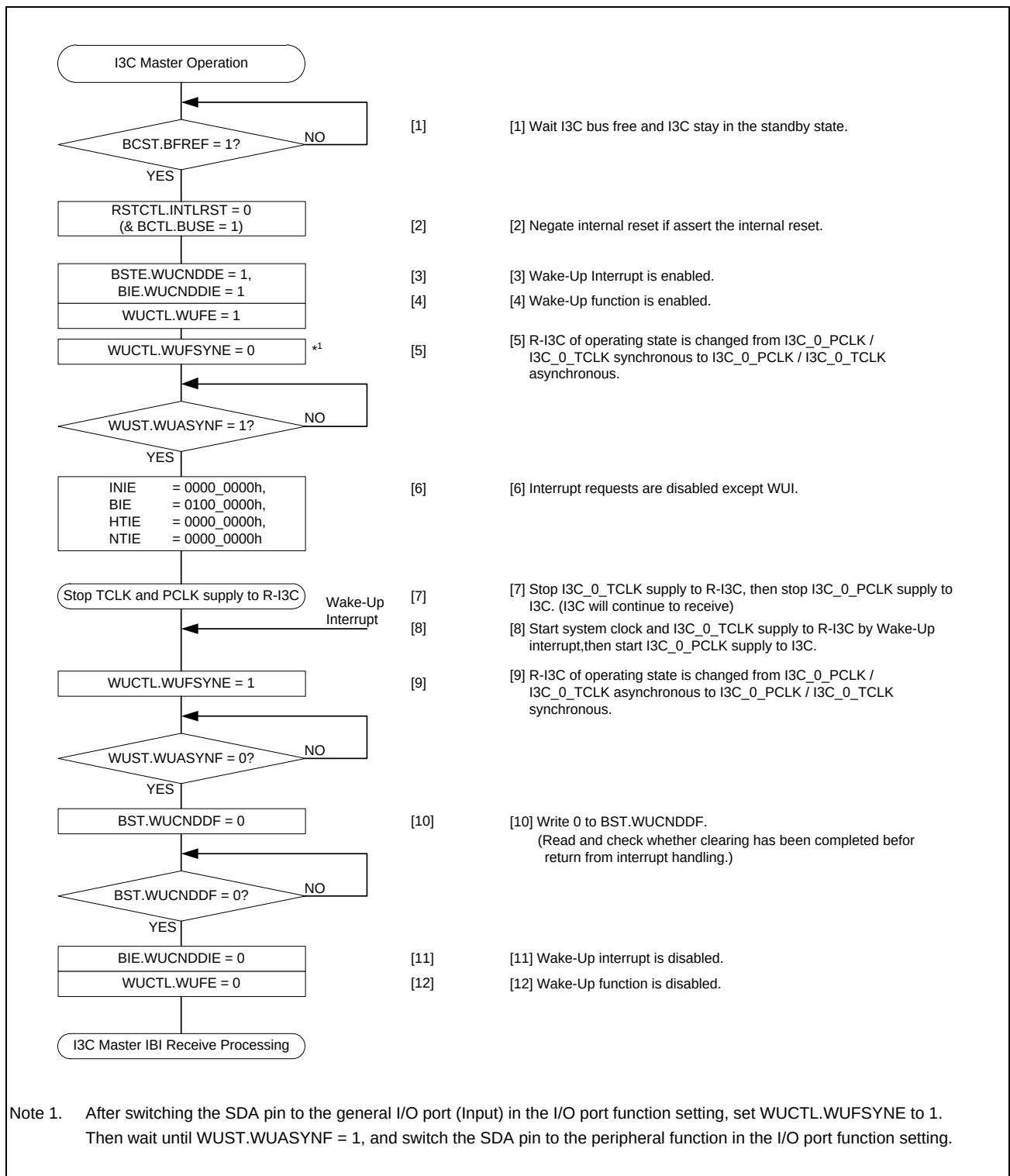


Figure 7.8-120 Use Case of I3C Master Wake-Up

7.8.4.3.4 Slave Mode Communication Flow

(1) I²C Slave Transmission Flow (Single Buffer Transfer)

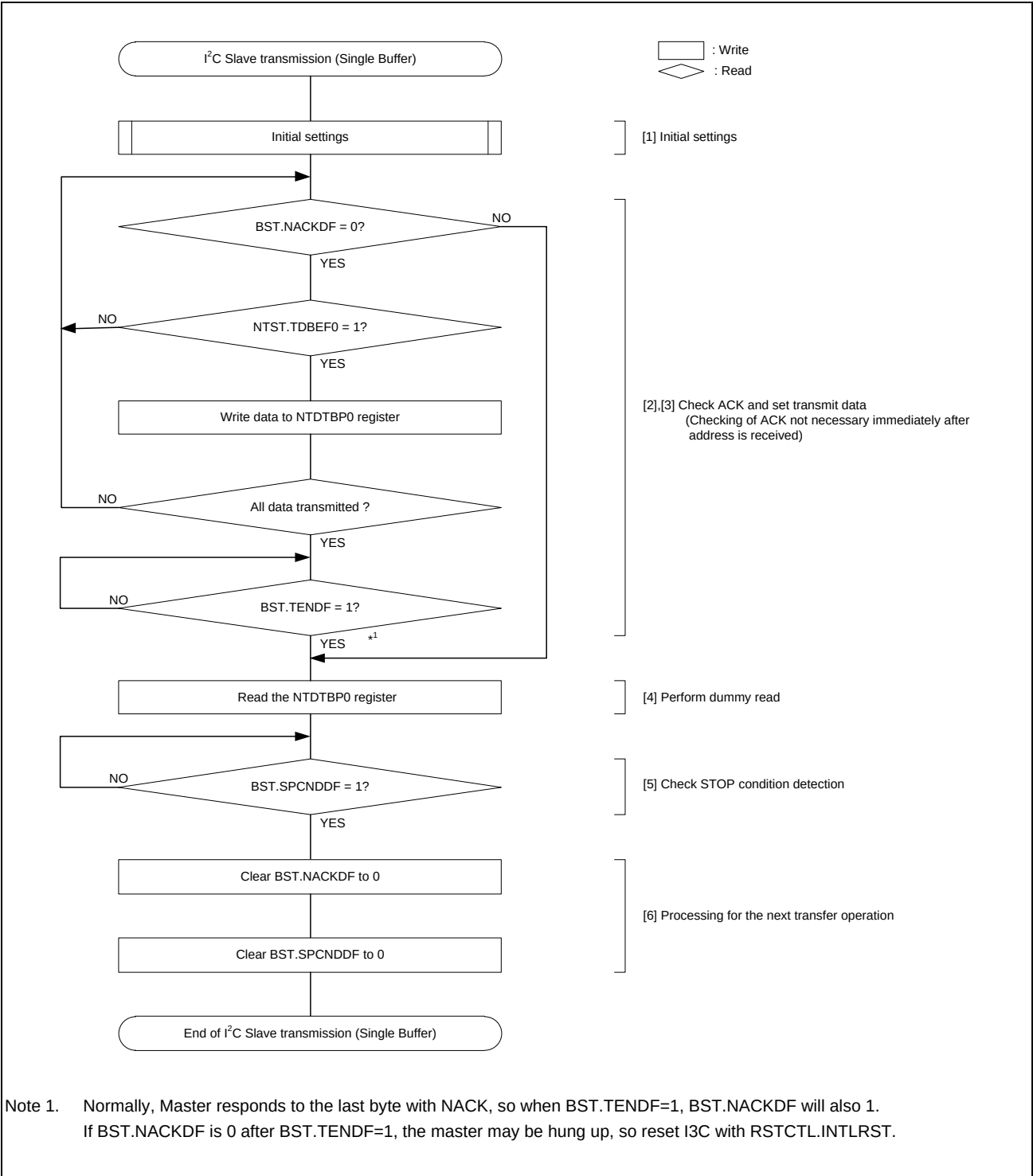


Figure 7.8-121 Example of I²C slave transmission flowchart (single buffer transfer)

(2) I²C Slave Reception Flow (Single Buffer Transfer)

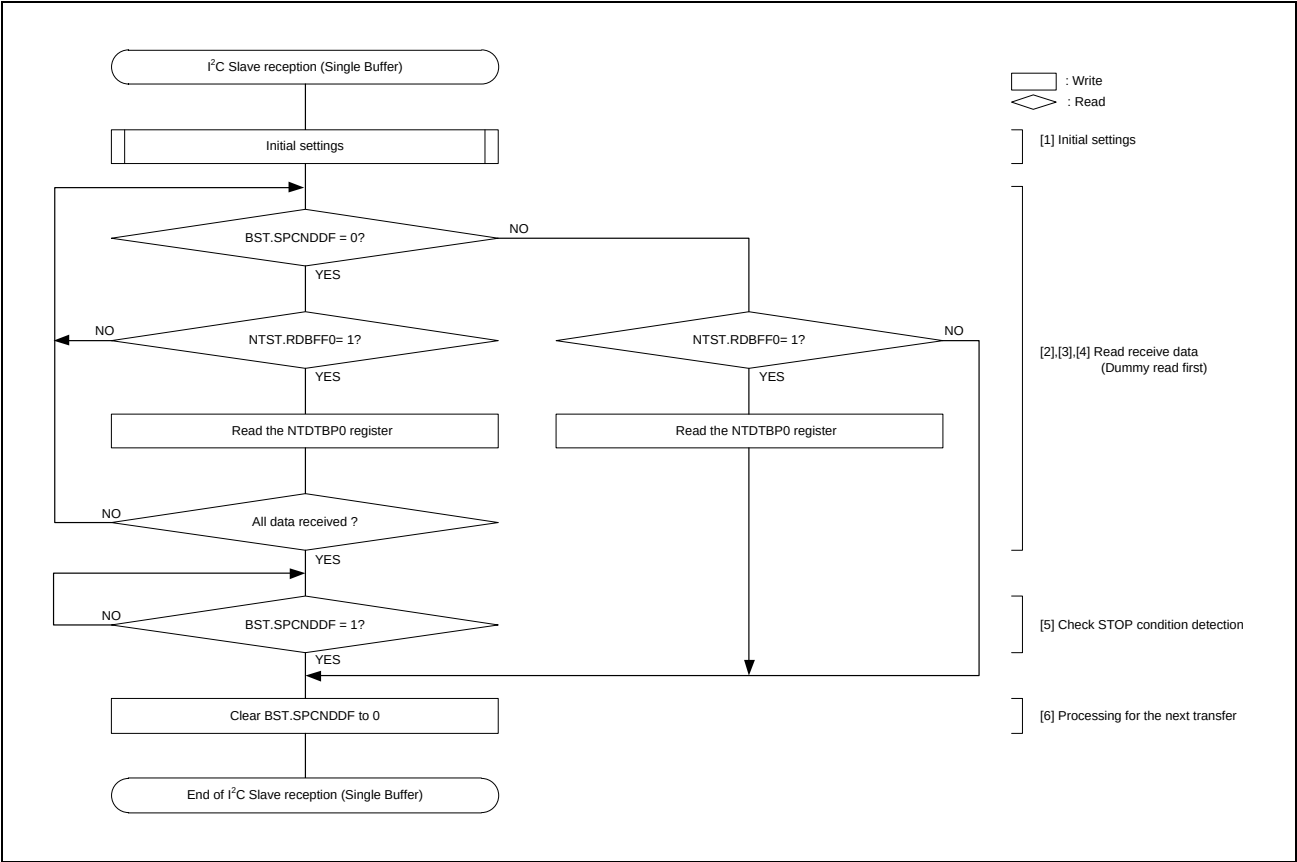


Figure 7.8-122 Example of I²C slave reception flowchart (single buffer transfer)

(3) I3C Slave Transmission Flow (Normal FIFO Buffer Transfer)

Slave Transmission Flow in I3C normal FIFO buffer transfer is common to Legacy I²C, SDR (Private Transfer, Broadcast CCC, Direct CCC).

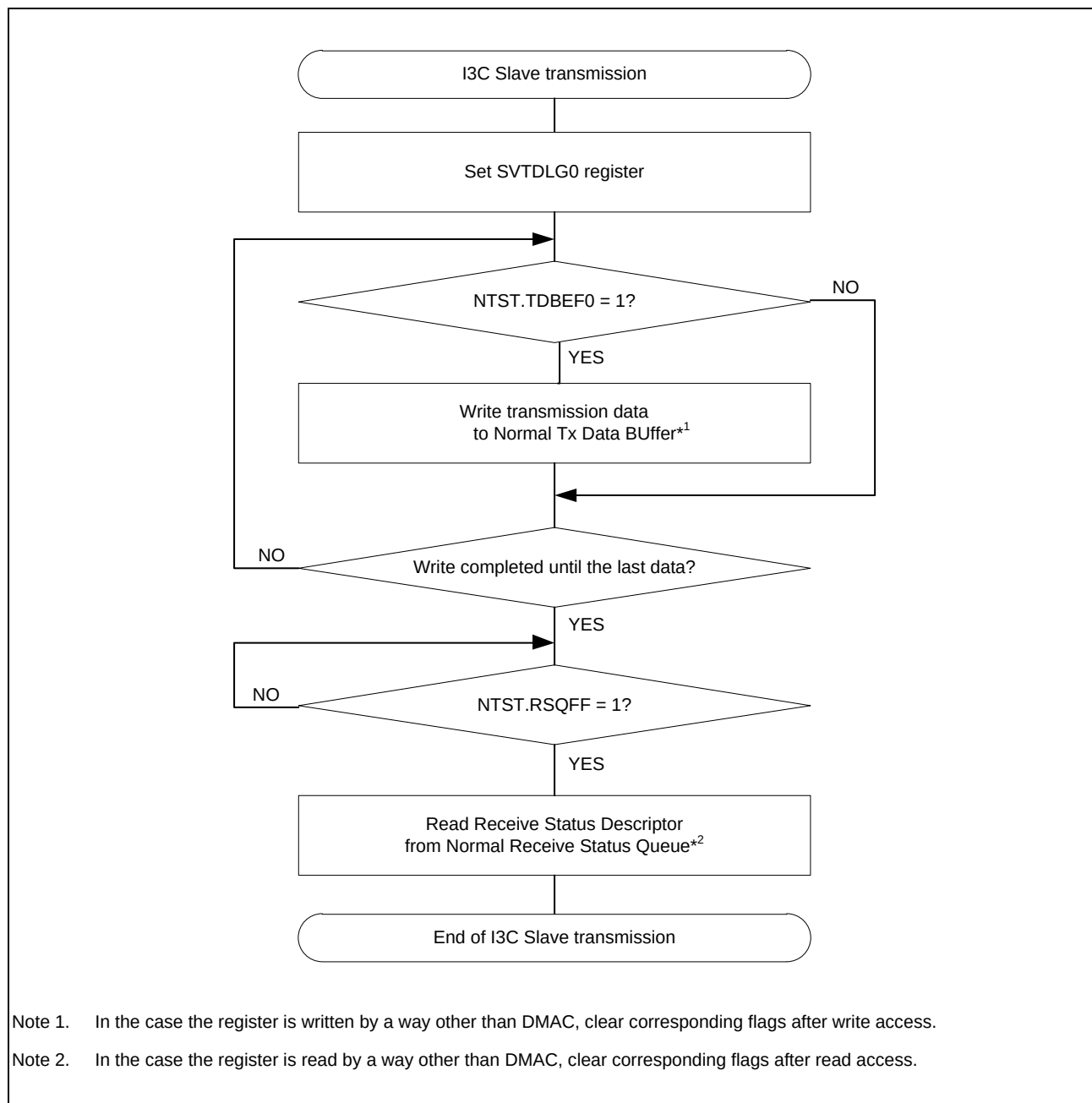


Figure 7.8-123 Example of I3C slave transmission flowchart (normal FIFO buffer transfer)

(4) I3C Slave Reception Flow (Normal FIFO Buffer Transfer)

Slave Reception Flow in I3C normal FIFO buffer transfer is common to Legacy I²C, SDR (Private Transfer, Broadcast CCC, Direct CCC).

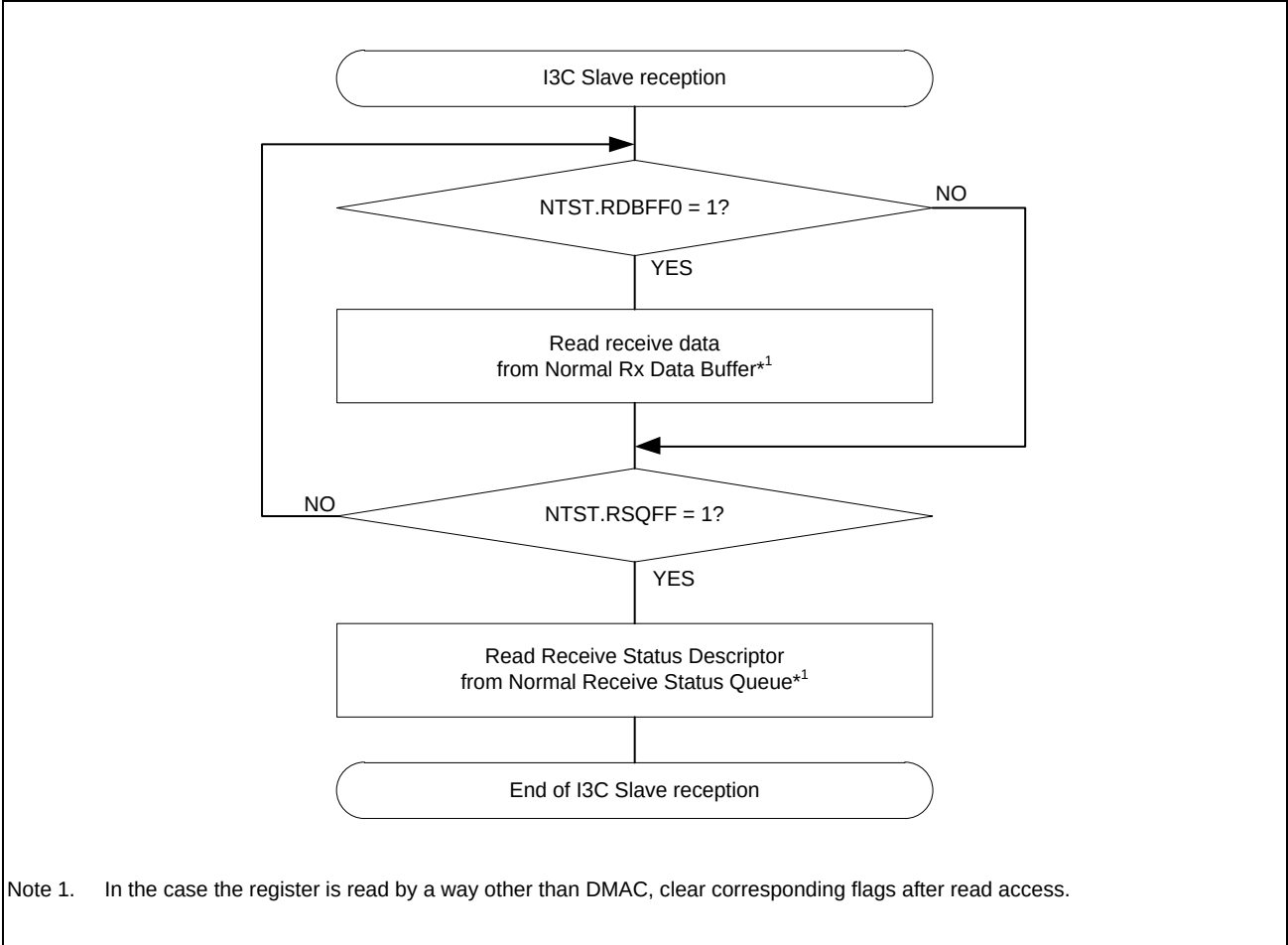


Figure 7.8-124 Example of I3C slave reception flowchart (normal FIFO buffer transfer)

(5) I3C Slave IBI Transmission Flow

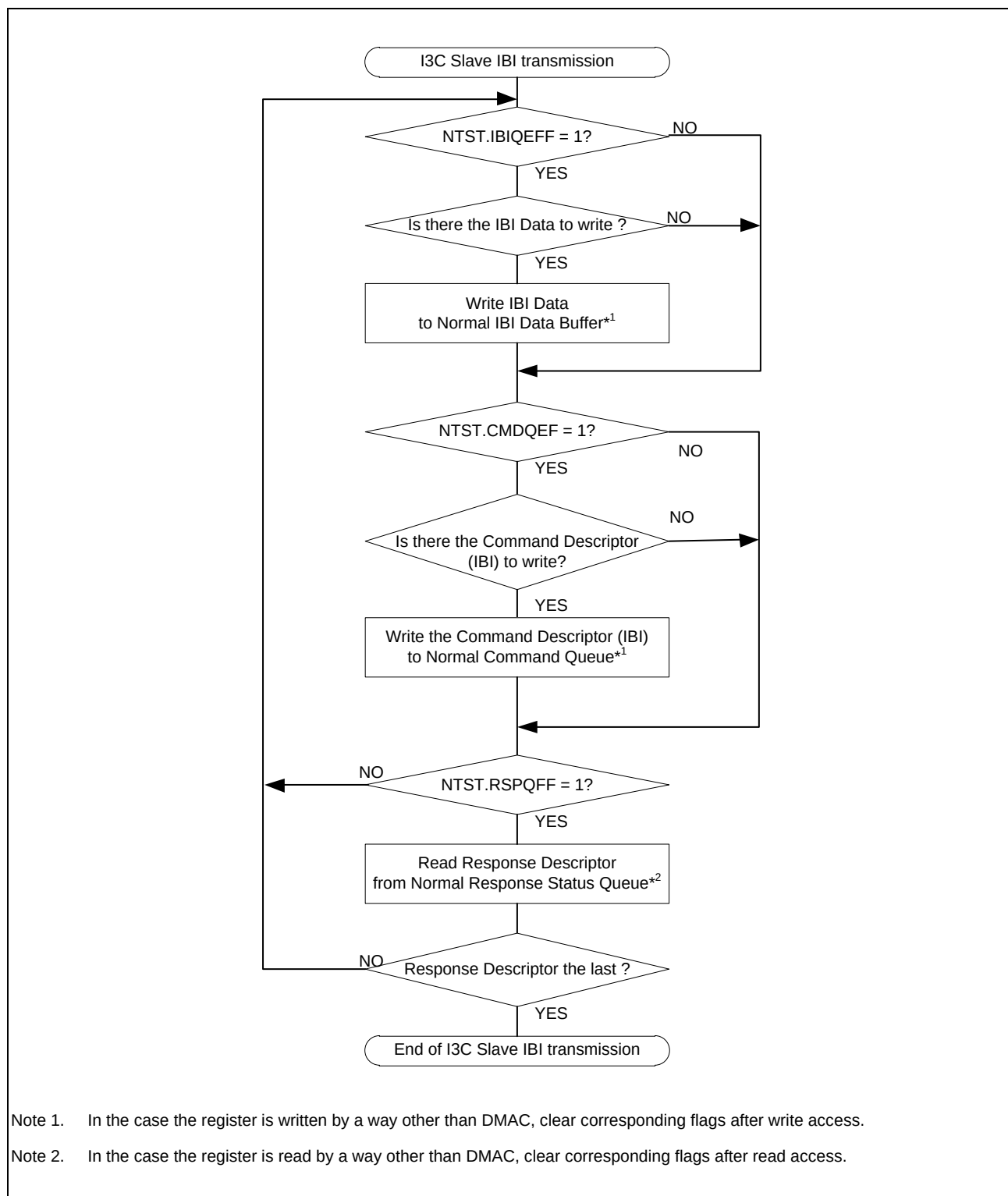


Figure 7.8-125 Example of I3C slave IBI transmission flowchart

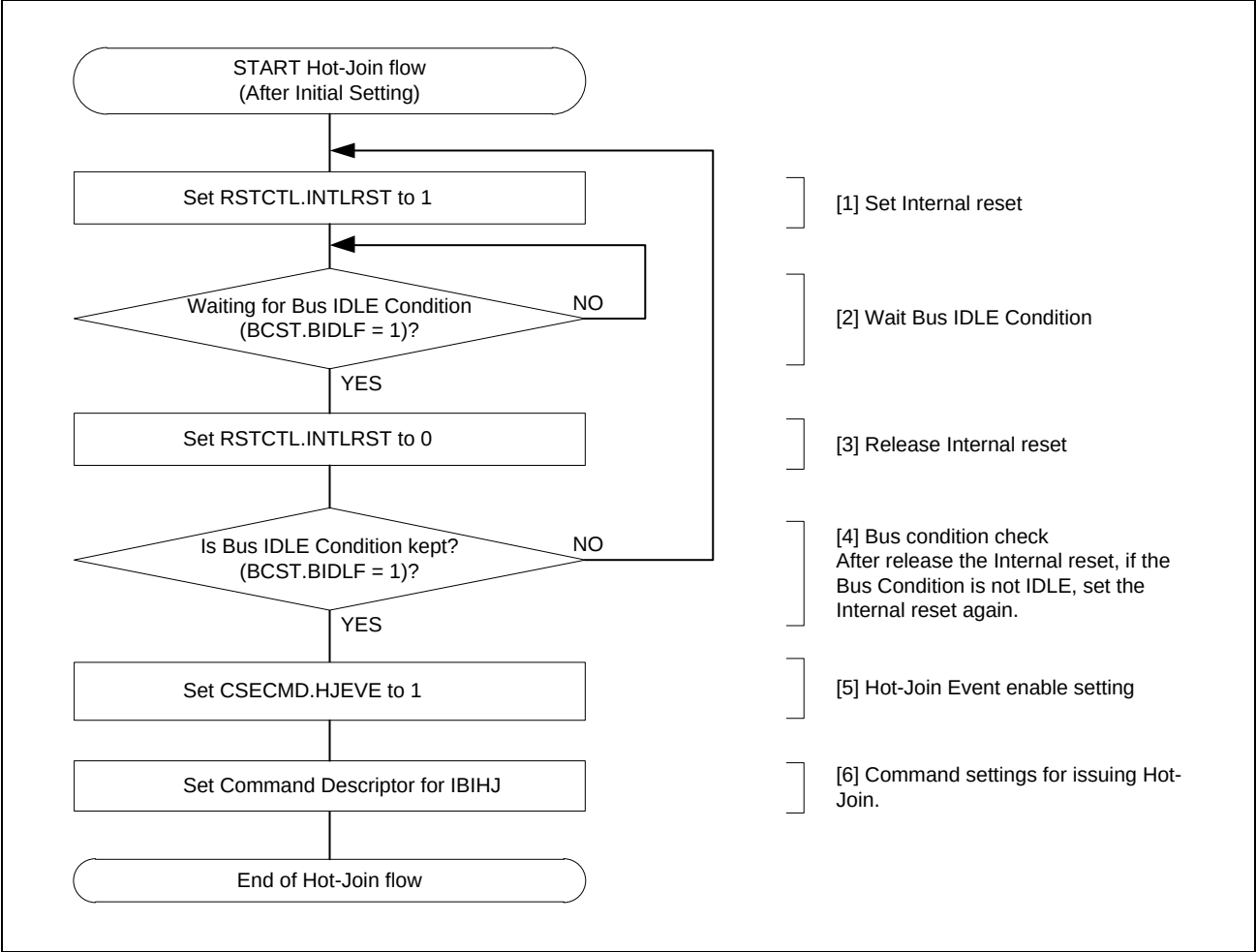


Figure 7.8-126 Hot-Join Flowchart after the I3C Bus has already been configured

(6) I3C Slave Wake-Up Flow

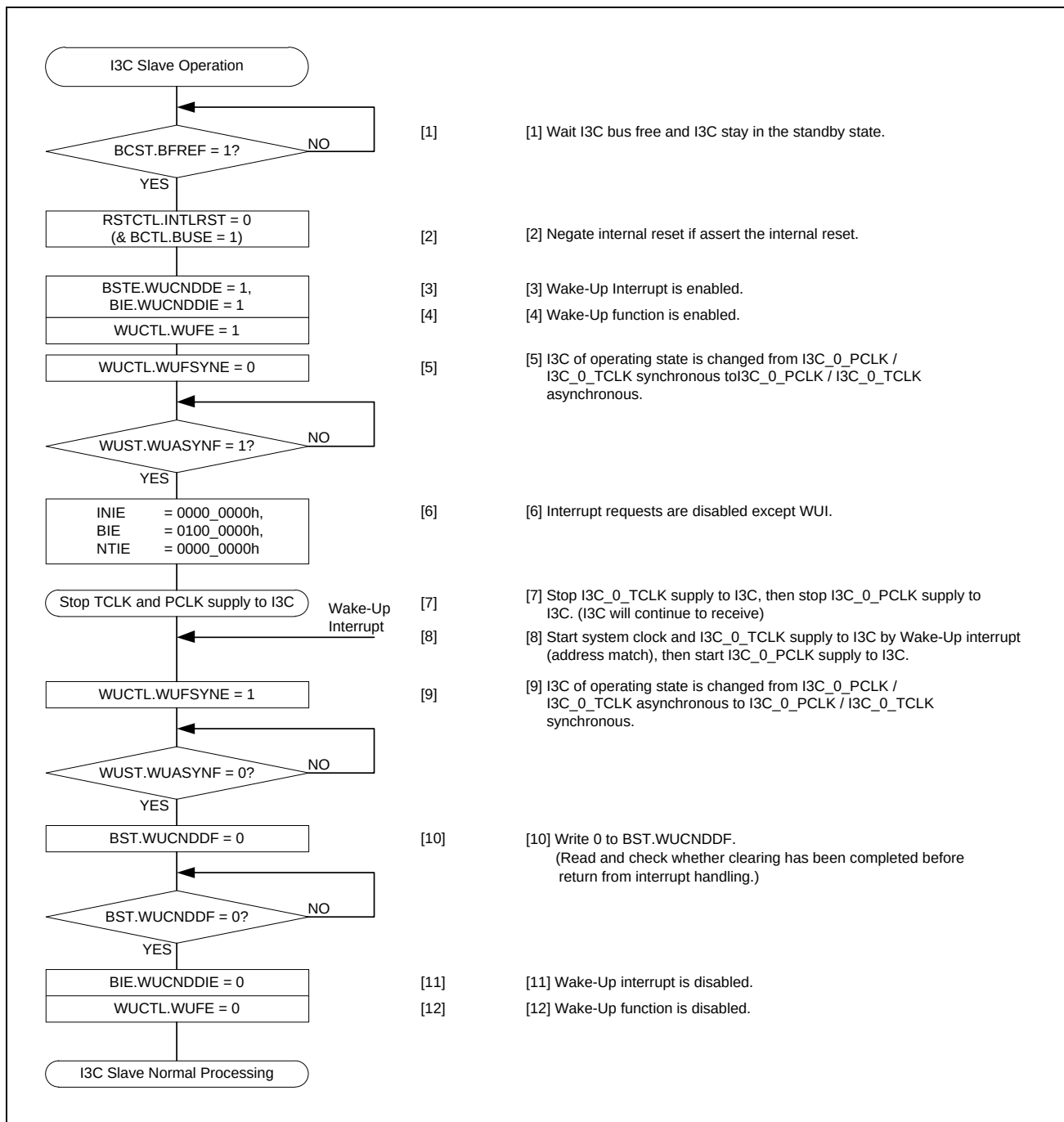


Figure 7.8-127 Use Case of I3C Slave Wake-Up (Wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

7.8.5 Interrupt Sources

I3C can generate the following interrupt requests:

7.8.5.1 Overview

The I3C has the interrupt factors shown in **Table 7.8-17**.

The interrupt indicated by Possible in the DMAC Activation column are capable of activating data transfer by the DMAC.

Table 7.8-17 Interrupt Generation

Symbol	Interrupt source	Interrupt flag	Support				DMAC Activation
			I ² C	I3CM	I3C2M	I3CS	
INT_ri3c_resp_n	Normal response status buffer full	NTST.RSPQFF	—	✓	✓	✓	Not possible
INT_ri3c_cmd_n	Normal command buffer empty	NTST.CMDQEF	—	✓	✓	✓	Not possible
INT_ri3c_ibi_n	Normal IBI status buffer empty/full	NTST.IBIQEFF	—	✓	✓	✓	Not possible
INT_ri3c_rx_n	Normal receive data buffer full	NTST.RDBFF0	✓	✓	✓	✓	Possible
INT_ri3c_tx_n	Normal transmit data buffer empty	NTST.TDBEF0	✓	✓	✓	✓	Possible
INT_ri3c_rcv_n	Normal receive status buffer full	NTST.RSQFF	—	—	✓	✓	Not possible
INT_ri3c_tend_n	Transmit end	BST.TENDF	✓	—	—	—	Not possible
INT_ri3c_st_n	Start condition detection	BST.STCNDDF	✓	✓	✓	✓	Not possible
INT_ri3c_sp_n	STOP condition detection	BST.SPCNDDF	✓	✓	✓	✓	Not possible
INT_ri3c_nack_n	NACK detection	BST.NACKDF	✓	—	—	—	Not possible
INT_ri3c_al_n	Arbitration lost	BST.ALF	✓	—	—	—	Not possible
INT_ri3c_tmo_n	Timeout detection	BST.TODF	✓	✓	✓	✓	Not possible
INT_ri3c_ierr_n	Non-recoverable internal error	INST.INEF	—	✓	✓	✓	Not possible
INT_ri3c_terr_n	Transfer Error	NTST.TEF	—	✓	✓	✓	Not possible
INT_ri3c_abort_n	Transfer Abort	NTST.TABTF	—	✓	✓	✓	Not possible
INT_ri3c_wu_n	Wake-up condition detection	BST.WUCNDDF	✓	✓	✓	✓	Not possible

Note: ✓: Support

—: Not support

Note: I²C: I²C Master/Slave (Single Buffer)

I3CM: I3C Master

I3C2M: I3C Secondary Master

I3CS: I3C Slave

7.8.6 Event Link Output

I3C handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Communication event

When a Communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition) occurs, the corresponding event signal can be output for another module via the ELC.

(2) Receive data full

When a receive data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

All event signals from I3C to ELC are shown in **Table 7.8-18**.

Table 7.8-18 Event Signals

Signal Name	Description	Support			
		I ² C	I3CM	I3C2M	I3CS
ri3c_elccommu	Communication event	✓	✓	✓	✓
ri3c_elcresp	Normal response status buffer full event	—	✓	✓	✓
ri3c_elccmd	Normal command buffer empty event	—	✓	✓	✓
ri3c_elcibi	Normal IBI status buffer empty/full event	—	✓	✓	✓
ri3c_elcrx	Normal receive data buffer full event	✓	✓	✓	✓
ri3c_elctx	Normal transmit data buffer empty event	✓	✓	✓	✓
ri3c_elcrxv	Normal receive status buffer full event	—	—	✓	✓
ri3c_elctend	Transmit end event	✓	—	—	—
ri3c_elcstev	Synchronous timing event	—	✓	✓	✓
ri3c_mrefovf	MREF counter overflow event	—	✓	✓	—
ri3c_mrefcpt	MREF capture event	—	✓	✓	—
ri3c_elcamev	Additional master-initiated bus event	—	✓	✓	—

Note: ✓: Support
— : Not support

Note: I²C: I²C Master/Slave (Single Buffer)
I3CM: I3C Master
I3C2M: I3C Secondary Master
I3CS: I3C Slave

7.8.6.1 Interrupt Handling and Event Linking

I3C module produces four kinds of interrupt: communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition), receive data full, transmit data empty, and transmit end interrupts. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits. For details on interrupt sources, see **7.8.5.1 Overview**.

7.8.7 Reset Descriptions

Table 7.8-19 Register states when issuing each condition (1)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
PRTS	PRTMD	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BCTL	BUSE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSM	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	ABT	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HJACKCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	INCBA	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDVAD	MDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MDYAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
RSTCTL	INTLRST	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	RDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	R13CRST	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
PRSST	PRSSTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TRMD	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	CRMS	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
INST	INEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
INSTE	INEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INIE	INEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INSTFC	INEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
DVCT	IDX[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
IBINCTL	NRSIRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NRMRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NRHJCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFCTL	HSME	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FMPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SMBS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-20 Register states when issuing each condition (2) (1/2)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
SVCTL	SVAE2	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE1	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HOAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	GCAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
REFCKCTL	IREFCKS[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
STDBR	DSBRPO	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
EXTBR	EBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFRECDT	FRECYC[8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BAVLCDT	AVLCYC[8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BIDLCDT	IDLCYC[17:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
OUTCTL	SDODCS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOD[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EXCYC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SOCWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	SCOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INCTL	DNFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DNFS[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
TMOCTL	TOMDS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOHCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOLCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODTS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
WUCTL	WUFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUFSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUANFS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUACKS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ACKCTL	ACKTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	ACKT	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	ACKR	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
SCSTRCTL	RWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKTWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 7.8-20 Register states when issuing each condition (2) (2/2)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
SCSTRCTL	PARPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AAPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STLCYC[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVTDLG0	STDLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
STCTL	STOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATCTL	CDIV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AMEOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MREFOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATTRG	ATSTRG	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
ATCCNTE	ATCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-21 Register states when issuing each condition (3)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
CNDCTL	SPCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	SRCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	STCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
NCMDQP	NCMDQP[31:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved
NRSPQP	NRSPQP[31:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved
NTDTBP0	NTDTBP0[31:0]	In reset	In reset	In reset	Saved	Saved	In reset	In reset	Saved	Saved
NIBIQP	NIBIQP[31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved
NRSQP	NRSQP[31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset
NQTHCTL	IBIQTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIDSSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQTH[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQTH[1: 0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTBTHCTL0	RXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RXDBTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXDBTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRQTHCTL	RSQTH	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BST	WUCNDDF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	ALF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TENDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
BSTE	WUCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-22 Register states when issuing each condition (4)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
BIE	WUCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BSTFC	WUCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTST	RSQFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset
	TEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TABTF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFF	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved
	CMDQEF	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved
	IBIQEFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved
	RDBFF0	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved
	TDBEF0	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved
NTSTE	RSQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTIE	RSQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-23 Register states when issuing each condition (5)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQCRST	RSPQCRST	TDBRST	RDBRST	IBIQCRST	RSQCRST
NTSTFC	RSQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-24 Register states when issuing each condition (6)

Register symbol	Register bit name	System reset	RSTCTL Register							
			RIBCRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
BCST	BIDLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BAVLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BFREF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVST	SVAF2	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF1	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF0	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	HOAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	GCAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
WUST	WUASYNF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MRCCPT	MRCCPT[31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
DATBAS _m (m = 0 to 7)	DVTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVNACK[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVDYAD[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBITS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVMRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSIRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSTAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-25 Register states when issuing each condition (7)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
EXDATBAS	EDTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDNACK[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDDYAD[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDSTAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDATBAS _n (n = 0 to 2)	SDDYAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDSTAD[9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDCT _m (m = 0 to 7)	RBCR[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVDCT	TBCR[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDCR[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-26 Register states when issuing each condition (8)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
SDCTPIDL	SDCTPIDL[31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDCTPIDH	SDCTPIDH[31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVDVADn (n = 0 to 2)	SDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SSTADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SADLG	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAD[9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CSECMD	HJEVE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MSRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVIRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CEACTST	ACTST[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMWLG	MWLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMRLG	IBIPSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSTMD	TSTMD[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CGDVST	VDRSV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACTMD[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PNDINT[3:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPW	MSWDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPR	CDTTIM[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MSRDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPPT	MRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRTTIM[23:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 7.8-27 Register states when issuing each condition (9)

Register symbol	Register bit name	System reset	RSTCTL Register							
			R13CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST
CETSM	INAC[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FREQ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTASYN[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTSYN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSS	ICOVF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	ASYNE[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BITCNT	BCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
NQSTLV	IBISCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved
	IBIQLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved
	RSPQLV[7:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved
	CMDQFLV[7:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved
NDBSTLV0	RDBLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved
	TDBFLV[7:0]	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved
NRSQSTLV	RSQLV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
PRSTDBG	SDOLV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCOLV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSERRCNT	M2ECNT[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SC1CPT	SC1C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
SC2CPT	SC2C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

SECTION 7 LOW-SPEED INTERFACE

7.9 CAN-FD Interface (CANFD)

7.9.1 Overview

The LSI has a 6-channel CAN-FD module (CANFD) that complies with ISO 11898-1 (2015) Standards. CANFD transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits). **Table 7.9-1** lists the specifications of the CANFD, **Figure 7.9-1** shows a block diagram of the CANFD, and **Table 7.9-2** lists the I/O pins.

Table 7.9-1 CANFD Specifications (1/2)

Parameter	Description
Number of channels	Six channels
Protocol	CAN-FD ISO 11898-1 (2015)
Communication speed	- Classical CAN mode: 1 Mbps - CAN FD mode*1 - Nominal bit rate: Max. 1 Mbps - Data bit rate: Max. 8 Mbps
Buffer	Total 320 buffers (when frame size is 76 bytes) - Individual buffers: 192 buffers (32 buffers × 6 channels) - Transmit buffer: 32 buffers per channel - Transmit queue: 4 queues per channel (shared with transmit buffer, up to 16 buffers allocatable) - Shared buffers: 384 buffers for all channels - Receive buffer: 0 to 96 buffers - Receive FIFO buffer: 8 FIFO buffers (up to 384 buffers allocatable to each) - Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 384 buffers allocatable to each)
Reception function	- Receives data frames and remote frames - Selects ID format (standard ID, extended ID, or both IDs) to be received - Sets interrupt enable/disable for each FIFO - Mirror function (reception of messages transmitted from the CAN node itself) - Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	- Selects receive message according to 768 receive rules - Sets the number of receive rules (0 to 127) for each channel - Acceptance filter processing: Sets ID and mask for each receive rule - DLC filter processing: Enables DLC filter check for each acceptance rule
Receive message transfer function	- Routing function - Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) - Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer - Label addition function - Stores label information together with a message in a receive buffer and FIFO buffer
Transmission function	- Transmits data frames and remote frames - Selects ID format (standard ID, extended ID, or both IDs) to be transmitted - Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer - Selects ID priority transmission or transmit buffer number priority transmission - Transmit request can be aborted (possible to confirm with a flag) - One-shot transmission function
Interval transmission function	Transmit message at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)

Table 7.9-1 CANFD Specifications (2/2)

Parameter	Description
Transmit queue function	Transmits all stored messages according to the ID priority
Transmit history function	Stores the history information of transmission-completed messages Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information
Gateway function	Transmits a received message automatically
Bus off recovery mode selection	Selects the method for returning from bus-off state. • ISO 11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from the bus-off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, from error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock) • Defects error status transitions (error warning, error passive, bus-off entry, and bus-off recovery) • Reads the error counter • Monitors DLC errors
Interrupt source	8 interrupt sources • Global interrupts (2 sources) RX FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) Channel transmit interrupt Channel error interrupt Common RX FIFO interrupt 14 DMA requests • 8 for RX FIFO • 6 for the first common FIFO (1 source/channel)
CAN clock source	CANFD_0_clkc (80 MHz)
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Inter-channel communication test (CRC error test enabled)

Note 1. 176 LQFP and 128 LQFP products do not support CAN-FD mode.

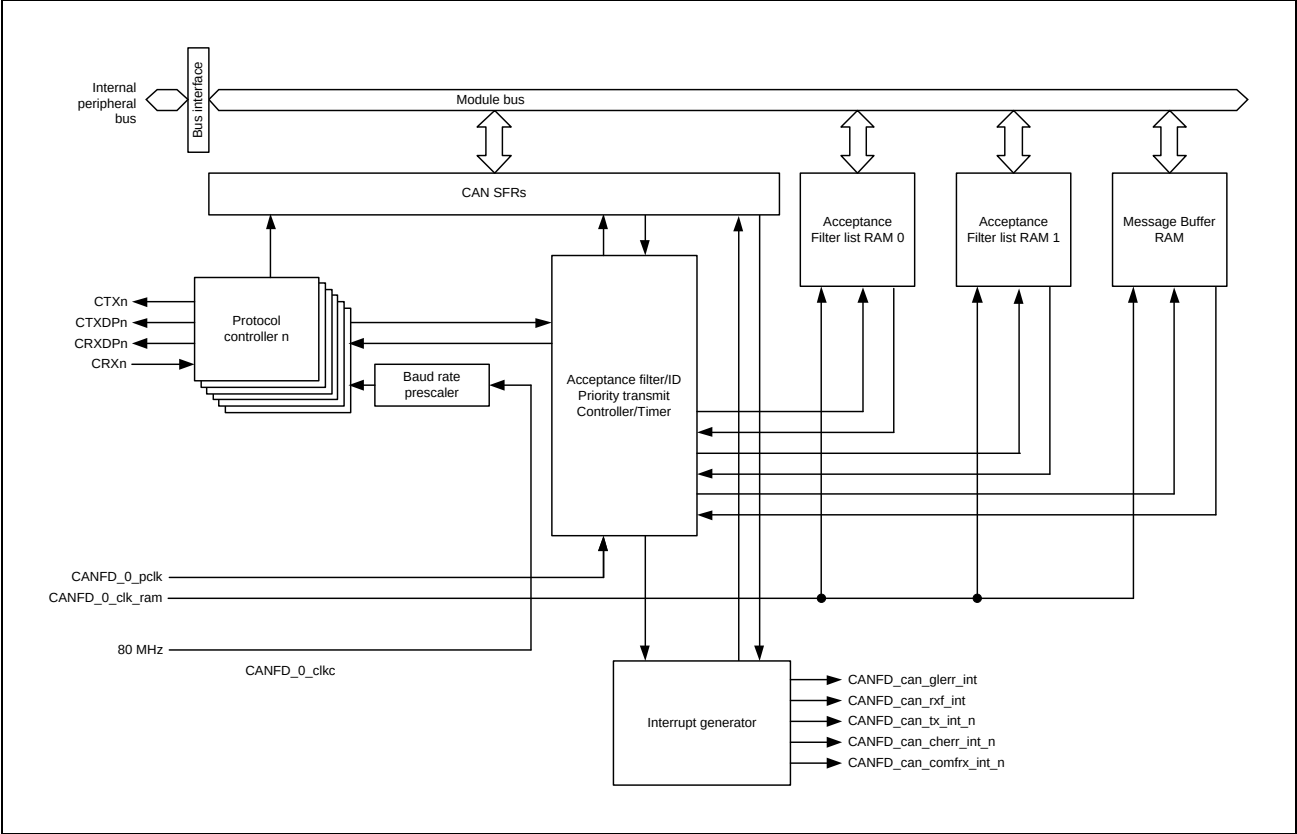


Figure 7.9-1 Block Diagram of CANFD

Table 7.9-2 lists the input/output pins of the CANFD module.

Table 7.9-2 Pin Configuration of CANFD

Channel	Pin Name	Input/Output	Function
CANFDn	CRXn	Input	CANFDn receive data input
	CTXn	Output	CANFDn transmit data output
	CRXDPn	Output	CANFDn receive data phase output This signal indicates the received data phase of a CAN-FD frame (active high).
	CTXDPn	Output	CANFDn transmit data phase output This signal indicates the transmission data phase of a CAN-FD frame (active high).

Note: n = 0 to 5

Table 7.9-3 CANFD Interrupt Sources

Name	Interrupt Sources	DMAC Activation
CANFD_can_rxf_int	RX FIFO interrupt	Not possible
CANFD_can_glerr_int	Global error interrupt	Not possible
CANFD_can_rf_dmareg_m	RX FIFO m DMA request	Possible
CANFD_can_tx_int_n	Channel n TX interrupt	Not possible
CANFD_can_cherr_int_n	Channel n CAN error interrupt	Not possible
CANFD_nan_comfrx_int_n	Channel n Common RX FIFO or TXQ interrupt	Not possible
CANFD_can_cf_dmareg_n	Channel n First common FIFO DMA request	Possible

Note: m = 0 to 7, n = 0 to 5

7.9.2 CANFD Registers

The base address for the CANFD is as follows.

Table 7.9-4 Register Base Address

Base Register Name	Base Address
<CFD_base>	0_1244_0000h (5244_0000h* ¹ , 4244_0000h* ²)

Note 1. CM33 Address Space Non-Secure

Note 2. CM33 Address Space Secure

7.9.2.1 List of Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]* ¹
Channel 0 Nominal Bit Rate Configuration Register	CFDC0NCFG	0000_0000h	0000h	8, 16, 32
Channel 0 Control Register	CFDC0CTR	0000_0005h	0004h	8, 16, 32
Channel 0 Status Register	CFDC0STS	0000_0005h	0008h	8, 16, 32
Channel 0 Error Flag Register	CFDC0ERFL	0000_0000h	000Ch	8, 16, 32
Channel 1 Nominal Bit Rate Configuration Register	CFDC1NCFG	0000_0000h	0010h	8, 16, 32
Channel 1 Control Register	CFDC1CTR	0000_0005h	0014h	8, 16, 32
Channel 1 Status Register	CFDC1STS	0000_0005h	0018h	8, 16, 32
Channel 1 Error Flag Register	CFDC1ERFL	0000_0000h	001Ch	8, 16, 32
Channel 2 Nominal Bit Rate Configuration Register	CFDC2NCFG	0000_0000h	0020h	8, 16, 32
Channel 2 Control Register	CFDC2CTR	0000_0005h	0024h	8, 16, 32
Channel 2 Status Register	CFDC2STS	0000_0005h	0028h	8, 16, 32
Channel 2 Error Flag Register	CFDC2ERFL	0000_0000h	002Ch	8, 16, 32
Channel 3 Nominal Bit Rate Configuration Register	CFDC3NCFG	0000_0000h	0030h	8, 16, 32
Channel 3 Control Register	CFDC3CTR	0000_0005h	0034h	8, 16, 32
Channel 3 Status Register	CFDC3STS	0000_0005h	0038h	8, 16, 32
Channel 3 Error Flag Register	CFDC3ERFL	0000_0000h	003Ch	8, 16, 32
Channel 4 Nominal Bit Rate Configuration Register	CFDC4NCFG	0000_0000h	0040h	8, 16, 32
Channel 4 Control Register	CFDC4CTR	0000_0005h	0044h	8, 16, 32
Channel 4 Status Register	CFDC4STS	0000_0005h	0048h	8, 16, 32
Channel 4 Error Flag Register	CFDC4ERFL	0000_0000h	004Ch	8, 16, 32
Channel 5 Nominal Bit Rate Configuration Register	CFDC5NCFG	0000_0000h	0050h	8, 16, 32
Channel 5 Control Register	CFDC5CTR	0000_0005h	0054h	8, 16, 32
Channel 5 Status Register	CFDC5STS	0000_0005h	0058h	8, 16, 32
Channel 5 Error Flag Register	CFDC5ERFL	0000_0000h	005Ch	8, 16, 32
Reserve	-	-	0060h to 007Fh	-
Global IP Version Register	CFDGIPV	1124_0043h	0080h	8, 16, 32
Global Configuration Register	CFDGCFG	0000_0000h	0084h	8, 16, 32
Global Control Register	CFDGCTR	0000_0005h	0088h	8, 16, 32
Global Status Register	CFDGSTS	0000_000Dh	008Ch	8, 16, 32
Global Error Flag Register	CFDGERFL	0000_0000h	0090h	8, 16, 32
Global Timestamp Counter Register	CFDGTSC	0000_0000h	0094h	16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Global Acceptance Filter List Entry Control Register	CFDGAFLECTR	0000_0000h	0098h	8, 16, 32
Global Acceptance Filter List Configuration Register 0	CFDGAFLCFG0	0000_0000h	009Ch	8, 16, 32
Global Acceptance Filter List Configuration Register 1	CFDGAFLCFG1	0000_0000h	00A0h	8, 16, 32
Global Acceptance Filter List Configuration Register 2	CFDGAFLCFG2	0000_0000h	00A4h	8, 16, 32
Reserve	-	-	00A8h to 00ABh	-
RX Message Buffer Number Register	CFDRMNB	0000_0000h	00ACh	8, 16, 32
RX Message Buffer New Data Register t	CFDRMNDt	0000_0000h	00B0h + t x 0004h	8, 16, 32
Reserve	-	-	00BCh to 00BFh	-
RX FIFO Configuration/Control Register 0	CFDRFCC0	0000_0000h	00C0h	8, 16, 32
RX FIFO Configuration/Control Register 1	CFDRFCC1	0000_0000h	00C4h	8, 16, 32
RX FIFO Configuration/Control Register 2	CFDRFCC2	0000_0000h	00C8h	8, 16, 32
RX FIFO Configuration/Control Register 3	CFDRFCC3	0000_0000h	00CCh	8, 16, 32
RX FIFO Configuration/Control Register 4	CFDRFCC4	0000_0000h	00D0h	8, 16, 32
RX FIFO Configuration/Control Register 5	CFDRFCC5	0000_0000h	00D4h	8, 16, 32
RX FIFO Configuration/Control Register 6	CFDRFCC6	0000_0000h	00D8h	8, 16, 32
RX FIFO Configuration/Control Register 7	CFDRFCC7	0000_0000h	00DCh	8, 16, 32
RX FIFO Status Register 0	CFDRFSTS0	0000_0001h	00E0h	8, 16, 32
RX FIFO Status Register 1	CFDRFSTS1	0000_0001h	00E4h	8, 16, 32
RX FIFO Status Register 2	CFDRFSTS2	0000_0001h	00E8h	8, 16, 32
RX FIFO Status Register 3	CFDRFSTS3	0000_0001h	00ECh	8, 16, 32
RX FIFO Status Register 4	CFDRFSTS4	0000_0001h	00F0h	8, 16, 32
RX FIFO Status Register 5	CFDRFSTS5	0000_0001h	00F4h	8, 16, 32
RX FIFO Status Register 6	CFDRFSTS6	0000_0001h	00F8h	8, 16, 32
RX FIFO Status Register 7	CFDRFSTS7	0000_0001h	00FCh	8, 16, 32
RX FIFO Pointer Control Register 0	CFDRFPCTR0	0000_0000h	0100h	8, 16, 32
RX FIFO Pointer Control Register 1	CFDRFPCTR1	0000_0000h	0104h	8, 16, 32
RX FIFO Pointer Control Register 2	CFDRFPCTR2	0000_0000h	0108h	8, 16, 32
RX FIFO Pointer Control Register 3	CFDRFPCTR3	0000_0000h	010Ch	8, 16, 32
RX FIFO Pointer Control Register 4	CFDRFPCTR4	0000_0000h	0110h	8, 16, 32
RX FIFO Pointer Control Register 5	CFDRFPCTR5	0000_0000h	0114h	8, 16, 32
RX FIFO Pointer Control Register 6	CFDRFPCTR6	0000_0000h	0118h	8, 16, 32
RX FIFO Pointer Control Register 7	CFDRFPCTR7	0000_0000h	011Ch	8, 16, 32
Common FIFO Configuration/Control Register n	CFDCFCCn	0000_0000h	0120h + n x 0004h	8, 16, 32
Reserve	-	-	016Ch to 017Fh	-
Common FIFO Configuration/Control Enhancement Register n	CFDCFCCEn	0000_0000h	0180h + n x 0004h	8, 16, 32
Reserve	-	-	01CCh to 01DFh	-
Common FIFO Status Register n	CFDCFSTSn	0000_0001h	01E0h + n x 0004h	8, 16, 32
Reserve	-	-	022Ch to 023Fh	-
Common FIFO Pointer Control Register n	CFDCFPCTRn	0000_0000h	0240h + n x 0004h	8, 16, 32
Reserve	-	-	028Ch to 029Fh	-
FIFO Empty Status Register	CFDFESTS	0000_3FFFh	02A0h	8, 16, 32
FIFO Full Status Register	CFDFFSTS	0000_0000h	02A4h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
FIFO Message Lost Status Register	CFDFMSTS	0000_0000h	02A8h	8, 16, 32
RX FIFO Interrupt Flag Status Register	CFDRFISTS	0000_0000h	02ACh	8, 16, 32
Common FIFO RX Interrupt Flag Status Register	CFDCFRISTS	0000_0000h	02B0h	8, 16, 32
Common FIFO TX Interrupt Flag Status Register	CFDCFTISTS	0000_0000h	02B4h	8, 16, 32
Common FIFO One Frame RX Interrupt Flag Status Register	CFDCFOFRISTS	0000_0000h	02B8h	8, 16, 32
Common FIFO One Frame TX Interrupt Flag Status Register	CFDCFOFTISTS	0000_0000h	02BCh	8, 16, 32
Common FIFO Message Overwrite Status Register	CFDCFMOWSTS	0000_0000h	02C0h	8, 16, 32
FIFO FDC Full Status Register	CFDFFFSTS	0000_0000h	02C4h	8, 16, 32
Reserve	-	-	02C8h to 02CFh	-
TX Message Buffer Control Register n	CFDTMCn	00h	02D0h + n x 0001h	8
Reserve	-	-	0450h to 07CFh	-
TX Message Buffer Status Register n	CFDTMSTSn	00h	07D0h + n x 0001h	8
Reserve	-	-	0950h to 0CCFh	-
TX Message Buffer Transmission Request Status Register f	CFDTMTRSTSf	0000_0000h	0CD0h + f x 0004h	8, 16, 32
Reserve	-	-	0D00h to 0D6Fh	-
TX Message Buffer Transmission Abort Request Status Register f	CFDTMTARSTSf	0000_0000h	0D70h + f x 0004h	8, 16, 32
Reserve	-	-	0DA0h to 0E0Fh	-
TX Message Buffer Transmission Completion Status Register f	CFDTMTCSTSf	0000_0000h	0E10h + f x 0004h	8, 16, 32
Reserve	-	-	0E40h to 0EAFh	-
TX Message Buffer Transmission Abort Status Register f	CFDTMTASTSf	0000_0000h	0EB0h + f x 0004h	8, 16, 32
Reserve	-	-	0D0h to 0F4Fh	-
TX Message Buffer Transmission Interrupt Enable Register f	CFDTMIECf	0000_0000h	0F50h + f x 0004h	8, 16, 32
Reserve	-	-	0F80h to 0FFFh	-
TX Queue Configuration/Control Register 00	CFDTXQCC00	0000_0000h	1000h	8, 16, 32
TX Queue Configuration/Control Register 01	CFDTXQCC01	0000_0000h	1004h	8, 16, 32
TX Queue Configuration/Control Register 02	CFDTXQCC02	0000_0000h	1008h	8, 16, 32
TX Queue Configuration/Control Register 03	CFDTXQCC03	0000_0000h	100Ch	8, 16, 32
TX Queue Configuration/Control Register 04	CFDTXQCC04	0000_0000h	1010h	8, 16, 32
TX Queue Configuration/Control Register 05	CFDTXQCC05	0000_0000h	1014h	8, 16, 32
Reserve	-	-	1018h to 101Fh	-
TX Queue Status Register 00	CFDTXQSTS00	0000_0001h	1020h	8, 16, 32
TX Queue Status Register 01	CFDTXQSTS01	0000_0001h	1024h	8, 16, 32
TX Queue Status Register 02	CFDTXQSTS02	0000_0001h	1028h	8, 16, 32
TX Queue Status Register 03	CFDTXQSTS03	0000_0001h	102Ch	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
TX Queue Status Register 04	CFDQXSTS04	0000_0001h	1030h	8, 16, 32
TX Queue Status Register 05	CFDQXSTS05	0000_0001h	1034h	8, 16, 32
Reserve	-	-	1038h to 104Fh	-
TX Queue Pointer Control Register 00	CFDQXPCTR00	0000_0000h	1040h	8, 16, 32
TX Queue Pointer Control Register 01	CFDQXPCTR01	0000_0000h	1044h	8, 16, 32
TX Queue Pointer Control Register 02	CFDQXPCTR02	0000_0000h	1048h	8, 16, 32
TX Queue Pointer Control Register 03	CFDQXPCTR03	0000_0000h	104Ch	8, 16, 32
TX Queue Pointer Control Register 04	CFDQXPCTR04	0000_0000h	1050h	8, 16, 32
TX Queue Pointer Control Register 05	CFDQXPCTR05	0000_0000h	1054h	8, 16, 32
Reserve	-	-	1058h to 105Fh	-
TX Queue Configuration/Control Register 10	CFDQXCC10	0000_0000h	1060h	8, 16, 32
TX Queue Configuration/Control Register 11	CFDQXCC11	0000_0000h	1064h	8, 16, 32
TX Queue Configuration/Control Register 12	CFDQXCC12	0000_0000h	1068h	8, 16, 32
TX Queue Configuration/Control Register 13	CFDQXCC13	0000_0000h	106Ch	8, 16, 32
TX Queue Configuration/Control Register 14	CFDQXCC14	0000_0000h	1070h	8, 16, 32
TX Queue Configuration/Control Register 15	CFDQXCC15	0000_0000h	1074h	8, 16, 32
Reserve	-	-	1078h to 107Fh	-
TX Queue Status Register 10	CFDQXSTS10	0000_0001h	1080h	8, 16, 32
TX Queue Status Register 11	CFDQXSTS11	0000_0001h	1084h	8, 16, 32
TX Queue Status Register 12	CFDQXSTS12	0000_0001h	1088h	8, 16, 32
TX Queue Status Register 13	CFDQXSTS13	0000_0001h	108Ch	8, 16, 32
TX Queue Status Register 14	CFDQXSTS14	0000_0001h	1090h	8, 16, 32
TX Queue Status Register 15	CFDQXSTS15	0000_0001h	1094h	8, 16, 32
Reserve	-	-	1098h to 109Fh	-
TX Queue Pointer Control Register 10	CFDQXPCTR10	0000_0000h	10A0h	8, 16, 32
TX Queue Pointer Control Register 11	CFDQXPCTR11	0000_0000h	10A4h	8, 16, 32
TX Queue Pointer Control Register 12	CFDQXPCTR12	0000_0000h	10A8h	8, 16, 32
TX Queue Pointer Control Register 13	CFDQXPCTR13	0000_0000h	10ACh	8, 16, 32
TX Queue Pointer Control Register 14	CFDQXPCTR14	0000_0000h	10B0h	8, 16, 32
TX Queue Pointer Control Register 15	CFDQXPCTR15	0000_0000h	10B4h	8, 16, 32
Reserve	-	-	10B8h to 10BFh	-
TX Queue Configuration/Control Register 20	CFDQXCC20	0000_0000h	10C0h	8, 16, 32
TX Queue Configuration/Control Register 21	CFDQXCC21	0000_0000h	10C4h	8, 16, 32
TX Queue Configuration/Control Register 22	CFDQXCC22	0000_0000h	10C8h	8, 16, 32
TX Queue Configuration/Control Register 23	CFDQXCC23	0000_0000h	10CCh	8, 16, 32
TX Queue Configuration/Control Register 24	CFDQXCC24	0000_0000h	10D0h	8, 16, 32
TX Queue Configuration/Control Register 25	CFDQXCC25	0000_0000h	10D4h	8, 16, 32
Reserve	-	-	10D8h to 10DFh	-
TX Queue Status Register 20	CFDQXSTS20	0000_0001h	10E0h	8, 16, 32
TX Queue Status Register 21	CFDQXSTS21	0000_0001h	10E4h	8, 16, 32
TX Queue Status Register 22	CFDQXSTS22	0000_0001h	10E8h	8, 16, 32
TX Queue Status Register 23	CFDQXSTS23	0000_0001h	10ECh	8, 16, 32
TX Queue Status Register 24	CFDQXSTS24	0000_0001h	10F0h	8, 16, 32
TX Queue Status Register 25	CFDQXSTS25	0000_0001h	10F4h	8, 16, 32
Reserve	-	-	10F8h to 10FFh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
TX Queue Pointer Control Register 20	CFDTPQPTR20	0000_0000h	1100h	8, 16, 32
TX Queue Pointer Control Register 21	CFDTPQPTR21	0000_0000h	1104h	8, 16, 32
TX Queue Pointer Control Register 22	CFDTPQPTR22	0000_0000h	1108h	8, 16, 32
TX Queue Pointer Control Register 23	CFDTPQPTR23	0000_0000h	110Ch	8, 16, 32
TX Queue Pointer Control Register 24	CFDTPQPTR24	0000_0000h	1110h	8, 16, 32
TX Queue Pointer Control Register 25	CFDTPQPTR25	0000_0000h	1114h	8, 16, 32
Reserve	-	-	1118h to 111Fh	-
TX Queue Configuration/Control Register 30	CFDTPQCC30	0000_0000h	1120h	8, 16, 32
TX Queue Configuration/Control Register 31	CFDTPQCC31	0000_0000h	1124h	8, 16, 32
TX Queue Configuration/Control Register 32	CFDTPQCC32	0000_0000h	1128h	8, 16, 32
TX Queue Configuration/Control Register 33	CFDTPQCC33	0000_0000h	112Ch	8, 16, 32
TX Queue Configuration/Control Register 34	CFDTPQCC34	0000_0000h	1130h	8, 16, 32
TX Queue Configuration/Control Register 35	CFDTPQCC35	0000_0000h	1134h	8, 16, 32
Reserve	-	-	1138h to 113Fh	-
TX Queue Status Register 30	CFDTPQSTS30	0000_0001h	1140h	8, 16, 32
TX Queue Status Register 31	CFDTPQSTS31	0000_0001h	1144h	8, 16, 32
TX Queue Status Register 32	CFDTPQSTS32	0000_0001h	1148h	8, 16, 32
TX Queue Status Register 33	CFDTPQSTS33	0000_0001h	114Ch	8, 16, 32
TX Queue Status Register 34	CFDTPQSTS34	0000_0001h	1150h	8, 16, 32
TX Queue Status Register 35	CFDTPQSTS35	0000_0001h	1154h	8, 16, 32
Reserve	-	-	1158h to 115Fh	-
TX Queue Pointer Control Register 30	CFDTPQPTR30	0000_0000h	1160h	8, 16, 32
TX Queue Pointer Control Register 31	CFDTPQPTR31	0000_0000h	1164h	8, 16, 32
TX Queue Pointer Control Register 32	CFDTPQPTR32	0000_0000h	1168h	8, 16, 32
TX Queue Pointer Control Register 33	CFDTPQPTR33	0000_0000h	116Ch	8, 16, 32
TX Queue Pointer Control Register 34	CFDTPQPTR34	0000_0000h	1170h	8, 16, 32
TX Queue Pointer Control Register 35	CFDTPQPTR35	0000_0000h	1174h	8, 16, 32
Reserve	-	-	1178h to 117Fh	-
TX Queue Empty Status Register	CFDTPQUESTS	0000_00FFh	1180h	8, 16, 32
TX Queue Full Interrupt Status Register	CFDTPQFISTS	0000_0000h	1184h	8, 16, 32
TX Queue Message Lost Status Register	CFDTPQMSTS	0000_0000h	1188h	8, 16, 32
Reserve	-	-	118Ch to 118Fh	-
TX Queue Interrupt Status Register	CFDTPQISTS	0000_0000h	1190h	8, 16, 32
TX Queue One Frame TX Interrupt Status Register	CFDTPQOFTISTS	0000_0000h	1194h	8, 16, 32
TX Queue One Frame RX Interrupt Status Register	CFDTPQOFRISTS	0000_0000h	1198h	8, 16, 32
TX Queue Full Status Register	CFDTPQFSTS	0000_0000h	119Ch	8, 16, 32
Reserve	-	-	11A0h to 11FFh	-
TX History List Configuration/Control Register 0	CFDTHLCC0	0000_0000h	1200h	8, 16, 32
TX History List Configuration/Control Register 1	CFDTHLCC1	0000_0000h	1204h	8, 16, 32
TX History List Configuration/Control Register 2	CFDTHLCC2	0000_0000h	1208h	8, 16, 32
TX History List Configuration/Control Register 3	CFDTHLCC3	0000_0000h	120Ch	8, 16, 32
TX History List Configuration/Control Register 4	CFDTHLCC4	0000_0000h	1210h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
TX History List Configuration/Control Register 5	CFDTHLCC5	0000_0000h	1214h	8, 16, 32
Reserve	-	-	1218h to 121Fh	-
TX History List Status Register 0	CFDTHLSTS0	0000_0001h	1220h	8, 16, 32
TX History List Status Register 1	CFDTHLSTS1	0000_0001h	1224h	8, 16, 32
TX History List Status Register 2	CFDTHLSTS2	0000_0001h	1228h	8, 16, 32
TX History List Status Register 3	CFDTHLSTS3	0000_0001h	122Ch	8, 16, 32
TX History List Status Register 4	CFDTHLSTS4	0000_0001h	1230h	8, 16, 32
TX History List Status Register 5	CFDTHLSTS5	0000_0001h	1234h	8, 16, 32
Reserve	-	-	1238h to 123Fh	-
TX History List Pointer Control Register 0	CFDTHLPCTR0	0000_0000h	1240h	8, 16, 32
TX History List Pointer Control Register 1	CFDTHLPCTR1	0000_0000h	1244h	8, 16, 32
TX History List Pointer Control Register 2	CFDTHLPCTR2	0000_0000h	1248h	8, 16, 32
TX History List Pointer Control Register 3	CFDTHLPCTR3	0000_0000h	124Ch	8, 16, 32
TX History List Pointer Control Register 4	CFDTHLPCTR4	0000_0000h	1250h	8, 16, 32
TX History List Pointer Control Register 5	CFDTHLPCTR5	0000_0000h	1254h	8, 16, 32
Reserve	-	-	1258h to 12FFh	-
Global TX Interrupt Status Register 0	CFDGTINTSTS0	0000_0000h	1300h	8, 16, 32
Global TX Interrupt Status Register 1	CFDGTINTSTS1	0000_0000h	1304h	8, 16, 32
Global Test Configuration Register	CFDGTSTCFG	0000_0000h	1308h	8, 16, 32
Global Test Control Register	CFDGTSTCTR	0000_0000h	130Ch	8, 16, 32
Reserve	-	-	1310h to 1313h	-
Global FD Configuration Register	CFDGFDCFG	0000_0000h	1314h	8, 16, 32
Reserve	-	-	1318h to 131Bh	-
Global Lock Key Register	CFDGLOCKK	0000_0000h	131Ch	16, 32
Reserve	-	-	1320h to 132Fh	-
DMA Transfer Control Register	CFDCDTCT	0000_0000h	1330h	8, 16, 32
DMA Transfer Status Register	CFDCDTSTS	0000_0000h	1334h	8, 16, 32
Reserve	-	-	1338h to 133Fh	-
DMA TX Transfer Control Register	CFDCDTTCT	0000_0000h	1340h	8, 16, 32
DMA TX Transfer Status Register	CFDCDTTSTS	0000_0000h	1344h	8, 16, 32
Reserve	-	-	1348h to 134Fh	-
Global RX Interrupt Status Register 0	CFDGRINTSTS0	0000_0000h	1350h	8, 16, 32
Global RX Interrupt Status Register 1	CFDGRINTSTS1	0000_0000h	1354h	8, 16, 32
Global RX Interrupt Status Register 2	CFDGRINTSTS2	0000_0000h	1358h	8, 16, 32
Global RX Interrupt Status Register 3	CFDGRINTSTS3	0000_0000h	135Ch	8, 16, 32
Global RX Interrupt Status Register 4	CFDGRINTSTS4	0000_0000h	1360h	8, 16, 32
Global RX Interrupt Status Register 5	CFDGRINTSTS5	0000_0000h	1364h	8, 16, 32
Reserve	-	-	1368h to 137Fh	-
Global Reset Control Register	CFDGRSTC	0000_0000h	1380h	16, 32
Global Flexible CAN Mode Configuration Register	CFDGFCMC	0000_0000h	1384h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Reserve	-	-	1388h to 138Bh	-
Global Flexible Transmission Buffer Assignment Configuration Register	CFDGFTBAC	0000_0000h	138Ch	8, 16, 32
Reserve	-	-	1390h to 13FFh	-
Channel 0 Data Bit Rate Configuration Register	CFDC0DCFG	0000_0000h	1400h	8, 16, 32
Channel 0 CAN-FD Configuration Register	CFDC0FDCFG	0000_0000h	1404h	8, 16, 32
Channel 0 CAN-FD Control Register	CFDC0FDCTR	0000_0000h	1408h	8, 16, 32
Channel 0 CAN-FD Status Register	CFDC0FDSTS	0000_0000h	140Ch	8, 16, 32
Channel 0 CAN-FD CRC Register	CFDC0FDCRC	0000_0000h	1410h	8, 16, 32
Reserve	-	-	1414h to 1417h	-
Channel 0 Bus Load Control Register	CFDC0BLCT	0000_0000h	1418h	8, 16, 32
Channel 0 Bus Load Status Register	CFDC0BLSTS	0000_0000h	141Ch	8, 16, 32
Channel 1 Data Bit Rate Configuration Register	CFDC1DCFG	0000_0000h	1420h	8, 16, 32
Channel 1 CAN-FD Configuration Register	CFDC1FDCFG	0000_0000h	1424h	8, 16, 32
Channel 1 CAN-FD Control Register	CFDC1FDCTR	0000_0000h	1428h	8, 16, 32
Channel 1 CAN-FD Status Register	CFDC1FDSTS	0000_0000h	142Ch	8, 16, 32
Channel 1 CAN-FD CRC Register	CFDC1FDCRC	0000_0000h	1430h	8, 16, 32
Reserve	-	-	1434h to 1437h	-
Channel 1 Bus Load Control Register	CFDC1BLCT	0000_0000h	1438h	8, 16, 32
Channel 1 Bus Load Status Register	CFDC1BLSTS	0000_0000h	143Ch	8, 16, 32
Channel 2 Data Bit Rate Configuration Register	CFDC2DCFG	0000_0000h	1440h	8, 16, 32
Channel 2 CAN-FD Configuration Register	CFDC2FDCFG	0000_0000h	1444h	8, 16, 32
Channel 2 CAN-FD Control Register	CFDC2FDCTR	0000_0000h	1448h	8, 16, 32
Channel 2 CAN-FD Status Register	CFDC2FDSTS	0000_0000h	144Ch	8, 16, 32
Channel 2 CAN-FD CRC Register	CFDC2FDCRC	0000_0000h	1450h	8, 16, 32
Reserve	-	-	1454h to 1457h	-
Channel 2 Bus Load Control Register	CFDC2BLCT	0000_0000h	1458h	8, 16, 32
Channel 2 Bus Load Status Register	CFDC2BLSTS	0000_0000h	145Ch	8, 16, 32
Channel 3 Data Bit Rate Configuration Register	CFDC3DCFG	0000_0000h	1460h	8, 16, 32
Channel 3 CAN-FD Configuration Register	CFDC3FDCFG	0000_0000h	1464h	8, 16, 32
Channel 3 CAN-FD Control Register	CFDC3FDCTR	0000_0000h	1468h	8, 16, 32
Channel 3 CAN-FD Status Register	CFDC3FDSTS	0000_0000h	146Ch	8, 16, 32
Channel 3 CAN-FD CRC Register	CFDC3FDCRC	0000_0000h	1470h	8, 16, 32
Reserve	-	-	1474h to 1477h	-
Channel 3 Bus Load Control Register	CFDC3BLCT	0000_0000h	1478h	8, 16, 32
Channel 3 Bus Load Status Register	CFDC3BLSTS	0000_0000h	147Ch	8, 16, 32
Channel 4 Data Bit Rate Configuration Register	CFDC4DCFG	0000_0000h	1480h	8, 16, 32
Channel 4 CAN-FD Configuration Register	CFDC4FDCFG	0000_0000h	1484h	8, 16, 32
Channel 4 CAN-FD Control Register	CFDC4FDCTR	0000_0000h	1488h	8, 16, 32
Channel 4 CAN-FD Status Register	CFDC4FDSTS	0000_0000h	148Ch	8, 16, 32
Channel 4 CAN-FD CRC Register	CFDC4FDCRC	0000_0000h	1490h	8, 16, 32
Reserve	-	-	1494h to 1497h	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Channel 4 Bus Load Control Register	CFDC4BLCT	0000_0000h	1498h	8, 16, 32
Channel 4 Bus Load Status Register	CFDC4BLSTS	0000_0000h	149Ch	8, 16, 32
Channel 5 Data Bit Rate Configuration Register	CFDC5DCFG	0000_0000h	14A0h	8, 16, 32
Channel 5 CAN-FD Configuration Register	CFDC5FDCFG	0000_0000h	14A4h	8, 16, 32
Channel 5 CAN-FD Control Register	CFDC5FDCTR	0000_0000h	14A8h	8, 16, 32
Channel 5 CAN-FD Status Register	CFDC5FDSTS	0000_0000h	14ACh	8, 16, 32
Channel 5 CAN-FD CRC Register	CFDC5FDCRC	0000_0000h	14B0h	8, 16, 32
Reserve	-	-	14B4h to 14B7h	-
Channel 5 Bus Load Control Register	CFDC5BLCT	0000_0000h	14B8h	8, 16, 32
Channel 5 Bus Load Status Register	CFDC5BLSTS	0000_0000h	14BCh	8, 16, 32
Reserve	-	-	14C0h to 17FFh	-
Global Acceptance Filter List ID Register n	CFDGAFLIDn	0000_0000h	1800h + (n-1) x 0010h	8, 16, 32
Global Acceptance Filter List Mask Register n	CFDGAFLMn	0000_0000h	1804h + (n-1) x 0010h	8, 16, 32
Global Acceptance Filter List Pointer 0 Register n	CFDGAFLP0n	0000_0000h	1808h + (n-1) x 0010h	8, 16, 32
Global Acceptance Filter List Pointer 1 Register n	CFDGAFLP1n	0000_0000h	180Ch + (n-1) x 0010h	8, 16, 32
Reserve	-	-	1900h to 1FFFh	-
RX Message Buffer ID Register n	CFDRMIDn	0000_0000h	2000h + n x 0080h	8, 16, 32
RX Message Buffer Pointer Register	CFDRMPTRn	0000_0000h	2004h + n x 0080h	8, 16, 32
RX Message Buffer CAN-FD Status Register	CFDRMFDSTSn	0000_0000h	2008h + n x 0080h	8, 16, 32
RX Message Buffer Data Field p Register n	CFDRMDFP_n	0000_0000h	200Ch + p x 0004h + n x 0080h	8, 16, 32
RX FIFO Access ID Register n	CFDRFIDn	0000_0000h	6000h + n x 0080h	8, 16, 32
RX FIFO Access Pointer Register n	CFDRFPTRn	0000_0000h	6004h + n x 0080h	8, 16, 32
RX FIFO Access CAN-FD Status Register n	CFDRFFDSTSn	0000_0000h	6008h + n x 0080h	8, 16, 32
RX FIFO Access Data Field p Register n	CFDRFDFpn	0000_0000h	600Ch + p x 0004h + n x 0080h	8, 16, 32
Common FIFO Access ID Register n	CFDCFIDn	0000_0000h	6400h + n x 0080h	8, 16, 32
Common FIFO Access Pointer Register n	CFDCFPTRn	0000_0000h	6404h + n x 0080h	8, 16, 32
Common FIFO Access CAN-FD Control/Status Register n	CFDCFFDCSTSn	0000_0000h	6408h + n x 0080h	8, 16, 32
Common FIFO Access Data Field p Register n	CFDCFDFpn	0000_0000h	640Ch + p x 0004h + n x 0080h	8, 16, 32
Channel 0 TX History List Access Register 0	CFDTHLACC00	0000_0000h	8000h	8, 16, 32
Channel 0 TX History List Access Register 1	CFDTHLACC10	0000_0000h	8004h	8, 16, 32
Channel 1 TX History List Access Register 0	CFDTHLACC01	0000_0000h	8008h	8, 16, 32
Channel 1 TX History List Access Register 1	CFDTHLACC11	0000_0000h	800Ch	8, 16, 32
Channel 2 TX History List Access Register 0	CFDTHLACC02	0000_0000h	8010h	8, 16, 32
Channel 2 TX History List Access Register 1	CFDTHLACC12	0000_0000h	8014h	8, 16, 32
Channel 3 TX History List Access Register 0	CFDTHLACC03	0000_0000h	8018h	8, 16, 32
Channel 3 TX History List Access Register 1	CFDTHLACC13	0000_0000h	801Ch	8, 16, 32
Channel 4 TX History List Access Register 0	CFDTHLACC04	0000_0000h	8020h	8, 16, 32
Channel 4 TX History List Access Register 1	CFDTHLACC14	0000_0000h	8024h	8, 16, 32
Channel 5 TX History List Access Register 0	CFDTHLACC05	0000_0000h	8028h	8, 16, 32
Channel 5 TX History List Access Register 1	CFDTHLACC15	0000_0000h	802Ch	8, 16, 32
Reserve	-	-	8030h to 83FFh	-
RAM Test Page Access Register n	CFDRPGACn	0000_0000h	8400h + n x 0004h	8, 16, 32
Reserve	-	-	8500h to FFFFh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]* ¹
TX Message Buffer ID Register n	CFDTMIDn	0000_0000h	1_0000h + n x 0080h	8, 16, 32
TX Message Buffer Pointer Register n	CFDTMPTRn	0000_0000h	1_0004h + n x 0080h	8, 16, 32
TX Message Buffer CAN-FD Control Register n	CFDTMFDCTRn	0000_0000h	1_0008h + n x 0080h	8, 16, 32
TX Message Buffer Data Field p Register n	CFDTMDFp_n	0000_0000h	1_000Ch + p x 0004h + n x 0080h	8, 16, 32

Note 1. The read access size is fixed at 32 bits.

7.9.2.2 CANFD Register Description

7.9.2.2.1 Channel n Nominal Bit Rate Configuration Register (CFDCnNCFG) (n = 0 to 5)

The CFDCnNCFG register is used to configure the transmission/reception nominal bit rate parameters of the channels. Do not write this register in CH_OPERATION or CH_SLEEP mode. Set this register in CH_RESET or CH_HALT mode.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0000h + n x 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NTSEG2[6:0]							NTSEG1[7:0]								NSJW
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NSJW[5:0]							NBRP[9:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 25	NTSEG2[6:0]		0h	RW	Nominal Bit Rate Time Segment 2 Control 00h: Setting prohibited 01h: 2 Tq ⋮ 7Eh: 127 Tq 7Fh: 128 Tq											
24 to 17	NTSEG1[7:0]		0h	RW	Nominal Bit Rate Time Segment 1 Control 00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq ⋮ FEh: 255 Tq FFh: 256 Tq											
16 to 10	NSJW[6:0]		0h	RW	Nominal Bit Rate Resynchronization Jump Width Control 00h: Setting prohibited 01h: 2 Tq ⋮ 7Eh: 127 Tq 7Fh: 128 Tq											
9 to 0	NBRP[9:0]		0h	RW	Nominal Bit Rate Prescaler Nominal bit rate prescaler division ratio setting											

NBRP[9:0] bits (Nominal Bit Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

NSJW[6:0] bits (Nominal Bit Rate Resynchronization Jump Width Control)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

NTSEG1[7:0] bits (Nominal Bit Rate Time Segment 1 Control)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment. Configure a Tq value only between 2 and 256, inclusive. See **7.9.4.1.2 CAN Bit Timing** for more details.

NTSEG2[6:0] bits (Nominal Bit Rate Time Segment 2 Control)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. Configure a Tq value only between 2 and 128, inclusive.

7.9.2.2.2 Channel n Control Register (CFDCnCTR) (n = 0 to 5)

The CFDCnCTR register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 0004h + n x 0010h

Initial Value : 0000_0005h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ROM	CRCT	-	-	-	CTMS[1:0]	CTME	ERRD	BOM[1:0]	-	TDCVFI E	SOCOI E	EOCOI E	TAIE		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPiE	EWiE	BEiE	-	-	-	-	RTBO	CSLPR	CHMDC[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	ROM ^{*1}	0h	RW	Restricted Operation Mode Enable 0b: Restricted operation mode disabled 1b: Restricted operation mode enabled
30	CRCT	0h	RW	CRC Error Test Enable 0b: The first bit of reception ID field is not inverted 1b: The first bit of reception ID field is inverted
29 to 27	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
26, 25	CTMS[1:0]	0h	RW	Communication Test Mode Select 00b: Standard test mode 01b: Listen-only mode 10b: Self-test mode 0 (External loopback mode) 11b: Self-test mode 1 (Internal loopback mode)
24	CTME	0h	RW	Communication Test Mode Enable 0b: Communication test mode is disabled 1b: Communication test mode is enabled
23	ERRD	0h	RW	Error Display Mode Select 0b: Error flags are displayed only for the first error information after bits [14:8] in the CFDCnERFL register are all cleared 1b: Error flag for all error information are displayed
22, 21	BOM[1:0]	0h	RW	Bus-Off Recovery Mode Select 00b: ISO11898-1 compliant 01b: Entry to Channel Halt mode automatically at bus-off entry 10b: Entry to Channel Halt mode automatically at bus-off end 11b: Entry to Channel Halt mode (in bus-off state) by program request
20	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
19	TDCVFI ^{*1}	0h	RW	Transceiver Delay Compensation Violation Interrupt Enable 0b: Transceiver delay compensation violation interrupt disabled 1b: Transceiver delay compensation violation interrupt enabled
18	SOCOI E	0h	RW	Successful Occurrence Counter Overflow Interrupt Enable 0b: Successful occurrence counter overflow interrupt disabled 1b: Successful occurrence counter overflow interrupt enabled
17	EOCOI E	0h	RW	Error Occurrence Counter Overflow Interrupt Enable 0b: Error occurrence counter overflow interrupt disabled 1b: Error occurrence counter overflow interrupt enabled
16	TAIE	0h	RW	Transmission Abort Interrupt Enable 0b: Transmission abort interrupt disabled 1b: Transmission abort interrupt enabled
15	ALIE	0h	RW	Arbitration Lost Interrupt Enable 0b: Arbitration lost interrupt disabled 1b: Arbitration lost interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
14	BLIE	0h	RW	Bus Lock Interrupt Enable 0b: Bus lock interrupt disabled 1b: Bus lock interrupt enabled
13	OLIE	0h	RW	Overload Interrupt Enable 0b: Overload interrupt disabled 1b: Overload interrupt enabled
12	BORIE	0h	RW	Bus-Off Recovery Interrupt Enable 0b: Bus-off recovery interrupt disabled 1b: Bus-off recovery interrupt enabled
11	BOEE	0h	RW	Bus-Off Entry Interrupt Enable 0b: Bus-off entry interrupt disabled 1b: Bus-off entry interrupt enabled
10	EPIE	0h	RW	Error Passive Interrupt Enable 0b: Error passive interrupt disabled 1b: Error passive interrupt enabled
9	EWIE	0h	RW	Error Warning Interrupt Enable 0b: Error warning interrupt disabled 1b: Error warning interrupt enabled
8	BIE	0h	RW	Bus Error Interrupt Enable 0b: Bus error interrupt disabled 1b: Bus error interrupt enabled
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	RTBO	0h	RW	Forcible Return from Bus-Off When this bit is set to 1b, forcible return from the bus-off state is made. This bit is always read as 0b.
2	CSLPR	1h	RW	Channel Stop Mode 0b: Other than Channel Stop mode 1b: Channel Stop mode
1, 0	CHMDC[1:0]	1h	RW	Mode Select 00b: Channel Communication mode 01b: Channel Reset mode 10b: Channel Halt mode 11b: Setting prohibited

Note 1. These bits are not available in the classical CAN function.

CHMDC[1:0] bits (Mode Select)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in **7.9.3.3 Channel Modes**.

Setting CHMDC[1:0] bits to 11b has no effect. When the CAN-FD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDCnCTR.BOM settings.

If CPU write access to CFDCnCTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDCnCTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDCnCTR.CHMDC[1:0] value is 00b (Operation mode).

CSLPR bit (Channel Stop Mode)

When the CSLPR bit is 1b, a Sleep mode request is generated for the corresponding CAN channel.

When this bit is 0b, a request to exit Sleep mode is generated for the related CAN-FD channel.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Forcible Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1b.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1b, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDCnSTS.BOSTS) is set to 0b.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDCnCTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0b.

Return from the Bus-Off command should be used only when CFDCnCTR.BOM[1:0] is set to 00b.

Only write to this bit when the related CAN-FD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDCnERFL.BEF bits are both 1b, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDCnERFL.EWF bits are both 1b, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDCnERFL.EPF are both 1b.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDCnERFL.BOEF bits are both 1b, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDCnERFL.BORF bits are both 1b, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDCnERFL.OVLF bits are both 1b, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDCnERFL.BLF bits are both 1b, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDCnERFL.ALF bits are both 1b, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1b and a transmission is successfully aborted from the transmit buffer, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1b and the CFDCnFDSTS.EOCO bit belonging to the corresponding CAN channel is 1b, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1b and the CFDCnFDSTS.SOCO bit belonging to the corresponding CAN channel is 1b, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1b and the CFDCnFDSTS.TDCVF bit belonging to the corresponding CAN channel is 1b, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Bus-Off Recovery Mode Select)

The BOM[1:0] bits control the timing of the recovery from Bus-Off mode of the CAN-FD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

ERRD bit (Error Display Mode Select)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDCnERFL).

If the ERRD bit is 0b and more than one errors occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until the error flag bits [14:8] are cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CTME bit (Communication Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

CTMS[1:0] bits (Communication Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CAN-FD channel is in CH_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

CRCT bit (CRC Error Test Enable)

The CRCT bit checks the internal CRC generator logic of the protocol controller.

This bit inverts the first bit (ID bit) of the CAN message data stream that is being received, so that the internal generated CRC result does not match the received CRC value of the frame. Refer to the bit stuffing rule when using this feature for the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDCnERFL.CRCREG (classical CAN frames)
- CFDCnFDCRC.CRCREG (CAN-FD frames)

Some limitations exist when using this bit:

- It is not possible to use this feature with CAN nodes connected to the device externally, only with nodes connected to the internal CAN bus communication can be used
- One CAN node can send a reference message and the receiver node can invert one bit of the incoming bit stream

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing this limitation.

The CRC Error Test mode is enabled if the CRCT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1b.

Do not write to the CRCT bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode Enable)

When the ROM and CTME bits are both 1b, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDCnCTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

7.9.2.2.3 Channel n Status Register (CFDCnSTS) (n = 0 to 5)

The CFDCnSTS register shows the mode, error, and transmission/reception status of the related channel together with its reception and transmission error count values.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0008h + n x 0010h														
Initial Value :		0000_0005h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPTS	CHLPTS	CRSTS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	TEC[7:0]	0h	R	Transmission Error Count The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	0h	R	Reception Error Count The receive error counter (REC) can be read.
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	ESIF ^{*1}	0h	RW	Error State Indication Flag 0b: No CANFD message whose ESI bit is recessive has been received 1b: At least one CANFD message whose ESI bit is recessive has been received
7	COMSTS	0h	R	Communication Status Flag 0b: Communication is not ready 1b: Communication is ready
6	RECSTS	0h	R	Receive Status Flag 0b: Bus idle, in transmission or bus off state 1b: In reception
5	TRMSTS	0h	R	Transmit Status Flag 0b: Bus idle or in reception 1b: In transmission or bus off state
4	BOSTS	0h	R	Bus-Off Status Flag 0b: Not in bus-off state 1b: In bus-off state
3	EPSTS	0h	R	Error Passive Status Flag 0b: Not in error passive state 1b: In error passive state
2	CSLPSTS	1h	R	Channel Stop Status Flag 0b: Not in Channel Sleep mode 1b: In Channel Sleep mode
1	CHLTSTS	0h	R	Channel Halt Status Flag 0b: Not in Channel Halt mode 1b: In Channel Halt mode
0	CRSTSTS	1h	R	Channel Reset Status Flag 0b: Not in Channel Reset mode 1b: In Channel Reset mode

Note 1. This bit is not available in the classical CAN function.

CRSTSTS bit (Channel Reset Status Flag)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1b.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status Flag)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Stop Status Flag)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CAN-FD channel enters Sleep mode, and is cleared automatically when the related CAN-FD channel exits Sleep mode.

EPSTS bit (Error Passive Status Flag)

The EPSTS bit indicates whether the related CAN-FD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 7Fh.

This bit is cleared automatically when the related CAN-FD channel exits the error passive state or enters Reset mode.

BOSTS bit (Bus-Off Status Flag)

The BOSTS bit indicates whether the related CAN-FD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds FFh and the related CAN-FD channel is in the bus-off state (CAN Transmission Error Count Register > FFh).

This bit is cleared automatically when the related CAN-FD channel exits bus-off state.

TRMSTS bit (Transmit Status Flag)

The TRMSTS bit indicates whether the related CAN-FD channel is transmitting a message.

This bit is set automatically when the related CAN-FD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Receive Status Flag)

The RECSTS bit indicates whether the related CAN-FD channel is receiving a message.

This bit is set automatically when the related CAN-FD channel is operating as a receiver node.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Communication Status Flag)

The COMSTS bit indicates whether the related CAN-FD channel is ready for communication.

This bit is set automatically when the related CAN-FD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

Note: This bit is 1b during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessive for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CAN-FD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0b to it. This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CAN-FD module enters GL_RESET or the CAN-FD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CAN-FD channel is in CH_HALT mode.

Do not read data from this register while CFDCnCTR.TRWE is set.

These bits are cleared automatically when CAN-FD module is in GL_RESET or CAN-FD channel is in CH_RESET mode.

7.9.2.2.4 Channel n Error Flag Register (CFDCnERFL) (n = 0 to 5)

The CFDCnERFL register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for a description of error occurrence conditions.

Only write to this register when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 000Ch + n x 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	CRCREG[14:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWf	BEF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
30 to 16	CRCREG[14:0]	0h	R	CRC Calculation Data (CRC length: 15 bits) These bits show the CRC value calculated based on the transmit message or receive message.
15	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
14	ADERR	0h	RW ^{*1}	Acknowledge Delimiter Error Flag 0b: Channel acknowledge delimiter error not detected 1b: Channel acknowledge delimiter error detected
13	B0ERR	0h	RW ^{*1}	Dominant Bit Error Flag 0b: Channel dominant bit error not detected 1b: Channel dominant bit error detected
12	B1ERR	0h	RW ^{*1}	Recessive Bit Error Flag 0b: Channel recessive bit error not detected 1b: Channel recessive bit error detected
11	CERR	0h	RW ^{*1}	CRC Error Flag 0b: Channel CRC error not detected 1b: Channel CRC error detected
10	AERR	0h	RW ^{*1}	Acknowledge Error Flag 0b: Channel acknowledge error not detected 1b: Channel acknowledge error detected
9	FERR	0h	RW ^{*1}	Form Error Flag 0b: Channel form error not detected 1b: Channel form error detected
8	SERR	0h	RW ^{*1}	Stuff Error Flag 0b: Channel stuff error not detected 1b: Channel stuff error detected
7	ALF	0h	RW ^{*1}	Arbitration Lost Flag 0b: Channel arbitration-lost not detected 1b: Channel arbitration-lost detected
6	BLF	0h	RW ^{*1}	Bus Lock Flag 0b: Channel bus lock not detected 1b: Channel bus lock detected
5	OVLF	0h	RW ^{*1}	Overload Flag 0b: Channel overload not detected 1b: Channel overload detected

Bit	Bit Name	Initial Value	R/W	Description
4	BORF	0h	RW ^{*1}	Bus-Off Recovery Flag 0b: Channel bus-off recovery not detected 1b: Channel bus-off recovery detected
3	BOEF	0h	RW ^{*1}	Bus-Off Entry Flag 0b: Channel bus-off entry not detected 1b: Channel bus-off entry detected
2	EPF	0h	RW ^{*1}	Error Passive Flag 0b: Channel error passive not detected 1b: Channel error passive detected
1	EWF	0h	RW ^{*1}	Error Warning Flag 0b: Channel error warning not detected 1b: Channel error warning detected
0	BEF	0h	RW ^{*1}	Bus Error Flag 0b: Channel bus error not detected 1b: Channel bus error detected

Note 1. To clear each flag of this register, software must write 0b to the corresponding bit. These flags cannot be set to 1 by software.

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when either TEC or REC exceeds 5Fh.

The setting of this bit only occurs when the TEC or REC initially exceeds 5Fh. Therefore, if the TEC or REC remains > 5Fh and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 60h and either TEC or REC crosses over again from a value 5Fh to a value > 5Fh.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 7Fh. Therefore, if the TEC or REC remains > 7Fh and the bit is cleared by software, it is not set again until both the TEC and REC go below 80h and either TEC or REC crosses over again from a value ≤ 7Fh to a value > 7Fh.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDCnCTR.BOM[1:0] is 00b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 10b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 11b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDCnCTR.RTBO is set to 1b (the CAN channel returns to error active)
- When CFDCnCTR.BOM[1:0] is 01b
- When CFDCnCTR.BOM[1:0] is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

OVLf bit (Overload Flag)

The OVLf flag indicates a detection of a CAN channel overload state.

The OVLf bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration-lost condition.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

The bit is set automatically when an arbitration-lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SERR bit (Stuff Error Flag)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

FERR bit (Form Error Flag)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

AERR bit (Acknowledge Error Flag)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.

2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

CERR bit (CRC Error Flag)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

B1ERR bit (Recessive Bit Error Flag)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0001b.

B0ERR bit (Dominant Bit Error Flag)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

ADERR bit (Acknowledge Delimiter Error Flag)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0b to it, and can only be set by CAN-FD module logic. Writing 1b has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDCnCTR.ERRD bit is 1b and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0b and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 000_0000b.

CRCREG[14:0] bits (CRC Calculation Data (CRC length: 15 bits))

The CRCREG[14:0] bits read the calculated CRC value when CFDCnCTR.CTME bit is 1b for the channel.

If CFDCnCTR.CTME bit is 0b, then these bits are always read as 0b.

These bits show the CAN2.0 CRC value calculated by the CAN-FD channel logic when the CTME bit is enabled.

The CFDCnERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.5 Channel n Data Bit Rate Configuration Register (CFDCnDCFG) (n = 0 to 5)

This register is not available in the classical CAN function.

The CFDCnDCFG register configures the transmission/reception data bit rate parameters of the channels.

The channel of Classical-only mode does not perform configuration of this register.

Do not write to this register in CH_OPERATION or CH_SLEEP mode. Only write to this register in CH_RESET or CH_HALT mode.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1400h + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	DSJW[3:0]				-	-	-	-	DTSEG2[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	DTSEG1[4:0]					DBRP[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name			Initial Value		R/W	Description									
31 to 28	-			All 0		RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
27 to 24	DSJW[3:0]			0h		RW	Data Bit Rate Resynchronization Jump Width Control 0h: 1 Tq 1h: 2 Tq ⋮ Eh: 15 Tq Fh: 16 Tq									
23 to 20	-			All 0		RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
19 to 16	DTSEG2[3:0]			0h		RW	Data Bit Rate Time Segment 2 Control 0h: Setting prohibited 1h: 2 Tq 2h: 3 Tq 3h: 4 Tq ⋮ Eh: 15 Tq Fh: 16 Tq									
15 to 13	-			All 0		RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
12 to 8	DTSEG1[4:0]			0h		RW	Data Bit Rate Time Segment 1 Control 00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq ⋮ 1Eh: 31 Tq 1Fh: 32 Tq									
7 to 0	DBRP[7:0]			0h		RW	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).									

DBRP[7:0] bits (Data Bit Rate Prescaler Division Ratio Setting)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

DTSEG1[4:0] bits (Data Bit Rate Time Segment 1 Control)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write any other value to these bits. See **7.9.4.1.2 CAN Bit Timing** for more details.

DTSEG2[3:0] bits (Data Bit Rate Time Segment 2 Control)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write any other value to these bits.

DSJW[3:0] bits (Data Bit Rate Resynchronization Jump Width Control)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

7.9.2.2.6 Channel n CAN-FD Configuration Register (CFDCnFDCFG) (n = 0 to 5)

The CFDCnFDCFG register configures which communication direction (transmitter/receiver) errors are counted.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1404h + n x 0020h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDTE	CLOE	REFE	FDOE	-	GWBR S	GWFD F	GWEN	TDCO[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	ESIC	TDCE	TDCOC	-	-	-	-	-	EOCCFG[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	CFDTE	0h	RW	CAN-FD Frame Distinction Enable 0b: CAN-FD frame distinction disabled 1b: CAN-FD frame distinction enabled
30	CLOE ^{*2}	0h	RW	Classical CAN-Only Enable ^{*1} 0b: Classical-only mode disabled 1b: Classical-only mode enabled
29	REFE	0h	RW	RX Edge Filter Enable 0b: RX edge filter disabled 1b: RX edge filter enabled
28	FDOE ^{*2}	0h	RW	FD-Only Enable 0b: FD-only mode disabled 1b: FD-only mode enabled
27	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
26	GWBR ^{*2}	0h	RW	Gateway BRS Configuration Bit 0b: Gateway frame is transmitted with BRS = 0 1b: Gateway frame is transmitted with BRS = 1
25	GWFD ^{*2}	0h	RW	Gateway FDF Configuration Bit 0b: Gateway frame is transmitted as Classical CAN frame 1b: Gateway frame is transmitted as CAN-FD frame
24	GWEN ^{*2}	0h	RW	CAN2.0, CAN-FD Multi Gateway Enable 0b: Multi gateway disabled 1b: Multi gateway enabled
23 to 16	TDCO ^{*2} [7:0]	0h	RW	Transceiver Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.
15 to 11	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10	ESIC ^{*2}	0h	RW	Error State Indication Configuration 0b: The ESI bit in the frame represents the error state of the node itself 1b: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.
9	TDCE ^{*2}	0h	RW	Transceiver Delay Compensation Enable 0b: Transceiver delay compensation disabled 1b: Transceiver delay compensation enabled
8	TDCOC ^{*2}	0h	RW	Transmitter Delay Compensation Offset Configuration 0b: Measurement and offset 1b: Offset-only
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	EOCCFG[2:0]	0h	RW	Error Occurrence Counter Configuration 000b: All transmit messages and receive messages 001b: All transmit messages 010b: All receive messages 011b: Setting prohibited 100b: Only data phase of transmitted or received CANFD message 101b: Only data phase of transmitted CANFD message 110b: Only data phase of received CANFD message 111b: Setting prohibited

Note 1. This bit is set to 1b for this LSI with CANFD disabled.

Note 2. These bits are not available in the classical CAN function.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transmitter Delay Compensation Offset Configuration)

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN-FD channel. If the bit is set to 0b, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1b, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCE bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN-FD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

ESIC bit (Error State Indication Configuration)

Bus controllers that are used as CAN-to-CAN gateway support that in every forwarded CAN-FD message. The ESI flag does not change to reflect the status of the gateway, bridge, or router but instead the flag is sent as it was in the original message.

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTSn.CFESI or CFDTMFDCTRn.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDCnFDCFG.TDCOC setting.

If CFDCnFDCFG.TDCOC = 0b, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDCnFDCFG.TDCO. See **7.9.4.1.5 Transmitter Delay Compensation** for details on how CFDCnFDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set to this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWEN bit (CAN2.0, CAN-FD Multi Gateway Enable)

When the GWEN bit is enabled, a multi gateway is enabled. Message received on one node can be routed to another node using the COM FIFO when they are configured as gateway FIFO (CFDCnFCCn.CFM[1:0] = 10b). Furthermore, when TX Queue is set as Gateway mode, the message received on one node can be stored in TX Queue and can be sent to another node.

The FDF and BRS bits of the routed message can be changed by the configuration value of the CFDCnFDCFG.GWFDF and CFDCnFDCFG.GWBRS bits. By this, the transmitted value of these bits can be replaced.

Example)

CFDCnFDCFG.GWEN = 1b on channel y
CFDCnFDCFG.GWFDF = 1b

If a Classical CAN frame is received on channel x and routed to a gateway FIFO or TX Queue of channel y, then this CAN frame is sent on channel y as a CAN-FD frame because of the CFDCnFDCFG.GWFDF bit.

Table 7.9-5 shows how the message information is changed depending on the received and configured data.

Table 7.9-5 Modified Message Information by Received and Configured Data

Routed CAN Frame	Routed Received DLC	CAN BRS Bit	Configured CFDCnFDCFG.GWFDF Bit	Gateway Message DLC	Gateway Message BRS Bit	Gateway Message Frame Type
CAN2.0	≤ 8	N/A	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	> 8	N/A	1	= 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	≤ 8	None	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	> 8	None	1	> 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	≤ 8	N/A	0	≤ 8	N/A	CAN2.0
CAN2.0	> 8	N/A	0	> 8	N/A	CAN2.0
CAN-FD	≤ 8	None	0	≤ 8	N/A	CAN2.0
CAN-FD	> 8	None	0	= 8	N/A	CAN2.0

Note: This gateway is limited to an 8-byte data payload for different frame type. If routing and target frame type is the same, the data length code (DLC) value remains the same. If the source frame is a CAN-FD with more than 8 data bytes, then on classical destination node, the data payload is reduced to 8 bytes. Only the first 8 bytes of data perform gateway transmission.

Note: Transmission buffers other than the gateway FIFO are not affected by this feature.

Do not route remote frames through the gateway when CFDCnFDCFG.GWEN is set. When a destination node is CFDCnFDCFG.FDOE = 1b, set CFDCnFDCFG.GWEN and CFDCnFDCFG.GWFDF to 1b. When a destination node is CFDCnFDCFG.CLOE = 1b, set CFDCnFDCFG.GWEN = 1b and CFDCnFDCFG.GWFDF = 0b.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

GWDF bit (Gateway FDF Configuration Bit)

When the GWEN bit is set to 1b, the FDF bit of the transmitting gateway frame is replaced by the value of the GWDF bit.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CAN-FD module is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWBR bit (Gateway BRS Configuration Bit)

When the GWEN bit is set to 1b, the BRS bit of the transmitting gateway frame is replaced by the value of CFDCnFDCFG.GWBR.

In classical CAN frames, the GWBR bit is invalid.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

FDOE bit (FD-Only Enable)

The FDOE bit enables the reception and transmission of CAN-FD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTSn.CFFDF/CFD TMFDCTRn.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with CCh.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDCnFDCFG.FDOE and CFDCnFDCFG.CLOE simultaneously.

This bit is not available in the classical CAN function.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical-only mode and when CFDCnFDCFG.CFDTE = 0b (disabled CAN-FD frame distinction).

CLOE bit (Classical CAN-Only Enable)

The CLOE bit enables the Classical CAN-only mode. If this bit is 1b, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDCnFDCFG.CLOE and CFDCnFDCFG.FDOE simultaneously.

CFDCnFDCFG.CLOE	CFDCnFDCFG.FDOE	Channel mode
0	0	CAN-FD mode (Dual mode)
0	1	FD-only mode
1	0	Classical CAN-only mode
1	1	Reserved

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

CFDTE bit (CAN-FD Frame Distinction Enable)

The CFDTE bit enables the CAN-FD frame distinction function. The CFDTE bit is required for Classical CAN-only mode (CFDCnFDCFG.CLOE = 1b).

If this bit is 0b, the protocol controller can only send Classical frames and response with a Form or CRC error on FD frames.

If this bit is 1b, the protocol controller behaves according to the ISO 11898-1 (DIS 2015) specification. If the FDF bit is detected recessive, then the protocol controller enters the protocol exception state, and attempts to integrate back to the CAN communication.

Do not write this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.7 Channel n CAN-FD Control Register (CFDCnFDCTR) (n = 0 to 5)

The CFDCnFDCTR register (n = 0 to 5) controls the error and successful occurrence counters.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1408h + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SOCCLR	EOCCLR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	SOCCLR	0h	RW	Successful Occurrence Counter Clear 0b: No successful occurrence counter clear 1b: Clear successful occurrence counter
0	EOCCLR	0h	RW	Error Occurrence Counter Clear 0b: No error occurrence counter clear 1b: Clear error occurrence counter

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0b.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0b.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.8 Channel n CAN-FD Status Register (CFDCnFDSTS) (n = 0 to 5)

The CFDCnFDSTS register (n = 0 to 5) indicates the transceiver compensation delay result and its related FIFO message lost status.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 140Ch + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOC[7:0]								EOC[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDCVF	-	-	-	-	-	SOCO	EOCO	TDCR[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SOC[7:0]	0h	R	Successful Occurrence Counter These bits show the successful occurrence counter value.
23 to 16	EOC[7:0]	0h	R	Error Occurrence Counter These bits show the error occurrence counter value.
15	TDCVF	0h	RW	Transceiver Delay Compensation Violation Flag 0b: Transceiver delay compensation violation has not occurred 1b: Transceiver delay compensation violation has occurred
14 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	SOCO	0h	RW	Successful Occurrence Counter Overflow Flag 0b: Successful occurrence counter has not overflowed 1b: Successful occurrence counter has overflowed
8	EOCO	0h	RW	Error Occurrence Counter Overflow Flag 0b: Error occurrence counter has not overflowed 1b: Error occurrence counter has overflowed
7 to 0	TDCR[7:0]	0h	R	Transceiver Delay Compensation Result

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDCnFDCFG.TDCOC configuration and the offset value in CFDCnFDCFG.TDCO. See **7.9.4.1.5 Transmitter Delay Compensation** for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between the FDF bit and the RES bit when CFDCnFDCFG.TDCOC = 0b and the transceiver delay compensation is enabled (CFDCnFDCFG.TDCE = 1b).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EOCO bit (Error Occurrence Counter Overflow Flag)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0b to it. Writing 1b has no effect.

This bit is set automatically when CFDCnFDSTS.EOC is FFh and a CAN bus error is detected based on the configuration defined in CFDCnFDCFG.EOCCFG.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

SOCO bit (Successful Occurrence Counter Overflow Flag)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0b to it. Writing 1b has no effect.

This bit is set automatically when CFDCnFDSTS.SOC is FFh and a successful message reception or successful message transmission occurs.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CAN-FD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDCnFDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0b to it. Writing 1b has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDCnFDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CAN-FD module logic. These bits are cleared by writing 1b to CFDCnFDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDCnFDCFG.EOCCFG bits. When the counter reaches the value of FFh, the update stops.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SOC[7:0] bits (Successful Occurrence Counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CAN-FD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of FFh, the update stops. In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1b to CFDCnFDCTR.SOCCLR.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.9 Channel n CAN-FD CRC Register (CFDCnFDCRC) (n = 0 to 5)

The CFDCnFDCRC register (n = 0 to 5) holds the CRC value calculated for the CAN-FD frame.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1410h + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	SCNT[3:0]				-	-	-	-	CRCREG[20:16]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRCREG[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read.
28 to 25	SCNT[3:0]	0h	R	Stuff Bit Count These bits shows the stuff bit count (mod 8) for the CAN-FD frame.
24 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read.
20 to 0	CRCREG[20:0]	0h	R	CRC Register Value These bits show the CRC value calculated for the CAN-FD frame.

CRCREG[20:0] bits (CRC Register Value)

The CRCREG[20:0] bits contain the CRC value calculated by the CAN-FD channel logic when the CFDCnCTR.CTME bit is enabled.

The CFDCnFDCRC.CRCREG value is updated in the first bit of the CRC field of the CAN-FD frame (reception and transmission).

When the CFDCnCTR.CTME bit is 0b, the CRCREG[20:0] bits are always read as 0b.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0b.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff Bit Count)

The SCNT[3:0] bits contain the stuff count value of the CAN-FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN-FD frame when the CFDCnCTR.CTME bit is enabled in CFDCnFDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDCnCTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0b.

The SCNT value is updated in the first bit of CRC field of the CAN-FD frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.10 Global IP Version Register (CFDGIPV)

The CFDGIPV register shows the release version of the CAN-FD module.

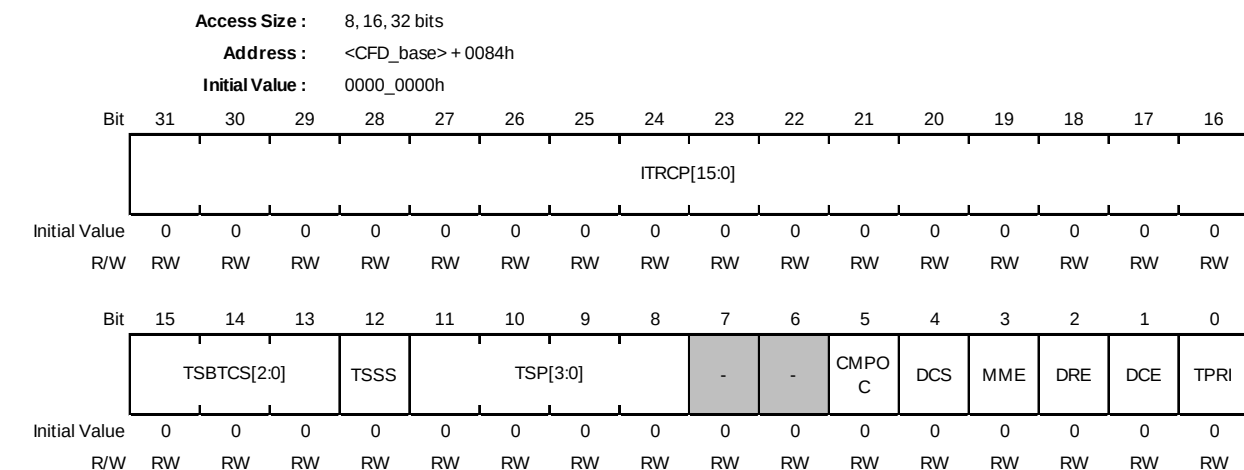
Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0080h														
Initial Value :		1124_0043h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	PSI[13:0]													
Initial Value	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	IPT[1:0]		IPV[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read.
29 to 16	PSI[13:0]	1124h	R	Parameter Status Information These bits show the IP configuration. Bits Description [29:27] Number of channel (010b: 6 channels) [26:24] Number of TXMB (001b: 32/channel) [23:21] Number of AFL entry (001b: 128/channel) [20:17] Number of pool buffer (0010b: 64/channel) [16] Same ID overwrite function of TXQ (0b: Support)
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read.
9, 8	IPT[1:0]	0h* ¹	R	IP Type These bits show the IP type used in the product. 00b: CAN IP 01b: CAN-FD IP 1xb: Reserved
7 to 0	IPV[7:0]	43h	R	IP Version These bits show the main release number.

Note 1. This value depends on the package type.

7.9.2.2.11 Global Configuration Register (CFDGCFG)

The CFDGCFG register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of all CAN channels. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	ITRCP[15:0]	0h	RW	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value
15 to 13	TSBTCS[2:0]	0h	RW	Timestamp Bit Time Channel Select 000b: Select clock from channel 0 001b: Select clock from channel 1 010b: select clock from Channel 2 011b: select clock from Channel 3 100b: select clock from Channel 4 101b: select clock from Channel 5 Others: Setting prohibited
12	TSSS	0h	RW	Timestamp Source Select 0b: Source clock for timestamp counter is peripheral clock 1b: Source clock for timestamp counter is bit time clock
11 to 8	TSP[3:0]	0h	RW	Timestamp Prescaler 0h: Timestamp prescaler = 1 1h: Timestamp prescaler = 2 2h: Timestamp prescaler = 4 3h: Timestamp prescaler = 8 ⋮ Dh: Timestamp prescaler = 8192 Eh: Timestamp prescaler = 16384 Fh: Timestamp prescaler = 32768
7, 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	CMPOC	0h	RW	CAN-FD Message Payload Overflow Configuration 0b: Message is rejected 1b: Message payload is cut to fit to configured message size
4	DCS	0h	RW	Data Link Controller Clock Select 0b: CANFD_0_clkc 1b: Setting prohibited
3	MME	0h	RW	Mirror Mode Enable 0b: Mirror mode disabled 1b: Mirror mode enabled
2	DRE	0h	RW	DLC Replacement Enable 0b: DLC replacement disabled 1b: DLC replacement enabled
1	DCE	0h	RW	DLC Check Enable 0b: DLC check disabled 1b: DLC check enabled

Bit	Bit Name	Initial Value	R/W	Description
0	TPRI	0h	RW	Transmission Priority 0b: ID priority 1b: Message buffer number priority

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1b and the DCE is 1b, the CAN-FD stores the configured value (CFDGAFLP0n.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects the clock source for CAN communication.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

CMPOC bit (CAN-FD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCn.RFPLS, and CFDCFCCn.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode. Additionally, do not set this bit to 1b when CAN-FD communication is used.

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

TSBTCS[2:0] bits (Timestamp Bit Time Channel Select)

The TSBTCS[2:0] bits allow the selection of the bit time clock of a particular channel for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0000h, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

7.9.2.2.12 Global Control Register (CFDGCTR)

The CFDGCTR register controls the global mode of the CAN-FD module and the timestamp function. The register also enables and disables the global error interrupts.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 0088h

Initial Value : 0000_0005h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TSRST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOWEIE	QMEIE	-	QOWEIE	CMPOFIE	THLEIE	MEIE	DEIE	-	-	-	-	-	GSLPR	GMDC[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	TSRST	0h	RW	Timestamp Reset 0b: Timestamp not reset 1b: Timestamp reset
15	MOWEIE	0h	RW	Message Lost Error Interrupt Enable 0b: GW FIFO message lost error interrupt disabled 1b: GW FIFO message lost error interrupt enabled
14	QMEIE	0h	RW	TXQ Message Lost Error Interrupt Enable 0b: TXQ message lost error interrupt disabled 1b: TXQ message lost error interrupt enabled
13	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
12	QOWEIE	0h	RW	TXQ Message Overwrite Error Interrupt Enable 0b: TXQ message overwrite error interrupt disabled 1b: TXQ message overwrite error interrupt enabled
11	CMPOFIE	0h	RW	CAN-FD Message Payload Overflow Flag Interrupt Enable 0b: CAN-FD message payload overflow flag interrupt disabled 1b: CAN-FD message payload overflow flag interrupt enabled
10	THLEIE	0h	RW	TX History List Entry Lost Interrupt Enable 0b: TX history list entry lost interrupt disabled 1b: TX history list entry lost interrupt enabled
9	MEIE	0h	RW	Message Lost Error Interrupt Enable 0b: Message lost error interrupt disabled 1b: Message lost error interrupt enabled
8	DEIE	0h	RW	DLC Check Interrupt Enable 0b: DLC check interrupt disabled 1b: DLC check interrupt enabled
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	GSLPR	1h	RW	Global Sleep Request 0b: Global sleep request disabled 1b: Global sleep request enabled
1, 0	GMDC[1:0]	1h	RW	Global Mode Control 00b: Global operation mode request 01b: Global reset mode request 10b: Global halt mode request 11b: Keep current value

GMDC[1:0] bits (Global Mode Control)

The GMDC[1:0] bits can be used to configure the modes for the CAN-FD module. Additionally, if CFDGCTR.GSLPR bit is 1b when the CAN-FD module is in Reset mode, the CAN-FD module transits to Global Sleep mode. Additionally, if CFDGCTR.GSLPR is 1b when the CAN-FD module is in Reset Mode, then the CAN-FD module transits to Global Sleep Mode.

Setting the GMDC[1:0] bits to 11b has no effect. Mode transition is described in detail in **7.9.3.2 Global Modes**.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CAN-FD module including all CAN channels. Channel sleep request is set automatically for all channels.

Only write to this bit when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1b, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1b, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1b, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CMPOFIE bit (CAN-FD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1b, an interrupt is generated when a CAN-FD message payload overflow condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QOWEIE bit (TXQ Message Overwrite Error Interrupt Enable)

When the QOWEIE bit is 1b, an interrupt is generated when a TXQ message overwrite condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QMEIE bit (TXQ Message Lost Error Interrupt Enable)

When the QMEIE bit is 1b, an interrupt is generated when a TXQ message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MOWEIE bit (Message Lost Error Interrupt Enable)

When the MOWEIE bit is 1b, an interrupt is generated in GW mode and a GW FIFO message overwrite condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

TSRST bit (Timestamp Reset)

When the TSRST bit is 1b, the Global Timestamp Register is reset to 0000h.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0b.

This bit is cleared automatically by the CAN-FD logic.

7.9.2.2.13 Global Status Register (CFDGSTS)

The CFDGSTS register indicates the global status of the CAN-FD module.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 008Ch														
Initial Value :		0000_000Dh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	GRAM INIT	GSLPS TS	GHLTS TS	GRSTS TS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read.
3	GRAMINIT	1h	R	Global RAM Initialization 0b: RAM initialization is complete 1b: RAM initialization is ongoing
2	GSLPSTS	1h	R	Global Sleep Status 0b: Not in Sleep mode 1b: In Sleep mode
1	GHLTSTS	0h	R	Global Halt Status 0b: Not in Halt mode 1b: In Halt mode
0	GRSTSTS	1h	R	Global Reset Status 0b: Not in Reset mode 1b: In Reset mode

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CAN-FD module Reset mode.

This bit is set automatically when the CAN-FD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CAN-FD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CAN-FD module Halt mode.

This bit is set automatically when the CAN-FD module enters GL_HALT mode.

This bit is cleared automatically when the CAN-FD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CAN-FD module Sleep mode.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode.

This bit is cleared automatically when the CAN-FD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CAN-FD module RAM initialization.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CAN-FD module completed RAM initialization.

7.9.2.2.14 Global Error Flag Register (CFDGERFL)

The CFDGERFL register indicates the detection of global errors.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0090h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	EEF5	EEF4	EEF3	EEF2	EEF1	EEF0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	MOWE S	QMES	-	QOWE S	CMPO F	THLES	MES	DEF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	RW	R	RW	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
21 to 16	EEFn	0h	RW	ECC Error Flag for Channel n (n = 0 to 5) 0b: ECC error not detected during TX-SCAN 1b: ECC error detected during TX-SCAN
15 to 8	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7	MOWES	0h	R	Message Overwrite Error Status 0b: Message overwrite error not detected 1b: Message overwrite error detected
6	QMES	0h	R	TXQ Message Lost Error Status 0b: TXQ message lost error not detected 1b: TXQ message lost error detected
5	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	QOWES	0h	R	TXQ Message Overwrite Error Status 0b: TXQ message overwrite error not detected 1b: TXQ message overwrite error detected
3	CMPOF	0h	RW	CAN-FD Message Payload Overflow Flag 0b: CAN-FD message payload overflow not detected 1b: CAN-FD message payload overflow detected
2	THLES	0h	R	TX History List Entry Lost Error Status 0b: TX history list entry lost error not detected 1b: TX history list entry lost error detected
1	MES	0h	R	Message Lost Error Status 0b: Message lost error not detected 1b: Message lost error detected
0	DEF	0h	RW	DLC Error Flag 0b: DLC error not detected 1b: DLC error detected

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when a DLC error is detected in a received frame.

The bit is cleared by writing 0b to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

CMPOF bit (CAN-FD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CAN-FD message payload overflow is detected on at least one channel.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0b to it. Writing 1b to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is cleared automatically in GL_RESET mode.

QOWES bit (TXQ Message Overwrite Error Status)

The QOWES bit is set automatically when a TXQ message overwrite error is detected.

This bit is cleared automatically when all TXQ message overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

QMES bit (TXQ Message Lost Error Status)

The QMES bit is set automatically when a TXQ message lost error is detected.

This bit is cleared automatically when all TXQ message lost flags are cleared.

This bit is cleared automatically in GL_RESET mode.

MOWES bit (Message Overwrite Error Status)

The MOWES bit is set automatically when GW mode and COMFIFO message overwrite error is detected.

This bit is cleared automatically when all COMFIFO Message Overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

EEFn bit (ECC Error Flag for Channel n) (n = 0 to 5)

The EEFn bit specifies whether an ECC error has occurred on Channel n.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1b to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

The bit is cleared by writing 0b to it. This bit is cleared automatically in GL_RESET mode.

7.9.2.2.15 Global TX Interrupt Status Register 0 (CFDGTINTSTS0)

The CFDGTINTSTS0 register indicates the detection of transmit specific interrupts.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1300h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	CFOTIF3	TQOFIF3	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	-	CFOTIF2	TQOFIF2	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CFOTIF1	TQOFIF1	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	-	CFOTIF0	TQOFIF0	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read.
30	CFOTIF3	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 3 0b: Channel 3 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 3 COM FIFO One Frame Transmission Interrupt flag set
29	TQOFIF3	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 3 0b: Channel 3 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 3 TX Queue One Frame Transmission Interrupt flag set
28	THIF3	0h	R	TX History List Interrupt Channel 3 0b: Channel 3 TX History List Interrupt flag not set 1b: Channel 3 TX History List Interrupt flag set
27	CFTIF3	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 3 0b: Channel 3 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 3 COM FIFO TX/GW Mode Interrupt flag set
26	TQIF3	0h	R	TX Queue Interrupt Flag Channel 3 0b: Channel 3 TX Queue Interrupt flag not set 1b: Channel 3 TX Queue Interrupt flag set
25	TAIF3	0h	R	TX Abort Interrupt Flag Channel 3 0b: Channel 3 TX Abort Interrupt flag not set 1b: Channel 3 TX Abort Interrupt flag set
24	TSIF3	0h	R	TX Successful Interrupt Flag Channel 3 0b: Channel 3 TX Successful Interrupt flag not set 1b: Channel 3 TX Successful Interrupt flag set
23	-	0h	R	Reserved Whenever it is read, 0b is read.
22	CFOTIF2	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 2 0b: Channel 2 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 2 COM FIFO One Frame Transmission Interrupt flag set
21	TQOFIF2	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 2 0b: Channel 2 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 2 TX Queue One Frame Transmission Interrupt flag set
20	THIF2	0h	R	TX History List Interrupt Channel 2 0b: Channel 2 TX History List Interrupt flag not set 1b: Channel 2 TX History List Interrupt flag set
19	CFTIF2	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 2 0b: Channel 2 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 2 COM FIFO TX/GW Mode Interrupt flag set
18	TQIF2	0h	R	TX Queue Interrupt Flag Channel 2 0b: Channel 2 TX Queue Interrupt flag not set 1b: Channel 2 TX Queue Interrupt flag set
17	TAIF2	0h	R	TX Abort Interrupt Flag Channel 2 0b: Channel 2 TX Abort Interrupt flag not set 1b: Channel 2 TX Abort Interrupt flag set

Bit	Bit Name	Initial Value	R/W	Description
16	TSIF2	0h	R	TX Successful Interrupt Flag Channel 2 0b: Channel 2 TX Successful Interrupt flag not set 1b: Channel 2 TX Successful Interrupt flag set
15	-	0h	R	Reserved Whenever it is read, 0b is read.
14	CFOTIF1	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 1 0b: Channel 1 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 1 COM FIFO One Frame Transmission Interrupt flag set
13	TQOFIF1	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 1 0b: Channel 1 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 1 TX Queue One Frame Transmission Interrupt flag set
12	THIF1	0h	R	TX History List Interrupt Channel 1 0b: Channel 1 TX History List Interrupt flag not set 1b: Channel 1 TX History List Interrupt flag set
11	CFTIF1	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 1 0b: Channel 1 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 1 COM FIFO TX/GW Mode Interrupt flag set
10	TQIF1	0h	R	TX Queue Interrupt Flag Channel 1 0b: Channel 1 TX Queue Interrupt flag not set 1b: Channel 1 TX Queue Interrupt flag set
9	TAIF1	0h	R	TX Abort Interrupt Flag Channel 1 0b: Channel 1 TX Abort Interrupt flag not set 1b: Channel 1 TX Abort Interrupt flag set
8	TSIF1	0h	R	TX Successful Interrupt Flag Channel 1 0b: Channel 1 TX Successful Interrupt flag not set 1b: Channel 1 TX Successful Interrupt flag set
7	-	0h	R	Reserved Whenever it is read, 0b is read.
6	CFOTIF0	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 0 0b: Channel 0 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 0 COM FIFO One Frame Transmission Interrupt flag set
5	TQOFIF0	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 0 0b: Channel 0 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 0 TX Queue One Frame Transmission Interrupt flag set
4	THIF0	0h	R	TX History List Interrupt Channel 0 0b: Channel 0 TX History List Interrupt flag not set 1b: Channel 0 TX History List Interrupt flag set
3	CFTIF0	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 0 0b: Channel 0 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 0 COM FIFO TX/GW Mode Interrupt flag set
2	TQIF0	0h	R	TX Queue Interrupt Flag Channel 0 0b: Channel 0 TX Queue Interrupt flag not set 1b: Channel 0 TX Queue Interrupt flag set
1	TAIF0	0h	R	TX Abort Interrupt Flag Channel 0 0b: Channel 0 TX Abort Interrupt flag not set 1b: Channel 0 TX Abort Interrupt flag set
0	TSIF0	0h	R	TX Successful Interrupt Flag Channel 0 0b: Channel 0 TX Successful Interrupt flag not set 1b: Channel 0 TX Successful Interrupt flag set

TSIFn bit (TX Successful Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.TSIFn bit is set to 1b when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAIFn bit (TX Abort Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.TAIFn bit is set to 1b when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIFn bit (TX Queue Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.TQIFn bit is set to 1b when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFTIFn bit (COM FIFO in TX/GW Mode Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.CFTIFn bit is set to 1b when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIFn bit (TX History List Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.THIFn bit is set to 1b when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQOFIFn bit (TX Queue One Frame Transmission Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.TQOFIFn bit is set to 1b when the TX Queue One Frame Transmission Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue One Frame Transmission Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFOTIFn bit (COM FIFO One Frame Transmission Interrupt Flag Channel n (n = 0 to 3))

The CFDGTINTSTS0.CFOTIFn bit is set to 1b when the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is set (when the Interrupt is enabled).

This bit is cleared automatically:

- When the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

7.9.2.2.16 Global TX Interrupt Status Register 1 (CFDGTINTSTS1)

The CFDGTINTSTS1 register indicates the detection of transmit specific interrupts.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1304h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CFOTIF5	TQOFIF5	THIF5	CFTIF5	TQIF5	TAIF5	TSIF5	-	CFOTIF4	TQOFIF4	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	-	All 0	R	Reserved Whenever it is read, 0b is read.
14	CFOTIF5	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 5 0b: Channel 5 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 5 COM FIFO One Frame Transmission Interrupt flag set
13	TQOFIF5	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 5 0b: Channel 5 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 5 TX Queue One Frame Transmission Interrupt flag set
12	THIF5	0h	R	TX History List Interrupt Channel 5 0b: Channel 5 TX History List Interrupt flag not set 1b: Channel 5 TX History List Interrupt flag set
11	CFTIF5	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 5 0b: Channel 5 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 5 COM FIFO TX/GW Mode Interrupt flag set
10	TQIF5	0h	R	TX Queue Interrupt Flag Channel 5 0b: Channel 5 TX Queue Interrupt flag not set 1b: Channel 5 TX Queue Interrupt flag set
9	TAIF5	0h	R	TX Abort Interrupt Flag Channel 5 0b: Channel 5 TX Abort Interrupt flag not set 1b: Channel 5 TX Abort Interrupt flag set
8	TSIF5	0h	R	TX Successful Interrupt Flag Channel 5 0b: Channel 5 TX Successful Interrupt flag not set 1b: Channel 5 TX Successful Interrupt flag set
7	-	0h	RW	Reserved Whenever it is read, 0b is read.
6	CFOTIF4	0h	R	COM FIFO One Frame Transmission Interrupt Flag Channel 4 0b: Channel 4 COM FIFO One Frame Transmission Interrupt flag not set 1b: Channel 4 COM FIFO One Frame Transmission Interrupt flag set
5	TQOFIF4	0h	R	TX Queue One Frame Transmission Interrupt Flag Channel 4 0b: Channel 4 TX Queue One Frame Transmission Interrupt flag not set 1b: Channel 4 TX Queue One Frame Transmission Interrupt flag set
4	THIF4	0h	R	TX History List Interrupt Channel 4 0b: Channel 4 TX History List Interrupt flag not set 1b: Channel 4 TX History List Interrupt flag set
3	CFTIF4	0h	R	COM FIFO TX/GW Mode Interrupt Flag Channel 4 0b: Channel 4 COM FIFO TX/GW Mode Interrupt flag not set 1b: Channel 4 COM FIFO TX/GW Mode Interrupt flag set
2	TQIF4	0h	R	TX Queue Interrupt Flag Channel 4 0b: Channel 4 TX Queue Interrupt flag not set 1b: Channel 4 TX Queue Interrupt flag set
1	TAIF4	0h	R	TX Abort Interrupt Flag Channel 4 0b: Channel 4 TX Abort Interrupt flag not set 1b: Channel 4 TX Abort Interrupt flag set

Bit	Bit Name	Initial Value	R/W	Description
0	TSIF4	0h	R	TX Successful Interrupt Flag Channel 4 0b: Channel 4 TX Successful Interrupt flag not set 1b: Channel 4 TX Successful Interrupt flag set

TSIFn bit (TX Successful Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.TSIFn bit is set to 1b when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAIFn bit (TX Abort Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.TAIFn bit is set to 1b when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIFn bit (TX Queue Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.TQIFn bit is set to 1b when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFTIFn bit (COM FIFO in TX/GW Mode Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.CFTIFn bit is set to 1b when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIFn bit (TX History List Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.THIFn bit is set to 1b when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQOFIFn bit (TX Queue One Frame Transmission Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.TQOFIFn bit is set to 1b when the TX Queue One Frame Transmission Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue One Frame Transmission Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFOTIFn bit (COM FIFO One Frame Transmission Interrupt Flag Channel n (n = 4, 5))

The CFDGTINTSTS0.CFOTIFn bit is set to 1b when the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is set (when the Interrupt is enabled).

This bit is cleared automatically:

- When the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

7.9.2.2.17 Global Timestamp Counter Register (CFDGTSC)

The CFDGTSC register stores the timestamp based on the selected configuration.

Access Size :		16, 32 bits														
Address :		<CFD_base> + 0094h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W	Description										
31 to 16	-		All 0		R	Reserved Whenever it is read, 0b is read.										
15 to 0	TS[15:0]		0h		R	Timestamp value										

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS, and TSP. The proper incrementing of the timestamp counter cannot be guaranteed when transitioning to halt state.

Do not write to bits TS[15:0] when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

7.9.2.2.18 Global Acceptance Filter List Entry Control Register (CFDGAFLECTR)

The CFDGAFLECTR register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0098h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	AFLDA E	-	-	AFLPN[5:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	AFLDAE	0h	RW	Acceptance Filter List Data Access Enable 0b: Acceptance Filter List data access disabled 1b: Acceptance Filter List data access enabled
7 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5 to 0	AFLPN[5:0]	0h	RW	Acceptance Filter List Page Number Select an Acceptance Filter List page (Set from 0 to 47.)

AFLPN[5:0] bits (Acceptance Filter List Page Number)

The AFLPN[5:0] bits select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode. Enter only the values between 0h and 2Fh, inclusive.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

7.9.2.2.19 Global Acceptance Filter List Configuration Register w (CFDGAFLCFGw) (w = 0, 1, 2)

The CFDGAFLCFGw register is used to define the number of rules for entries in the Acceptance Filter List, applicable for channels 0 to 5.

The total number of available entries in the Acceptance Filter List is 256 for 2 CAN channels. However, the filters can be allocated flexibly to the different channels depending on requirements as long as both of the following conditions are satisfied:

- The maximum number of acceptance filter per channel is 256
- The total number of rules defined for all channels is not exceeding the number of available entries in the Acceptance Filter List.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 009Ch + w x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	RNC0[8:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	RNC1[8:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0 is read. The written value should always be 0b.
24 to 16	RNC0[8:0]	0h	RW	Rule Number for Channel 0 Number of rules dedicated to channel 0
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0 is read. The written value should always be 0b.
8 to 0	RNC1[8:0]	0h	RW	Rule Number for Channel 1 Number of rules dedicated to channel 1

RNCn[8:0] bits (Rule Number for Channel n (n = 0 to 5))

The RNCn[8:0] bits define the number of rules in the Acceptance Filter List for channel n.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

7.9.2.2.20 Global Acceptance Filter List ID Register n (CFDGAFLIDn) (n = 1 to 16)

The CFDGAFLIDn register (n = 1 to 16) is used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1800h + (n - 1) x 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLI DE	GAFLR TR	GAFL LB	GAFLID[28:16]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	GAFLIDE	0h	RW	Global Acceptance Filter List Entry IDE Field 0b: Standard identifier of rule entry ID is valid for acceptance filtering 1b: Extended identifier of rule entry ID is valid for acceptance filtering
30	GAFLRTR	0h	RW	Global Acceptance Filter List Entry RTR Field 0b: Data frame 1b: Remote frame
29	GAFLLB	0h	RW	Global Acceptance Filter List Entry Loopback Configuration 0b: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1b: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX
28 to 0	GAFLID[28:0]	0h	RW	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

The acceptance filter process compares this field against the ID of a received CAN message. For alignment of these bits in standard and extended frame formats, see **7.9.2.2.89 Identifier Bits Alignment**.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See **7.9.5.5 Loopback Modes** for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

7.9.2.2.21 Global Acceptance Filter List Mask Register n (CFDGAFLMn) (n = 1 to 16)

The CFDGAFLMn register (n = 1 to 16) is used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 1804h + (n - 1) x 0010h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	GAFLI DEM	GAFLR TRM	GAFLIF L1	GAFLIDM[28:16]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	GAFLIDM[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31	GAFLIDEM	0h	RW	Global Acceptance Filter List IDE Mask 0b: IDE bit is not used for ID matching 1b: IDE bit is used for ID matching
30	GAFLRTRM	0h	RW	Global Acceptance Filter List Entry RTR Mask 0b: RTR bit is not used for ID matching 1b: RTR bit is used for ID matching
29	GAFLIFL1	0h	RW	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1
28 to 0	GAFLIDM[28:0]	0h	RW	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

- 0: Corresponding STD-ID/EXT-ID bit is not used for ID matching
- 1: Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTS.RMIFL[1], CFDRFFDSTS.RFIFL[1], CFDCFFDCSTSn.CFIFL[1]) of the storage location of an incoming message.

Note: This bit is stored in CFDTHLACC1n.TIFL[1] when CFDTHLCCn.THLDGE = 1b is set up using Gateway function.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0b, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0b, the STD-ID comparison takes place.

If the received IDE bit is 1b, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

7.9.2.2.22 Global Acceptance Filter List Pointer 0 Register n (CFDGAFLP0n) (n = 1 to 16)

The CFDGAFLP0n register (n = 1 to 16) is used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 1808h + (n - 1) x 0010h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	GAFLPTR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	GAFLRMV	-	-	GAFLRMDP[4:0]				GAFLIFL0	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLDLC[3:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	GAFLPTR [15:0]	0h	RW	Global Acceptance Filter List Pointer
15	GAFLRMV	0h	RW	Global Acceptance Filter List RX Message Buffer Valid 0b: Single message buffer direction pointer is invalid 1b: Single message buffer direction pointer is valid
14, 13	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
12 to 8	GAFLRMDP [4:0]	0h	RW	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages
7	GAFLIFL0	0h	RW	Global Acceptance Filter List Information Label 0
6	GAFLSRD2	0h	RW	Global Acceptance Filter List Select Routing Destination 2 0b: Routing target is CFIFO2 1b: Routing target is TX Queue 2 instead of CFIFO2
5	GAFLSRD1	0h	RW	Global Acceptance Filter List Select Routing Destination 1 0b: Routing target is CFIFO1 1b: Routing target is TX Queue 1 instead of CFIFO1
4	GAFLSRD0	0h	RW	Global Acceptance Filter List Select Routing Destination 0 0b: Routing target is CFIFO0 1b: Routing target is TX Queue 0 instead of CFIFO0
3 to 0	GAFLDLC[3:0]	0h	RW	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry.

Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0b.

Table 7.9-6 shows DLC value that can be configured.

Table 7.9-6 Configuration of DLC Value

Format	DLC[3:0]	Description
CAN and CAN-FD	0000b	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CAN-FD	0001b	DLC of received message = 1 or more
CAN and CAN-FD	0010b	DLC of received message = 2 or more
CAN and CAN-FD	0011b	DLC of received message = 3 or more
CAN and CAN-FD	0100b	DLC of received message = 4 or more
CAN and CAN-FD	0101b	DLC of received message = 5 or more
CAN and CAN-FD	0110b	DLC of received message = 6 or more
CAN and CAN-FD	0111b	DLC of received message = 7 or more
CAN	1xxx b	DLC of received message = 8 or more
CAN-FD	1000b	DLC of received message = 8 or more* ¹
CAN-FD	1001b	DLC of received message = 12 or more* ¹
CAN-FD	1010b	DLC of received message = 16 or more* ¹
CAN-FD	1011b	DLC of received message = 20 or more* ¹
CAN-FD	1100b	DLC of received message = 24 or more* ¹
CAN-FD	1101b	DLC of received message = 32 or more* ¹
CAN-FD	1110b	DLC of received message = 48 or more* ¹
CAN-FD	1111b	DLC of received message = 64* ¹

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD0 bit (Global Acceptance Filter List Select Routing Destination 0)

The GAFLSRD0 bit changes a copy destination to CFIFO0 or TXQ0 by routing.

If this bit is set as 1b, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ0.

If this bit is set to 0b, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO0.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD1 bit (Global Acceptance Filter List Select Routing Destination 1)

The GAFLSRD1 bit changes a copy destination to CFIFO1 or TXQ1 by routing.

If this bit is set to 1b, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ1.

If this bit is set to 0b, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO1.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD2 bit (Global Acceptance Filter List Select Routing Destination 2)

The GAFLSRD2 bit changes a copy destination to CFIFO2 or TXQ2 by routing.

If this bit is set to 1b, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ2.

If this bit is set to 0b, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO2.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to the bit when the CAN-FD module is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTS.RMIFL[0], CFDRFFDSTS.RFIFL[0], CFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

This bit is stored in CFDTHLACC1n.TIFL[0] when CFDTHLCCn.THLDGE = 1b is set up using the gateway function.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[7:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0n.GAFLRMDP[4:0] bits should only be between 00h and CFDRMNB.NRXMB[7:0] to 1 less.

If CFDRMNB.NRXMB[7:0] = 00h, the GAFLRMV bit should be configured as 0b.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

7.9.2.2.23 Global Acceptance Filter List Pointer 1 Register n (CFDGAFLP1n) (n = 1 to 16)

The CFDGAFLP1n register (n = 1 to 16) is used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 180Ch + (n - 1) x 0010h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	GAFLFDP[25:16]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	GAFLFDP[15:0]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
25 to 0	GAFLFDP [25:0]	0h	RW	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage

GAFLFDP[25:0] bits (Global Acceptance Filter List FIFO Direction Pointer)

The GAFLFDP[25:0] bits allow the configuration of FIFO buffers as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. Each bit of the CFDGAFLP1n.GAFLFDP[25:0] is configured as dedicated FIFO.

Bit	Value (binary)	Function
0	0	Disable RX FIFO 0 as target for reception
	1	Enable RX FIFO 0 as target for reception
1	0	Disable RX FIFO 1 as target for reception
	1	Enable RX FIFO 1 as target for reception
2	0	Disable RX FIFO 2 as target for reception
	1	Enable RX FIFO 2 as target for reception
3	0	Disable RX FIFO 3 as target for reception
	1	Enable RX FIFO 3 as target for reception
4	0	Disable RX FIFO 4 as target for reception
	1	Enable RX FIFO 4 as target for reception
5	0	Disable RX FIFO 5 as target for reception
	1	Enable RX FIFO 5 as target for reception
6	0	Disable RX FIFO 6 as target for reception
	1	Enable RX FIFO 6 as target for reception
7	0	Disable RX FIFO 7 as target for reception
	1	Enable RX FIFO 7 as target for reception
8	0	Disable Common FIFO 0 and Channel 0 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 0 as target for reception GAFLSRD0 = 1b: Enable Channel 0 TX Queue 0 as target for reception

Bit	Value (binary)	Function
9	0	Disable Common FIFO 1 and Channel 0 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 1 as target for reception GAFLSRD1 = 1b: Enable Channel 0 TX Queue 1 as target for reception
10	0	Disable Common FIFO 2 and Channel 0 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 2 as target for reception GAFLSRD2 = 1b: Enable Channel 0 TX Queue 2 as target for reception
11	0	Disable Common FIFO 3 and Channel 1 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 3 as target for reception GAFLSRD0 = 1b: Enable Channel 1 TX Queue 0 as target for reception
12	0	Disable Common FIFO 4 and Channel 1 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 4 as target for reception GAFLSRD1 = 1b: Enable Channel 1 TX Queue 1 as target for reception
13	0	Disable Common FIFO 5 and Channel 1 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 5 as target for reception GAFLSRD2 = 1b: Enable Channel 1 TX Queue 2 as target for reception
14	0	Disable Common FIFO 6 and Channel 2 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 6 as target for reception GAFLSRD0 = 1b: Enable Channel 2 TX Queue 0 as target for reception
15	0	Disable Common FIFO 7 and Channel 2 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 7 as target for reception GAFLSRD1 = 1b: Enable Channel 2 TX Queue 1 as target for reception
16	0	Disable Common FIFO 8 and Channel 2 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 8 as target for reception GAFLSRD2 = 1b: Enable Channel 2 TX Queue 2 as target for reception
17	0	Disable Common FIFO 9 and Channel 3 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 9 as target for reception GAFLSRD0 = 1b: Enable Channel 3 TX Queue 0 as target for reception
18	0	Disable Common FIFO 10 and Channel 3 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 10 as target for reception GAFLSRD1 = 1b: Enable Channel 3 TX Queue 1 as target for reception
19	0	Disable Common FIFO 11 and Channel 3 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 11 as target for reception GAFLSRD2 = 1b: Enable Channel 3 TX Queue 2 as target for reception
20	0	Disable Common FIFO 12 and Channel 4 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 12 as target for reception GAFLSRD0 = 1b: Enable Channel 4 TX Queue 0 as target for reception
21	0	Disable Common FIFO 13 and Channel 4 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 13 as target for reception GAFLSRD1 = 1b: Enable Channel 4 TX Queue 1 as target for reception
22	0	Disable Common FIFO 14 and Channel 4 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 14 as target for reception GAFLSRD2 = 1b: Enable Channel 4 TX Queue 2 as target for reception
23	0	Disable Common FIFO 15 and Channel 5 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0b: Enable Common FIFO 15 as target for reception GAFLSRD0 = 1b: Enable Channel 5 TX Queue 0 as target for reception
24	0	Disable Common FIFO 16 and Channel 5 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0b: Enable Common FIFO 16 as target for reception GAFLSRD1 = 1b: Enable Channel 5 TX Queue 1 as target for reception

Bit	Value (binary)	Function
25	0	Disable Common FIFO 17 and Channel 5 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0b: Enable Common FIFO 17 as target for reception GAFLSRD2 = 1b: Enable Channel 5 TX Queue 2 as target for reception

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0b. Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

For storage in TX queue, TX queue buffers of a target that is in GW mode is possible.

Only one of the following configurations is valid:

- Up to 8 destination FIFO buffers
- 7 destination FIFO buffers plus one RX message buffer
- 8 destination TX queue buffers
- 7 destination TX queue buffers plus one RX message buffer
- A maximum of 8 destinations in all at FIFO buffer and TX queue buffer.

7.9.2.2.24 RX Message Buffer Number Register (CFDRMNB)

The CFDRMNB register is used to configure the total number of RX message buffers allocated to all channels.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 00ACh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RMPLS[2:0]			NRXMB[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10 to 8	RMPLS* ¹ [2:0]	0h	RW	Reception Message Buffer Payload Data Size 000b: 8 bytes 001b: 12 bytes 010b: 16 bytes 011b: 20 bytes 100b: 24 bytes 101b: 32 bytes 110b: 48 bytes 111b: 64 bytes
7 to 0	NRXMB[7:0]	0h	RW	Number of RX Message Buffers

Note 1. These bits are not available in the classical CAN function.

NRXMB[7:0] bits (Number of RX Message Buffers)

The NRXMB[7:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Enter only values between 0 and 32 inclusive, with 00h indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

7.9.2.2.25 RX Message Buffer New Data Register t (CFDRMNDt) (t = 0 to 2)

The CFDRMNDt specifies the new data storage status of the RX message buffers.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 00B0h + t x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	RMNS[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	RMNS[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RMNS[31:0]	0h	RW	RX Message Buffer New Data Status 0b: New data not stored in corresponding RX message buffer 1b: New data stored in corresponding RX message buffer

RMNS[31:0] bits (RX Message Buffer New Data Status)

The RMNSu[31:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

no_of_channels = 6

no_of_CFDRMBBCPs_per_channel = (number of RX message buffer components per channel) = 16

no_of_CFDRMBBCPs = no_of_channels × no_of_CFDRMBBCPs_per_channel (6 × 16 = 96)

no_of_bits_per_register = 32

no_of_CFDRMNDs (number of CFDRMND Registers) = no_of_CFDRMBBCPs/no_of_bits_per_register (96/32 = 3)

t = [0 ... no_of_CFDRMNDs - 1]

u = [t × 32 ... (no_of_CFDRMBBCPs - ((no_of_CFDRMNDs - 1 - t) × 32) - 1)]

t can be calculated from the target of the New Data Status flag (u) using the formula $t = \text{floor}(u/32)$

Bit position can be calculated using the formula $(u - (t \times 32))$

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode. Writing 1b has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.

Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0b. These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 CANFD_0_pclk cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 CANFD_0_pclk cycles + 1 for each 4 bytes (maximum of 20 CANFD_0_pclk cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

7.9.2.2.26 RX FIFO Configuration/Control Register n (CFDRFCCn) (n = 0 to 7)

The CFDRFCCn registers (n = 0 to 7) are used to configure and control the eight RX FIFOs.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 00C0h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RFFIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	-	RFDC[2:0]			-	RFPLS[2:0]			-	-	RFIE	RFE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	RFFIE	0h	RW	RX FIFO Full Interrupt Enable 0b: FIFO interrupt generation disabled 1b: FIFO interrupt generation enabled
15 to 13	RFIGCV[2:0]	0h	RW	RX FIFO Interrupt Generation Counter Value 000b: Interrupt generated when FIFO is 1/8th full 001b: Interrupt generated when FIFO is 1/4th full 010b: Interrupt generated when FIFO is 3/8th full 011b: Interrupt generated when FIFO is 1/2 full 100b: Interrupt generated when FIFO is 5/8th full 101b: Interrupt generated when FIFO is 3/4th full 110b: Interrupt generated when FIFO is 7/8th full 111b: Interrupt generated when FIFO is full
12	RFIM	0h	RW	RX FIFO Interrupt Mode 0b: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1b: Interrupt generated at the end of every received message storage
11	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
10 to 8	RFDC[2:0]	0h	RW	RX FIFO Depth Configuration 000b: FIFO Depth = 0 messages 001b: FIFO Depth = 4 messages 010b: FIFO Depth = 8 messages 011b: FIFO Depth = 16 messages 100b: FIFO Depth = 32 messages 101b: FIFO Depth = 48 messages 110b: FIFO Depth = 64 messages 111b: FIFO Depth = 128 messages
7	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
6 to 4	RFPLS* ¹ [2:0]	0h	RW	Rx FIFO Payload Data Size Configuration 000b: 8 bytes 001b: 12 bytes 010b: 16 bytes 011b: 20 bytes 100b: 24 bytes 101b: 32 bytes 110b: 48 bytes 111b: 64 bytes
3, 2	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Bit	Bit Name	Initial Value	R/W	Description
1	RFIE	0h	RW	RX FIFO Interrupt Enable 0b: FIFO interrupt generation disabled 1b: FIFO interrupt generation enabled
0	RFE	0h	RW	RX FIFO Enable 0b: FIFO disabled 1b: FIFO enabled

Note 1. These bits are not available in the classical CAN function.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0b, the RX FIFO is cleared to empty.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 000b (CFDRFCCn.RFDC[2:0] > 000b).

Set the RFE bit with a separate write access to the CFDRFCCn register, after all the other bits in the CFDRFCCn register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFFIE bit (RX FIFO Full Interrupt Enable)

The RFFIE bit enables generation of the RX FIFO full interrupt. Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

- Interruption output in number of arbitrary stages (CFDRFCCn.RFIGCV)
- Interruption output in FIFO full state.

Note: Management of the receiving data of FIFO can be performed by these notices of interruption.

7.9.2.2.27 RX FIFO Status Register n (CFDRFSTSn) (n = 0 to 7)

The CFDRFSTSn register (n = 0 to 7) shows the status of messages stored in the corresponding FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 00E0h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RFFIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								-	-	-	-	RFIF	RFMLT	RFFLL	RFEMP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	RFFIF	0h	RW	RX FIFO Full Interrupt Flag 0b: FIFO full interrupt condition not satisfied 1b: FIFO full interrupt condition satisfied
15 to 8	RFMC[7:0]	0h	R	RX FIFO Message Count Number of messages stored in FIFO
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	RFIF	0h	RW	RX FIFO Interrupt Flag 0b: FIFO interrupt condition not satisfied 1b: FIFO interrupt condition satisfied
2	RFMLT	0h	RW	RX FIFO Message Lost 0b: No message lost in FIFO 1b: FIFO message lost
1	RFFLL	0h	R	RX FIFO Full 0b: FIFO not full 1b: FIFO full
0	RFEMP	1h	R	RX FIFO Empty 0b: FIFO not empty 1b: FIFO empty

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0b
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0b
- The CAN-FD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth

- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0b
- The CAN-FD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0b to it. The bit is also cleared when CAN-FD module is in GL_RESET mode.

RFMC[7:0] bits (RX FIFO Message Count)

The RFMC[7:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU. These bits are cleared automatically when the FIFO is disabled and when the CAN-FD module is in GL_RESET mode.

RFFIF bit (RX FIFO Full Interrupt Flag)

The RFFIF bit is not cleared automatically when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the FIFO full interrupt condition is satisfied. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared by writing 0b to it.

The bit is also cleared when the CAN-FD module is in GL_RESET mode.

7.9.2.2.2 RX FIFO Pointer Control Register n (CFDRFPCTRn) (n = 0 to 7)

The CFDRFPCTRn register (n = 0 to 7) can be used to increment the read pointer of the corresponding RX FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0100h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RFPC[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	RFPC* ¹ [7:0]	0h	RW	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers

Note 1. These bits are not available in the classical CAN function.

RFPC[7:0] bits (RX FIFO Pointer Control)

When the value FFh is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write FFh to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 00h.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

7.9.2.2.29 Common FIFO Configuration/Control Register n (CFDCFCCn) (n = 0 to 17)

The CFDCFCCn register (n = 0 to 17) is used to configure the Common FIFOs.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0120h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFDC[2:0]			CFTML[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFGICV[2:0]			CFIM	CFITR	CFITSS	CFM[1:0]		-	CFPLS[2:0]			-	CFTXIE	CFRXIE	CFE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	CFITT[7:0]	0h	RW	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX or GW mode, delay is a multiple of basic Interval Timer Clock Source unit
23 to 21	CFDC[2:0]	0h	RW	Common FIFO Depth Configuration 000b: FIFO Depth = 0 messages 001b: FIFO Depth = 4 messages 010b: FIFO Depth = 8 messages 011b: FIFO Depth = 16 messages 100b: FIFO Depth = 32 messages 101b: FIFO Depth = 48 messages 110b: FIFO Depth = 64 messages 111b: FIFO Depth = 128 messages
20 to 16	CFTML[4:0]	0h	RW	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel
15 to 13	CFGICV[2:0]	0h	RW	Common FIFO Interrupt Generation Counter Value 000b: Interrupt generated when FIFO is 1/8th full 001b: Interrupt generated when FIFO is 1/4th full 010b: Interrupt generated when FIFO is 3/8th full 011b: Interrupt generated when FIFO is 1/2 full 100b: Interrupt generated when FIFO is 5/8th full 101b: Interrupt generated when FIFO is 3/4th full 110b: Interrupt generated when FIFO is 7/8th full 111b: Interrupt generated when FIFO is full
12	CFIM	0h	RW	Common FIFO Interrupt Mode 0b: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFGICV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully GW FIFO mode: For RX interrupt flag: Interrupt generated when FIFO counter increments and reaches the value configured in CFGICV For TX interrupt flag: Interrupt generated when FIFO transmits the last message successfully 1b: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: TX interrupt generated for every successfully transmitted message GW FIFO mode: For RX interrupt flag: Interrupt generated when a message is stored in the FIFO For TX interrupt flag: Interrupt generated when a message is successfully transmitted from the FIFO
11	CFITR	0h	RW	Common FIFO Interval Timer Resolution 0b: Reference clock period × 1 1b: Reference clock period × 10

Bit	Bit Name	Initial Value	R/W	Description
10	CFTSS	0h	RW	Common FIFO Interval Timer Source Select 0b: Reference clock ($\times 1 / \times 10$ period) 1b: Bit time clock of related channel (FIFO is linked to fixed channel)
9, 8	CFM[1:0]	0h	RW	Common FIFO Mode 00b: RX FIFO mode 01b: TX FIFO mode 10b: CAN – CAN GW FIFO mode 11b: Reserved
7	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
6 to 4	CFPLS* ¹ [2:0]	0h	RW	Common FIFO Payload Data Size Configuration 000b: 8 bytes 001b: 12 bytes 010b: 16 bytes 011b: 20 bytes 100b: 24 bytes 101b: 32 bytes 110b: 48 bytes 111b: 64 bytes
3	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	CCTXIE	0h	RW	Common FIFO TX Interrupt Enable 0b: FIFO interrupt generation disabled for Frame TX 1b: FIFO interrupt generation enabled for Frame TX
1	CFRXIE	0h	RW	Common FIFO RX Interrupt Enable 0b: FIFO interrupt generation disabled for Frame RX 1b: FIFO interrupt generation enabled for Frame RX
0	CFE	0h	RW	Common FIFO Enable 0b: FIFO disabled 1b: FIFO enabled

Note 1. These bits are not available in the classical CAN function.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode or GW mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.

This bit can only be set if the configured FIFO depth is greater than 0 (CFDC bit > 0).

Set the CFE bit with a separate write access to the CFDCFCCn register, after all the other bits in this register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CCTXIE bit (Common FIFO TX Interrupt Enable)

The CCTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see **7.9.6 FIFO Buffers and Normal Message Buffer Configuration**.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM[1:0] bits (Common FIFO Mode)

The CFM[1:0] bits select the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode. Do not configure these bits to 11b.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1b.

Do not write 1b to this bit when CAN-FD communication is used.*¹

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1b.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFTML[4:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[4:0] bits select the normal transmit message buffer position where the TX or GW FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX or GW mode. The delay is a multiple of the basic interval timer clock source period (reference clock $\times$ 1, reference clock $\times$ 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1b.

When CFDCCFG.ITRCP[15:0] = 0000h, set the CFITT[7:0] bits to 0000h.

7.9.2.2.30 Common FIFO Configuration/Control Enhancement Register n (CFDCFCCEn) (n = 0 to 17)

The CFDCFCCEn register (n = 0 to 17) is used to configure the Common FIFOs.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 0180h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFBME
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CFMO WM	-	-	-	-	-	CFOFT XIE	CFOFR XIE	CFFIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
16	CFBME	0h	RW	Common FIFO Buffering Mode Enable 0b: Transmission from Common FIFO 1b: Transmission halt from Common FIFO
15 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	CFMOWM	0h	RW	Common FIFO Message Overwrite Mode 0b: Message discarded mode 1b: Message overwrite mode
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	CFOFTXIE	0h	RW	Common FIFO One Frame Transmission Interrupt Enable 0b: One Frame TX interrupt generation disabled 1b: One Frame TX interrupt generation enabled
1	CFOFRXIE	0h	RW	Common FIFO One Frame Reception Interrupt Enable 0b: One Frame RX interrupt generation disabled 1b: One Frame RX interrupt generation enabled
0	CFFIE	0h	RW	Common FIFO Full Interrupt Enable 0b: FIFO Interrupt generation disabled 1b: FIFO Interrupt generation enabled

CFFIE bit (Common FIFO Full Interrupt Enable)

The CFFIE bit enables generation of the FIFO full interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

1. Interruption output in number of arbitrary stages (CFDCFCCEn.CFIGCV).
2. Interruption output in FIFO full state.

Management of the receiving data of FIFO can be performed by these notices of interruption.

CFOFRXIE bit (Common FIFO One Frame Reception Interrupt Enable)

The CFOFRXIE bit enables generation of the one frame reception interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFOFTXIE bit (Common FIFO One Frame Transmission Interrupt Enable)

The CFOFTXIE bit enables generation of the one frame transmission interrupt when the interrupt flag is set after transmission of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFMOWM bit (Common FIFO Message Overwrite Mode)

When the CFMOWM bit is 0b, a receiving message is discarded and FIFO is full. When the CFMOWM bit is 1b, a receiving message is overwritten and FIFO is full.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write 1b to this bit when the common FIFO is in GW mode.

Do not write 1b to this bit when the CFE bit is 1b.

CFBME bit (Common FIFO Buffering Mode Enable)

When the CFBME bit is 0b, messages are transmitted from FIFO. When the CFBME bit is 1b, messages are not transmitted from FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Additionally, do not write 1b to this bit when the CFE bit is 1b.

7.9.2.2.31 Common FIFO Status Register n (CFDCFSTSn) (n = 0 to 17)

The CFDCFSTSn register (n = 0 to 17) shows the status of messages stored in the corresponding FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 01E0h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	CFMO W	-	-	-	-	-	CFOFT XIF	CFOFR XIF	CFEIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								-	-	-	CFTXIF	CFRXIF	CFMLT	CFLL	CFEMP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
24	CFMOW	0h	RW	Common FIFO Message Overwrite 0b: No message overwrite occurred in FIFO 1b: Message overwrite occurred in FIFO
23 to 19	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18	CFOFTXIF	0h	RW	Common FIFO One Frame Transmission Interrupt Flag For each FIFO that transmits a frame, a corresponding interrupt is set.
17	CFOFRXIF	0h	RW	Common FIFO One Frame Reception Interrupt Flag For each FIFO that receives a frame, a corresponding interrupt is set.
16	CFEIF	0h	RW	Common FIFO Full Interrupt Flag 0b: Interrupt condition not satisfied for FIFO full interrupt 1b: Interrupt condition satisfied for FIFO full interrupt
15 to 8	CFMC[7:0]	0h	R	Common FIFO Message Count Number of messages stored in FIFO
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	CFTXIF	0h	RW	Common TX FIFO Interrupt Flag 0b: FIFO interrupt condition not satisfied after frame transmission 1b: FIFO interrupt condition satisfied after frame transmission
3	CFRXIF	0h	RW	Common RX FIFO Interrupt Flag 0b: FIFO interrupt condition not satisfied after frame reception 1b: FIFO interrupt condition satisfied after frame reception
2	CFMLT	0h	RW	Common FIFO Message Lost 0b: FIFO message not lost 1b: FIFO message lost
1	CFLL	0h	R	Common FIFO Full 0b: FIFO not full 1b: FIFO full
0	CFEMP	1h	R	Common FIFO Empty 0b: FIFO not empty 1b: FIFO empty

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX or GW mode
- The FIFO is disabled by setting the CFE bit to 0b

- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET when FIFO configured in TX or GW mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX or GW mode.

CFLL bit (Common FIFO Full)

The CFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0b
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode when FIFO buffer is configured in TX or GW mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX or GW mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

The CFMLT bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in GW mode or RX mode.

The CFRXIF bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in GW mode.

CCTXIF bit (Common TX FIFO Interrupt Flag)

The CCTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in GW or TX mode.

The CCTXIF bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFMC[7:0] bits (Common FIFO Message Count)

The CFMC[7:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CAN-FD to be read by the CPU
- Number of CAN messages stored by the CAN-FD in the GW FIFO pending for transmission.

The CFMC[7:0] bits are cleared automatically when:

- The FIFO is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFFIF bit (Common FIFO Full Interrupt Flag)

The CFFIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the FIFO Full Interrupt condition is satisfied for Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFFIF bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFOFRXIF bit (Common FIFO One Frame Reception Interrupt Flag)

The CFOFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for the FIFO buffer configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the One Frame Reception Interrupt condition is satisfied for the Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFOFRXIF bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

This bit is not affected by the CFIM bit of CFDCFCn.

CFOFTXIF bit (Common FIFO One Frame Transmission Interrupt Flag)

The CFOFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is set automatically when the One Frame Transmission Interrupt condition is satisfied for Common FIFO buffers configured in GW mode or TX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

This bit is not affected by the CFIM bit of CFDCFCn.

CFMOW bit (Common FIFO Message Overwrite)

The CFMOW bit is set automatically whenever a message is an overwrite storage of a new message when CFDCFCEn.CFMOWM = 1b and FIFO is already full in GW mode.

If the set from the CAN channel occurs simultaneously with the clear by a write access, the bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

The CFMOW bit is cleared:

- By writing 0b to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

7.9.2.2.32 Common FIFO Pointer Control Register n (CFDCFPCTRn) (n = 0 to 17)

The CFDCFPCTRn register (n = 0 to 17) can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0240h + n x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	CFPC[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	
Bit	Bit Name		Initial Value		R/W		Description										
31 to 8	-		All 0		W		Reserved The written value should always be 0b.										
7 to 0	CFPC[7:0]		0h		W		Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.										

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value FFh is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 00h.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Only write FFh to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode
- The Common FIFO buffer is enabled and is not configured in GW mode.

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

7.9.2.2.33 FIFO Empty Status Register (CFDFESTS)

The CFDFESTS register shows status of the empty bits of the FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02A0h														
Initial Value :		0000_3FFFh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	CFXEMP[17:8]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXEMP[7:0]								RFXEMP[7:0]							
Initial Value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read.
25 to 8	CFXEMP[17:0]	3Fh	R	Common FIFO Empty Status 0b: Corresponding FIFO not empty 1b: Corresponding FIFO empty
7 to 0	RFXEMP[7:0]	FFh	R	RX FIFO Empty Status 0b: Corresponding FIFO not empty 1b: Corresponding FIFO empty

RFXEMP[7:0] bits (RX FIFO Empty Status)

Bit [7] (RFXEMP[7]) is associated with FIFO index 7 and bit [0] (RFXEMP[0]) is associated with FIFO index 0.

The RFXEMP[7:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXEMP[17:0] bits (Common FIFO Empty Status)

Bit [25] (CFXEMP[17]) is associated with common FIFO index 17 and bit [8] (CFXEMP[0]) is associated with common FIFO index 0.

The CFXEMP[17:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

7.9.2.2.34 FIFO Full Status Register (CFDFFSTS)

The CFDFFSTS register shows status of the full bits of the FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02A4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	CFXFLL[17:8]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXFLL[7:0]								RFXFLL[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read.
25 to 8	CFXFLL[17:0]	0h	R	Common FIFO Full Status 0b: Corresponding FIFO not full 1b: Corresponding FIFO full
7 to 0	RFXFLL[7:0]	0h	R	RX FIFO Full Status 0b: Corresponding FIFO not full 1b: Corresponding FIFO full

RFXFL[7:0] bits (RX FIFO Full Status)

The RFXFL[7:0] bits are cleared when CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXFL[17:0] bits (Common FIFO Full Status)

The CFXFL[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

7.9.2.2.35 FIFO Message Lost Status Register (CFDFMSTS)

The CFDFMSTS register shows status of the message lost bits of the FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02A8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	CFXMLT[17:8]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXMLT[7:0]								RFXMLT[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read.
25 to 8	CFXMLT[17:0]	0h	R	Common FIFO Message Lost Status 0b: Corresponding FIFO Message Lost flag not set 1b: Corresponding FIFO Message Lost flag set
7 to 0	RFXMLT[7:0]	0h	R	RX FIFO Message Lost Status 0b: Corresponding FIFO Message Lost flag not set 1b: Corresponding FIFO Message Lost flag set

RFXMLT[7:0] bits (RX FIFO Message Lost Status)

The RFXMLT[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXMLT[17:0] bits (Common FIFO Message Lost Status)

The CFXMLT[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

7.9.2.2.36 RX FIFO Interrupt Flag Status Register (CFDRFISTS)

The CFDRFISTS register shows status of the interrupt flag bits of the RX FIFO buffers.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 02ACh																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	RFXFFLL[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	RFXIF[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read.
23 to 16	RFXFFLL[7:0]	0h	R	RX FIFO[x] Interrupt Full Flag Status 0b: Corresponding RX FIFO Interrupt Full flag not set 1b: Corresponding RX FIFO Interrupt Full flag set
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read.
7 to 0	RFXIF[7:0]	0h	R	RX FIFO[x] Interrupt Flag Status 0b: Corresponding RX FIFO Interrupt flag not set 1b: Corresponding RX FIFO Interrupt flag set

RFXIF[7:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

RFXFFLL[7:0] bits (RX FIFO[x] Interrupt Full Flag Status)

Each bit is set automatically when the corresponding interrupt full flag bit is set in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt full flag bit is cleared in the RX FIFO Status Registers.

7.9.2.2.37 Common FIFO RX Interrupt Flag Status Register (CFDCFRISTS)

The CFDCFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02B0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFXRXIF[17:16]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXRXIF[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	31 to 18		Bit Name	Initial Value	R/W	Description										
			-	All 0	R	Reserved Whenever it is read, 0b is read.										
	17 to 0		CFXRXIF[17:0]	0h	R	Common FIFO RX Interrupt Flag Status 0b: Corresponding Common FIFO RX Interrupt flag not set 1b: Corresponding Common FIFO RX Interrupt flag set										

CFXRXIF[17:0] bits (Common FIFO RX Interrupt Flag Status)

Each bit is set automatically when the corresponding RX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXRXIF[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding RX interrupt flag bit is cleared in the Common FIFO Status Registers.

7.9.2.2.38 Common FIFO TX Interrupt Flag Status Register (CFDCFTISTS)

The CFDCFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02B4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFXTXIF[17:16]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXTXIF[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read.
17 to 0	CFXTXIF[17:0]	0h	R	Common FIFO TX Interrupt Flag Status 0b: Corresponding Common FIFO TX Interrupt flag not set 1b: Corresponding Common FIFO TX Interrupt flag set

CFXTXIF[17:0] bits (Common FIFO TX Interrupt Flag Status)

Each bit is set automatically when the corresponding TX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXTXIF[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding TX interrupt flag bit is cleared in the Common FIFO Status Registers.

7.9.2.2.39 FIFO FDC Full Status Register (CFDFFFSTS)

The CFDFFFSTS register shows status of the full interrupt flag bits of the FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02C4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	CFXFFLL[17:8]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXFFLL[7:0]								RFXFFLL[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read.
25 to 8	CFXFFLL[17:0]	0h	R	COMMON FIFO FDC Level Full Status 0b: Corresponding FIFO full interrupt not set 1b: Corresponding FIFO full interrupt set
7 to 0	RFXFFLL[7:0]	0h	R	RX FIFO FDC Level Full Status 0b: Corresponding FIFO full interrupt not set 1b: Corresponding FIFO full interrupt set

RFXFFLL[7:0] bits (RX FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

CFXFFLL[17:0] bits (COMMON FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

The CFXFFLL[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

7.9.2.2.40 Common FIFO Message Overwrite Status Register (CFDCFMOWSTS)

The CFDCFMOWSTS register shows the status of the Interrupt Flag bits of the Common FIFO Buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02C0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFXMOW[17:16]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXMOW[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read.
17 to 0	CFXMOW[17:0]	0h	R	Common FIFO Message Overwrite Status 0b: Corresponding FIFO Overwrite flag not set 1b: Corresponding FIFO Overwrite flag set

CFXMOW[17:0] bits (Common FIFO Message Overwrite Status)

The CFXMOW[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode. This register is only valid in GW mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

7.9.2.2.41 Common FIFO One Frame RX Interrupt Flag Status Register (CFDCFOFRISTS)

The CFDCFOFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02B8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFXOFRXIF [17:16]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXOFRXIF[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read.
17 to 0	CFXOFRXIF [17:0]	0h	R	Common FIFO One Frame RX Interrupt Flag Status 0b: Corresponding Common FIFO One Frame RX Interrupt flag not set 1b: Corresponding Common FIFO One Frame RX Interrupt flag set

CFXOFRXIF[17:0] bits (Common FIFO One Frame RX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame RX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFRXIF[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame RX Interrupt flag bit is cleared in the Common FIFO Status Registers.

7.9.2.2.42 Common FIFO One Frame TX Interrupt Flag Status Register (CFDCFOFTISTS)

The CFDCFOFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 02BCh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CFXOFTXIF [17:16]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFXOFTXIF[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read.
17 to 0	CFXOFTXIF [17:0]	0h	R	Common FIFO One Frame TX Interrupt Flag Status 0b: Corresponding Common FIFO One Frame TX Interrupt flag not set 1b: Corresponding Common FIFO One Frame TX Interrupt flag set

CFXOFTXIF[17:0] bits (Common FIFO One Frame TX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame TX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFTXIF[17:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame TX Interrupt flag bit is cleared in the Common FIFO Status Registers.

7.9.2.2.43 DMA Transfer Control Register (CFDCDTCT)

The CFDCDTCT register controls the start and stop of DMA transfer operation.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1330h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CFDM AE5	CFDM AE4	CFDM AE3	CFDM AE2	CFDM AE1	CFDM AE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13	CFDMAE5	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 5 0b: DMA transfer request disabled 1b: DMA transfer request enabled
12	CFDMAE4	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 4 0b: DMA transfer request disabled 1b: DMA transfer request enabled
11	CFDMAE3	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 3 0b: DMA transfer request disabled 1b: DMA transfer request enabled
10	CFDMAE2	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 2 0b: DMA transfer request disabled 1b: DMA transfer request enabled
9	CFDMAE1	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 1 0b: DMA transfer request disabled 1b: DMA transfer request enabled
8	CFDMAE0	0h	RW	DMA Transfer Enable for Common FIFO 0 of Channel 0 0b: DMA transfer request disabled 1b: DMA transfer request enabled
7	RFDMAE7	0h	RW	DMA Transfer Enable for RX FIFO 7 0b: DMA transfer request disabled 1b: DMA transfer request enabled
6	RFDMAE6	0h	RW	DMA Transfer Enable for RX FIFO 6 0b: DMA transfer request disabled 1b: DMA transfer request enabled
5	RFDMAE5	0h	RW	DMA Transfer Enable for RX FIFO 5 0b: DMA transfer request disabled 1b: DMA transfer request enabled
4	RFDMAE4	0h	RW	DMA Transfer Enable for RX FIFO 4 0b: DMA transfer request disabled 1b: DMA transfer request enabled
3	RFDMAE3	0h	RW	DMA Transfer Enable for RX FIFO 3 0b: DMA transfer request disabled 1b: DMA transfer request enabled
2	RFDMAE2	0h	RW	DMA Transfer Enable for RX FIFO 2 0b: DMA transfer request disabled 1b: DMA transfer request enabled
1	RFDMAE1	0h	RW	DMA Transfer Enable for RX FIFO 1 0b: DMA transfer request disabled 1b: DMA transfer request enabled

Bit	Bit Name	Initial Value	R/W	Description
0	RFDMAEn	0h	RW	DMA Transfer Enable for RX FIFO 0 0b: DMA transfer request disabled 1b: DMA transfer request enabled

RFDMAEn (n = 0 to 7) bit (DMA Transfer Enable for RX FIFO n)

Number of RX FIFOs = 8

The RFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0 to 5) bit (DMA Transfer Enable for Common FIFO 0 of Channels 0 to 5)

The CFDMAEn bit enables or disables DMA transfer request for common FIFO 0 of channels 0 to 5. Only Common FIFO 0 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 2, see bit CFDCDTTCT.CFDMAEn in **7.9.2.2.45 DMA TX Transfer Control Register (CFDCDTTCT)**.

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX or GW FIFO.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

7.9.2.2.44 DMA Transfer Status Register (CFDCDTSTS)

The CFDCDTSTS register shows the status of the DMA transfer.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1334h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CFDM ASTS5	CFDM ASTS4	CFDM ASTS3	CFDM ASTS2	CFDM ASTS1	CFDM ASTS0	RFDMA ASTS7	RFDMA ASTS6	RFDMA ASTS5	RFDMA ASTS4	RFDMA ASTS3	RFDMA ASTS2	RFDMA ASTS1	RFDMA ASTS0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read.
13	CFDMASTS5	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 5 0b: DMA transfer stopped 1b: DMA transfer ongoing
12	CFDMASTS4	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 4 0b: DMA transfer stopped 1b: DMA transfer ongoing
11	CFDMASTS3	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 3 0b: DMA transfer stopped 1b: DMA transfer ongoing
10	CFDMASTS2	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 2 0b: DMA transfer stopped 1b: DMA transfer ongoing
9	CFDMASTS1	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 1 0b: DMA transfer stopped 1b: DMA transfer ongoing
8	CFDMASTS0	0h	R	DMA Transfer Status only for Common FIFO 0 of Channel 0 0b: DMA transfer stopped 1b: DMA transfer ongoing
7	RFDMASTS7	0h	R	DMA Transfer Status for RX FIFO 7 0b: DMA transfer stopped 1b: DMA transfer ongoing
6	RFDMASTS6	0h	R	DMA Transfer Status for RX FIFO 6 0b: DMA transfer stopped 1b: DMA transfer ongoing
5	RFDMASTS5	0h	R	DMA Transfer Status for RX FIFO 5 0b: DMA transfer stopped 1b: DMA transfer ongoing
4	RFDMASTS4	0h	R	DMA Transfer Status for RX FIFO 4 0b: DMA transfer stopped 1b: DMA transfer ongoing
3	RFDMASTS3	0h	R	DMA Transfer Status for RX FIFO 3 0b: DMA transfer stopped 1b: DMA transfer ongoing
2	RFDMASTS2	0h	R	DMA Transfer Status for RX FIFO 2 0b: DMA transfer stopped 1b: DMA transfer ongoing
1	RFDMASTS1	0h	R	DMA Transfer Status for RX FIFO 1 0b: DMA transfer stopped 1b: DMA transfer ongoing

Bit	Bit Name	Initial Value	R/W	Description
0	RFDMASTS0	0h	R	DMA Transfer Status for RX FIFO 0 0b: DMA transfer stopped 1b: DMA transfer ongoing

RFDMASTS_n (n = 0 to 7) bit (DMA Transfer Status for RX FIFO n)

Number of RX FIFOs = 8

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEn (see CFDCDTCT.RFDMAEn bit in **7.9.2.2.43 DMA Transfer Control Register (CFDCDTCT)**) is set to 0b while DMA transfer for the corresponding FIFO is ongoing, the RFDMASTS_n bit becomes 0b when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMASTS_n bit (DMA Transfer Status only for Common FIFO 0 of Channels 0 to 5)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAEn (see CFDCDTCT.CFDMAEn bit in **7.9.2.2.43 DMA Transfer Control Register (CFDCDTCT)**) is set to 0b while DMA transfer for the corresponding FIFO is ongoing, the CFDMASTS_n bit becomes 0b when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

7.9.2.2.45 DMA TX Transfer Control Register (CFDCDTTCT)

The CFDCDTTCT register controls the start and stop of DMA transfer operation.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1340h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	CFDM AE5	CFDM AE4	CFDM AE3	CFDM AE2	CFDM AE1	CFDM AE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TQ3D MAE5	TQ3D MAE4	TQ3D MAE3	TQ3D MAE2	TQ3D MAE1	TQ3D MAE0	-	-	TQ0D MAE5	TQ0D MAE4	TQ0D MAE3	TQ0D MAE2	TQ0D MAE1	TQ0D MAE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
21	CFDMAE5	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 5 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
20	CFDMAE4	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 4 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
19	CFDMAE3	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 3 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
18	CFDMAE2	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 2 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
17	CFDMAE1	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 1 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
16	CFDMAE0	0h	RW	DMA TX Transfer Enable for Common FIFO 2 of Channel 0 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
15 to 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13	TQ3DMAE5	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 5 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
12	TQ3DMAE4	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 4 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
11	TQ3DMAE3	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 3 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
10	TQ3DMAE2	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 2 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
9	TQ3DMAE1	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 1 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
8	TQ3DMAE0	0h	RW	DMA TX Transfer Enable for TXQ 3 of Channel 0 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
7 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

Bit	Bit Name	Initial Value	R/W	Description
5	TQ0DMAE5	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 5 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
4	TQ0DMAE4	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 4 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
3	TQ0DMAE3	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 3 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
2	TQ0DMAE2	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 2 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
1	TQ0DMAE1	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 1 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled
0	TQ0DMAE0	0h	RW	DMA TX Transfer Enable for TXQ 0 of Channel 0 0b: DMA TX transfer request disabled 1b: DMA TX transfer request enabled

TQ0DMAEn (n = 0 to 5) bit (DMA TX Transfer Enable for TXQ 0 of Channel n)

The TQ0DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

TQ3DMAEn (n = 0 to 5) bit (DMA TX Transfer Enable for TXQ 3 of Channel n)

The TQ3DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0 to 5) bit (DMA TX Transfer Enable for Common FIFO 2 of Channel n)

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Only common FIFO 2 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel. To link Common FIFO 0, see CFDCDTCT.CFDMAEn bit in **7.9.2.2.43 DMA Transfer Control Register (CFDCDTCT)**.

Do not enable a DMA transfer for a Common FIFO that is configured as RX or GW FIFO.

The CFDMAEn bit is cleared when the CAN-FD module is in GL_RESET mode.

7.9.2.2.46 DMA TX Transfer Status Register (CFDCDTTSTS)

The CFDCDTTSTS register shows the status of the DMA TX transfer.

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1344h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	CFDM ASTS5	CFDM ASTS4	CFDM ASTS3	CFDM ASTS2	CFDM ASTS1	CFDM ASTS0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TQ3D MAST S5	TQ3D MAST S4	TQ3D MAST S3	TQ3D MAST S2	TQ3D MAST S1	TQ3D MAST S0	-	-	TQ0D MAST S5	TQ0D MAST S4	TQ0D MAST S3	TQ0D MAST S2	TQ0D MAST S1	TQ0D MAST S0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	-	All 0	R	Reserved Whenever it is read, 0b is read.
21	CFDMASTS5	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 5 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
20	CFDMASTS4	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 4 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
19	CFDMASTS3	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 3 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
18	CFDMASTS2	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 2 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
17	CFDMASTS1	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 1 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
16	CFDMASTS0	0h	R	DMA TX Transfer Status for Common FIFO 2 of Channel 0 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
15 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read.
13	TQ3DMASTS5	0h	R	DMA TX Transfer Status for TXQ3 of Channel 5 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
12	TQ3DMASTS4	0h	R	DMA TX Transfer Status for TXQ3 of Channel 4 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
11	TQ3DMASTS3	0h	R	DMA TX Transfer Status for TXQ3 of Channel 3 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
10	TQ3DMASTS2	0h	R	DMA TX Transfer Status for TXQ3 of Channel 2 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
9	TQ3DMASTS1	0h	R	DMA TX Transfer Status for TXQ3 of Channel 1 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
8	TQ3DMASTS0	0h	R	DMA TX Transfer Status for TXQ3 of Channel 0 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
7 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read.

Bit	Bit Name	Initial Value	R/W	Description
5	TQ0DMASTS5	0h	R	DMA TX Transfer Status for TXQ0 of Channel 5 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
4	TQ0DMASTS4	0h	R	DMA TX Transfer Status for TXQ0 of Channel 4 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
3	TQ0DMASTS3	0h	R	DMA TX Transfer Status for TXQ0 of Channel 3 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
2	TQ0DMASTS2	0h	R	DMA TX Transfer Status for TXQ0 of Channel 2 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
1	TQ0DMASTS1	0h	R	DMA TX Transfer Status for TXQ0 of Channel 1 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled
0	TQ0DMASTS0	0h	R	DMA TX Transfer Status for TXQ0 of Channel 0 0b: DMA TX transfer stopped 1b: DMA TX transfer enabled

TQ0DMASTS_n (n = 0 to 5) bit (DMA TX Transfer Status for TXQ 0 of Channel n)

The TQ0DMASTS_n bit is set when the CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is set (see **7.9.2.2.45 DMA TX Transfer Control Register (CFDCDTTCT)**).

This bit is cleared when:

- The CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

TQ3DMASTS_n (n = 0 to 5) bit (DMA TX Transfer Status for TXQ 3 of Channel n)

The TQ3DMASTS_n bit is set when the CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is set (see **7.9.2.2.45 DMA TX Transfer Control Register (CFDCDTTCT)**).

This bit is cleared when:

- The CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

CFDMASTS_n (n = 0 to 5) bit (DMA TX Transfer Status only for Common FIFO 2 of Channel n)

The CFDMASTS_n bit is set when the CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is set (see **7.9.2.2.45 DMA TX Transfer Control Register (CFDCDTTCT)**).

This bit is cleared when:

- The CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

7.9.2.2.47 Global RX Interrupt Status Register n (CFDGRINTSTSn) (n = 0 to 5)

Access Size : 8, 16, 32 bits

Address : <CFD_base> + 1350h + n x 0004h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	CFOFRIF[2:0]			-	CFRFIF[2:0]			-	-	-	-	-	CFRIF[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	BQOFRIF[1:0]		-	QOFRIF[2:0]			-	-	BQFIF[1:0]		-	QFIF[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read.
30 to 28	CFOFRIF[2:0]	0h	R	Common FIFO One Frame RX Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding Common FIFO One Frame RX Interrupt flag is not set 1b: Corresponding Common FIFO One Frame RX Interrupt flag is set
27	-	0h	R	Reserved Whenever it is read, 0b is read.
26 to 24	CFRFIF[2:0]	0h	R	Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding Common FIFO Full Interrupt flag is not set 1b: Corresponding Common FIFO Full Interrupt flag is set
23 to 19	-	All 0	R	Reserved Whenever it is read, 0b is read.
18 to 16	CFRIF[2:0]	0h	R	Common FIFO RX Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding Common FIFO RX Interrupt flag is not set 1b: Corresponding Common FIFO RX Interrupt flag is set
15 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read.
13 to 12	BQOFRIF[1:0]	0h	R	Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding TXQ One Frame RX Interrupt flag is not set 1b: Corresponding TXQ One Frame RX Interrupt flag is set
11	-	0h	R	Reserved Whenever it is read, 0b is read.
10 to 8	QOFRIF[2:0]	0h	R	TXQ One Frame RX Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding TXQ One Frame RX Interrupt flag is not set 1b: Corresponding TXQ One Frame RX Interrupt flag is set
7 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read.
5 to 4	BQFIF[1:0]	0h	R	Borrowed TXQ Full Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding TXQ Full Interrupt flag is not set 1b: Corresponding TXQ Full Interrupt flag is set
3	-	0h	R	Reserved Whenever it is read, 0b is read.
2 to 0	QFIF[2:0]	0h	R	TXQ Full Interrupt Flag Channel n (n = 0 to 5) 0b: Corresponding TXQ Full Interrupt flag is not set 1b: Corresponding TXQ Full Interrupt flag is set

QFIF[2:0] bits (TXQ Full Interrupt Flag Channel n (n = 0 to 5))

The QFIF[2:0] bits are set automatically when the TXQ Full Interrupt flag of the related channel is set when the interrupt is enabled.

The QFIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled

- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

BQFIF[1:0] bits (Borrowed TXQ Full Interrupt Flag Channel n (n = 0 to 5))

The BQFIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ is in full status. Operation is the same as CFDGRINTSTSn.QFIF.

The bit of the channel that lends TXMB is a reserved bit.

QOFRIF[2:0] bits (TXQ One Frame RX Interrupt Flag Channel n (n = 0 to 5))

The QOFRIF[2:0] bits are set automatically when the TXQ One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The QOFRIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

BQOFRIF[1:0] bits (Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0 to 5))

The BQOFRIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ receives one frame. Operation is the same as CFDGRINTSTSn.QOFRIF.

The bit of the channel that lends TXMB is a reserved bit.

CFRIF[2:0] bits (Common FIFO RX Interrupt Flag Channel n (n = 0 to 5))

The CFRIF[2:0] bits are set automatically when the Common FIFO RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

CFRFIF[2:0] bits (Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0 to 5))

The CFRFIF[2:0] bits are set automatically when the Common FIFO Full Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRFIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

CFOFRIF[2:0] bits (Common FIFO One Frame RX Interrupt Flag Channel n (n = 0 to 5))

The CFOFRIF[2:0] bits are set automatically when the Common FIFO One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFOFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

7.9.2.2.48 TX Message Buffer Control Register n (CFDTMCn) (n = 0 to 383)

The CFDTMCn register (n = 0 to 383) configures the TX message buffer functions.

Access Size :		8 bits						
Address :		<CFD_base> + 02D0h + n x 0001h						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	TMOM	TMTAR	TMTR
Initial Value	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TMOM	0h	RW	TX Message Buffer One-shot Mode 0b: TX message buffer not configured in one-shot mode 1b: TX message buffer configured in one-shot mode
1	TMTAR	0h	RW	TX Message Buffer Transmission Abort Request 0b: TX message buffer transmission request abort not requested 1b: TX message buffer transmission request abort requested
0	TMTR	0h	RW	TX Message Buffer Transmission Request 0b: TX Message buffer transmission not requested 1b: TX message buffer transmission requested

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CAN-FD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CAN-FD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX or GW mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the CFDTMSTSn register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CAN-FD module logic at the end of a successful transmission
- CAN-FD module logic at the end of a transmission abort, requested by the corresponding CFDTMCn.TMTAR bit
- CAN-FD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCn.TMOM bit is set for the message buffer
- CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CAN-FD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CAN-FD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CAN-FD module logic at the end of a successful transmission
- The CAN-FD module logic at the end of a transmission abort
- The CAN-FD module logic when there is detection of a CAN bus error or arbitration loss
- The CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CAN-FD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSn.TMTRF[1:0] bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

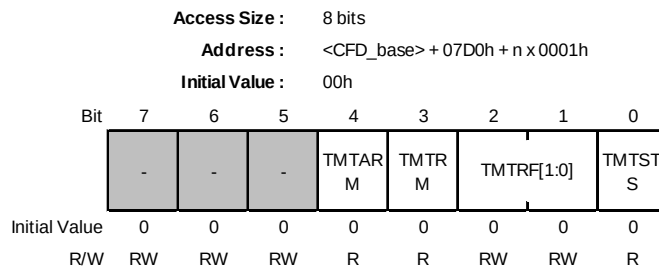
Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

7.9.2.2.49 TX Message Buffer Status Register n (CFDTMSTSn) (n = 0 to 383)

The CFDTMSTSn register (n = 0 to 383) shows status of the transmission and transmission abort for the corresponding message buffers.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4	TMTARM	0h	R	TX Message Buffer Transmission Abort Request Mirrored 0b: TX message buffer transmission request abort not requested 1b: TX message buffer transmission request abort requested
3	TMTRM	0h	R	TX Message Buffer Transmission Request Mirrored 0b: TX message buffer transmission not requested 1b: TX message buffer transmission requested
2, 1	TMTRF[1:0]	0h	RW	TX Message Buffer Transmission Result Flag 00b: No result 01b: Transmission aborted from the TX message buffer 10b: Transmission successful from the TX message buffer and transmission abort was not requested 11b: Transmission successful from the TX message buffer and transmission abort was requested
0	TMTSTS	0h	R	TX Message Buffer Transmission Status 0b: No on-going transmission 1b: On-going transmission

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00b: Transmission in progress or has not been requested
- 01b: Transmission has been aborted from the corresponding TX message buffer
- 10b: Transmission was successful from the corresponding TX message buffer and the CFDTMCn.TMTAR bit was not set for this TX message buffer
- 11b: Transmission was successful from the corresponding TX message buffer, but the CFDTMCn.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

The TMTARM bit is set when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is cleared.

7.9.2.2.50 TX Message Buffer Transmission Request Status Register f (CFDTMTRSTSf) (f = 0 to 11)

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0CD0h + f x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TMTRSTS[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	TMTRSTS[15:0]	0h	R	TX Message Buffer Transmission Request Status 0b: Transmission not requested for corresponding TX message buffer 1b: Transmission requested for corresponding TX message buffer

TMTRSTS[15:0] bits (TX Message Buffer Transmission Request Status)

The TMTRSTS[15:0] bits show status of the CFDTMCn.TMTR bits of the TX Message Buffer Control Registers.

Alignment of the TMTRSTS[15:0] bits is shown in **Table 7.9-7**.

Table 7.9-7 Alignment of TMTRSTS[15:0] Mirror Bits

Bit Position	TX Message Buffer Number
$n \times 64 - f_{min} \times 32$	$n \times 64 + 0$
$n \times 64 + 1 - f_{min} \times 32$	$n \times 64 + 1$
⋮	⋮
$n \times 64 + 15 - f_{min} \times 32$	$n \times 64 + 15$
$n \times 64 + 32 - f_{max} \times 32$	$n \times 64 + 32$
⋮	⋮
$n \times 64 + 46 - f_{max} \times 32$	$n \times 64 + 46$
$n \times 64 + 47 - f_{max} \times 32$	$n \times 64 + 47$

Note: When $n = 0$, $f_{min} = 0$, $f_{max} = 1$
 When $n = 1$, $f_{min} = 2$, $f_{max} = 3$
 When $n = 2$, $f_{min} = 4$, $f_{max} = 5$
 When $n = 3$, $f_{min} = 6$, $f_{max} = 7$
 When $n = 4$, $f_{min} = 8$, $f_{max} = 9$
 When $n = 5$, $f_{min} = 10$, $f_{max} = 11$

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCn), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

7.9.2.2.51 TX Message Buffer Transmission Abort Request Status Register f (CFD TMTARSTsf) (f = 0 to 11)

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0D70h + f x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TMTARSTS[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	TMTARSTS [15:0]	0h	R	TX Message Buffer Transmission Abort Request Status 0b: Transmission abort not requested for corresponding TX message buffer 1b: Transmission abort requested for corresponding TX message buffer

TMTARSTS[15:0] bits (TX Message Buffer Transmission Abort Request Status)

The TMTARSTS[15:0] bits show status of the CFDTMCn.TMTAR bits of the TX Message Buffer Control Registers. Alignment of the TMTARSTS[15:0] bits is shown in **Table 7.9-8**.

Table 7.9-8 Alignment of TMTARSTS[15:0] Mirror Bits

Bit Position	TX Message Buffer Number
$n \times 64 - f_{min} \times 32$	$n \times 64 + 0$
$n \times 64 + 1 - f_{min} \times 32$	$n \times 64 + 1$
⋮	⋮
$n \times 64 + 15 - f_{min} \times 32$	$n \times 64 + 15$
$n \times 64 + 32 - f_{max} \times 32$	$n \times 64 + 32$
⋮	⋮
$n \times 64 + 46 - f_{max} \times 32$	$n \times 64 + 46$
$n \times 64 + 47 - f_{max} \times 32$	$n \times 64 + 47$

Note: When $n = 0$, $f_{min} = 0$, $f_{max} = 1$
 When $n = 1$, $f_{min} = 2$, $f_{max} = 3$
 When $n = 2$, $f_{min} = 4$, $f_{max} = 5$
 When $n = 3$, $f_{min} = 6$, $f_{max} = 7$
 When $n = 4$, $f_{min} = 8$, $f_{max} = 9$
 When $n = 5$, $f_{min} = 10$, $f_{max} = 11$

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

7.9.2.2.52 TX Message Buffer Transmission Completion Status Register f (CFDTMTCSTsf) (f = 0 to 11)

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0E10h + f x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TMTCSTS[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	TMTCSTS [15:0]	0h	R	TX Message Buffer Transmission Completion Status 0b: Transmission not completed for corresponding TX message buffer 1b: Transmission completed for corresponding TX message buffer

TMTCSTS[15:0] bits (TX Message Buffer Transmission Completion Status)

The TMTCSTS[15:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Alignment of the TMTCSTS[15:0] bits is shown in **Table 7.9-9**.

Table 7.9-9 Alignment of TMTCSTS[15:0] Mirror Bits

Bit Position	TX Message Buffer Number
$n \times 64 - f_{min} \times 32$	$n \times 64 + 0$
$n \times 64 + 1 - f_{min} \times 32$	$n \times 64 + 1$
⋮	⋮
$n \times 64 + 15 - f_{min} \times 32$	$n \times 64 + 15$
$n \times 64 + 32 - f_{max} \times 32$	$n \times 64 + 32$
⋮	⋮
$n \times 64 + 46 - f_{max} \times 32$	$n \times 64 + 46$
$n \times 64 + 47 - f_{max} \times 32$	$n \times 64 + 47$

Note: When $n = 0$, $f_{min} = 0$, $f_{max} = 1$
 When $n = 1$, $f_{min} = 2$, $f_{max} = 3$
 When $n = 2$, $f_{min} = 4$, $f_{max} = 5$
 When $n = 3$, $f_{min} = 6$, $f_{max} = 7$
 When $n = 4$, $f_{min} = 8$, $f_{max} = 9$
 When $n = 5$, $f_{min} = 10$, $f_{max} = 11$

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

7.9.2.2.53 TX Message Buffer Transmission Abort Status Register f (CFD TMTASTSf) (f = 0 to 11)

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0EB0h + f x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TMTASTS[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	TMTASTS [15:0]	0h	R	TX Message Buffer Transmission Abort Status 0b: Transmission not aborted for corresponding TX message buffer 1b: Transmission aborted for corresponding TX message buffer

TMTASTS[15:0] bits (TX Message Buffer Transmission Abort Status)

The TMTASTS[15:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Alignment of the TMTASTS[15:0] bits is shown in **Table 7.9-10**.

Table 7.9-10 Alignment of TMTASTS[15:0] Mirror Bits

Bit Position	TX Message Buffer Number
$n \times 64 - f_{min} \times 32$	$n \times 64 + 0$
$n \times 64 + 1 - f_{min} \times 32$	$n \times 64 + 1$
⋮	⋮
$n \times 64 + 15 - f_{min} \times 32$	$n \times 64 + 15$
$n \times 64 + 32 - f_{max} \times 32$	$n \times 64 + 32$
⋮	⋮
$n \times 64 + 46 - f_{max} \times 32$	$n \times 64 + 46$
$n \times 64 + 47 - f_{max} \times 32$	$n \times 64 + 47$

Note: When $n = 0$, $f_{min} = 0$, $f_{max} = 1$
 When $n = 1$, $f_{min} = 2$, $f_{max} = 3$
 When $n = 2$, $f_{min} = 4$, $f_{max} = 5$
 When $n = 3$, $f_{min} = 6$, $f_{max} = 7$
 When $n = 4$, $f_{min} = 8$, $f_{max} = 9$
 When $n = 5$, $f_{min} = 10$, $f_{max} = 11$

Each bit is set automatically when the CFDTMSTSn.TMTRF[1:0] bits are set to 01b in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when:

- The CFDTMSTSn.TMTRF[1:0] bits are cleared in the corresponding TX Message Buffer Status Register
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

7.9.2.2.54 TX Message Buffer Transmission Interrupt Enable Register f (CFDTMIEcf) (f = 0 to 11)

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 0F50h + f x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TMIE[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 0	TMIE[15:0]	0h	RW	TX Message Buffer Interrupt Enable 0b: TX message buffer interrupt disabled for corresponding TX message buffer 1b: TX message buffer interrupt enabled for corresponding TX message buffer

TMIE[15:0] bits (TX Message Buffer Interrupt Enable)

If the TMIE[15:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See **7.9.7 Interrupt and DMA** for TX Message Buffer Interrupt specification.

Alignment of the TMIE[15:0] bits is shown in **Table 7.9-11**.

Table 7.9-11 Alignment of TMIE[15:0] bits

Bit Position	TX Message Buffer Number
$n \times 64 - f_{min} \times 32$	$n \times 64 + 0$
$n \times 64 + 1 - f_{min} \times 32$	$n \times 64 + 1$
⋮	⋮
$n \times 64 + 15 - f_{min} \times 32$	$n \times 64 + 15$
$n \times 64 + 32 - f_{max} \times 32$	$n \times 64 + 32$
⋮	⋮
$n \times 64 + 46 - f_{max} \times 32$	$n \times 64 + 46$
$n \times 64 + 47 - f_{max} \times 32$	$n \times 64 + 47$

Note: When $n = 0$, $f_{min} = 0$, $f_{max} = 1$
 When $n = 1$, $f_{min} = 2$, $f_{max} = 3$
 When $n = 2$, $f_{min} = 4$, $f_{max} = 5$
 When $n = 3$, $f_{min} = 6$, $f_{max} = 7$
 When $n = 4$, $f_{min} = 8$, $f_{max} = 9$
 When $n = 5$, $f_{min} = 10$, $f_{max} = 11$

Do not write to the TMIE[15:0] bits when:

- The CAN-FD module is in GL_SLEEP mode
- The related CAN-FD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCCn.CFTML[4:0] bits.

7.9.2.2.55 TX Queue Configuration/Control Register 0n (CFDTXQCC0n) (n = 0 to 5)

The CFDTXQCC0n register (n = 0 to 5) is used to configure the TX Queue transmission.

TXQ0 is composed of TXMB0 to TXMB15 (at the maximum) when TXQE is enabled.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1000h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	TXQOF TXIE	TXQOF RXIE	TXQFIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TXQDC[4:0]				TXQIM	-	TXQTX IE	-	-	TXQO WE	TXQG WE	TXQE	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18	TXQOFTXIE	0h	RW	TXQ One Frame Transmission Interrupt Enable 0b: One Frame TX interrupt generation disabled 1b: One Frame TX interrupt generation enabled
17	TXQOFRXIE	0h	RW	TXQ One Frame Reception Interrupt Enable 0b: One Frame RX interrupt generation disabled 1b: One Frame RX interrupt generation enabled
16	TXQFIE	0h	RW	TXQ Full Interrupt Enable 0b: TX Queue full interrupt generation disabled 1b: TX Queue full interrupt generation enabled
15 to 13	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
12 to 8	TXQDC[4:0]	0h	RW	TX Queue Depth Configuration 00h: 0 messages 01h: Reserved 02h: 3 messages 03h: 4 messages ⋮ 0Fh: 16 messages Others: Setting prohibited
7	TXQIM	0h	RW	TX Queue Interrupt Mode 0b: When the last message is successfully transmitted 1b: At every successful transmission
6	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	TXQTXIE	0h	RW	TX Queue TX Interrupt Enable 0b: TX Queue TX interrupt disabled 1b: TX Queue TX interrupt enabled
4, 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQOWE	0h	RW	TX Queue Overwrite Mode Enable 0b: TX Queue OW mode disabled 1b: TX Queue OW mode enabled
1	TXQGWE	0h	RW	TX Queue Gateway Mode Enable 0b: TX Queue GW mode disabled 1b: TX Queue GW mode enabled
0	TXQE	0h	RW	TX Queue Enable 0b: TX Queue disabled 1b: TX Queue enabled

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 00h (CFDTXQCC0n.TXQDC[4:0] = 00h).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue GW mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC0n.TXQE bit is 1b.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP

- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[15] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 16 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit in Gateway mode (CFDTXQCC0n.TXQGWE = 1b).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit in GW mode (CFDTXQCC0n.TXQGWE = 1b).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

7.9.2.2.56 TX Queue Configuration/Control Register 1n (CFDTXQCC1n) (n = 0 to 5)

The CFDTXQCC1n register (n = 0 to 5) is used to configure the TX Queue transmission.

TXQ1 is composed of TXMB15 to TXMB0 (at the maximum) when TXQE is enabled.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1060h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	TXQOF TXIE	TXQOF RXIE	TXQFIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TXQDC[4:0]				TXQIM	-	TXQTX IE	-	-	TXQO WE	TXQG WE	TXQE	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Reserved															
31 to 19	-		All 0		RW		Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
18	TXQOFTXIE		0h		RW		TXQ One Frame Transmission Interrupt Enable 0b: One Frame TX interrupt generation disabled 1b: One Frame TX interrupt generation enabled									
17	TXQOFRXIE		0h		RW		TXQ One Frame Reception Interrupt Enable 0b: One Frame RX interrupt generation disabled 1b: One Frame RX interrupt generation enabled									
16	TXQFIE		0h		RW		TXQ Full Interrupt Enable 0b: TX Queue full interrupt generation disabled 1b: TX Queue full interrupt generation enabled									
15 to 13	-		All 0		RW		Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
12 to 8	TXQDC[4:0]		0h		RW		TX Queue Depth Configuration 00h: 0 messages 01h: Reserved 02h: 3 messages 03h: 4 messages ⋮ 0Fh: 16 messages Others: Setting prohibited									
7	TXQIM		0h		RW		TX Queue Interrupt Mode 0b: When the last message is successfully transmitted 1b: At every successful transmission									
6	-		0h		RW		Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
5	TXQTXIE		0h		RW		TX Queue TX Interrupt Enable 0b: TX Queue TX interrupt disabled 1b: TX Queue TX interrupt enabled									
4, 3	-		All 0		RW		Reserved Whenever it is read, 0b is read. The written value should always be 0b.									
2	TXQOWE		0h		RW		TX Queue Overwrite Mode Enable 0b: TX Queue OW mode disabled 1b: TX Queue OW mode enabled									
1	TXQGWE		0h		RW		TX Queue Gateway Mode Enable 0b: TX Queue GW mode disabled 1b: TX Queue GW mode enabled									
0	TXQE		0h		RW		TX Queue Enable 0b: TX Queue disabled 1b: TX Queue enabled									

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 00h (CFDTXQCC1n.TXQDC[4:0] = 00h).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC1n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC1n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP

- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[15] down to MB[0] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 16 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1b).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1b).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

7.9.2.2.57 TX Queue Configuration/Control Register 2n (CFDTXQCC2n) (n = 0 to 5)

The CFDTXQCC2n register (n = 0 to 5) is used to configure the TX Queue transmission.

TXQ2 is composed of TXMB32 to TXMB47 (at the maximum) when TXQE is enabled.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 10C0h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	TXQOF TXIE	TXQOF RXIE	TXQFIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TXQDC[4:0]				TXQIM	-	TXQTX IE	-	-	TXQO WE	TXQG WE	TXQE	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Reserved															
31 to 19	-			All 0			RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.								
18	TXQOFTXIE		0h		RW	TXQ One Frame Transmission Interrupt Enable 0b: One Frame TX interrupt generation disabled 1b: One Frame TX interrupt generation enabled										
17	TXQOFRXIE		0h		RW	TXQ One Frame Reception Interrupt Enable 0b: One Frame RX interrupt generation disabled 1b: One Frame RX interrupt generation enabled										
16	TXQFIE		0h		RW	TXQ Full Interrupt Enable 0b: TX Queue full interrupt generation disabled 1b: TX Queue full interrupt generation enabled										
15 to 13	-			All 0			RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.								
12 to 8	TXQDC[4:0]		0h		RW	TX Queue Depth Configuration 00h: 0 messages 01h: Reserved 02h: 3 messages 03h: 4 messages ⋮ 0Fh: 16 messages Others: Setting prohibited										
7	TXQIM		0h		RW	TX Queue Interrupt Mode 0b: When the last message is successfully transmitted 1b: At every successful transmission										
6	-		0h		RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.										
5	TXQTXIE		0h		RW	TX Queue TX Interrupt Enable 0b: TX Queue TX interrupt disabled 1b: TX Queue TX interrupt enabled										
4, 3	-			All 0			RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.								
2	TXQOWE		0h		RW	TX Queue Overwrite Mode Enable 0b: TX Queue OW mode disabled 1b: TX Queue OW mode enabled										
1	TXQGWE		0h		RW	TX Queue Gateway Mode Enable 0b: TX Queue GW mode disabled 1b: TX Queue GW mode enabled										
0	TXQE		0h		RW	TX Queue Enable 0b: TX Queue disabled 1b: TX Queue enabled										

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 00h (CFDTXQCC2n.TXQDC[4:0] = 00h).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode. When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC2n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC2n.TXQE bit is 1b.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP

- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[32] up to MB[47] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 16 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1b).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1b to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1b).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

7.9.2.2.58 TX Queue Configuration/Control Register 3n (CFDTXQCC3n) (n = 0 to 5)

The CFDTXQCC3n register (n = 0 to 5) is used to configure the TX Queue transmission.

TXQ3 is composed of TXMB47 to TXMB32 (at the maximum) when TXQE is enabled.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1120h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	TXQOF TXIE	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TXQDC[4:0]				TXQIM	-	TXQTX IE	-	-	TXQO WE	-	TXQE	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18	TXQOFTXIE	0h	RW	TXQ One Frame Transmission Interrupt Enable 0b: One Frame TX interrupt generation disabled 1b: One Frame TX interrupt generation enabled
17 to 13	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
12 to 8	TXQDC[4:0]	0h	RW	TX Queue Depth Configuration 00h: 0 messages 01h: Reserved 02h: 3 messages 03h: 4 messages ⋮ 0Fh: 16 messages Others: Setting prohibited
7	TXQIM	0h	RW	TX Queue Interrupt Mode 0b: When the last message is successfully transmitted 1b: At every successful transmission
6	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	TXQTXIE	0h	RW	TX Queue TX Interrupt Enable 0b: TX Queue TX interrupt disabled 1b: TX Queue TX interrupt enabled
4, 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQOWE	0h	RW	TX Queue Overwrite Mode Enable 0b: TX Queue OW mode disabled 1b: TX Queue OW mode enabled
1	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	TXQE	0h	RW	TX Queue Enable 0b: TX Queue disabled 1b: TX Queue enabled

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 00h (CFDTXQCC3n.TXQDC[4:0] = 00h).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC3n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC3n.TXQE bit is 1b.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[47] down to MB[32] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 16 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS3n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

7.9.2.2.59 TX Queue Status Register 0n (CFDTXQSTS0n) (n = 0 to 5)

The CFDTXQSTS0n register (n = 0 to 5) shows the status of the TX Queue of corresponding CAN channel.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1020h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	TXQM OW	TXQM LT	TXQOF TXIF	TXQOF RXIF	TXQFIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TXQMC[5:0]					-	-	-	-	-	TXQT XIF	TXQFL L	TXQE MP	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TXQMOW	0h	RW	TXQ Message Overwrite 0b: No message overwrite in TXQ 1b: Message overwrite in TXQ
19	TXQMLT	0h	RW	TXQ Message Lost 0b: No message lost in TXQ 1b: TXQ message lost
18	TXQOFTXIF	0h	RW	TXQ One Frame Transmission Interrupt Flag When one frame is transmitted from TXQ, an interrupt is set.
17	TXQOFRXIF	0h	RW	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.
16	TXQFIF	0h	RW	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.
15, 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	TXQMC[5:0]	0h	R	TX Queue Message Count Number of messages in the TX Queue.
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQTXIF	0h	RW	TX Queue TX Interrupt Flag 0b: TX Queue interrupt condition not satisfied after a frame TX 1b: TX Queue interrupt condition satisfied after a frame TX
1	TXQFLL	0h	R	TX Queue Full 0b: TX Queue not full 1b: TX Queue full
0	TXQE MP	1h	R	TX Queue Empty 0b: TX Queue not empty 1b: TX Queue empty

TXQEEMP bit (TX Queue Empty)

The TXQEEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC0n.TXQGWE = 1b) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC0n.TXQOWE = 1b and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.60 TX Queue Status Register 1n (CFDTXQSTS1n) (n = 0 to 5)

The CFDTXQSTS1n register (n = 0 to 5) shows the status of the TX Queue of corresponding CAN channel.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1080h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	TXQM OW	TXQM LT	TXQOF TXIF	TXQOF RXIF	TXQFIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TXQMC[5:0]					-	-	-	-	-	TXQTX IF	TXQFL L	TXQE MP	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TXQMOW	0h	RW	TXQ Message Overwrite 0b: No message overwrite in TXQ 1b: Message overwrite in TXQ
19	TXQMLT	0h	RW	TXQ Message Lost 0b: No message lost in TXQ 1b: TXQ message lost
18	TXQOFTXIF	0h	RW	TXQ One Frame Transmission Interrupt Flag When one frame is transmitted from TXQ, an interrupt is set.
17	TXQOFRXIF	0h	RW	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.
16	TXQFIF	0h	RW	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.
15, 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	TXQMC[5:0]	0h	R	TX Queue Message Count Number of messages in the TX Queue
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQTXIF	0h	RW	TX Queue TX Interrupt Flag 0b: TX Queue interrupt condition not satisfied after a frame TX 1b: TX Queue interrupt condition satisfied after a frame TX
1	TXQFLL	0h	R	TX Queue Full 0b: TX Queue not full 1b: TX Queue full
0	TXQEEMP	1h	R	TX Queue Empty 0b: TX Queue not empty 1b: TX Queue empty

TXQEEMP bit (TX Queue Empty)

The TXQEEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only in Gateway mode (CFDCTXQCC1n.TXQGWE = 1b) that this bit is set automatically when TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC1n.TXQOWE = 1b and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.61 TX Queue Status Register 2n (CFDTXQSTS2n) (n = 0 to 5)

The CFDTXQSTS2n register (n = 0 to 5) shows the status of the TX Queue of corresponding CAN Channel.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 10E0h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	TXQM OW	TXQM LT	TXQOF TXIF	TXQOF RXIF	TXQFIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TXQMC[5:0]					-	-	-	-	-	-	TXQTX IF	TXQFL L	TXQE MP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TXQMOW	0h	RW	TXQ Message Overwrite 0b: No message overwrite in TXQ 1b: Message overwrite in TXQ
19	TXQMLT	0h	RW	TXQ Message Lost 0b: No message lost in TXQ 1b: TXQ message lost
18	TXQOFTXIF	0h	RW	TXQ One Frame Transmission Interrupt Flag When one frame is transmitted from TXQ, an interrupt is set.
17	TXQOFRXIF	0h	RW	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.
16	TXQFIF	0h	RW	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.
15, 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	TXQMC[5:0]	0h	R	TX Queue Message Count Number of messages in the TX Queue.
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQTXIF	0h	RW	TX Queue TX Interrupt Flag 0b: TX Queue interrupt condition not satisfied after a frame TX 1b: TX Queue interrupt condition satisfied after a frame TX
1	TXQFLL	0h	R	TX Queue Full 0b: TX Queue not full 1b: TX Queue full
0	TXQEEMP	1h	R	TX Queue Empty 0b: TX Queue not empty 1b: TX Queue empty

TXQEEMP bit (TX Queue Empty)

The TXQEEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDCTXQCC2n.TXQGWE = 1b) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC2n.TXQOWE = 1b and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.62 TX Queue Status Register 3n (CFDTXQSTS3n) (n = 0 to 5)

The CFDTXQSTS3n register (n = 0 to 5) shows the status of the TX Queue of corresponding CAN Channel.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1140h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	TXQMOW	-	TXQOF TXIF	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TXQMC[5:0]					-	-	-	-	-	TXQTXIF	TXQFL L	TXQEMP	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
20	TXQMOW	0h	RW	TXQ Message Overwrite 0b: No message overwrite in TXQ 1b: Message overwrite in TXQ
19	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
18	TXQOFTXIF	0h	RW	TXQ One Frame Transmission Interrupt Flag When one frame is transmitted from TXQ, an interrupt is set.
17 to 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	TXQMC[5:0]	0h	R	TX Queue Message Count Number of messages in the TX Queue
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TXQTXIF	0h	RW	TX Queue TX Interrupt Flag 0b: TX Queue interrupt condition not satisfied after a frame TX 1b: TX Queue interrupt condition satisfied after a frame TX
1	TXQFLL	0h	R	TX Queue Full 0b: TX Queue not full 1b: TX Queue full
0	TXQEMP	1h	R	TX Queue Empty 0b: TX Queue not empty 1b: TX Queue empty

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC3n.TXQOWE = 1b and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b. Writing 1b has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.63 TX Queue Pointer Control Register 0n (CFDTXQPCTR0n) (n = 0 to 5)

The CFDTXQPCTR0n register (n = 0 to 5) is used to confirm storage of a full message in the corresponding TX Queue buffers.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 1040h + n x 0004h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TXQPC[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	W	Reserved The written value should always be 0b.
7 to 0	TXQPC[7:0]	0h	W	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value FFh is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 00h. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write FFh to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

7.9.2.2.64 TX Queue Pointer Control Register 1n (CFDTXQPCTR1n) (n = 0 to 5)

The CFDTXQPCTR1n register (n = 0 to 5) is used to confirm storage of a full message in the corresponding TX Queue buffers.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 10A0h + n x 0004h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TXQPC[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	W	Reserved The written value should always be 0b.
7 to 0	TXQPC[7:0]	0h	W	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value FFh is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 00h. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write FFh to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

7.9.2.2.65 TX Queue Pointer Control Register 2n (CFDTXQPCTR2n) (n = 0 to 5)

The CFDTXQPCTR2n register (n = 0 to 5) is used to confirm storage of a full message in the corresponding TX Queue buffers.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 1100h + n x 0004h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TXQPC[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	Bit Name		Initial Value		R/W		Description											
31 to 8	-		All 0		W		Reserved The written value should always be 0b.											
7 to 0	TXQPC[7:0]		0h		W		TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel											

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value FFh is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 00h. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write FFh to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

7.9.2.2.66 TX Queue Pointer Control Register 3n (CFDTXQPCTR3n) (n = 0 to 5)

The CFDTXQPCTR3n register (n = 0 to 5) is used to confirm storage of a full message in the corresponding TX Queue buffers.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 1160h + n x 0004h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TXQPC[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	W	Reserved The written value should always be 0b.
7 to 0	TXQPC[7:0]	0h	W	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value FFh is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 00h. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write FFh to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in Gateway mode.

7.9.2.2.67 TX Queue Empty Status Register (CFDTXQESTS)

The CFDTXQESTS register shows the status of the empty bits of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1180h														
Initial Value :		0000_00FFh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	TXQxEMP[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQxEMP[15:0]															
Initial Value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read.
23 to 0	TXQxEMP[23:0]	FFh	R	TXQ Empty Status 0b: TXQ not empty 1b: TXQ empty

TXQxEMP[23:0] bits (TXQ Empty Status)

Each bit is set automatically when the corresponding bit is set in the TX Queue Empty Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Empty Status Register.

This bit is set when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Channel 2 TX Queue 3
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Channel 3 TX Queue 3
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2

Bit Position	Corresponding TX Queue
19	Channel 4 TX Queue 3
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Channel 5 TX Queue 3

7.9.2.2.68 TX Queue Full Interrupt Status Register (CFDTXQFISTS)

The CFDTXQFISTS register shows the status of the full interrupt bits of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1184h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	TXQ5FULL[2:0]			-	TXQ4FULL[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TXQ3FULL[2:0]			-	TXQ2FULL[2:0]			-	TXQ1FULL[2:0]			-	TXQ0FULL[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read.
22 to 20	TXQ5FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 5 0b: TXQ full interrupt not set 1b: TXQ full interrupt set
19	-	0h	R	Reserved Whenever it is read, 0b is read.
18 to 16	TXQ4FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 4 0b: TXQ full interrupt not set 1b: TXQ full interrupt set
15	-	0h	R	Reserved Whenever it is read, 0b is read.
14 to 12	TXQ3FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 3 0b: TXQ full interrupt not set 1b: TXQ full interrupt set
11	-	0h	R	Reserved Whenever it is read, 0b is read.
10 to 8	TXQ2FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 2 0b: TXQ full interrupt not set 1b: TXQ full interrupt set
7	-	0h	R	Reserved Whenever it is read, 0b is read.
6 to 4	TXQ1FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 1 0b: TXQ full interrupt not set 1b: TXQ full interrupt set
3	-	0h	R	Reserved Whenever it is read, 0b is read.
2 to 0	TXQ0FULL[2:0]	0h	R	TXQ Full Interrupt Status Flag for Channel 0 0b: TXQ full interrupt not set 1b: TXQ full interrupt set

TXQnFULL[2:0] (n = 0 to 5) bits (TXQ Full Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Reserved
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Reserved
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2
19	Reserved
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Reserved

7.9.2.2.69 TX Queue Message Lost Status Register (CFDTXQMSTS)

The CFDTXQMSTS register shows the status of the message lost bits of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1188h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	TXQ5ML[2:0]			-	TXQ4ML[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TXQ3ML[2:0]			-	TXQ2ML[2:0]			-	TXQ1ML[2:0]			-	TXQ0ML[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read.
22 to 20	TXQ5ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 5 0b: TXQ message lost flag not set 1b: TXQ message lost flag set
19	-	0h	R	Reserved Whenever it is read, 0b is read.
18 to 16	TXQ4ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 4 0b: TXQ message lost flag not set 1b: TXQ message lost flag set
15	-	0h	R	Reserved Whenever it is read, 0b is read.
14 to 12	TXQ3ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 3 0b: TXQ message lost flag not set 1b: TXQ message lost flag set
11	-	0h	R	Reserved Whenever it is read, 0b is read.
10 to 8	TXQ2ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 2 0b: TXQ message lost flag not set 1b: TXQ message lost flag set
7	-	0h	R	Reserved Whenever it is read, 0b is read.
6 to 4	TXQ1ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 1 0b: TXQ message lost flag not set 1b: TXQ message lost flag set
3	-	0h	R	Reserved Whenever it is read, 0b is read.
2 to 0	TXQ0ML[2:0]	0h	R	TXQ Message Lost Status Flag for Channel 0 0b: TXQ message lost flag not set 1b: TXQ message lost flag set

TXQnML[2:0] (n = 0 to 5) bits (TXQ Message Lost Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Message Lost Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Message Lost Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Reserved
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Reserved
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2
19	Reserved
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Reserved

7.9.2.2.70 TX Queue Interrupt Status Register (CFDTXQISTS)

The CFDTXQISTS register shows the status of the interrupt flag of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1190h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	TXQ5ISF[3:0]				TXQ4ISF[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ3ISF[3:0]				TXQ2ISF[3:0]				TXQ1ISF[3:0]				TXQ0ISF[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read.
23 to 0	TXQnISF[3:0]	0h	R	TXQ Interrupt Status Flag for Channel n (n = 0 to 5) 0b: TXQ Interrupt flag not set 1b: TXQ Interrupt flag set

TXQnISF[3:0] (n = 0 to 5) bits (TXQ Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Channel 2 TX Queue 3
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Channel 3 TX Queue 3
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2

Bit Position	Corresponding TX Queue
19	Channel 4 TX Queue 3
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Channel 5 TX Queue 3

7.9.2.2.71 TX Queue One Frame TX Interrupt Status Register (CFDTXQOFTISTS)

The CFDTXQOFTISTS register shows the status of the One Frame TX Interrupt flag of the TXQ buffers.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 1194h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	TXQ5OFTISF[3:0]				TXQ4OFTISF[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TXQ3OFTISF[3:0]				TXQ2OFTISF[3:0]				TXQ1OFTISF[3:0]				TXQ0OFTISF[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read.
23 to 0	TXQnOFTISF [3:0]	0h	R	TXQ One Frame TX Interrupt Status Flag for Channel n (n = 0 to 5) 0b: TXQ One Frame TX Interrupt flag not set 1b: TXQ One Frame TX Interrupt flag set

TXQnOFTISF[3:0] (n = 0 to 5) bits (TXQ One Frame TX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame TX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame TX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Channel 2 TX Queue 3
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Channel 3 TX Queue 3
16	Channel 4 TX Queue 0

Bit Position	Corresponding TX Queue
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2
19	Channel 4 TX Queue 3
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Channel 5 TX Queue 3

7.9.2.2.72 TX Queue One Frame RX Interrupt Status Register (CFDTXQOFRISTS)

The CFDTXQOFRISTS register shows the status of the One Frame RX Interrupt flag of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1198h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	TXQ5OFRISF[2:0]			-	TXQ4OFRISF[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TXQ3OFRISF[2:0]			-	TXQ2OFRISF[2:0]			-	TXQ1OFRISF[2:0]			-	TXQ0OFRISF[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read.
22 to 20	TXQ5OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 5 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set
19	-	0h	R	Reserved Whenever it is read, 0b is read.
18 to 16	TXQ4OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 4 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set
15	-	0h	R	Reserved Whenever it is read, 0b is read.
14 to 12	TXQ3OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 3 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set
11	-	0h	R	Reserved Whenever it is read, 0b is read.
10 to 8	TXQ2OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 2 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set
7	-	0h	R	Reserved Whenever it is read, 0b is read.
6 to 4	TXQ1OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 1 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set
3	-	0h	R	Reserved Whenever it is read, 0b is read.
2 to 0	TXQ0OFRISF [2:0]	0h	R	TXQ One Frame RX Interrupt Status Flag for Channel 0 0b: TXQ One Frame RX Interrupt flag not set 1b: TXQ One Frame RX Interrupt flag set

TXQnOFRISF[2:0] (n = 0 to 5) bits (TXQ One Frame RX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame RX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame RX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Reserved
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Reserved
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2
19	Reserved
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Reserved

7.9.2.2.73 TX Queue Full Status Register (CFDCTXQFSTS)

The CFDCTXQFSTS register shows the status of the Full Status flag bits of the TXQ buffers.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 119Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	TXQ5FSF[3:0]				TXQ4FSF[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ3FSF[3:0]				TXQ2FSF[3:0]				TXQ1FSF[3:0]				TXQ0FSF[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0 is read.
23 to 0	TXQnFSF[3:0]	0h	R	TXQ Full Status Flag for Channel n (n = 0 to 5) 0b: TXQ Full flag not set 1b: TXQ Full flag set

TXQnFSF[3:0] (n = 0 to 5) bits (TXQ Full Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit Position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3
8	Channel 2 TX Queue 0
9	Channel 2 TX Queue 1
10	Channel 2 TX Queue 2
11	Channel 2 TX Queue 3
12	Channel 3 TX Queue 0
13	Channel 3 TX Queue 1
14	Channel 3 TX Queue 2
15	Channel 3 TX Queue 3
16	Channel 4 TX Queue 0
17	Channel 4 TX Queue 1
18	Channel 4 TX Queue 2

Bit Position	Corresponding TX Queue
19	Channel 4 TX Queue 3
20	Channel 5 TX Queue 0
21	Channel 5 TX Queue 1
22	Channel 5 TX Queue 2
23	Channel 5 TX Queue 3

7.9.2.2.74 TX History List Configuration/Control Register n (CFDTHLCCn) (n = 0 to 5)

The CFDTHLCCn register (n = 0 to 5) configures the TX History List functions.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 1200h + n x 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	THLDG E	THLDT E	THLIM	THLIE	-	-	-	-	-	-	-	THLE	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11	THLDGE	0h	RW	TX History List Dedicated Gateway Enable 0b: Not dedicated Gateway FIFO + Gateway TX Queue 1b: Dedicated Gateway FIFO + Gateway TX Queue
10	THLDTE	0h	RW	TX History List Dedicated TX Enable 0b: TX FIFO + TX Queue 1b: Flat TX MB + TX FIFO + TX Queue
9	THLIM	0h	RW	TX History List Interrupt Mode 0b: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1b: Interrupt generated for every successfully stored entry
8	THLIE	0h	RW	TX History List Interrupt Enable 0b: TX History List Interrupt disabled 1b: TX History List Interrupt enabled
7 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	THLE	0h	RW	TX History List Enable 0b: TX History List disabled 1b: TX History List enabled

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDGE bit (TX History List Dedicated Gateway Enable)

The THLDGE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

7.9.2.2.75 TX History List Status Register n (CFDTHLSTSn) (n = 0 to 5)

The CFDTHLSTSn register (n = 0 to 5) shows the status of data stored in the TX History List buffer.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1220h + n x 0004h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	THLMC[5:0]					-	-	-	-	THLIF	THLELT	THLFL	THLEMP	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	THLMC[5:0]	0h	R	TX History List Message Count Number of messages stored in TX History List
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3	THLIF	0h	RW	TX History List Interrupt Flag 0b: TX History List interrupt condition not satisfied 1b: TX History List interrupt condition satisfied
2	THLELT	0h	RW	TX History List Entry Lost 0b: No entry lost in TX History List 1b: TX History List entry Lost
1	THLFL	0h	R	TX History List Full 0b: TX History List not full 1b: TX History List full
0	THLEMP	1h	R	TX History List Empty 0b: TX History List not empty 1b: TX History List empty

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

THLFL bit (TX History List Full)

The THLFL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 32 entries (each channel has a dedicated TX History List).

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled

- The related CAN-FD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1b has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1b.

This bit is cleared:

- By writing 0b to it
- When the related CAN-FD channel is in CH_RESET mode.

This bit is automatically cleared in CH_RESET mode.

THLMC[5:0] bits (TX History List Message Count)

The THLMC[5:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

7.9.2.2.76 Channel n TX History List Access Register 0 (CFDTHLACC0n) (n = 0 to 5)

The CFDTHLACC0n register (n = 0 to 5) provides access to the entry in the TX History List based on the read timestamp value.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 8000h + n x 0008h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TGW	-	-	-	-	-	BN[6:0]						BT[2:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TMTS[15:0]	0h	R	Transmit Timestamp Transmit timestamp value for software drivers
15	TGW	0h	R	Transmit Gateway Buffer Indication 0b: No transmission from gateway 1b: Transmission from gateway
14 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read.
9 to 3	BN[6:0]	0h	R	Buffer Number Number of the message buffer
2 to 0	BT[2:0]	0h	R	Buffer Type 001b: Flat TX message buffer 010b: TX FIFO message buffer number and gateway FIFO message number 100b: TX Queue message buffer number Others: Setting prohibited

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[6:0] bits (Buffer Number)

The BN[6:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TGW bit (Transmit Gateway Buffer Indication)

The TGW bit is automatically set to 1b when transmission is completed in GW mode.

TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

7.9.2.2.77 Channel n TX History List Access Register 1 (CFDTHLACC1n) (n = 0 to 5)

The CFDTHLACC1n register (n = 0 to 5) provides access to entry in the TX History List based on the read pointer value.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 8004h + n x 0008h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TIFL[1:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TID[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read.
17, 16	TIFL[1:0]	0h	R	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.
15 to 0	TID[15:0]	0h	R	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRn.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTSn.CFPTR) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLP0n.GAFLPTR) instead of the message buffer reference ID (CFDTMFDCTRn.TMPTR).

TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRn.TMIFL) or the TX FIFO information label (CFDCFFDCSTSn.CFIFL) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLMn.GAFLIFL1 and CFDGAFLP0n.GAFLIFL0) instead of the MB information label (CFDTMFDCTRn.TMIFL).

7.9.2.2.78 TX History List Pointer Control Register n (CFDTHLPCTRn) (n = 0 to 5)

The CFDTHLPCTRn register (n = 0 to 5) is used to increment the read pointer of the TX History List.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1240h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	THLPC[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	Bit Name		Initial Value	R/W	Description											
31 to 8	-		All 0	W	Reserved The written value should always be 0b.											
7 to 0	THLPC[7:0]		0h	W	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel											

THLPC[7:0] bits (TX History List Pointer Control)

When FFh is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

The read value from these bits is always 00h. Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Only write FFh to these registers when the corresponding TX History List is enabled and not empty.

7.9.2.2.79 Global Reset Control Register (CFDGRSTC)

Access Size :		16, 32 bits														
Address :		<CFD_base> + 1380h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY[7:0]								-	-	-	-	-	-	-	SRST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 8	KEY[7:0]	All 0	W	Key Code These bits control the validity of rewriting of the SRST bit.
7 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	SRST	0h	RW	Software Reset 0b: Normal state 1b: Software reset state

SRST bit (Software Reset)

When the SRST bit is set, the CAN-FD module is in the same state as hardware reset. When a reset is required, write 1b then write 0b to this bit.

This bit is cleared when the CAN-FD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When C4h is written in the KEY[7:0] bits, a write to the SRST bit is valid.

The read value from these bits is always 00h.

7.9.2.2.80 Global Flexible CAN Mode Configuration Register (CFDGFCMC)

Flexible CAN mode configured in the CFDGFCMC register and Flexible transmission buffer assignment configured in the CFDGFTBAC register should not be used simultaneously.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1384h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	FLXC2	FLXC1	FLXC0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name	Initial Value	R/W	Description												
31 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.												
2	FLXC2	0h	RW	Flexible CAN mode between Channel 4 and Channel 5 0b: Normal mode 1b: Flexible CAN mode												
1	FLXC1	0h	RW	Flexible CAN mode between Channel 2 and Channel 3 0b: Normal mode 1b: Flexible CAN mode												
0	FLXC0	0h	RW	Flexible CAN mode between Channel 0 and Channel 1 0b: Normal mode 1b: Flexible CAN mode												

FLXC0 bit (Flexible CAN Mode between Channel 0 and Channel 1)

When the FLXC0 bit is set, Channel 0 and Channel 1 of a CAN-FD module are in Flexible CAN mode.

Channel 1 uses TX/RX terminal of Channel 0. The TX/RX terminal of Channel 1 cannot be used.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

FLXC1 bit (Flexible CAN Mode between Channel 2 and Channel 3)

When the FLXC1 bit is set, Channel 2 and Channel 3 of a CAN-FD module are in Flexible CAN mode.

Channel 3 uses TX/RX terminal of Channel 2. The TX/RX terminal of Channel 3 cannot be used.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

FLXC2 bit (Flexible CAN Mode between Channel 4 and Channel 5)

When the FLXC2 bit is set, Channel 4 and Channel 5 of a CAN-FD module are in Flexible CAN mode.

Channel 5 uses TX/RX terminal of Channel 4. The TX/RX terminal of Channel 5 cannot be used.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

7.9.2.2.81 Global Flexible Transmission Buffer Assignment Configuration Register (CFDGFTBAC)

Flexible transmission buffer assignment configured in the CFDGFTBAC register and Flexible CAN mode configured in the CFDGFCMC register should not be used simultaneously.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 138Ch															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	FLXMB3[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	FLXMB1[3:0]				-	-	-	-	FLXMB0[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
19 to 16	FLXMB3[3:0]	0h	RW	Flexible transmission buffer assignment between Channel 4 and Channel 5 By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel 0000b: 0 0001b: 4 0010b: 8 0011b: 12 0100b: 16 0101b: 20 0110b: 24 0111b: 28 1000b: 32
15 to 12	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11 to 8	FLXMB1[3:0]	0h	RW	Flexible transmission buffer assignment between Channel 2 and Channel 3 By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel 0000b: 0 0001b: 4 0010b: 8 0011b: 12 0100b: 16 0101b: 20 0110b: 24 0111b: 28 1000b: 32
7 to 4	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
3 to 0	FLXMB0[3:0]	0h	RW	Flexible transmission buffer assignment between Channel 0 and Channel 1 By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel 0000b: 0 0001b: 4 0010b: 8 0011b: 12 0100b: 16 0101b: 20 0110b: 24 0111b: 28 1000b: 32

FLXMB0[3:0] bits (Flexible Transmission Buffer Assignment between Channel 0 and Channel 1)

Channel 0 can use the number TXMB of channel 1 from 0 to 32 by the configuration of these bits.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

7.9.2.2.82 Global Test Configuration Register (CFDGTSTCFG)

The CFDGTSTCFG register is used to configure the CAN channels joining the internal CAN bus communication test mode and the RAM test mode page.

Access Size :		8, 16, 32 bits																
Address :		<CFD_base> + 1308h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	RTMPS[9:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	-	-	C5ICB CE	C4ICB CE	C3ICB CE	C2ICB CE	C1ICB CE	C0ICB CE		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
25 to 16	RTMPS[9:0]	0h	RW	RAM Test Mode Page Select Select a RAM test mode page
15 to 6	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5 to 0	CnICBCE	0h	RW	Channel n Internal CAN Bus Communication Test Mode Enable (n = 0 to 5) 0b: Channel n internal CAN bus communication disabled 1b: Channel n internal CAN bus communication enabled

CnICBCE (n = 0 to 5) bits (Channel n Internal CAN Bus Communication Test Mode Enable)

When the CnICBCE bits are set and CAN-FD module is configured in the internal CAN bus communication test mode, then CAN channel n joins the internal CAN bus communication test mode operation.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

RTMPS[9:0] bits (RAM Test Mode Page Select)

The RTMPS[9:0] bits select the RAM page mode for CPU read/write access when the CAN-FD module is configured in RAM test mode.

See **7.9.9.2.1 RAM Test Mode** for the RAM test mode specification.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 47 (02Fh) for the AFL RAM and 48 to 228 (0E4h) for the message buffer RAM.

The setting range of these bits depends on the combination of parameters.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in GL_RESET mode.

7.9.2.2.83 Global Test Control Register (CFDGTSTCTR)

The CFDGTSTCTR register is used to control the global test modes of the CAN-FD module.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 130Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	RTME	-	ICBCTME
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	RTME	0h	RW	RAM Test Mode Enable 0b: RAM test mode disabled 1b: RAM test mode enabled
1	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	ICBCTME	0h	RW	Internal CAN Bus Communication Test Mode Enable 0b: Internal CAN Bus Communication test mode disabled 1b: Internal CAN Bus Communication test mode enabled

ICBCTME bit (Internal CAN Bus Communication Test Mode Enable)

When the ICBCTME bit is set, internal CAN bus communication is enabled for the CAN channels that are configured for internal CAN bus communication participation. See **7.9.9.2.2 Internal CAN Bus Communication Mode** for the specification of internal CAN bus communication test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CAN-FD module is configured in RAM test mode. See **7.9.9.2.1 RAM Test Mode** for RAM test mode specification.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

7.9.2.2.84 Global FD Configuration Register (CFDGFDCFG)

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1314h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TSCCFG[1:0]		-	-	-	-	-	-	-	RPED
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9, 8	TSCCFG[1:0]	0h	RW	Timestamp Capture Configuration 00b: Timestamp capture at the sample point of SOF (start of frame) 01b: Timestamp capture at frame valid indication 10b: Timestamp capture at the sample point of RES bit 11b: Reserved
7 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	RPED	0h	RW	RES Bit Protocol Exception Disable 0b: Protocol exception event detection enabled 1b: Protocol exception event detection disabled

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CAN-FD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CAN-FD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

7.9.2.2.85 Global Lock Key Register (CFDGLOCKK)

The CFDGLOCKK register is a write-only register that is used to unlock the protection for special test bits. See 7.9.9.2 Global Test Modes for Lock key specification.

Access Size :		16, 32 bits														
Address :		<CFD_base> + 131Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	Bit Name															
31 to 16	-		All 0		W		Reserved The written value should always be 0b.									
15 to 0	LOCK[15:0]		0h		W		Lock Key Key bits for unlocking the protection of test modes									

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CAN-FD module in RAM test modes.

The read value from these bits is always 0000h.

You cannot write to these bits when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CAN-FD module is in GL_OPERATION mode.

7.9.2.2.86 RAM Test Page Access Register n (CFDRPGACCn) (n = 0 to 63)

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 8400h + n x 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDTA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDTA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	RDTA[31:0]															
31 to 0	RDTA[31:0]															
		0h														
		RW														
		RAM Data Test Access RAM data bytes														

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CAN-FD module is configured in RAM test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode and RAM test mode is enabled.

If data is read when RAM test mode is not enabled, then it is always read as 0000_0000h.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

7.9.2.2.87 Channel n Bus Load Control Register (CFDCnBLCT) (n = 0 to 5)

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1418h + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	BLCLD	-	-	-	-	-	-	-	BLCE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	W	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
8	BLCLD	0h	W	Bus Load Counter Load When CFDCnBLCT.BLCLD is set, it is reset after a bus load counter value is loaded.
7 to 1	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	BLCE	0h	RW	Bus Load Counter Enable 0b: Bus load counter disabled 1b: Bus load counter enabled

BLCE bit (Bus Load Counter Enable)

The BLCE bit indicates the enabling signal of a bus load counter. Write to this bit after setting up the CFDCnNCFG register.

When this bit is 0b, the bus load counter stops but the counter is not clear.

The bit is cleared when the related CAN-FD channel is in CH_RESET mode.

Do not write to this bit in CH_SLEEP mode.

BLCLD bit (Bus Load Counter Load)

When the BLCLD bit is set, the bus load counter value is loaded into CFDCnBLSTS.BLC and the bus load counter is reset.

The read value is always 0b.

7.9.2.2.88 Channel n Bus Load Status Register (CFDCnBLSTS) (n = 0 to 5)

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 141Ch + n x 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLC[28:13]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLC[12:0]														-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 3	BLC[28:0]															
	Initial Value															
	0h															
	R/W															
	R															
	Description															
	Bus Load Counter Status															
	These bits indicate the bus load counter value.															
2 to 0	-															
	Initial Value															
	All 0															
	R/W															
	R															
	Description															
	Reserved															
	Whenever it is read, 0b is read.															

BLC[28:0] bits (Bus Load Counter Status)

When the CFDCnBLCT.BLCLD bit is set, the bus load counter value is loaded to BLC[28:0] bits and the bus load counter is reset.

These bits are set only by the CAN-FD module. Writing any value has no effect.

7.9.2.2.89 Identifier Bits Alignment

Standard Identifier (11 bits) format: ID28 – ID18 is aligned to b10 to b0, b11 to b28 should be 0b.

Extended Identifier (29 bits) format: ID28 – ID0 is aligned to b28 – b0

Table 7.9-12 Standard Identifier (11-Bit Format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0b	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

Table 7.9-13 Extended Identifier (29-Bit Format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1b	RTR	—	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

7.9.2.3 Message Buffer Component Structure

7.9.2.3.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components and the number of channels.

- number_of_channels: 6
- number_of_RMBCPs_per_channel: 16
- number_of_RFMBCPs: 8
- number_of_CFMBCPs_per_channel: 3
- number_of_TMBCPs_per_channel: 64

The start addresses for each register in the Message Buffer component are depicted in **Table 7.9-14**.

Table 7.9-14 Message Buffer Component Register Start Addresses

b = Message Buffer Component Index	MBCP	Register	p	Start Address (n = 0 to 5)
0 to number_of_RMBCPs_per_channel - 1	RMBCPb[i]	RMID	x	$2000h + b \times 0080h + n \times 800h$
		RMPTR	x	$2004h + b \times 0080h + n \times 800h$
		RMFDSTS	x	$2008h + b \times 0080h + n \times 800h$
		RMDFp	0 to 15	$200Ch + p \times 0004h + b \times 0080h + n \times 800h$
0 to number_of_RFMBCPs - 1	RFMBCPb[i]	RFID	x	$6000h + b \times 0080h$
		RFPTR	x	$6004h + b \times 0080h$
		RFFDSTS	x	$6008h + b \times 0080h$
		RFDFp	0 to 15	$600Ch + p \times 0004h + b \times 0080h$
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb[i]	CFID	x	$6400h + b \times 0080h + n \times 180h$
		CFPTR	x	$6404h + b \times 0080h + n \times 180h$
		CFFDCSTS	x	$6408h + b \times 0080h + n \times 180h$
		CFDFp	0 to 15	$640Ch + p \times 0004h + b \times 0080h + n \times 180h$
0 to number_of_TMBCPs_per_channel - 1	TMBCPb[i]	TMID	x	$10000h + b \times 0080h + n \times 2000h$
		TMPTR	x	$10004h + b \times 0080h + n \times 2000h$
		TMFDCTR	x	$10008h + b \times 0080h + n \times 2000h$
		TMDfP	0 to 15	$1000Ch + p \times 0004h + b \times 0080h + n \times 2000h$

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[i])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[i])
- Common FIFO Access Message Buffer Component (CFDCFMBCPb[i])
- TX Message Buffer Component (CFDTMBCPb[i]).

Where b = the Message Buffer component index having a range that varies based on the type of Message Buffer component and i = channel index that has a range from 0 to n.

For a summary of this configuration, see **Figure 7.9-28**. For a detailed description of the number of and the different types of message buffers, see **7.9.6 FIFO Buffers and Normal Message Buffer Configuration**.

As described in **7.9.2 CANFD Registers**, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

p is the Data Field register index having a range that varies based on the type of Message Buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index having a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains ‘-’ means reserved and has the same behavior as reserved bits for registers in **7.9.2.3 Message Buffer Component Structure**.

(1) RX Message Buffer ID Register n (CFDRMIDn) (n = 0 to 95)

The CFDRMIDn register (n = 0 to 95) stores the ID field, IDE bit, and RTR bit of the received message.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 2000h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRT R	-	RMID[28:16]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	RMIDE	0h	R	RX Message Buffer IDE Bit 0b: STD-ID is stored 1b: EXT-ID is stored
30	RMRT R	0h	R	RX Message Buffer RTR Bit 0b: Data frame 1b: Remote frame
29	-	0h	R	Reserved Whenever it is read, 0b is read.
28 to 0	RMID[28:0]	0h	R	RX Message Buffer ID Field STD-ID/EXT-ID fields

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

For alignment of these bits in standard and extended frame format, see **7.9.2.2.89 Identifier Bits Alignment**.

See **7.9.2.3.1 Start Addresses** for details on how to interpret the structure of this buffer component.

RMRT bit (RX Message Buffer RTR Bit)

The RMRT bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

(2) RX Message Buffer Pointer Register (CFDRMPTRn) (n = 0 to 95)

The CFDRMPTRn register (n = 0 to 95) stores the DLC and Timestamp fields for the received message.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 2004h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	RMDLC[3:0]	0h	RW	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.
27 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 0	RMTS[15:0]	0h	RW	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

(3) RX Message Buffer CAN-FD Status Register (CFDRMFDSTSn) (n = 0 to 95)

The CFDRMFDSTSn register (n = 0 to 95) shows the status of the FDF, BRS, and ESI bits, and pointer of the received CAN-FD frame.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 2008h + n x 0080h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	RMPTR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	RMIFL[1:0]		-	-	-	-	-	RMFDF	RMBRS	RMESI	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RMPTR[15:0]	0h	RW	RX Message Buffer Pointer Field
15 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9, 8	RMIFL[1:0]	0h	RW	RX Message Buffer Information Label Field
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	RMFDF ^{*1}	0h	RW	CAN FD Format bit 0b: No CAN-FD frame received 1b: CAN-FD frame received
1	RMBRS ^{*1}	0h	RW	Bit Rate Switch bit 0b: CAN-FD frame received with no bit rate switch 1b: CAN-FD frame received with bit rate switch
0	RMESI ^{*1}	0h	RW	Error State Indicator bit 0b: CAN-FD frame received from error active node 1b: CAN-FD frame received from error passive node

Note 1. This bit is not available in the classical CAN function.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0b, this means a CAN2.0 frame is received and 0b is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0b, this means a CAN2.0 frame is received and 0b is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

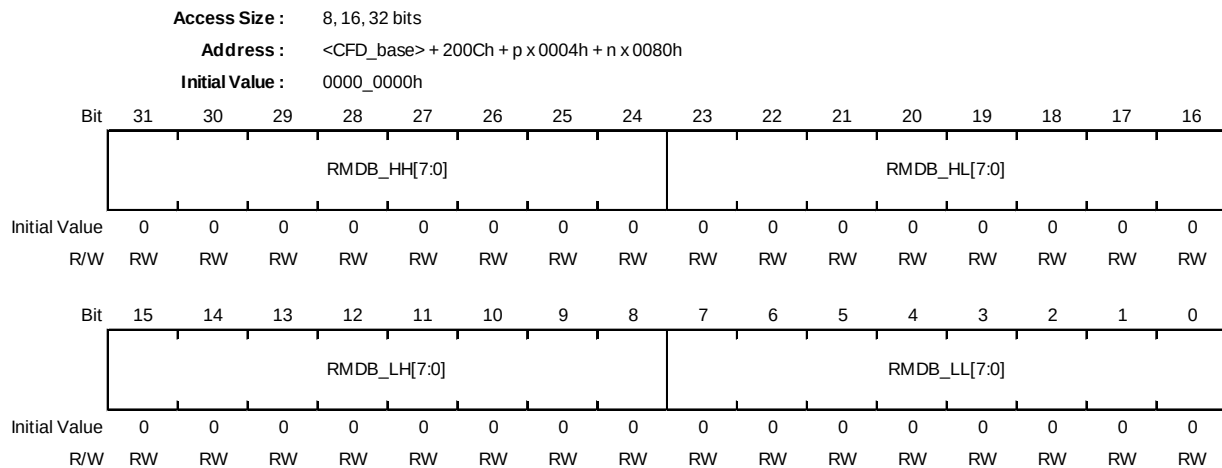
The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

(4) RX Message Buffer Data Field p Register n (CFDRMDFp_n) (p = 0 to 15, n = 0 to 95)

The CFDRMDFp_n register (p = 0 to 15, n = 0 to 95) stores the data bytes (4 × p) to data bytes (4 × p + 3) of the received message.



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	RMDB_HH ¹ [7:0]	0h	RW	RX Message Buffer Data Byte (4 × p + 3)
23 to 16	RMDB_HL[7:0]	0h	RW	RX Message Buffer Data Byte (4 × p + 2)
15 to 8	RMDB_LH[7:0]	0h	RW	RX Message Buffer Data Byte (4 × p + 1)
7 to 0	RMDB_LL[7:0]	0h	RW	RX Message Buffer Data Byte (4 × p)

Note 1. These bits are not available in the classical CAN function.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (4 × p))

The RMDB_LL[7:0] bits store data bytes (4 × p) of the message in the RX message buffer. Unused data bytes are filled with 00h.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte (4 × p + 1))

The RMDB_LH[7:0] bits store data bytes (4 × p + 1) of the message in the RX message buffer. Unused Data Bytes are filled with 00h.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte (4 × p + 2))

The RMDB_HL[7:0] bits store data bytes (4 × p + 2) of the message in the RX message buffer. Unused data bytes are filled with 00h.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte (4 × p + 3))

The RMDB_HH[7:0] bits store data bytes (4 × p + 3) of the message in the RX message buffer. Unused data bytes are filled with 00h.

These bits are not available in the classical CAN function.

(5) RX FIFO Access ID Register n (CFDRFIDn) (n = 0 to 7)

The CFDRFIDn register (n = 0 to 7) stores the ID field, IDE bit, and RTR bit of the message.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 6000h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	-	RFID[28:16]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	RFIDE	0h	R	RX FIFO Buffer IDE bit 0b: STD-ID has been received 1b: EXT-ID has been received
30	RFRTR	0h	R	RX FIFO Buffer RTR bit 0b: Data frame 1b: Remote frame
29	-	0h	R	Reserved Whenever it is read, 0b is read.
28 to 0	RFID[28:0]	0h	R	RX FIFO Buffer ID Field STD-ID/EXT-ID fields

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see 7.9.2.2.89 Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

(6) RX FIFO Access Pointer Register n (CFDRFPTRn) (n = 0 to 7)

The CFDRFPTRn register (n = 0 to 7) stores the DLC and Timestamp fields for the received message.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 6004h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	RFDLC[3:0]	0h	R	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame
27 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read.
15 to 0	RFTS[15:0]	0h	R	RX FIFO Timestamp Value Timestamp value of the received CAN frame

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

(7) RX FIFO Access CAN-FD Status Register n (CFDRFFDSTSn) (n = 0 to 7)

The CFDRFFDSTSn register (n = 0 to 7) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 6008h + n x 0080h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CFDRFPTR[15:0]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	RFIFL[1:0]		-	-	-	-	-	RFFDF	RFBRS	RFESI	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	CFDRFPTR [15:0]	0h	R	RX FIFO Buffer Pointer Field
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read.
9, 8	RFIFL[1:0]	0h	R	RX FIFO Buffer Information Label Field
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read.
2	RFFDF* ¹	0h	R	CAN FD Format bit 0b: No CAN-FD frame received 1b: CAN-FD frame received
1	RFBRS* ¹	0h	R	Bit Rate Switch bit 0b: CAN-FD frame received with no bit rate switch 1b: CAN-FD frame received with bit rate switch
0	RFESI* ¹	0h	R	Error State Indicator bit 0b: CAN-FD frame received from error active node 1b: CAN-FD frame received from error passive node

Note 1. This bit is not available in the classical CAN function.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0b, this means a CAN2.0 frame is received and 0b is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0b, this means a CAN2.0 frame is received and 0b is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

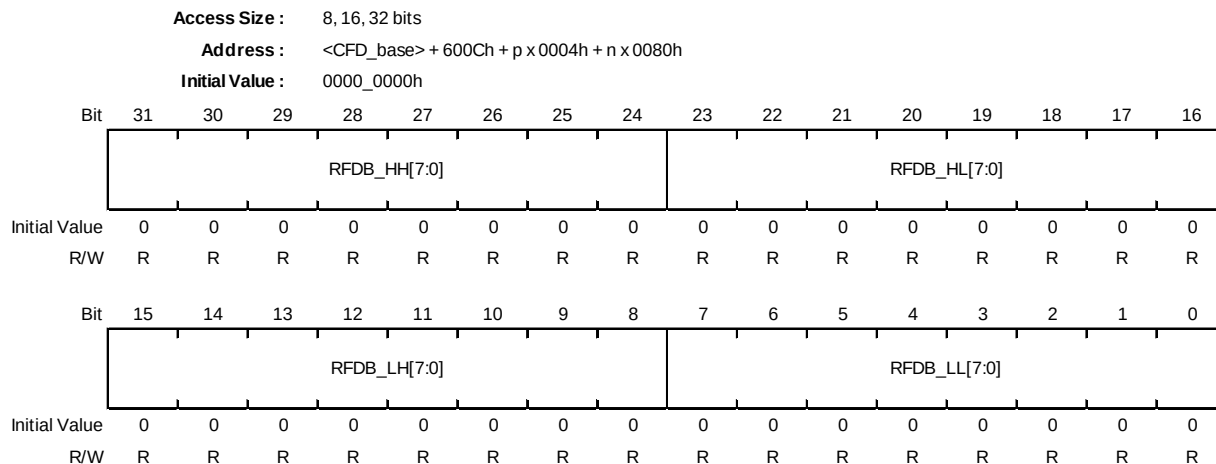
The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

(8) RX FIFO Access Data Field p Register n (CFDRFDFpn) (p = 0 to 15, n = 0 to 7)

The CFDRFDFpn register (p = 0 to 15, n = 0 to 7) stores data bytes ($4 \times p$) to data bytes ($4 \times p + 3$) of the received message.



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	RFDB_HH[7:0]	0h	R	RX FIFO Buffer Data Byte ($4 \times p + 3$)
23 to 16	RFDB_HL[7:0]	0h	R	RX FIFO Buffer Data Byte ($4 \times p + 2$)
15 to 8	RFDB_LH[7:0]	0h	R	RX FIFO Buffer Data Byte ($4 \times p + 1$)
7 to 0	RFDB_LL[7:0]	0h	R	RX FIFO Buffer Data Byte ($4 \times p$)

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte ($4 \times p$))

The RFDB_LL[7:0] bits store data bytes ($4 \times p$) of the message present in the FIFO buffer. Unused data bytes are filled with 00h according to the configured data payload size CFDRFCCn.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte ($4 \times p + 1$))

The RFDB_LH[7:0] bits store data bytes ($4 \times p + 1$) of the message present in the FIFO buffer. Unused data bytes are filled with 00h.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte ($4 \times p + 2$))

The RFDB_HL[7:0] bits store data bytes ($4 \times p + 2$) of the message present in the FIFO buffer. Unused data bytes are filled with 00h.

RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte ($4 \times p + 3$))

The RFDB_HH[7:0] bits store data bytes ($4 \times p + 3$) of the message present in the FIFO buffer. Unused data bytes are filled with 00h.

(9) Common FIFO Access ID Register n (CFDCFDn) (n = 0 to 17)

The CFDCFDn register (n = 0 to 17) stores the ID field, IDE bit, and RTR bit of the message.

Remark In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 6400h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	CFIDE	0h	RW	Common FIFO Buffer IDE bit 0b: STD-ID is to be transmitted or has been received 1b: EXT-ID is to be transmitted or has been received
30	CFRTR	0h	RW	Common FIFO Buffer RTR bit 0b: Data frame 1b: Remote frame
29	THLEN	0h	RW	THL Entry Enable TX FIFO mode: 0b: Entry is not to be stored in THL after successful TX 1b: Entry is to be stored in THL after successful TX RX FIFO mode: Reserved, This bit is read as 0b.
28 to 0	CFID[28:0]	0h	RW	Common FIFO Buffer ID Field STD-ID/EXT-ID fields

CFID[28:0] bits (Common FIFO Buffer ID Field)

The CFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see **7.9.2.2.89 Identifier Bits Alignment**.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

THLEN bit (THL Entry Enable)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR bit)

The CFRTR bit selects whether a data frame or a remote frame is to be transmitted from or was received in the FIFO buffer.

Note: There are no remote frames in CAN FD format. When a CAN-FD frame is received (RX mode), the register reflects the state of the received value (RRS bit in FD frame format). When CAN-FD transmission (TX or GW mode CFDCFD.CFFDF = 1b), the bit is always transmitted dominant (data frame).

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE bit)

The CFIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from or was received in the FIFO buffer.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

(10) Common FIFO Access Pointer Register n (CFDCFPTRn) (n = 0 to 17)

The CFDCFPTRn register (n = 0 to 17) stores the DLC and Timestamp fields.

Remark In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 6404h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	CFDLC[3:0]	0h	RW	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or transmitted in a CAN frame.
27 to 16	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
15 to 0	CFTS[15:0]	0h	RW	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write data from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

(11) Common FIFO Access CAN-FD Control/Status Register n (CFDCFFDCSTSn) (n = 0 to 17)

The CFDCFFDCSTSn register (n = 0 to 17) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame or the CAN-FD frame to transmit.

Remark In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 6408h + n x 0080h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	CFPTR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	CFIFL[1:0]	-	-	-	-	-	-	CFFDF	CFBRS	CFESI	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	CFPTR[15:0]	0h	RW	Common FIFO Buffer Pointer Field
15 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9, 8	CFIFL[1:0]	0h	RW	COMMON FIFO Buffer Information Label Field
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	CFFDF ^{*1}	0h	RW	CAN FD Format bit 0b: No CAN-FD frame received or to transmit 1b: CAN-FD frame received or to transmit
1	CFBRS ^{*1}	0h	RW	Bit Rate Switch bit 0b: CAN-FD frame received or to transmit with no bit rate switch 1b: CAN-FD frame received or to transmit with bit rate switch
0	CFESI ^{*1}	0h	RW	Error State Indicator bit 0b: CAN-FD frame received or to transmit by error active node 1b: CAN-FD frame received or to transmit by error passive node

Note 1. This bit is not available in the classical CAN function.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, when the CAN-FD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFESI bit is updated with the ESI bit value of the CAN-FD frame when it has been received, indicating the error state of the transmitting node. In RX or GW mode, 0b is stored to this bit when the received FDF bit is 0b, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0b to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFBRS bit is updated with the BRS bit value of the CAN-FD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CAN-FD frame.

In RX or GW mode, 0b is stored to the CFBRS bit when the received FDF bit is 0b, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0b to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CAN-FD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CAN-FD frame (1).

This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS_n.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS_n.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

(12) Common FIFO Access Data Field p Register n (CFDCFDpFn) (p = 0 to 15, n = 0 to 17)

The CFDCFDpFn register (p = 0 to 15, n = 0 to 17) stores data bytes ($4 \times p$) to data bytes ($4 \times p + 3$) of the message.

Remark In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 640Ch + p x 0004h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB_HH[7:0]								CFDB_HL[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB_LH[7:0]								CFDB_LL[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	CFDB_HH[7:0]	0h	RW	Common FIFO Buffer Data Byte (4 × p + 3)
23 to 16	CFDB_HL[7:0]	0h	RW	Common FIFO Buffer Data Byte (4 × p + 2)
15 to 8	CFDB_LH[7:0]	0h	RW	Common FIFO Buffer Data Byte (4 × p + 1)
7 to 0	CFDB_LL[7:0]	0h	RW	Common FIFO Buffer Data Byte (4 × p)

CFDB_LL[7:0] bits (Common FIFO Buffer Data Byte ($4 \times p$))

The CFDB_LL[7:0] bits store data bytes ($4 \times p$) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 00h, according to their configured data payload size CFDCFCCn.CFPLS.*¹

CFDB_LH[7:0] bits (Common FIFO Buffer Data Byte ($4 \times p + 1$))

The CFDB_LH[7:0] bits store data bytes ($4 \times p + 1$) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 00h, according to their configured data payload size CFDCFCCn.CFPLS.*¹

CFDB_HL[7:0] bits (Common FIFO Buffer Data Byte ($4 \times p + 2$))

The CFDB_HL[7:0] bits store data bytes ($4 \times p + 2$) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 00h, according to their configured data payload size CFDCFCCn.CFPLS.*¹

CFDB_HH[7:0] bits (Common FIFO Buffer Data Byte ($4 \times p + 3$))

The CFDB_HH[7:0] bits store data bytes ($4 \times p + 3$) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 00h, according to their configured data payload size CFDCFCCn.CFPLS.*¹

Note 1. In RX or GW mode, unused data bytes are filled with 00h according to the configured data payload size CFDCFCCn.CFPLS, which is a CAN-FD feature not found in classical CAN.

(13) TX Message Buffer ID Register n (CFDTMIDn) (n = 0 to 383)

The CFDTMIDn register (n = 0 to 383) stores the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1_0000h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	TMIDE	0h	RW	TX Message Buffer IDE bit 0b: STD-ID is transmitted 1b: EXT-ID is transmitted
30	TMRTR	0h	RW	TX Message Buffer RTR bit 0b: Data frame 1b: Remote frame
29	THLEN	0h	RW	Tx History List Entry 0b: Entry not stored in THL after successful TX 1b: Entry stored in THL after successful TX
28 to 0	TMID[28:0]	0h	RW	TX Message Buffer ID Field STD-ID/EXT-ID fields

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

For alignment of these bits in standard and extended frame format, see **7.9.2.2.89 Identifier Bits Alignment**.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CAN-FD format. For a CAN-FD transmission (CFDTMFDCTR.CFFDF = 1b), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

(14) TX Message Buffer Pointer Register n (CFDTMPTRn) (n = 0 to 383)

The CFDTMPTRn register (n = 0 to 383) stores the DLC fields of the message to transmit from the associated buffer.

Access Size :		8, 16, 32 bits														
Address :		<CFD_base> + 1_0004h + n x 0080h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	TMDLC[3:0]	0h	RW	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.
27 to 0	-	All 0	RW	Reserved Whenever it is read, undefined value is read. The written value should always be 0b.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0b.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

(15) TX Message Buffer CAN-FD Control Register n (CFDTMFDCTRn) (n = 0 to 383)

The CFDTMFDCTRn register (n = 0 to 383) shows the status of the FDF, BRS, and ESI bits, including the pointer fields of the CAN-FD frame to be transmitted.

Access Size :		8, 16, 32 bits															
Address :		<CFD_base> + 1_0008h + n x 0080h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	TMPTR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	TMIFL[1:0]		-	-	-	-	-	TMFDF	TMBRS	TMESI	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TMPTR[15:0]	0h	RW	TX Message Buffer Pointer Field
15 to 10	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9, 8	TMIFL[1:0]	0h	RW	TX Message Buffer Information Label Field
7 to 3	-	All 0	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	TMFDF ^{*1}	0h	RW	CAN FD Format bit 0b: No CAN-FD frame to transmit 1b: CAN-FD frame to transmit
1	TMBRS ^{*1}	0h	RW	Bit Rate Switch bit 0b: CAN-FD frame to transmit with no bit rate switch 1b: CAN-FD frame to transmit with bit rate switch
0	TMESI ^{*1}	0h	RW	Error State Indicator bit 0b: CAN-FD frame to transmit by error active node 1b: CAN-FD frame to transmit by error passive node

Note 1. This bit is not available in the classical CAN function.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

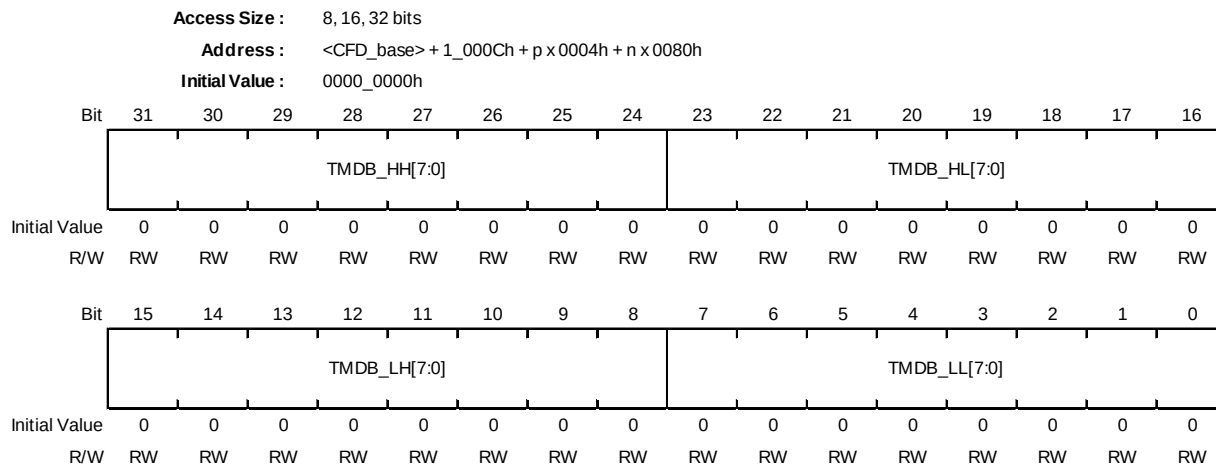
TMPTR[15:0] bits (TX Message Buffer Pointer Field)

The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

(16) TX Message Buffer Data Field p Register n (CFDTMDFp_n) (p = 0 to 15, n = 0 to 383)

The CFDTMDFp_n register (p = 0 to 15, n = 0 to 383) stores data bytes ($4 \times p$) to data bytes ($4 \times p + 3$) of the message to transmit from the associated buffer.



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	TMDB_HH[7:0]	0h	RW	TX Message Buffer Data Byte ($4 \times p + 3$)
23 to 16	TMDB_HL[7:0]	0h	RW	TX Message Buffer Data Byte ($4 \times p + 2$)
15 to 8	TMDB_LH[7:0]	0h	RW	TX Message Buffer Data Byte ($4 \times p + 1$)
7 to 0	TMDB_LL[7:0]	0h	RW	TX Message Buffer Data Byte ($4 \times p$)

TMDB_LL[7:0] bits (TX Message Buffer Data Byte ($4 \times p$))

Data bytes ($4 \times p$)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte ($4 \times p + 1$))

Data bytes ($4 \times p + 1$)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte ($4 \times p + 2$))

Data bytes ($4 \times p + 2$)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte ($4 \times p + 3$))

Data bytes ($4 \times p + 3$)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

7.9.3 Operation

7.9.3.1 Overview

The modes of the CAN-FD module can be classified into 2 groups:

- Global modes
- Channel modes

7.9.3.2 Global Modes

Global modes are applicable for the complete CAN-FD module. The global modes of the CAN-FD module are:

- Global Sleep
- Global Reset
- Global Halt
- Global Operation

Figure 7.9-2 shows the possible transitions between the Global modes.

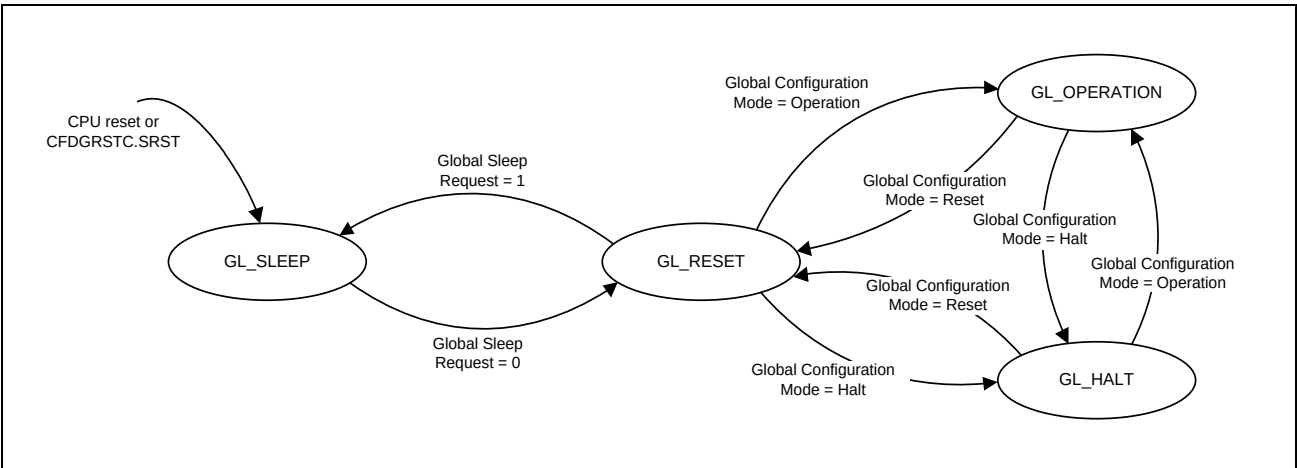


Figure 7.9-2 Transition between CAN-FD Global Modes

Changes in the Global mode can affect the Channel mode. **Table 7.9-15** shows the effect of a Global mode transition on a Channel mode.

Table 7.9-15 Possible CAN-FD Channel Modes and Global Modes

Current Global Mode	Target Global Mode			
	Sleep	Reset	Halt	Operation
Sleep	—	Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A	—	—
Reset	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	—

7.9.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module automatically enters Global Sleep mode.

The CAN-FD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets all Channel Sleep Request bits and forces all channels into the Channel Sleep mode. Sleep mode is used for power saving purpose. When CAN-FD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CAN-FD module are suspended.

Read access from all registers is still possible and all register values are preserved.

In Global Sleep mode, RAM access is prohibited because the logic which generates a RAM address does not operate.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

Figure 7.9-3 shows the procedure for entering Global Sleep mode and **Figure 7.9-4** shows the procedure for exiting Global Sleep mode.

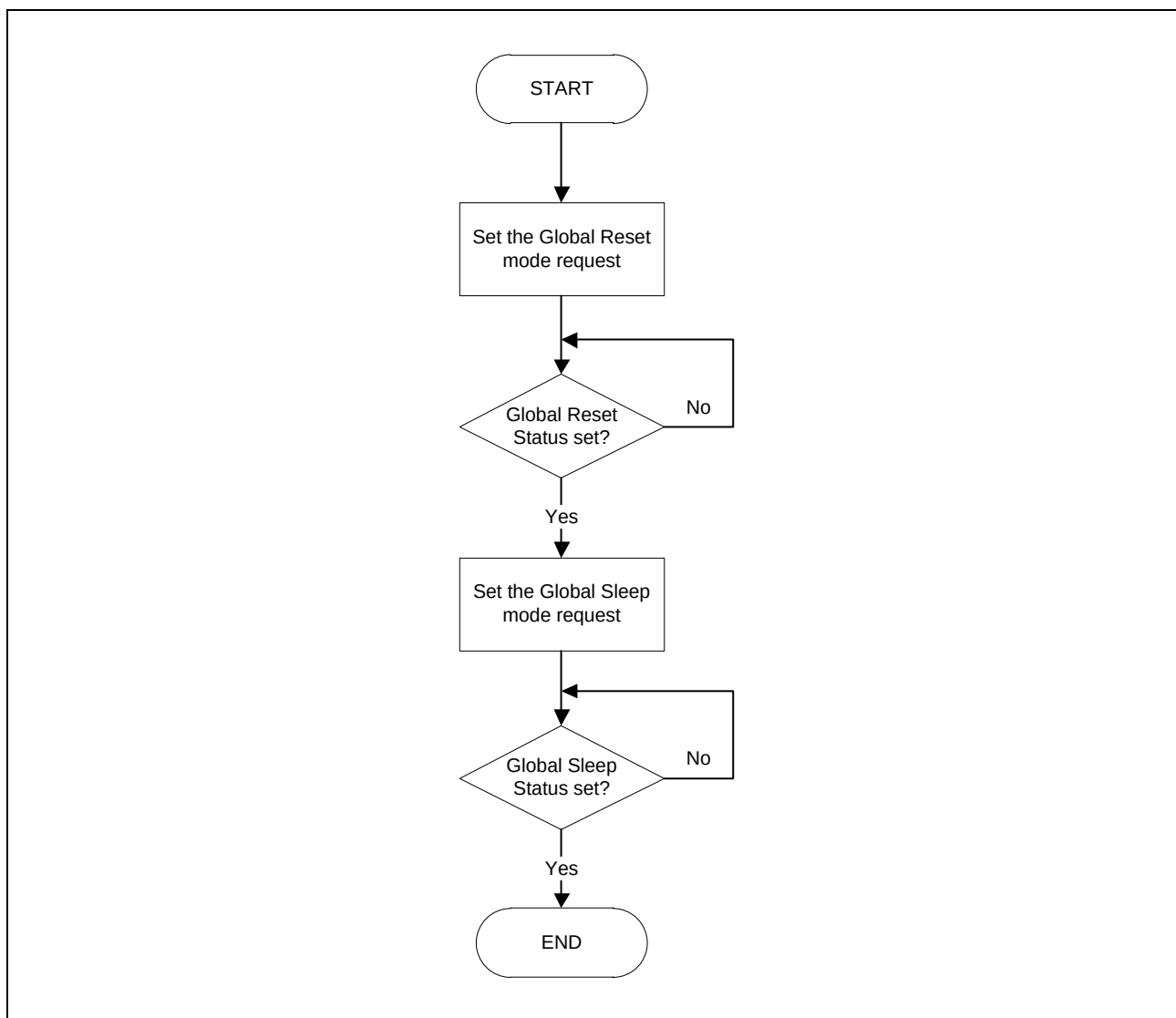


Figure 7.9-3 Procedure for Entering Global Sleep Mode

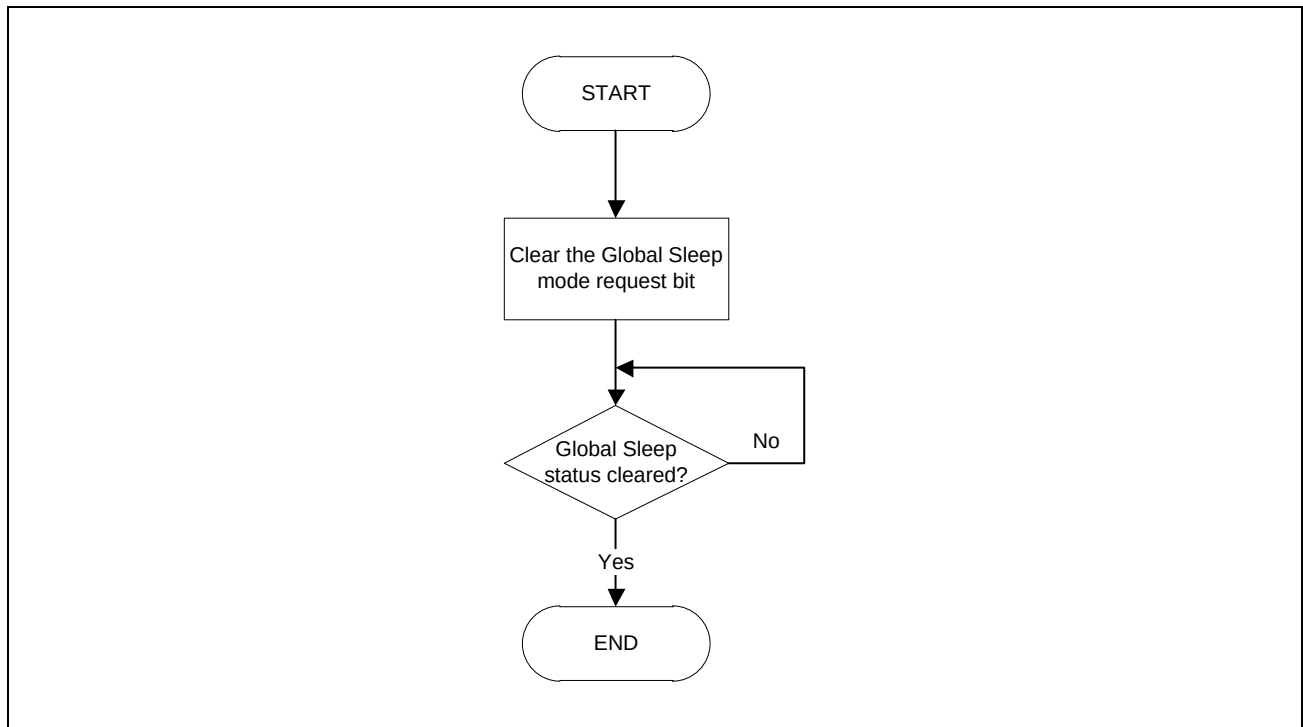


Figure 7.9-4 Procedure for Exiting Global Sleep Mode

7.9.3.2.2 Global Reset Mode

The CAN-FD module enters the Global Reset mode in the following ways:

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Reset mode while the CAN-FD module is in Global Halt or Global Operation mode.
- Global Sleep Mode Request bit is cleared while CAN-FD module is in Global Sleep mode.

In Global Reset mode, all CAN-FD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and all channel TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their device reset values and the CAN-FD module can be configured.

See **7.9.3.4 Global Mode and Channel Mode Transition Interactions** for a detailed description of the behavior of all registers when transition to Global Reset mode.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register to 01b sets all Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers to 01b and forces all channels into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDCnCTR.CHMDC[1:0] of related channel already set to 01b).

After setting Global Mode Control bits CFDGCTR.GMDC[1:0] to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

Figure 7.9-5 shows the procedure for entering Global Reset mode and **Figure 7.9-6** shows the procedure for exiting Global Reset mode.

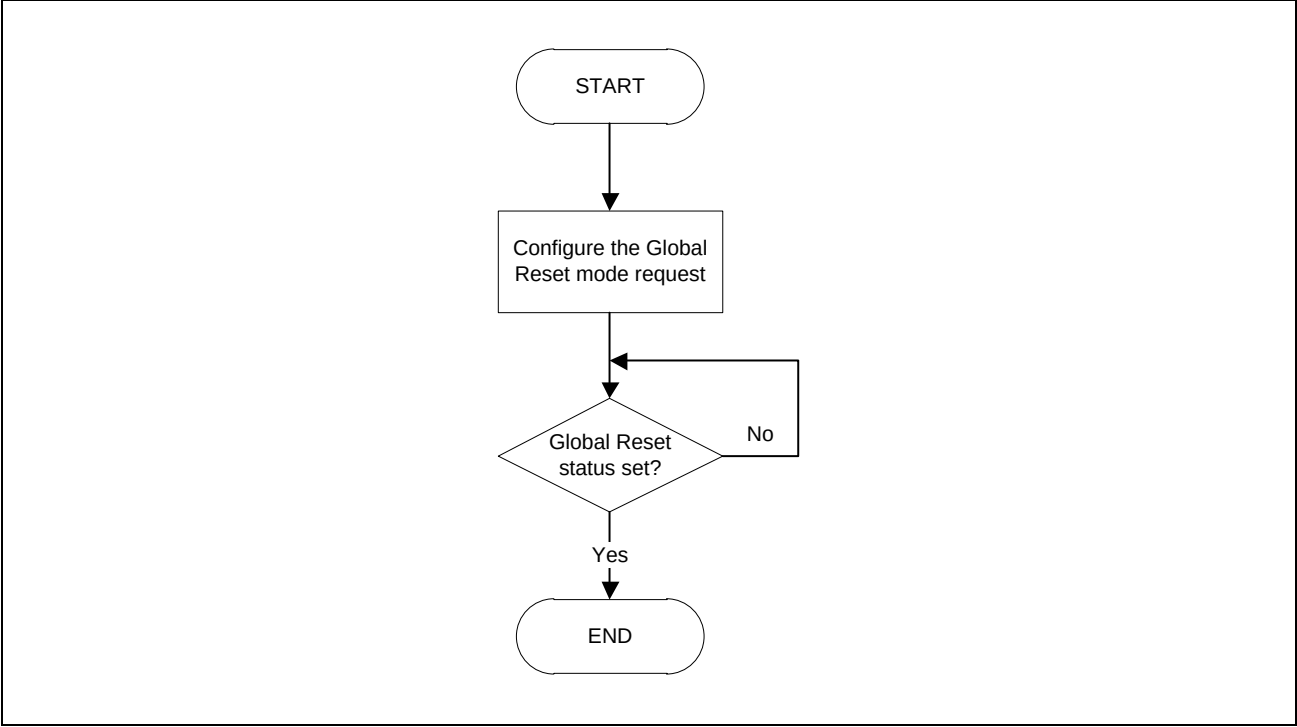


Figure 7.9-5 Procedure for Entering Global Reset Mode

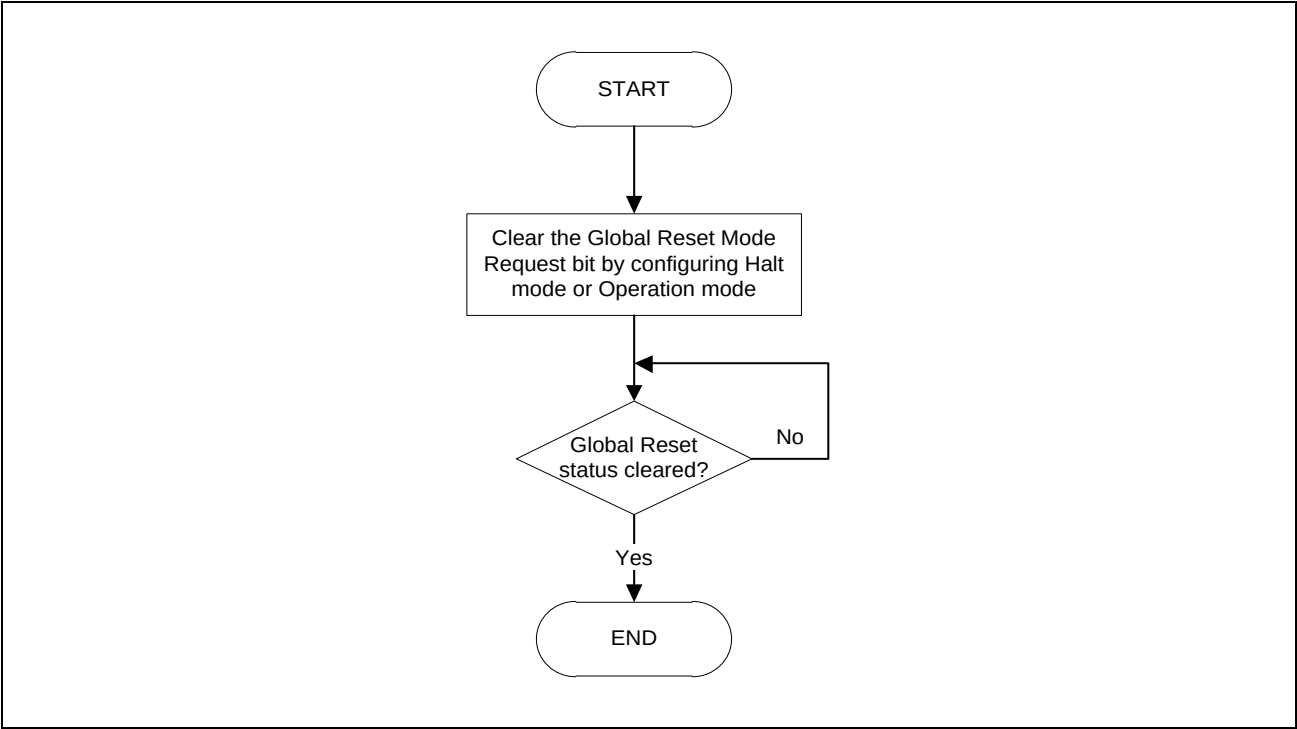


Figure 7.9-6 Procedure for Exiting Global Reset Mode

7.9.3.2.3 Global Halt Mode

The CAN-FD module enters the Global Halt mode in the following ways:

- Global Mode Control bits `CFDGCTR.GMDC[1:0]` in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Reset mode:
 - The channel is in either Channel Reset or Channel Sleep mode and remains in this mode.
- Global Mode Control bits `CFDGCTR.GMDC[1:0]` in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Operation mode:
 - All channels in Channel Reset, Channel Halt, or Channel Sleep mode remain in this mode
 - All channels in Channel Operation mode transit to Channel Halt mode
 - Global Halt Mode Status bit is set when all channels have left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off state, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CAN-FD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off state that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See **7.9.3.4 Global Mode and Channel Mode Transition Interactions** for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt mode by setting the Global Mode Control bits `CFDGCTR.GMDC[1:0]` in the Global Control Register to 10b sets all Channel Mode Control bits `CFDCnCTR.CHMDC[1:0]` in the Channel Control Registers to 10b for the channels that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For channels that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CAN-FD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bits `CFDGCTR.GMDC[1:0]` to Halt mode, it is necessary to confirm that the Halt Mode Status bit `CFDGSTS.GHLTSTS` in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming `CFDGSTS.GHLTSTS` is set.

Figure 7.9-7 shows the procedure for entering Global Halt mode and **Figure 7.9-8** shows the procedure for exiting Global Halt mode.

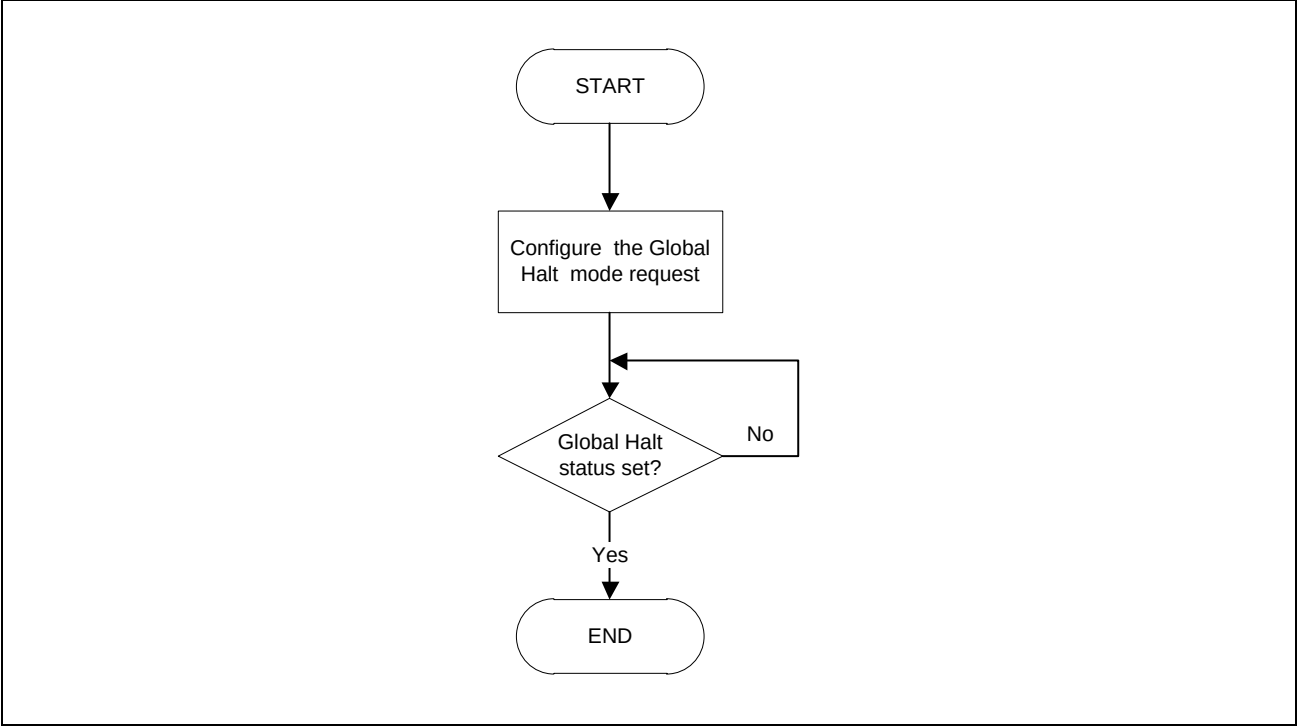


Figure 7.9-7 Procedure for Entering Global Halt Mode

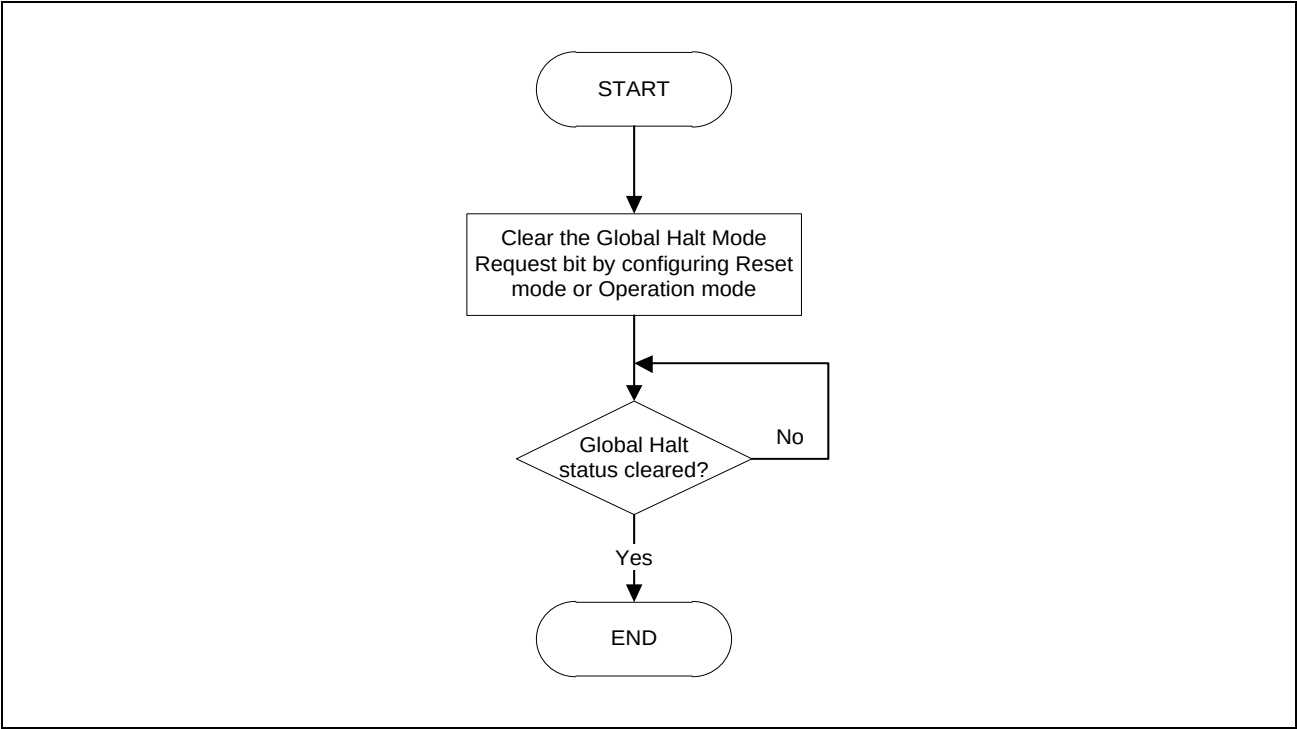


Figure 7.9-8 Procedure for Exiting Global Halt Mode

7.9.3.2.4 Global Operation Mode

The CAN-FD module enters the Global Operation mode when the Global Mode Configuration bits are set to Global Operation mode.

The CAN-FD channels can only be set to Channel Operation mode and start CAN communication when CAN-FD is in Global Operation mode.

After setting the Global Mode Control bits `CFDGCTR.GMDC[1:0]` to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit `CFDGSTS.GRSTSTS` and the Global Halt Mode Status bit `CFDGSTS.GHLTSTS` in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before `CFDGCTR.GMDC` can be modified again.

Figure 7.9-9 shows the procedure for entering Global Operation mode and **Figure 7.9-10** shows the procedure for exiting Global Operation mode.

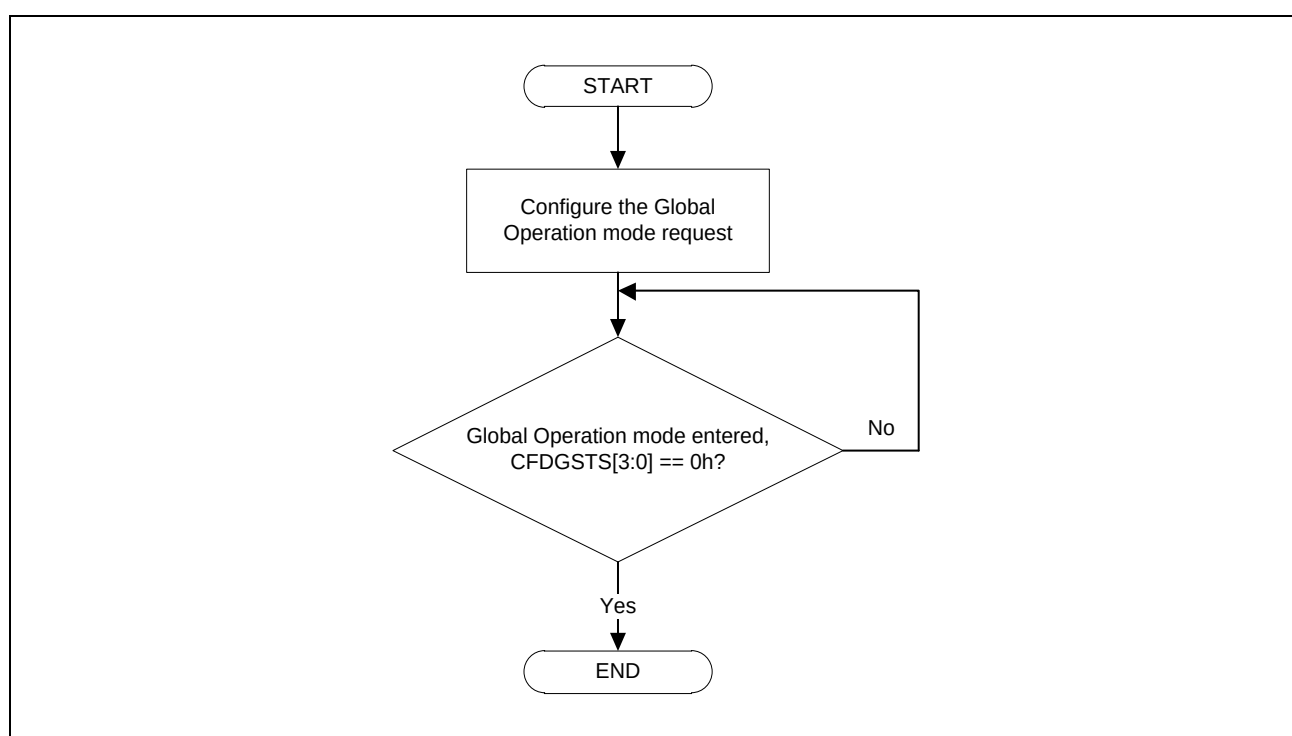


Figure 7.9-9 Procedure for Entering Global Operation Mode

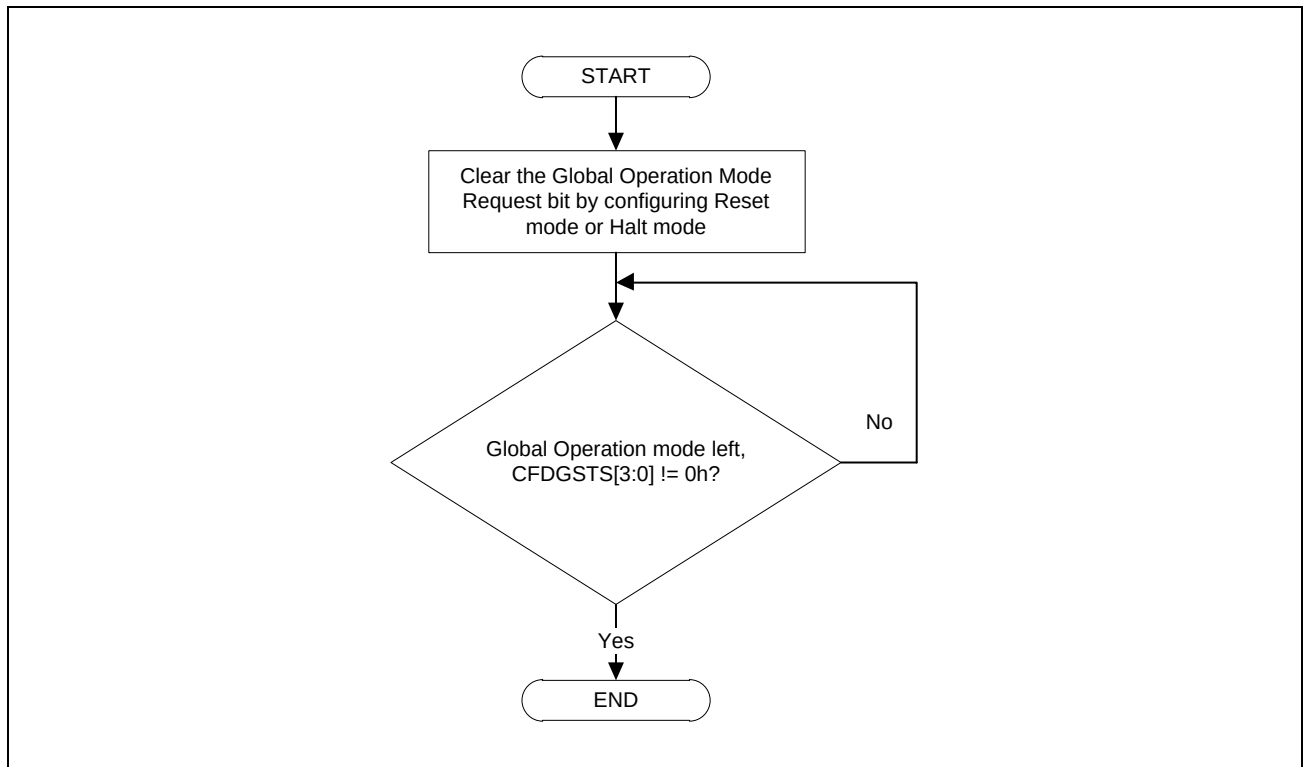


Figure 7.9-10 Procedure for Exiting Global Operation Mode

7.9.3.3 Channel Modes

Each CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep

Figure 7.9-11 shows possible transitions between the channel modes.

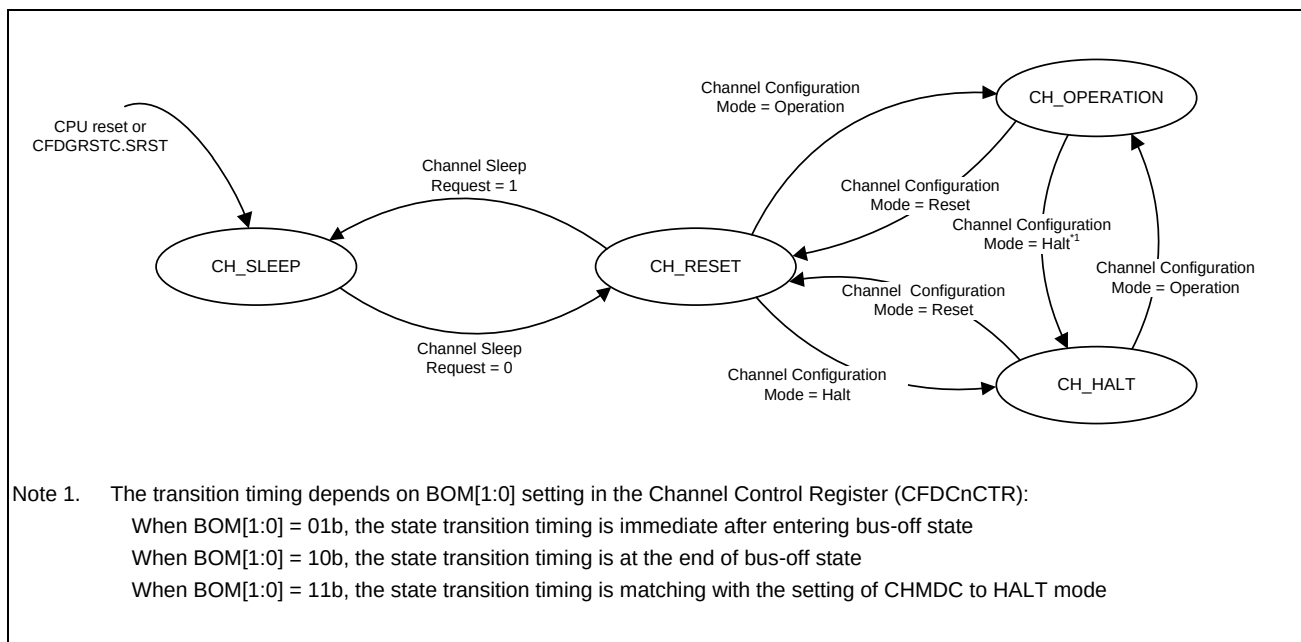


Figure 7.9-11 Transition between CAN Channel Modes

7.9.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, each CAN channel of the CAN-FD module automatically enters Channel Sleep mode.

Each CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

7.9.3.3.2 CAN Channel Reset Mode

A CAN-FD CAN channel enters the CAN Channel Reset mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel-related transmission control bits are cleared and the channel-related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See **7.9.3.4 Global Mode and Channel Mode Transition Interactions** for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDCnSTS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See **Table 7.9-16** for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

Table 7.9-16 Transition Behavior in CAN Reset Mode and Halt Mode

Mode	State		
	Receiver	Transmitter	Bus-Off
CAN Channel Reset mode (CFDCnCTR.CHMDC[1:0] = 01b)	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing reception.* ¹	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing transmission.* ¹	The CAN channel transits to Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt Mode (CFDCnCTR.CHMDC[1:0] = 10b)	The CAN channel transits to Channel Halt mode at the end of the ongoing reception or error.* ²	The CAN channel transits to Channel Halt mode after completion of the ongoing transmission.	When CFDCnCTR.BOM[1:0] is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDCnCTR.BOM[1:0] is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 11b, the CAN channel transits to Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at a dominant level after an error flag, software can detect this situation by monitoring the channel related Bus Lock flag and resolve the lock condition by setting the CAN channel to Channel Reset mode.

7.9.3.3.3 CAN Channel Halt Mode

A CAN-FD CAN channel enters the CAN Channel Halt mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers are configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] are set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for the channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure Channel Test modes.

See **7.9.3.4 Global Mode and Channel Mode Transition Interactions** for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDCnSTS.CHLTSTS in the related Channel Status Register is updated to indicate a successful transition to Channel Halt mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See **Table 7.9-16** for the behavior of transitioning to Channel Halt mode while CAN communication is ongoing.

7.9.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDCnCTR.CHMDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering CAN Channel Operation mode, the CFDCnSTS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see **Figure 7.9-12**):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

Note: The channel may receive its own message simultaneously when Self Test mode is enabled.

- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDCnSTS.CRSTSTS and the Channel Halt Mode Status bit CFDCnSTS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDCnCTR.CHMDC[1:0] bits can be changed again.

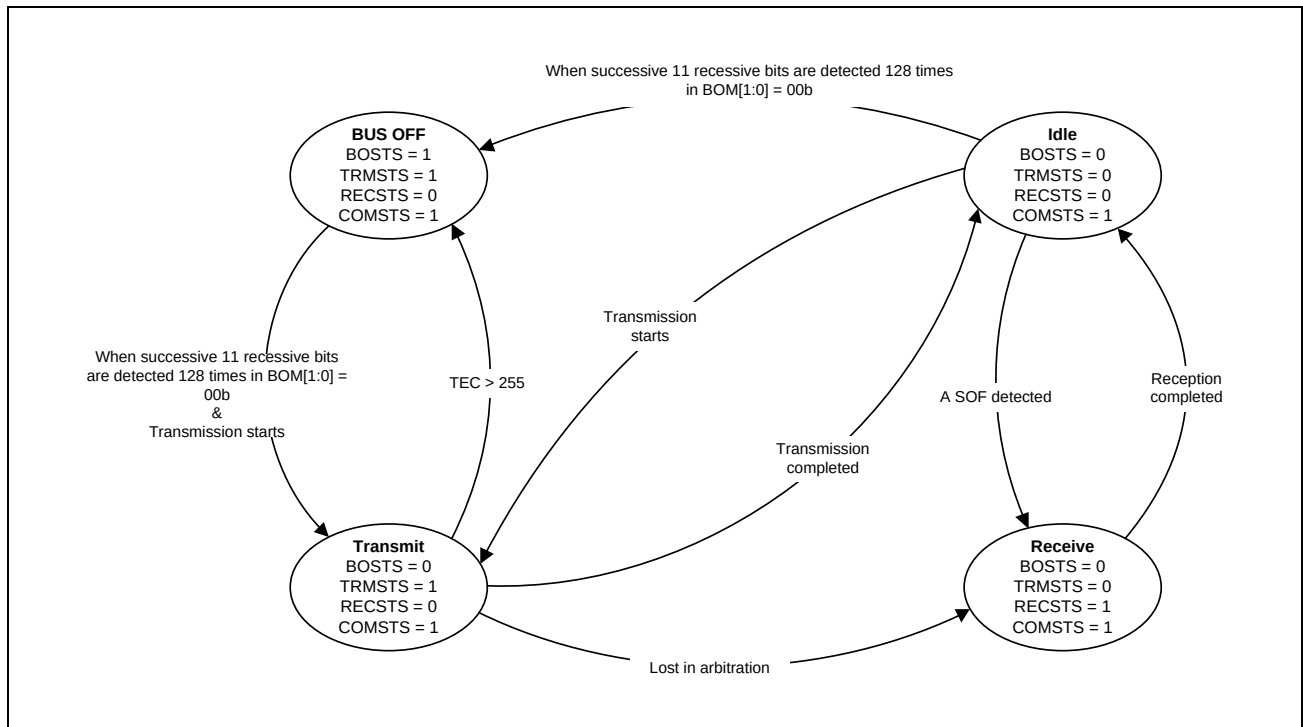


Figure 7.9-12 Sub-Modes of CAN Channel Operation Mode (Only when BOM[1:0] = 00b)

7.9.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDCnCTR.BOM[1:0] = 00b:
Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0b. The Bus- Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM[1:0] = 01b:
The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set in this case.
- CFDCnCTR.BOM[1:0] = 10b:
The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0b and the Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM[1:0] = 11b:
Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
TEC and REC counters are initialized to 0b and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set. Without setting CFDCnCTR.CHMDC[1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDCnCTR.BOM[1:0] = 00b.

Note: If the recovery from bus-off occurs normally in Channel Halt mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDCnERFL.BORF is set.

When software writes to the CFDCnCTR.CHMDC[1:0] bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDCnCTR.CHMDC[1:0] bits to Channel Halt mode request is performed when the CFDCnCTR.CHMDC[1:0] bits value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDCnCTR.RTBO to 1b. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0b.

Before setting CFDCnCTR.RTBO to 1b, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX or GW mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSn.TMTRF[1:0]). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTSn.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTSn.CFEMP).

The CFDCnCTR.RTBO bit should be used for bus-off recovery only when CFDCnCTR.BOM[1:0] is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 7.9-17 shows the settings for the Bus-Off Entry flag CFDCnERFL.BOEF and the Bus-Off Recovery flag CFDCnERFL.BORF for the different configurations of CFDCnCTR.BOM[1:0].

Table 7.9-17 Behavior of Bus-off Entry and Recovery Flags

BOM[1:0]	BOEF Bit Setting	BORF Bit Setting
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDCnCTR.RTBO set to 1b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDCnCTR.RTBO to 1b
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in **Figure 7.9-13**.

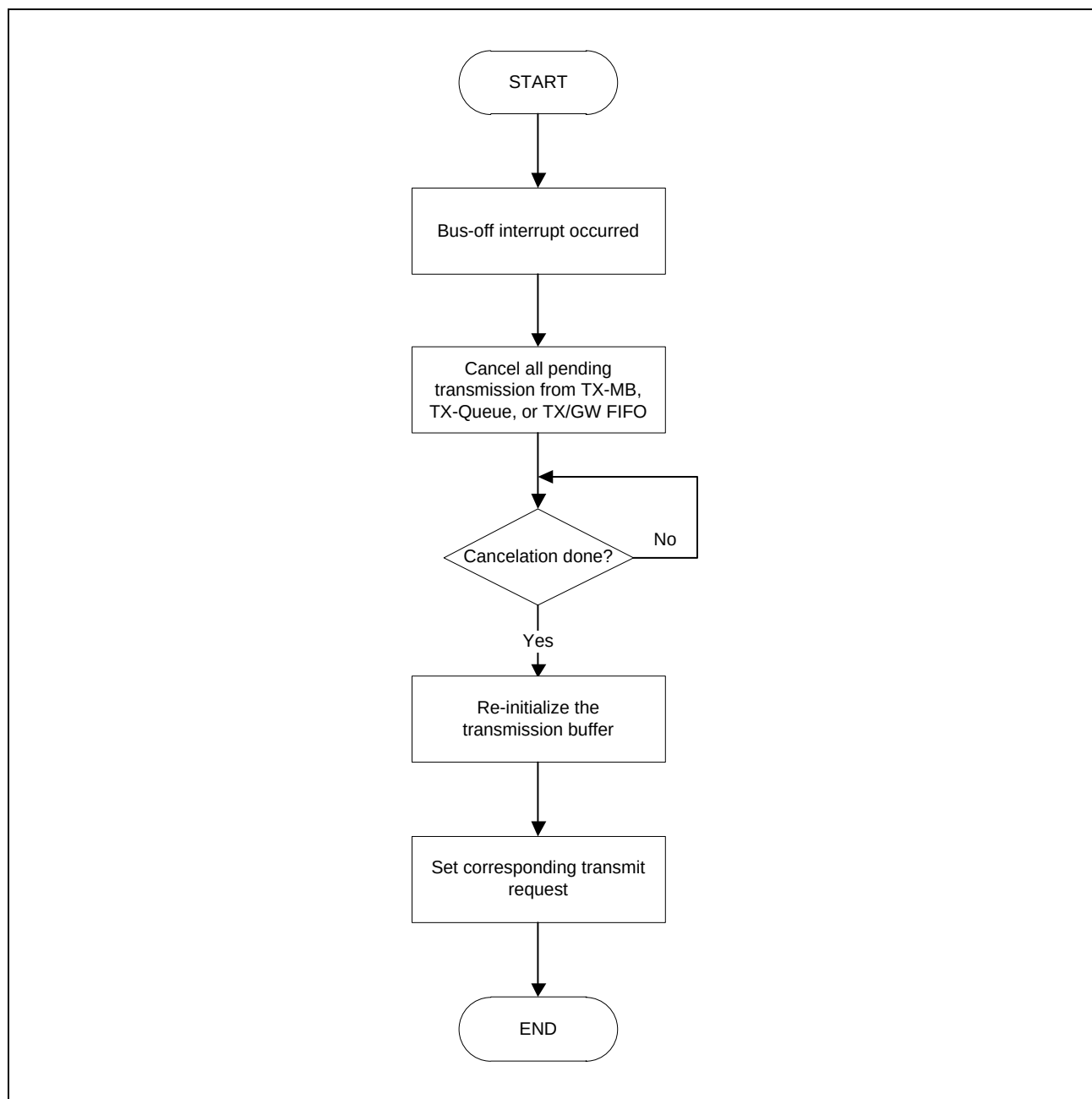


Figure 7.9-13 Transmission Re-Initialization during Bus-Off State

7.9.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers does not affect the Global Mode Control bits CFDGCTR.GMDC[1:0].
- Changing the Global Mode Control bits CFDGCTR.GMDC[1:0] affects the channel mode control as described in **Table 7.9-18**.

Table 7.9-18 Instruction between Global and Channel Mode Transition

Global Mode Change	Channel Mode	Channel Mode Transition Action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel transits to Sleep mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
	Operation	Channel mode control is set to Reset mode, channel transits to Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel transits to Halt mode after communication finished

7.9.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

From	To	Maximum Transition Time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1,*3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.

Note 3. TQ, CAN frame, and CAN bits are related to the individual channels. For maximum transition time, the channel with the lowest baud rate must be used.

7.9.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

From	To	Maximum transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*2
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1,*3

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDCnCTR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 3. In general, if the baud rate prescaler value CFDCnNCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

7.9.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud rate setting (nominal and data rate)
- CAN-FD setting
- Acceptance filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

7.9.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

7.9.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

- Each segment setting SS = Fixed to 1 TQ
 TSEG1 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
 TSEG2 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
 SJW = Refer to (CFDCnNCFG) and (CFDCnDCFG)
 SS + TSEG1 + TSEG2 = 5 to 49 TQs for data bit rate and 8 to 385 for nominal bit rate
- Limitations on TSEG1, TSEG2 and SJW
 TSEG1 (N) > TSEG2 (N) ≥ SJW (N)
 TSEG1 (D) ≥ TSEG2 (D) ≥ SJW (D)
 When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDCnDCFG to valid values.

Table 7.9-19 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 7.9-19 Transition Behavior in CAN Reset Mode and Halt Mode (1/2)

1 Bit	Set Value (TQ)				Sample Point*1 (%)
	SS	TSEG1	TSEG2	SJW	
5 TQ	1	2	2	1	60.00
8 TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10 TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12 TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15 TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16 TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20 TQ	1	12	7	1	65.00
	1	13	6	1	70.00

Table 7.9-19 Transition Behavior in CAN Reset Mode and Halt Mode (2/2)

1 Bit	Set Value (TQ)				Sample Point*1 (%)
	SS	TSEG1	TSEG2	SJW	
24 TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50 TQ	1	39	10	4	80.00

Note 1. Sample Point (in Case of 75%)

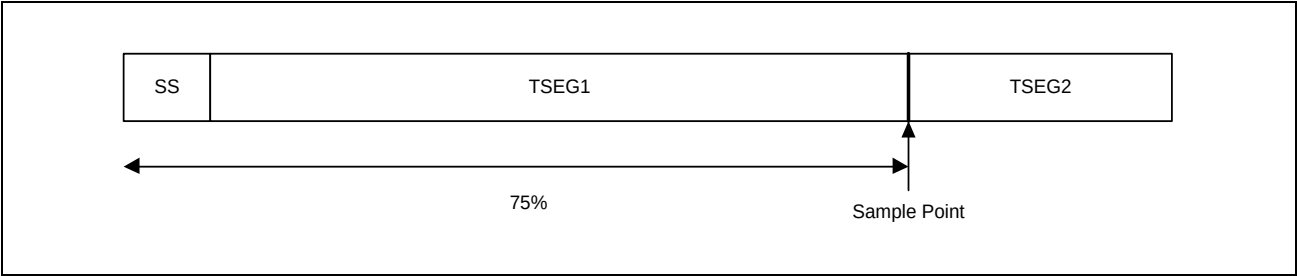


Figure 7.9-14 Sample Point (in Case of 75%)

7.9.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for each channel using the related CFDCnNCFG and CFDCnDCFG registers.

Figure 7.9-15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

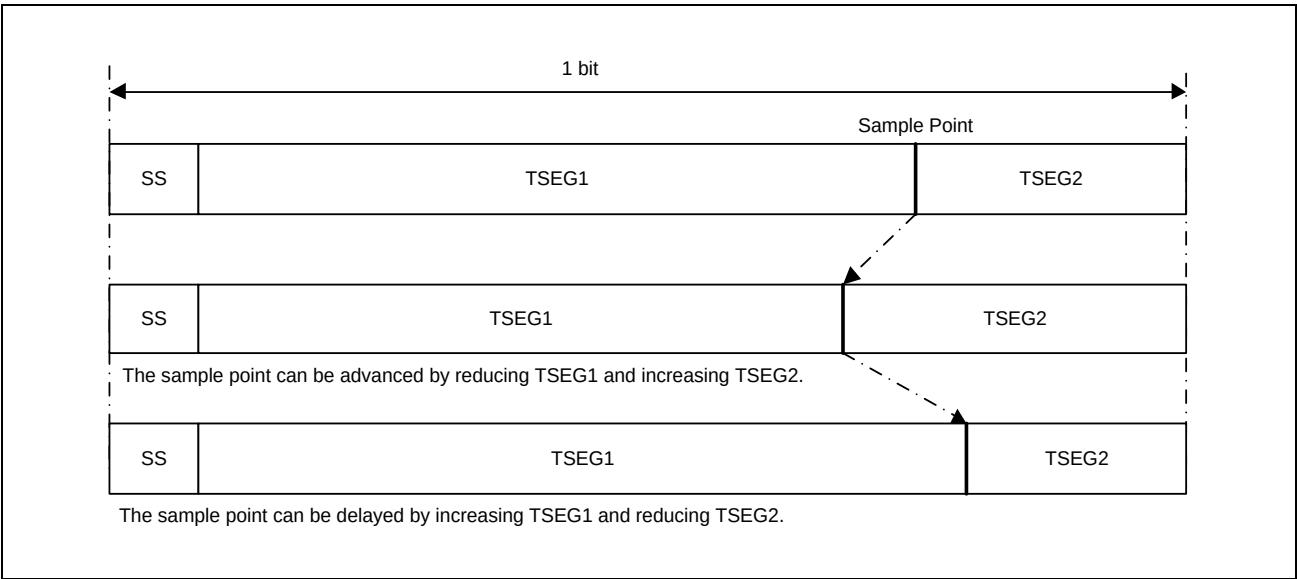


Figure 7.9-15 Segment Composition of a Bit and the Sample Point

1. SS: Synchronization Segment
This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 7.9-15 shows only one symbolic sample point.

7.9.4.1.3 Baud Rate

Either CANFD_0_pclk or CANFD_0_clkc can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baud rate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 7.9-16 shows a block diagram of the circuit that generates the CAN channel system clock, Table 7.9-20 and Table 7.9-21 show baud rate examples.

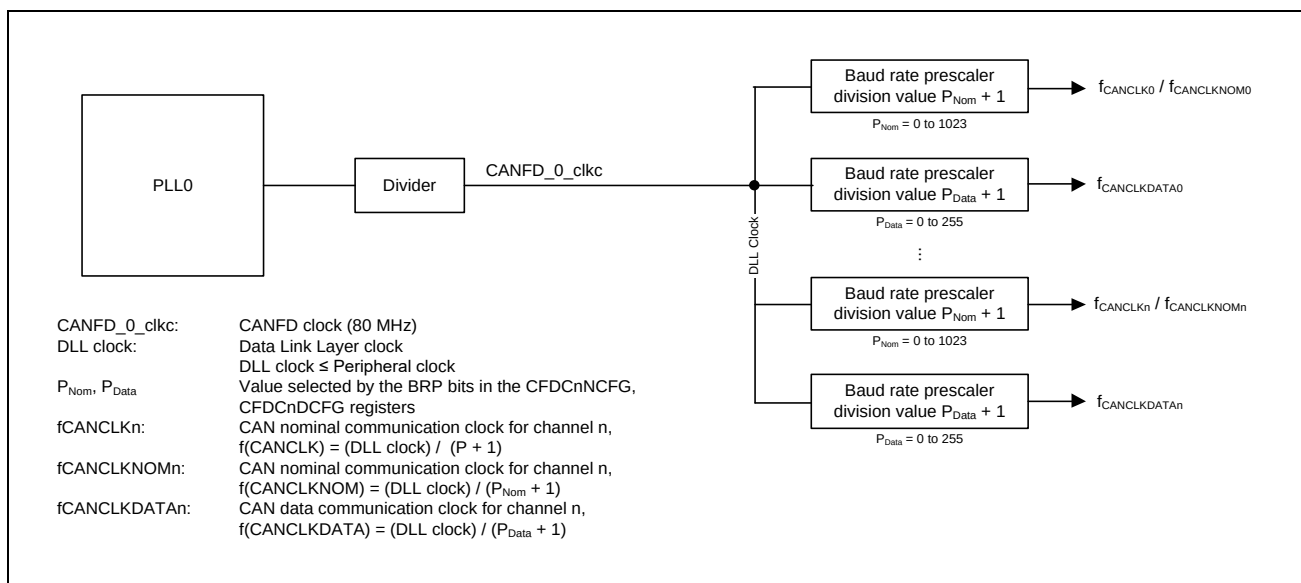


Figure 7.9-16 Block Diagram of the Circuit that Generates the CAN Channel Communication Clock

Table 7.9-20 Nominal Baud Rate Calculation Formula and Example CAN Communication Configurations

Baud Rate Calculation Formula	(DLL Clock) (Baud Rate Prescaler Divide-By-N Value*1) × (Number of TQs in One Bit)			
	80 MHz	40 MHz	100 MHz	75 MHz
1 Mbps	8 TQ (10) 20 TQ (4)	8 TQ (5) 20 TQ (2)	10 TQ (10) 20 TQ (5)	15 TQ (5)
500 Kbps	8 TQ (20) 20 TQ (8)	8 TQ (10) 20 TQ (4)	10 TQ (20) 20 TQ (10)	15 TQ (10)
250 Kbps	8 TQ (40) 20 TQ (16)	8 TQ (20) 20 TQ (8)	10 TQ (40) 20 TQ (20)	15 TQ (20)
125 Kbps	8 TQ (80) 20 TQ (32)	8 TQ (40) 20 TQ (16)	10 TQ (80) 20 TQ (40)	15 TQ (40)
83.3 Kbps	8 TQ (120) 12 TQ (80) 16 TQ (60) 24 TQ (40)	8 TQ (60) 12 TQ (40) 16 TQ (30) 24 TQ (20)	8 TQ (150) 10 TQ (120) 12 TQ (100) 15 TQ (80) 16 TQ (75) 20 TQ (60) 24 TQ (50)	10 TQ (90) 12 TQ (75) 15 TQ (60) 20 TQ (45)
33.3 Kbps	8 TQ (300) 12 TQ (200) 16 TQ (150) 20 TQ (120) 24 TQ (100)	8 TQ (150) 12 TQ (100) 16 TQ (75) 20 TQ (60) 24 TQ (50)	8 TQ (375) 10 TQ (300) 12 TQ (250) 15 TQ (200) 20 TQ (150) 24 TQ (125)	10 TQ (225) 15 TQ (150) 18 TQ (125)

Note: Shown in () are the baud rate prescaler divided-by-N values.

Note 1. Baud rate prescaler divide-by-N value = $P + 1$ ($P = 0$ to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

Table 7.9-21 Baud Rate Calculation Example for Nominal and Data Bit Rate CAN Communication Configurations

Baud Rate Calculation Formula	(DLL Clock) (Baud Rate Prescaler Divide-By-N Value*1) × (Number of TQs in One Bit)			
	80 MHz	40 MHz	100 MHz	75 MHz
Nominal 1 Mbps	80 TQ (1)	40 TQ (1)	100 TQ (1)	75 TQ (1)
Data 8 Mbps	10 TQ (1)	5 TQ (1)	Not possible	Not possible
Nominal 1 Mbps	80 TQ (1)	40 TQ (1)	100 TQ (1)	75 TQ (1)
Data 5 Mbps	16 TQ (1)	8 TQ (1)	20 TQ (1)	15 TQ (1)
Nominal 500 Kbps	160 TQ (1)	80 TQ (1)	200 TQ (1)	150 TQ (1)
Data 2 Mbps	40 TQ (1)	20 TQ (1)	50 TQ (1)	Not possible

Note: Shown in () are the baud rate prescaler divided-by-N values.

Note 1. Baud rate prescaler divide-by-N value = $P + 1$ ($P = 0$ to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means $CFDCnNCFG.NBRP = CFDCnDCFG.DBRP$.

Additionally, if transceiver delay compensation is used, do not program the $CFDCnDCFG.DBRP[7:0]$ bits to be greater than 1, as 1 means divide by 2.

7.9.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 7.9-17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

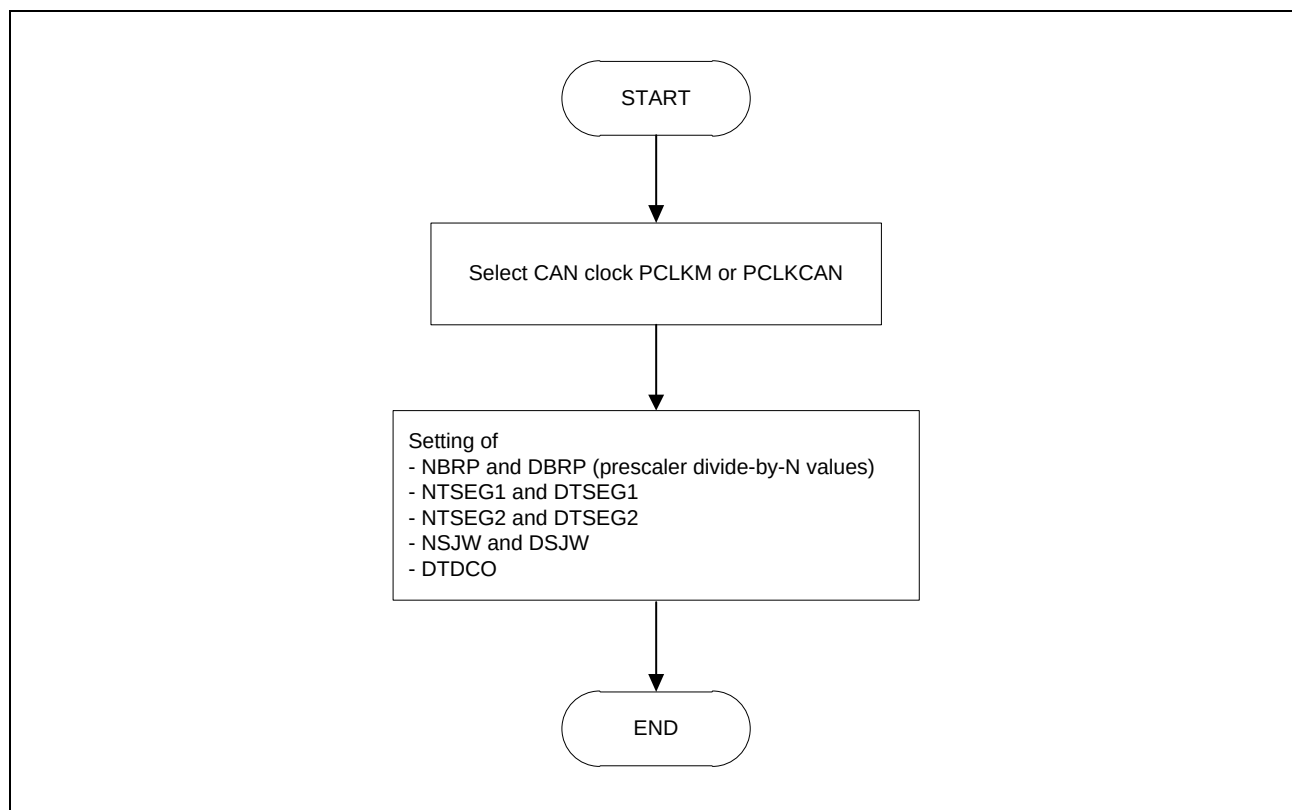


Figure 7.9-17 Procedure for Setting the CAN Bit Timing and Baud Rate

7.9.4.1.5 Transmitter Delay Compensation

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN-FD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN-FD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDCnFDSTS.TDCR) as shown in **Figure 7.9-18**.

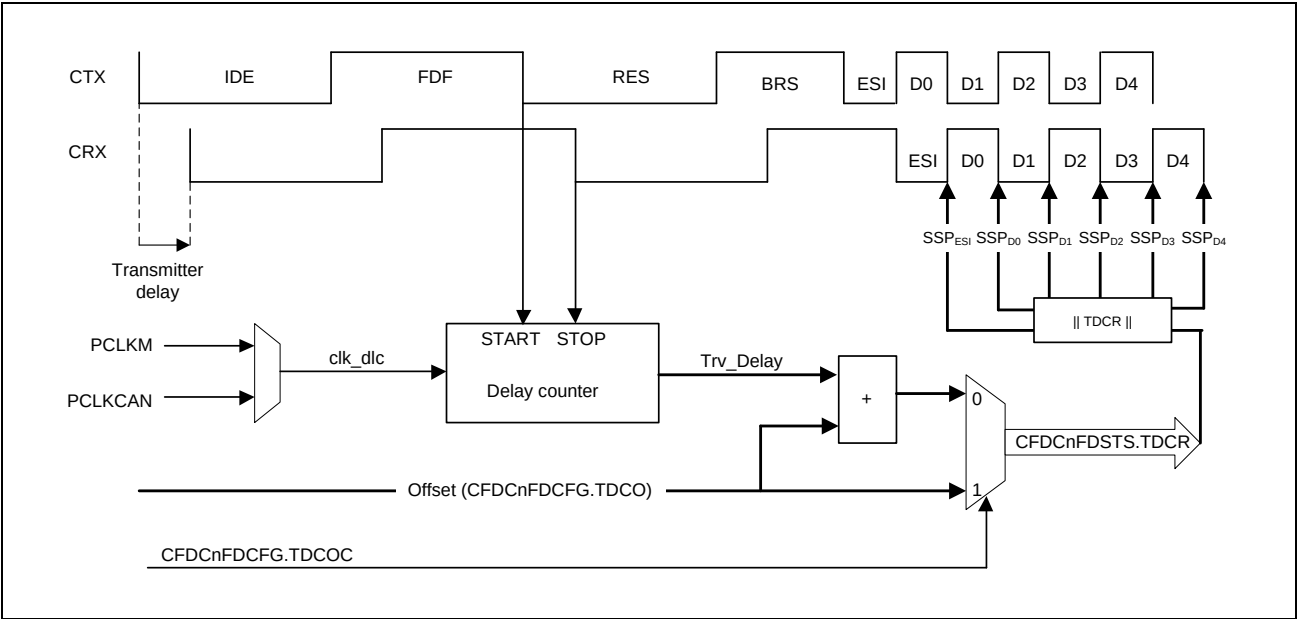


Figure 7.9-18 Transmitter Delay Compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles. The delay is counted up by 1 for each started clock until the dominant value is seen on CAN_RX. **Figure 7.9-19** shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

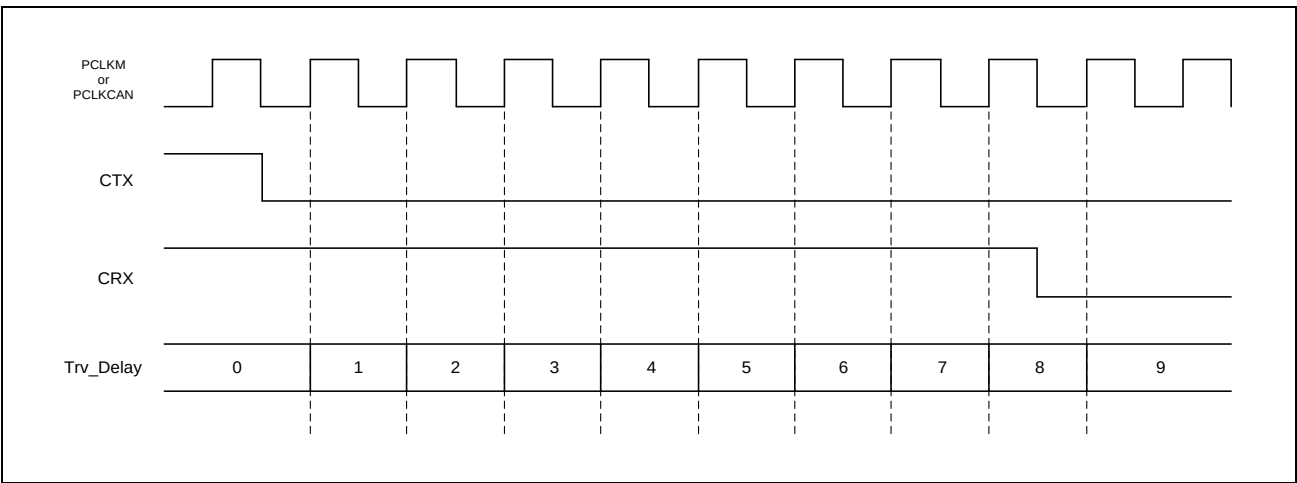


Figure 7.9-19 Trv_Delay Measurement Example

The SSP is calculated by taking the result from CFDCnFDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 7.9-20 shows the positioning of the secondary sample point. When CFDCnFDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (Sync Segment data + TSEG1 data) to position the SSP to a theoretical location of the sample point.

If the CFDCnFDCFG.TDCOC is equal to 1, the SSP is defined by CFDCnFDCFG.TDCO. If CFDCnDCFG.DBRP[7:0] is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

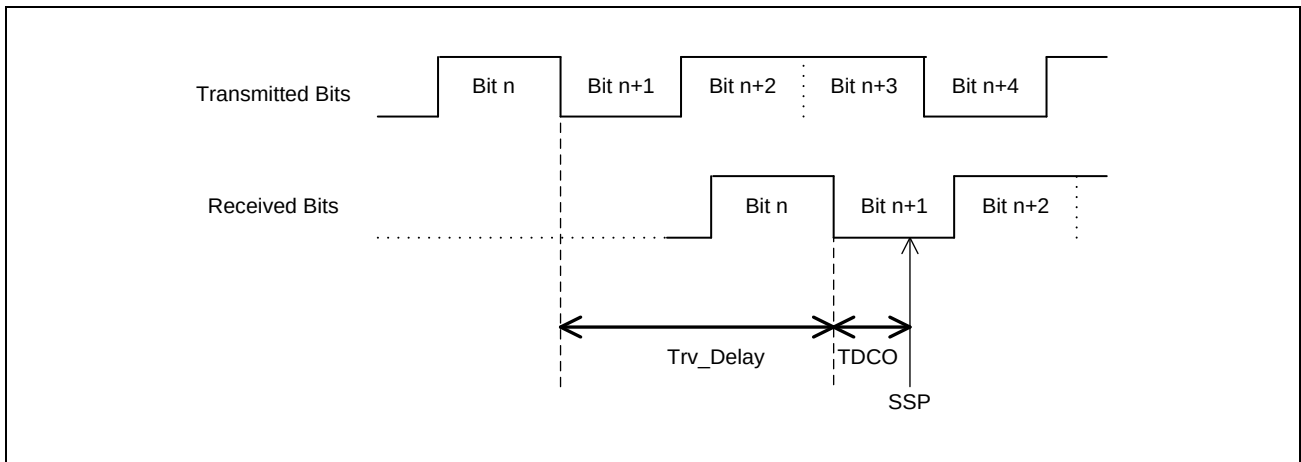


Figure 7.9-20 Position of the Secondary Sample Point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CAN-FD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDCnNCFG.NBRP and CFDCnDCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in **Figure 7.9-21**.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means $CFDCnNCFG.NBRP = CFDCnDCFG.DBRP$.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

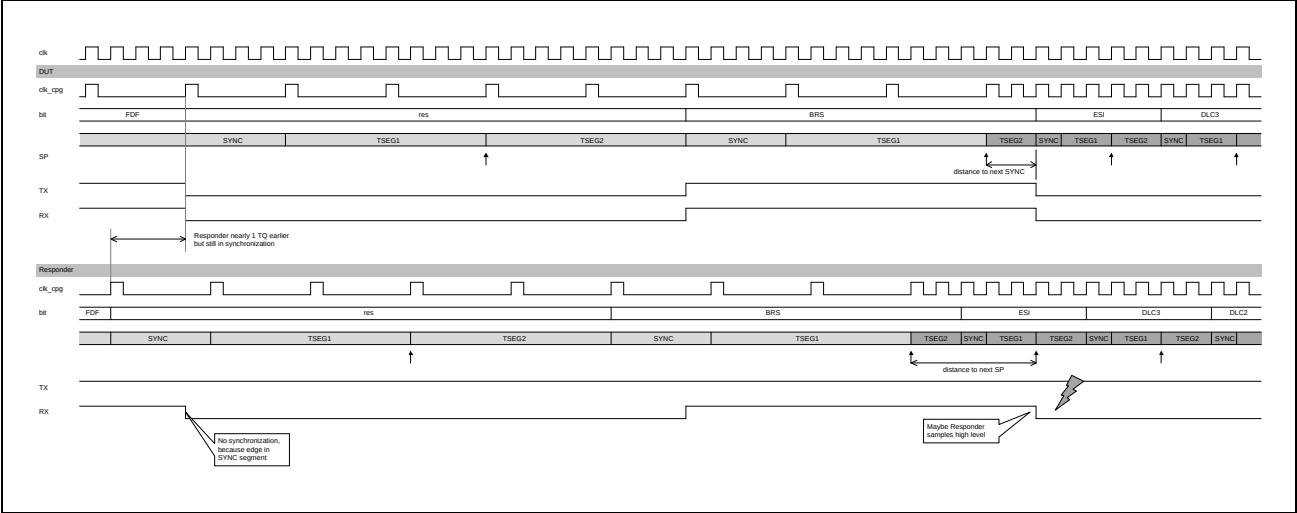


Figure 7.9-21 Loss of Synchronization between 2 CAN Nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDCnFDCFG.TDCE = 1b, CFDCnFDCFG.TDCOC = 0b).

Figure 7.9-22 shows the read flow to get the measured transmitter delay compensation result.

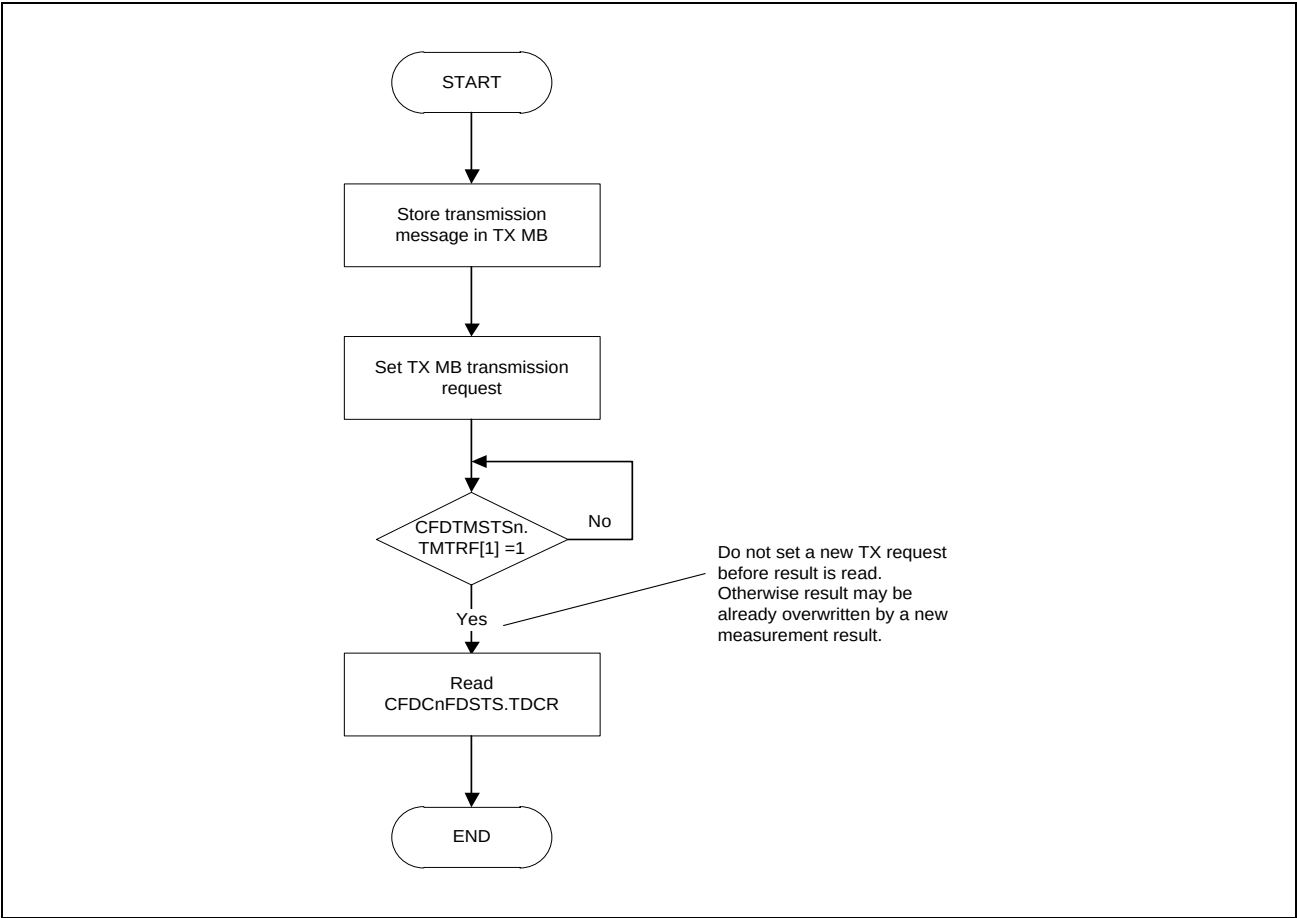


Figure 7.9-22 TDC Result Read Flow

7.9.4.1.6 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module enters Global Sleep mode automatically.

To enable configuration of the CAN-FD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0b.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CAN-FD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after hardware resets the random data presented in the RAM.

Do not access registers of the CAN-FD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, each required CAN channel must be configured such as CAN bit timing. For this configuration, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

For this, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 7.9-23 shows the configuration procedure. For details about each step, see **7.9.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)**, **7.9.6 FIFO Buffers and Normal Message Buffer Configuration**, **7.9.7 Interrupt and DMA**, and **7.9.4.1.3 Baud Rate**.

The CAN-FD module does not perform RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

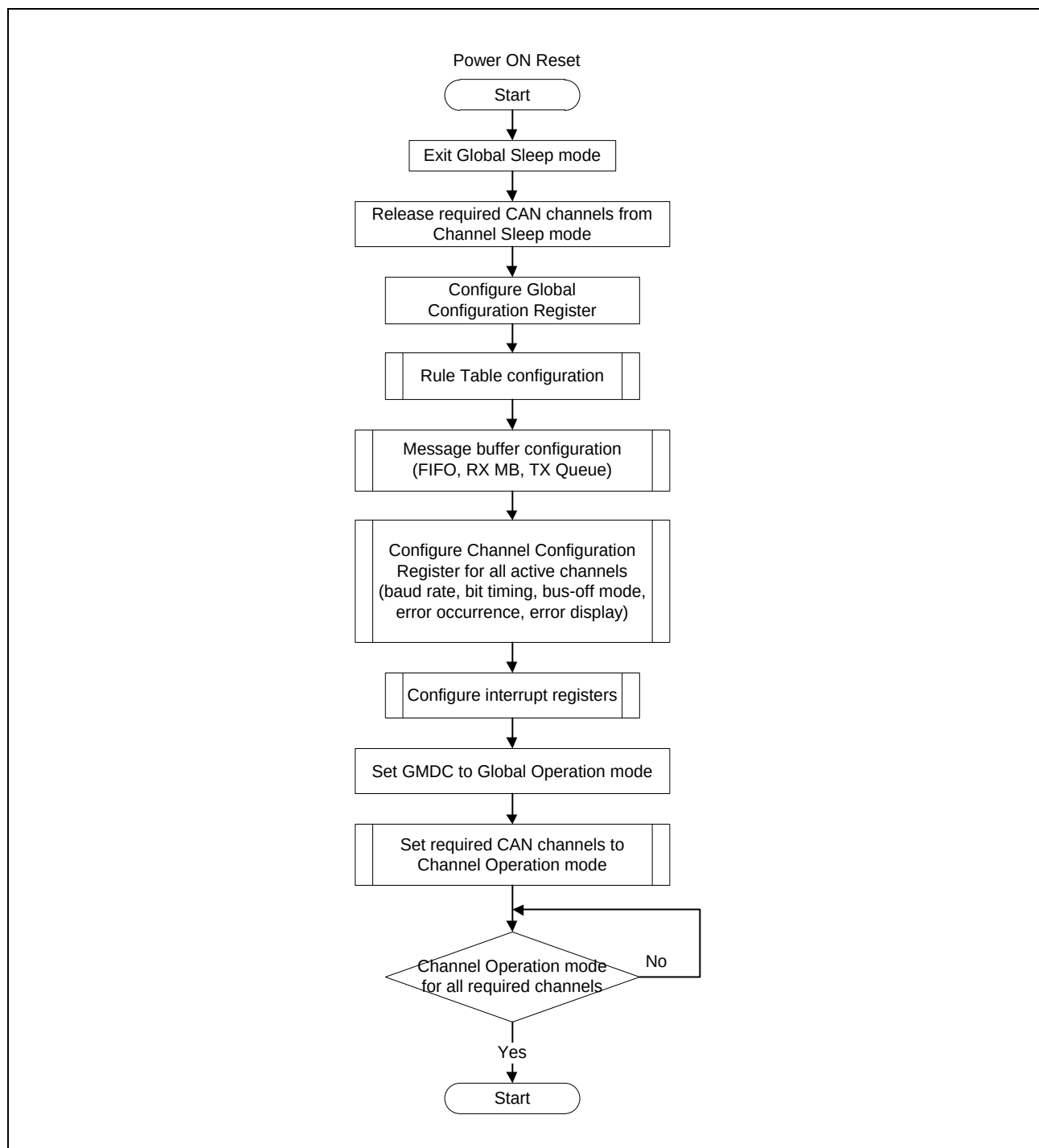


Figure 7.9-23 Configuration Procedure after a Hardware Reset

7.9.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

7.9.5.1 Overview

The CAN-FD module can handle message acceptance filtering for all channels with a global Acceptance Filter List (AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the `CFDGCFG.CMPOC` bit
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

The 6-channel CAN-FD module allows a maximum of 768 AFL entries across all channels with a maximum of 128 AFL entries per channel.

During the acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

The AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (`CFDGCFG.DRE` bit) is enabled, DLC value configured in the matching AFL entry is greater than 0h and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 00h in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0h, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the `CFDGCFG.DRE` bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception or gateway function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 8 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than eight target destinations is not allowed. When more destinations are programmed then internal timing, a race condition can occur and received message may not be stored to the message RAM. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]).

If CFDGCFG.CMPOC = 0b, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0b and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[17:0] bits are not set to 1b, respectively.

When CFDGCFG.CMPOC = 1b, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1b and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[17:0] bits are set to 1b, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1b if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF.

7.9.5.2 Allocation of AFL Entries to Each CAN Channel

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see **Figure 7.9-24**).

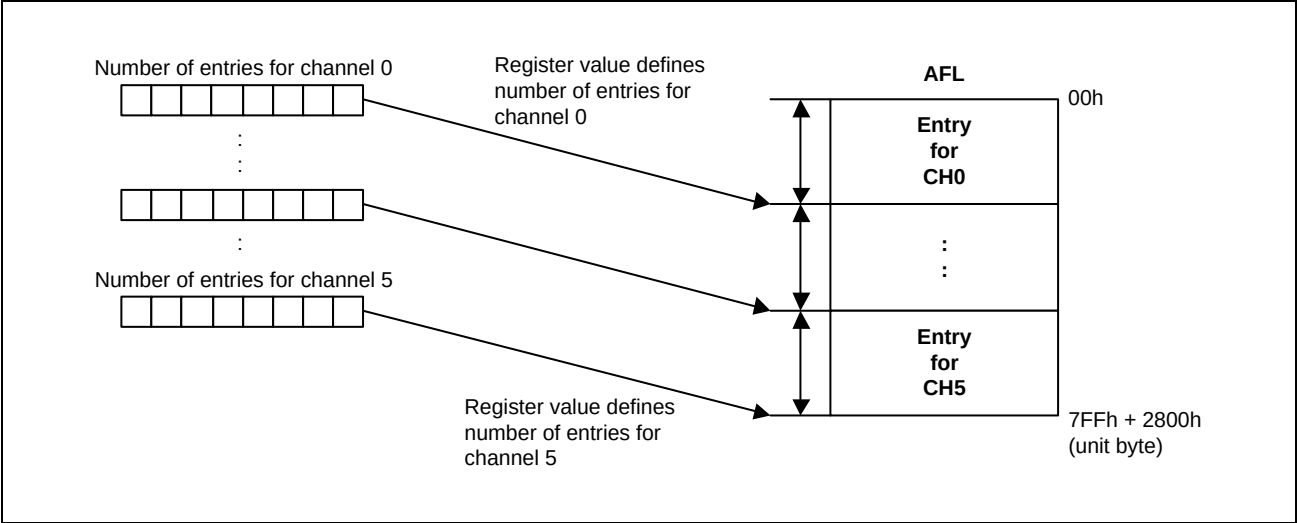


Figure 7.9-24 Configuration of AFL for Each Channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 128. The total number of entries for all channels should not exceed the maximum limit of $(n + 1) \times 128$.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CAN-FD module does not flag errors related to the configuration of the AFL.

7.9.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).
- RTR bit:
Acceptance filter unit only accepts data frames (RTR = 0b) or remote frames (RTR = 1b) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):
Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see **Figure 7.9-25**.
- Mask for IDE bit:
If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- Mask for RTR bit:
If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.
- Pointer information (16 bits):
This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.
- Information label (2 bits):
This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.
- DLC value for automatic DLC filtering:
If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed. If the DLC value in this AFL entry is configured to 0b, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

Note: A message received on channel A can be routed to Common FIFO buffer of another channel. If this Common FIFO buffer is configured in Gateway mode, then the message stored in this Common FIFO Buffer is transmitted on that channel because Common FIFO buffer is associated with channel.

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

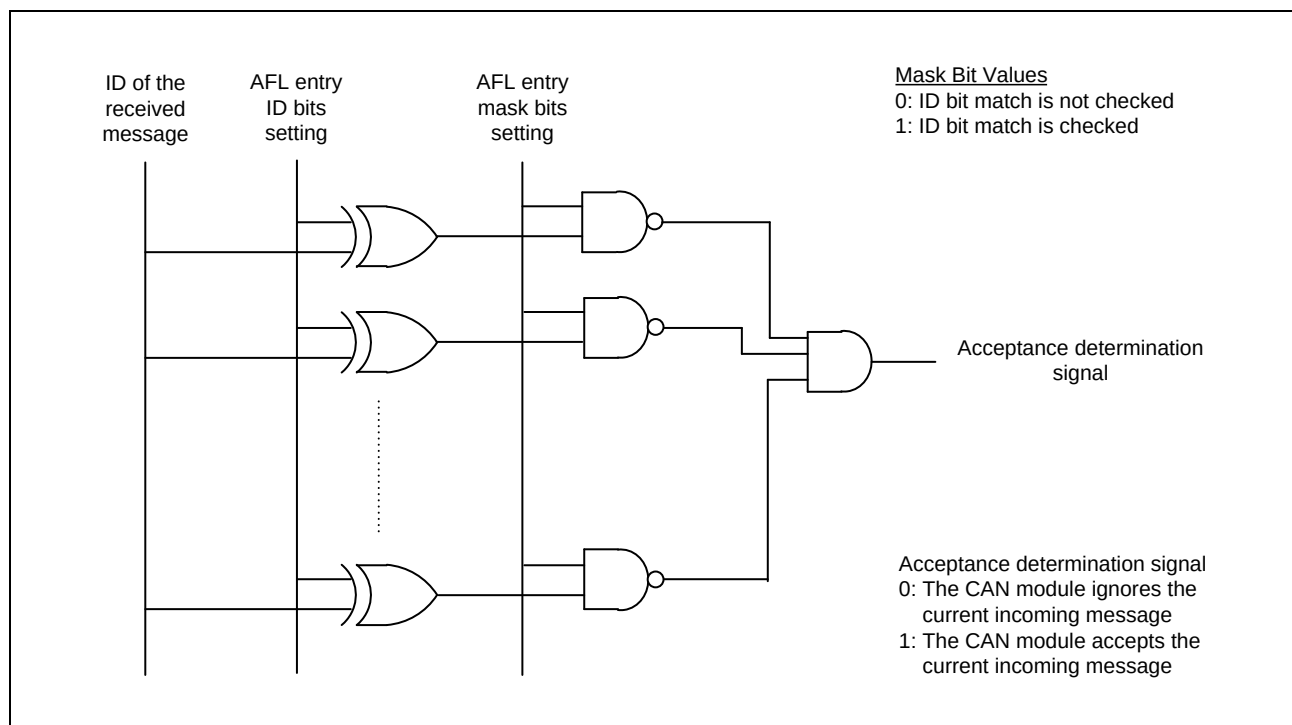


Figure 7.9-25 Acceptance Function

7.9.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry

16 sets of these registers form a group of AFL entries. Each group can be accessed through a page mechanism. For the CAN-FD module, 6-channel version, 48 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT mode. Pages are linked to the AFL entries in the following way:

Table 7.9-22 Mapping between Pages and AFL Entries

Page 0	Entry 0 - 15
Page 1	Entry 16 - 31
Page 2	Entry 32 - 47
Page 3	Entry 48 - 63
Page 4	Entry 64 - 79
Page 5	Entry 78 - 95
Page 6	Entry 96 - 111
Page 7	Entry 112 - 127
Page 8	Entry 128 - 143
Page 9	Entry 144 - 159
Page 10	Entry 160 - 175
Page 11	Entry 176 - 191
Page 12	Entry 192 - 207
Page 13	Entry 208 - 223
Page 14	Entry 224 - 239
Page 15	Entry 240 - 255
Page 16	Entry 256 - 271
Page 17	Entry 272 - 287
Page 18	Entry 288 - 303
Page 19	Entry 304 - 319
Page 20	Entry 320 - 335
Page 21	Entry 336 - 351
Page 22	Entry 352 - 367
Page 23	Entry 368 - 383
Page 24	Entry 384 - 399
Page 25	Entry 400 - 415
Page 26	Entry 416 - 431
Page 27	Entry 432 - 447
Page 28	Entry 448 - 463
Page 29	Entry 464 - 479
Page 30	Entry 480 - 495
Page 31	Entry 496 - 511
Page 32	Entry 512 - 527

Page 33	Entry 528 - 543
Page 34	Entry 544 - 559
Page 35	Entry 560 - 575
Page 36	Entry 576 - 591
Page 37	Entry 592 - 607
Page 38	Entry 608 - 623
Page 39	Entry 624 - 639
Page 40	Entry 640 - 655
Page 41	Entry 656 - 671
Page 42	Entry 672 - 687
Page 43	Entry 688 - 703
Page 44	Entry 704 - 719
Page 45	Entry 720 - 735
Page 46	Entry 736 - 751
Page 47	Entry 752 - 767

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) (see **Figure 7.9-26**). This register has the following fields:

- 4 bits to select the AFL page number
- 1 bit to enable or disable the AFL data access to prevent unwanted write access to the AFL

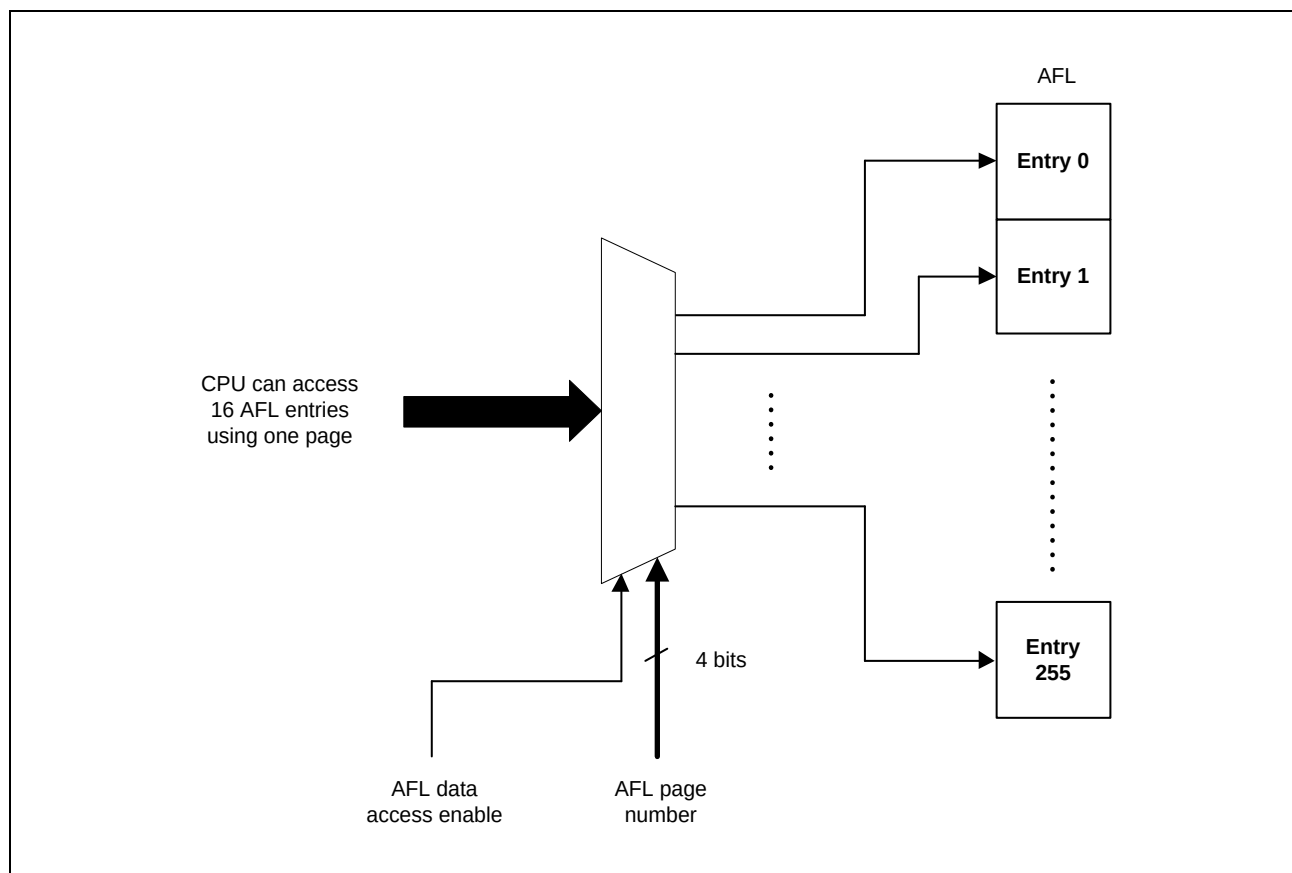


Figure 7.9-26 AFL Page Access

Application software should not write numbers higher than Fh for the AFL page number.

Follow the configuration shown in **Figure 7.9-27** to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

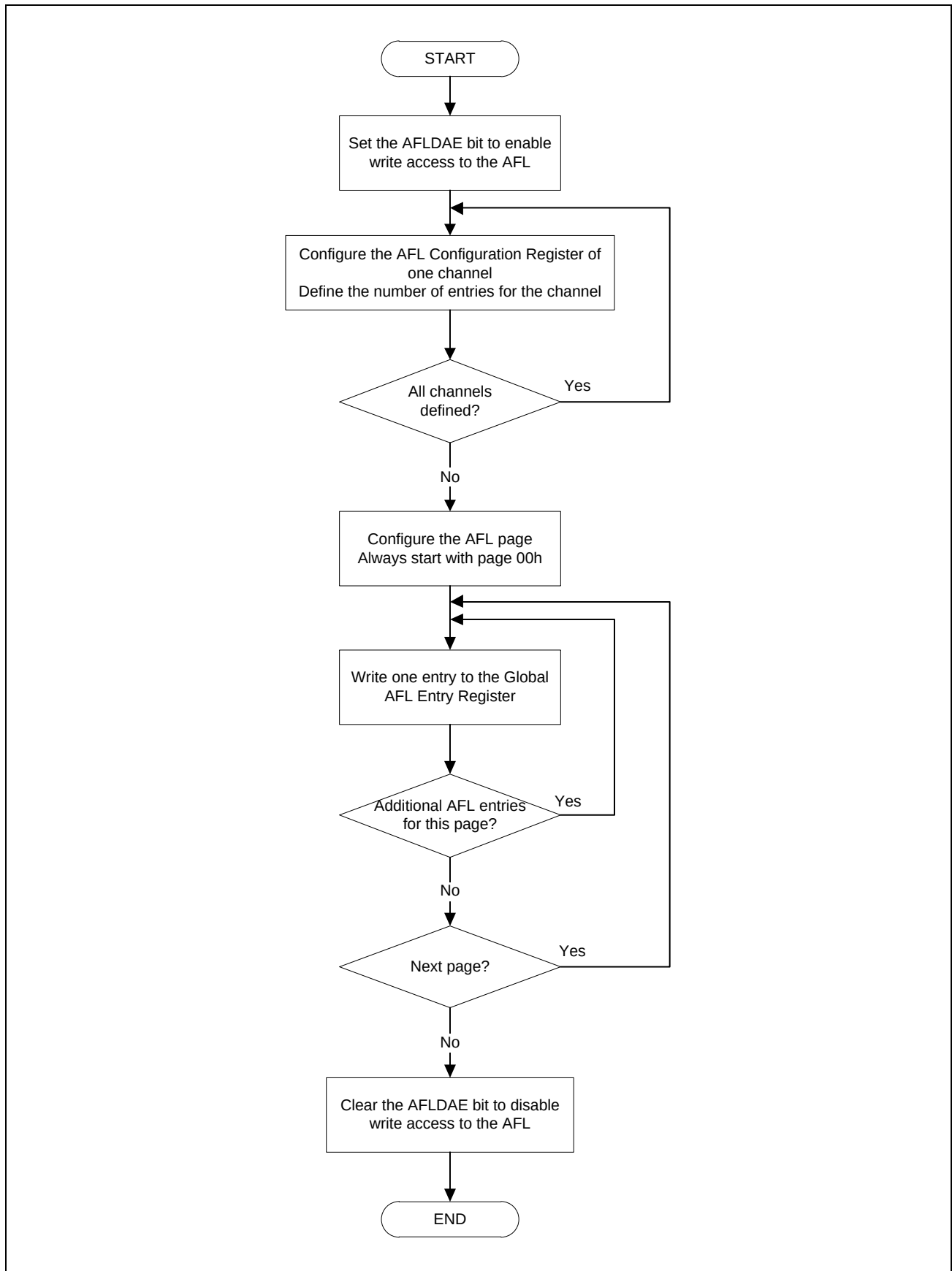


Figure 7.9-27 AFL Configuration Flow

7.9.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in loopback test mode (Self test mode 0 or Self test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0b, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If mirror mode and loopback test mode are configured at the same time, the loopback test mode behavior applies.

Table 7.9-23 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 7.9-23 Behavior of Acceptance Filter Based on the Loopback Configuration Setting in AFL Entry

Mirror Mode Enable (MME Configuration Bit)	Loopback in Test Mode (Self test mode 0 or Self test mode 1)	Channel Mode	Loopback Configuration Bit in AFL Entry	AFL Entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

7.9.5.6 IDE Masking

When the GAFLIDEM bit is 0b in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
 - CFDGAFLID [x] = C055_3A20h → IDE = 1b, RTR = 1b, LLB = 0b, ID[10:0] = 220h/ID[28:0] = 055_3A20h
 - CFDGAFLMn = 0000_FFFFh → IDEM = 0b, RTRM = 0b, IDM[10:0] = 7FFh/IDM[28:0] = 0000_FFFFh
- The comparison result for the four different received IDs with AFL entry x is described as follows:
 - If a frame with IDE = 0b and ID = 220h is received, this is considered as a match
 - If a frame with IDE = 0b and ID = 320h is received, this is not a match
 - If a frame with IDE = 1b and ID = 1FFF_3A20h is received, this is considered as a match
 - If a frame with IDE = 1b and ID = 0880_3220h is received, this is not a match.

7.9.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CAN-FD module. The message buffers are mapped as shown in **Figure 7.9-28**.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode, TX mode, or GW mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in GW mode or RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

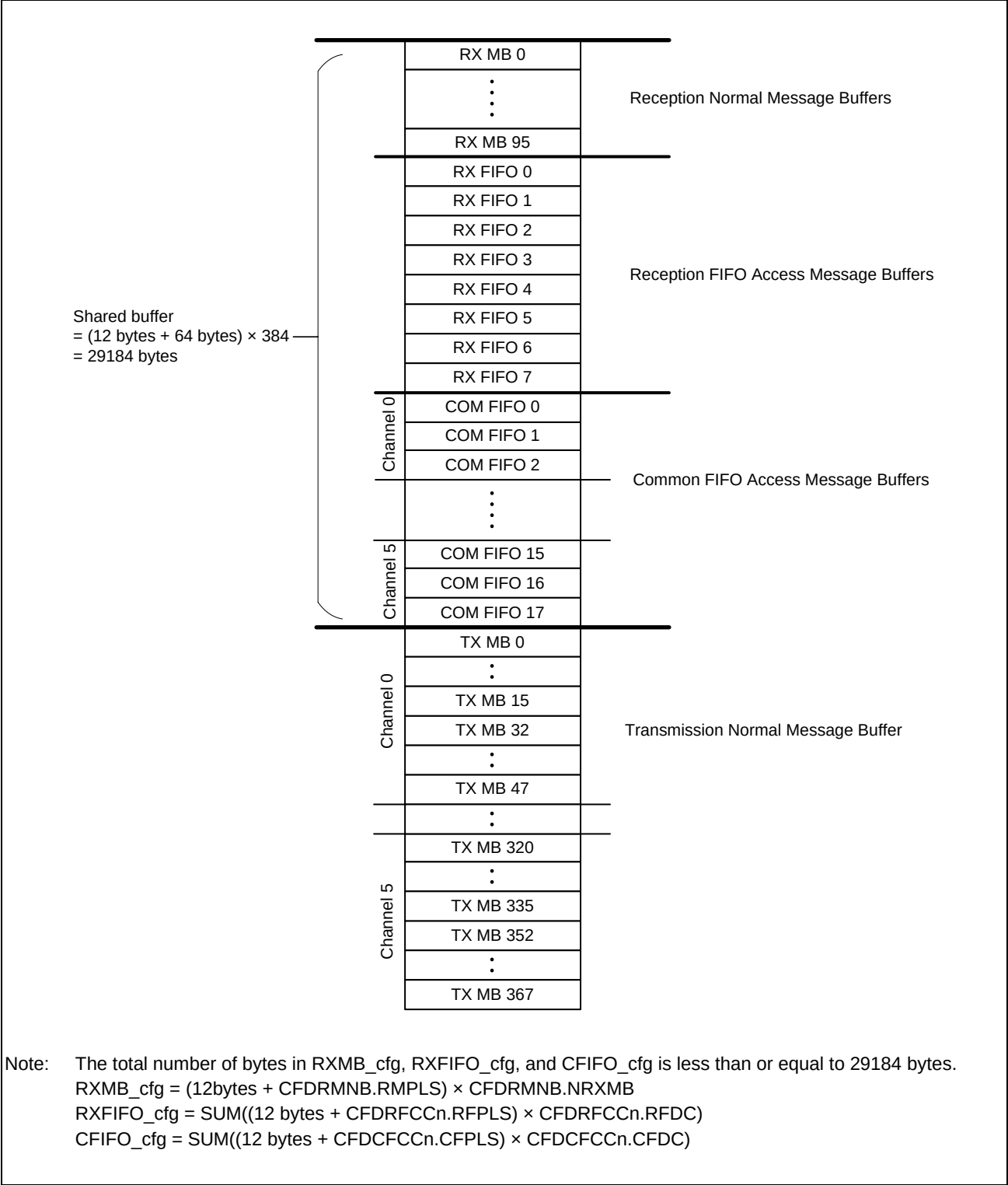


Figure 7.9-28 Message Buffer Configuration

7.9.6.1 Normal RX Message Buffers

In CAN-FD module, the frames received by various channels can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

7.9.6.1.1 Normal RX Message Buffer Configuration

In CAN-FD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 00h (no normal RX MB)
- Maximum value = $(16 \times \text{number of CAN channels}) = 16 \times 6 = 96 = 60\text{h}$ (96 flat RX MBs for 6 channels)

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CAN-FD module against incorrect configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS[2:0] bits. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

7.9.6.2 FIFO Buffer

The CAN-FD module provides a fixed number of FIFO buffers to support storage of frames for reception, transmission and gateway functions for various CAN channels.

The number of reception-only FIFO buffers is fixed to 8. However, 3 common FIFO buffers per channel can be configured to store messages for transmission, reception, or gateway function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO or GW FIFO

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

7.9.6.2.1 FIFO Buffers Configuration

In CAN-FD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 8 RX FIFO buffers + 18 common FIFO buffers = 26 FIFO buffers for 6 channels and message overwrite mechanism.

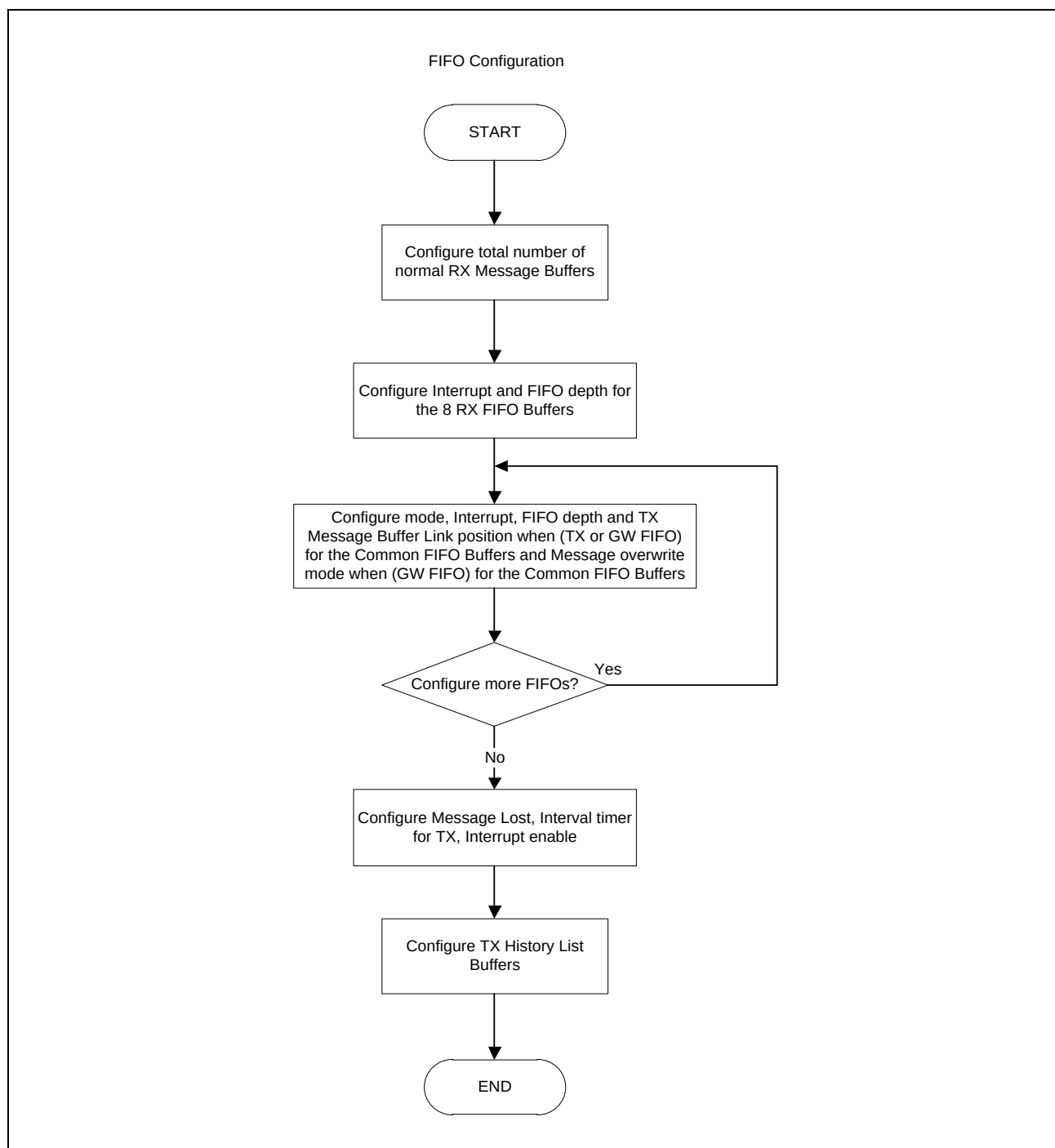


Figure 7.9-29 FIFO Buffer Configuration Flow in CAN-FD Module

As shown in **Figure 7.9-29**, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 8 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size

For the common FIFO buffers, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position

(1) FIFO Mode Configuration of Common FIFO Buffers

The mode of the common FIFO buffers can be configured by writing to the CFDCFCCn.CFM[1:0] bits in the Common FIFO Configuration/Control Registers. The possible modes of configuration for Common FIFO buffers are:

- 00b: RX mode (default mode after hardware reset)
- 01b: TX mode
- 10b: GW mode
- 11b: Reserved (do not write this value to the register bits)

Messages can only be read from the RX FIFO buffers and the Common FIFO buffers configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffers configured in TX mode. These messages are transmitted on the appropriate CAN channel.

Messages can only be read from the Common FIFO buffers configured in GW mode. However, the CPU read access has no impact on the read or write pointers. The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CAN-FD module.

After a hardware reset, all the Common FIFO buffers are configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffers in the required modes.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting the CFDCFCCn.CFMOWM bit.

- When CFDCFCCn.CFMOWM = 0b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded, and CFDCFSTSn.CFMLT bit is set to 1b.

- When CFDCFCCn.CFMOWM = 1b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message. The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message. The CFDCFSTSn.CFMOW bit is then set to 1b, which notifies that the oldest message has been overwritten with the received message.

In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive

FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point is then moves to the next message automatically. Do not write to this bit when the CFDCFCCn.CFE bit is 1b.

(2) FIFO TX Message Buffer Link Configuration

When the common FIFO is configured as TX or GW FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan of a CAN channel.

The link to a normal TX message buffer must be unique, for example the same TX message buffer cannot be shared between 2 or more common FIFO buffers.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 00h: TX Message Buffer 32
- 01h: TX Message Buffer 33
- ⋮
- 0Fh: TX Message Buffer 47

(3) FIFO Depth Configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCn.RFDC[2:0] bits and CFDCFCCn.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The 8 available options for depth configuration are:

- 000b: 0 Messages (FIFO buffer cannot be enabled)
- 001b: 4 Messages
- 010b: 8 Messages
- 011b: 16 Messages
- 100b: 32 Messages
- 101b: 48 Messages
- 110b: 64 Messages
- 111b: 128 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to $(n + 1) \times 256$ messages. Configuration of the RX message buffers, along with FIFO buffers, which exceed this maximum limit should not be done.

CAN-FD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCCn.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.

Note: If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) FIFO Payload Size Configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCn.RFPLS[2:0] bits and CFDCFCCn.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

(5) FIFO Interrupt Configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCn.RFIM bit and the CFDCFCCn.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0b:
 - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCn.RFIGCV/CFDCFCCn.CFIGCV value
 - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
 - GW FIFO mode
 - Frame RX: Interrupt generated when message counter increments and reaches the interrupt threshold value
 - Frame TX: Interrupt generated when the last message is transmitted successfully from FIFO
- 1b:
 - RX FIFO mode: Interrupt generated at the end of storage of every received message
 - TX FIFO mode: Interrupt generated for every successfully transmitted message
 - GW FIFO mode
 - Frame RX: Interrupt generated when message is stored in the FIFO
 - Frame TX: Interrupt generated when message is successfully transmitted from the FIFO

If the Interrupt Mode bit is 0b for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCn.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0b for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCCn.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full

- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCn.RFIGCV[2:0] and CFDCFCCn.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see **Table 7.9-24**.

Table 7.9-24 FIFO Interrupt Generation Counter and FIFO Depth Configuration

RFDC[2:0] (FDC[2:0])	RFIGCV[2:0] (CFGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

Common FIFO buffer can set an interrupt output at the completion time of transmitting one frame, or the completion of reception. In addition, Common FIFO and RX FIFO can set an interrupt output, when stored to the setup number (CFDC/ RFDC) of FIFO stages.

7.9.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCn.RFIE
- CFDRFCCn.RFFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Registers:

- CFDCFCCn.CFRXIE
- CFDCFCCn.CFTXIE
- CFDCFCCEn.CFFIE
- CFDCFCCEn.CFOFRXIE
- CFDCFCCEn.CFOFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCn.RFE and CFDCFCCn.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers to allow transmission and reception of messages.

When CFDCFCCEn.CFBME = 1b, it becomes FIFO buffering mode, send data is stored in the Common FIFO, and transmission is stopped. Transmission starts when CFDCFCCEn.CFBME = 0b.

Do not write 1b from 0b for this bit when the CFDCFCCn.CFE bit is 1b.

7.9.7 Interrupt and DMA

7.9.7.1 Interrupts

The CAN-FD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional Status Flag register. These status bits are only set when the corresponding interrupt enables are set.

The Status Flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CAN-FD module can be classified into 2 groups, global interrupts and channel interrupts:

- Global interrupts

The CAN-FD module can generate 2 global interrupts:

1. One Global interrupt for successful reception into the 8 RX FIFO buffers
2. One Global error interrupt

- Channel interrupts:

Each channel of the CAN-FD module can generate 3 channel interrupts:

1. Channel transmission
 - Transmission completion from channel
 - Transmission abort from channel
 - Transmission from TX Queue for a channel
 - Channel THL Interrupt
 - Successful transmission from a Common FIFO in TX or GW mode for a channel
2. Channel error interrupt
3. Successful reception in a Common FIFO in RX or GW mode for a channel or successful routing in a TXQ

The interrupts are cleared when the corresponding flag bits are cleared or interrupt enable bits are cleared.

Table 7.9-25 provides an overview of interrupt.

Table 7.9-25 Overview of Interrupt Source

Parameter	Interrupt	Interrupt Source	Interrupt Clearing
Global Interrupts	Successful reception into at least one RX FIFO	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	FIFO full into at least one RX FIFO	FIFO Full Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the FIFO Full Interrupt flags of corresponding RX FIFO buffer for which interrupt is enabled
	Global error	Any of the following: • DLC Error flag • Message Lost Status bit • Message Overwrite Status bit • TXQ Message Lost Status bit • TXQ Message Overwrite Status bit • TX History Entry Lost Status bit • CAN-FD Message Payload overflow flag	Clear all of: • DLC Error flag • Message Lost flags in all of the FIFO Status Registers • Message Overwrite flags in all of the Common FIFO Status Registers • Message Lost flags in all of the TXQ Status Registers • Message Overwrite Flags in all of the TXQ Status Registers • TX History List Entry Lost flag • CAN-FD Message Payload Overflow flag
Channel Transmission Interrupts	Channel n successful transmission	Any channel related TX MB Successful flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue* ¹	Clear all channel related TX MB Result Status bits for which the interrupt is enabled
	Channel n abort	Any channel related TX MB Abort flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue* ¹	Clear all channel related TX MB Result Status bits for which the interrupt is enabled globally
	Channel n transmission from TX Queue	Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel n THL Interrupt	Channel n THL Interrupt Status flag	Clear the relevant THL Interrupt Status flag
	Channel n COM FIFO TX Interrupt	Interrupt flag for Common FIFOs in TX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame TX Interrupt	One Frame Transmission Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of Common FIFOs belonging to the related channel
	Channel n TXQ One Frame TX Interrupt	One Frame Transmission Interrupt flag for TXQs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of TXQs belonging to the related channel
Channel COM RX FIFO Interrupt	Channel n COM FIFO RX Interrupt	Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame RX Interrupt	One Frame Reception Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Reception Interrupt flags of Common FIFOs belonging to the related channel
	Channel n COM FIFO Full Interrupt	FIFO Full Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n TXQ One Frame Routing Interrupt	One Frame Routing Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the One Frame Routing Interrupt flags of TXQs in GW mode belonging to the related channel
	Channel n TXQ Full Interrupt	TXQ Full Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of TXQs in GW mode belonging to the related channel
Channel Error Interrupt	Channel n Error	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register

Note 1. These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a common FIFO.

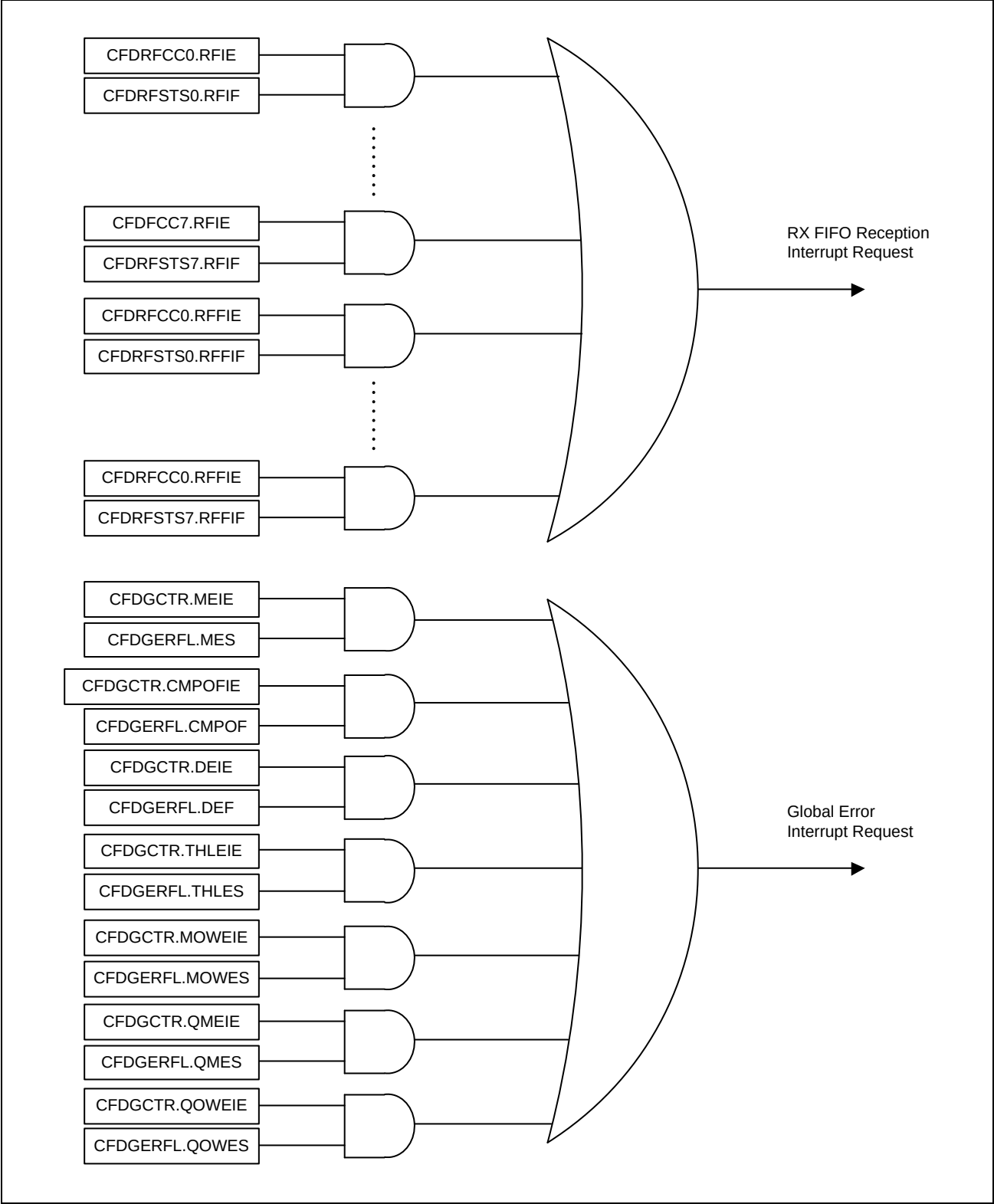


Figure 7.9-30 Block Diagram of Global Interrupt

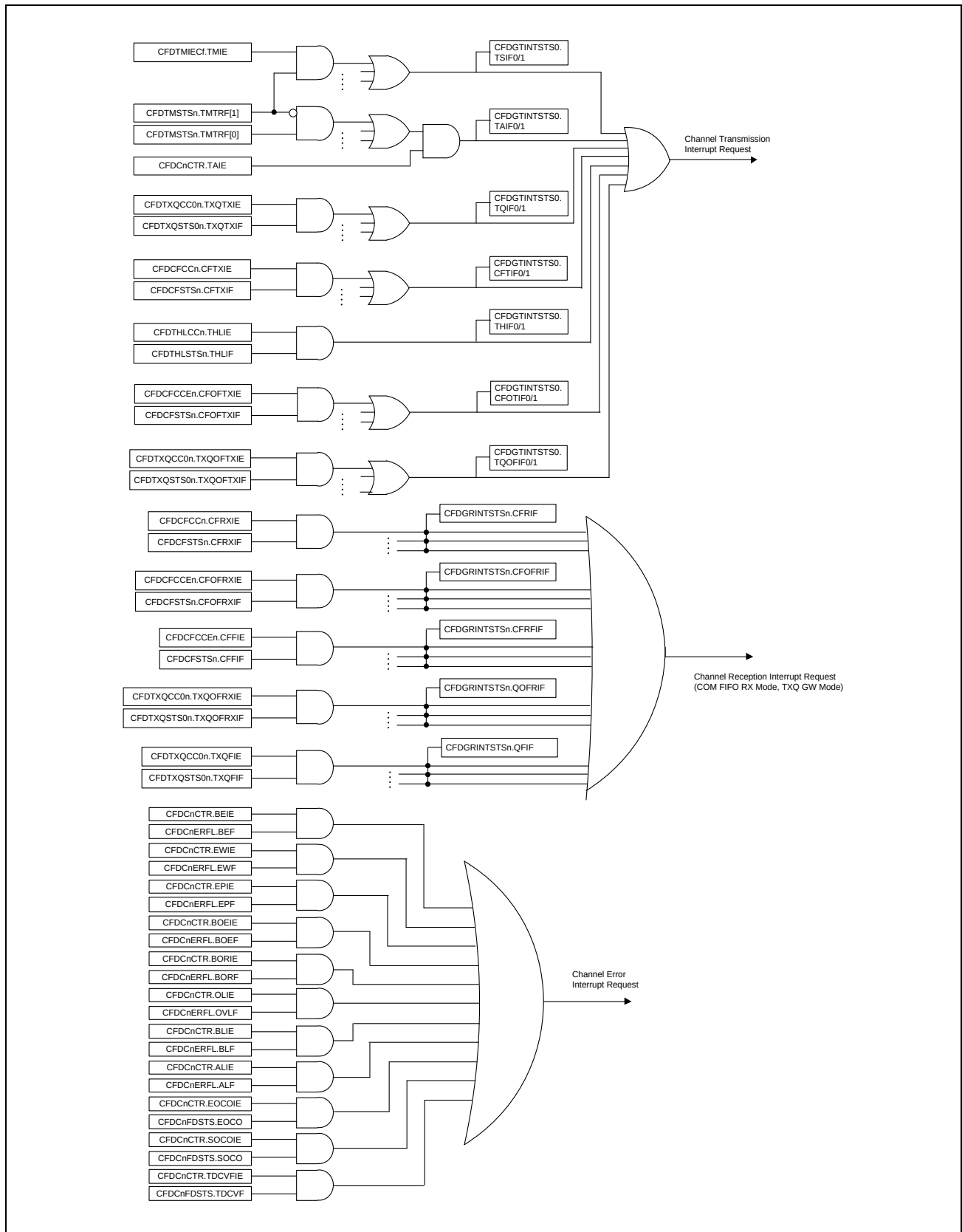


Figure 7.9-31 Block Diagram of Channel Interrupt

7.9.7.2 DMA Transfer

The CAN-FD module has some message buffer which can be associated with a DMA channel:

- Reception DMA
 - 8 RX FIFO Message Buffers
 - 6 Common FIFO Message Buffers
- Transmission DMA
 - 16 TXQ Message Buffers (TXQ0, TXQ3)
 - 6 Common FIFO Message Buffer

Figure 7.9-32 shows the potential DMA channels.

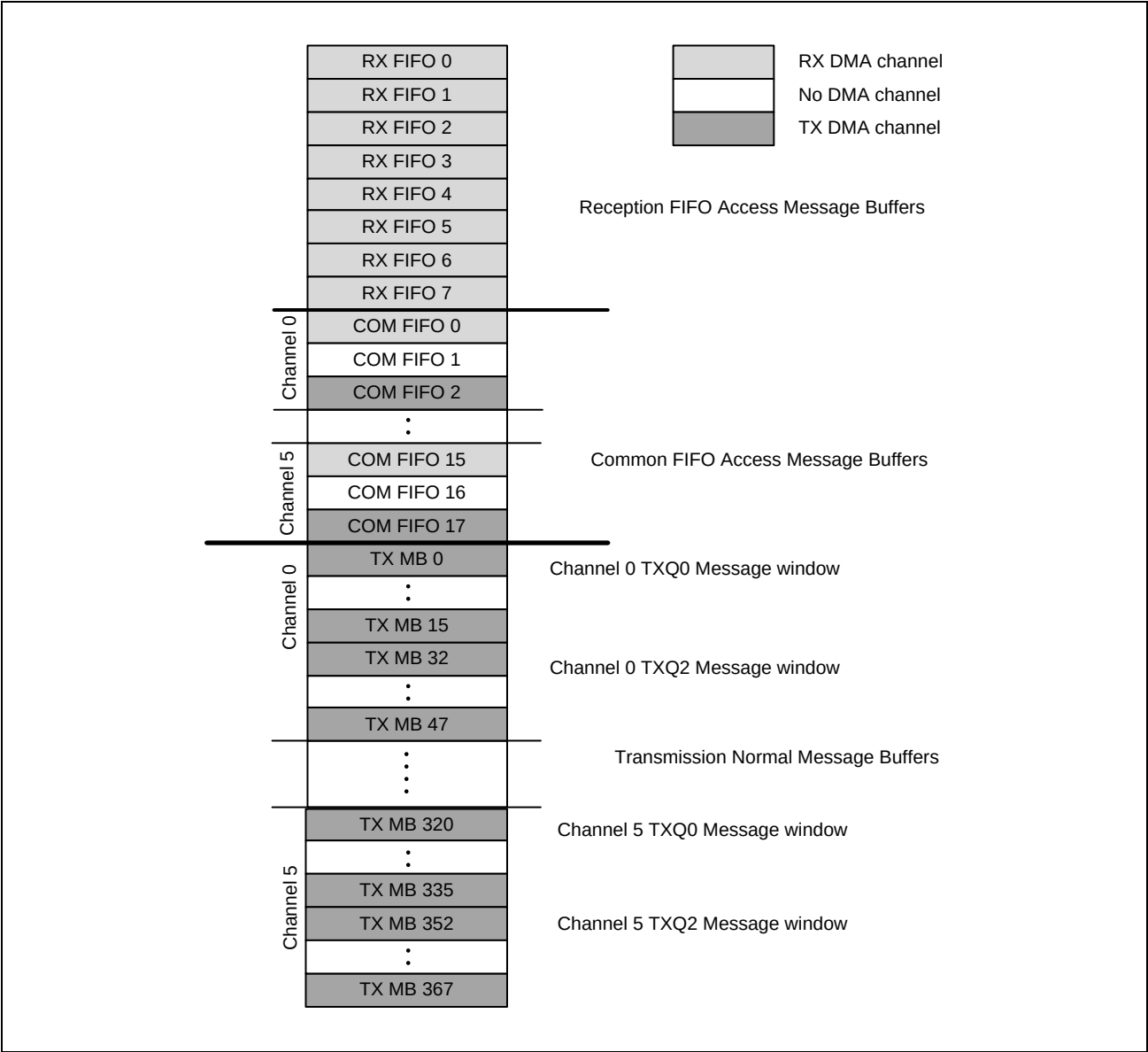


Figure 7.9-32 Message Buffer Connectable to DMA Channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1b and the belonging FIFO is not empty.

Reception FIFO interrupt should be disabled for this particular FIFO (CFDRFCCn.RFIE or CFDCFCCn.CFRXIE).

Use the regular start address for the DMA access window address and add 8000h to the regular start address for the debugger access window. See **Table 7.9-26**.

Table 7.9-26 DMA Channel Access Window Address

b = Message Buffer Component.Index	MBCP	Register	p	Start Address n = 0, 1
0 to number_of_RFMBCPs - 1	RFMBCPb	RFID	x	6000h + b × 0080h
		RFPTR	x	6004h + b × 0080h
		RFFDSTS	x	6008h + b × 0080h
		RFDfP	0 to 15	600Ch + p × 0004h + b × 0080h
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb	CFID	x	6400h + b × 0080h + n × 0180h
		CFPTR	x	6404h + b × 0080h + n × 0180h
		CFFDCSTS	x	6408h + b × 0080h + n × 0180h
		CFDfP	0 to 15	640Ch + p × 0004h + b × 0080h + n × 0180h

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

The software debugger must access outside of the regular SFR address range from E000h to EFFFh.

Do not write to the FIFO and TXQ control registers when DMA is enabled.

The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time, the procedure in this section is a configuration flow for an initial setup.

When CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE is set, the messages of the corresponding TXQ or Common FIFO can be handled by DMA controller.

Use the following procedure when the TXQ or the Common FIFO can be handled by DMA controller.

1. CPU checks the TXQ or the Common FIFO is not full.
2. When transmit data can be used, CPU sets this data to Common FIFO or TXQ.
When using Common FIFO, transmit data is write in CFDCFID, CFDCFPTR, CFDCFFDCSTS, and CFDTMBCPb[i] registers.
When using TXQ, transmit data is write in CFDTMID, CFDTMPTR, CFDTMFDCTR, and CFDTMDFp registers.
3. For Common FIFO, the common FIFO pointer is incremented automatically when DMA controller writes the last data payload byte configured by CFDCFCCn.CFPLS.
For TXQ, if the data of 64 data payload is written, a TXQ pointer increases automatically. When payload data is less than 64 bytes, dummy data must be written in and 64 data payload size must be done.

Note: Only 32-bit write-access can be possible on the DMA message handling.

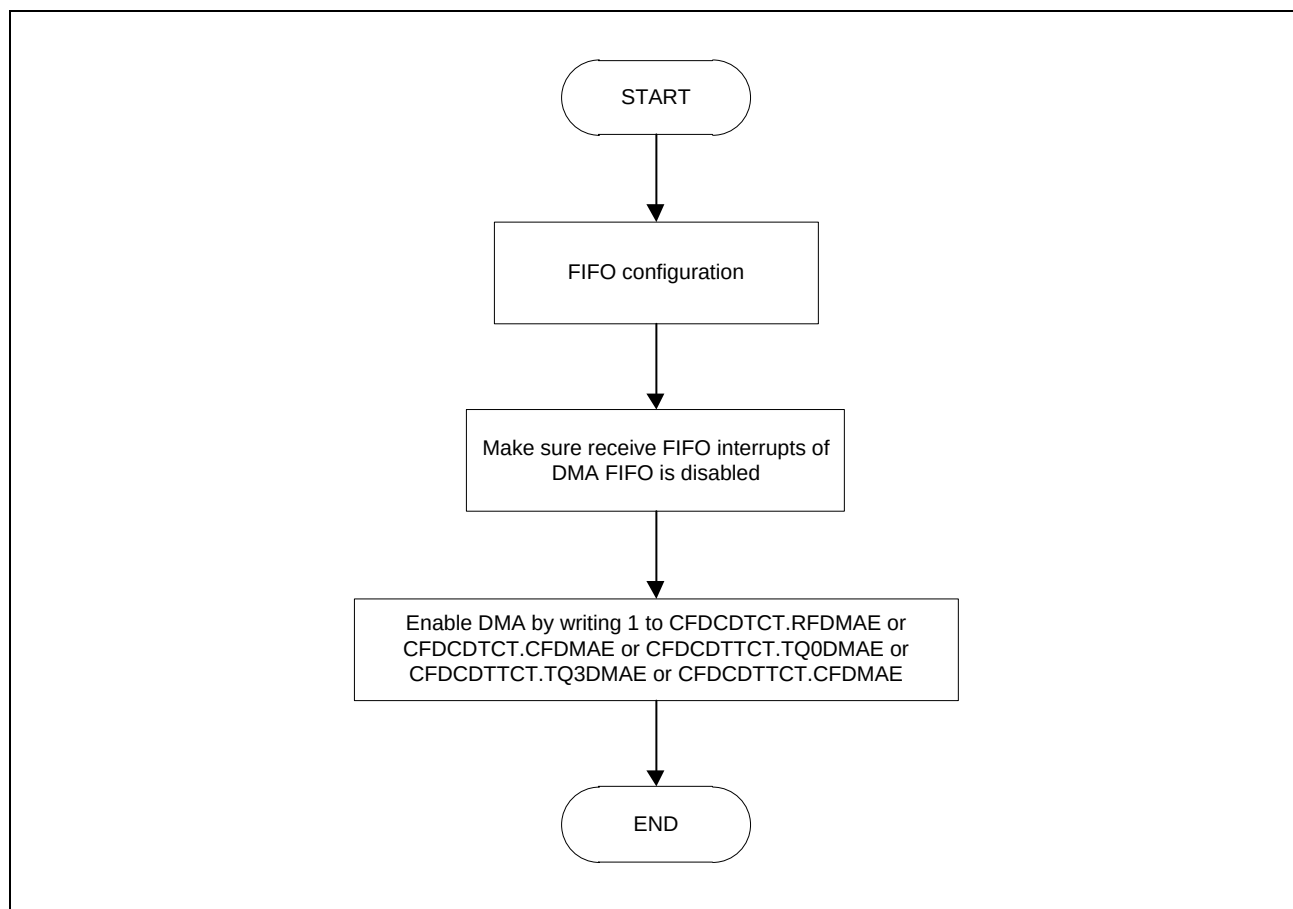


Figure 7.9-33 DMA Enable Flow

To disable a DMA transfer requested, disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then this must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS. For reference, see the flow in **Figure 7.9-34**. When the DMA is disabled, then consider what to do with the remaining or new incoming messages to this particular FIFO reception.

When the FIFO is not disabled, reception to the FIFO continues.

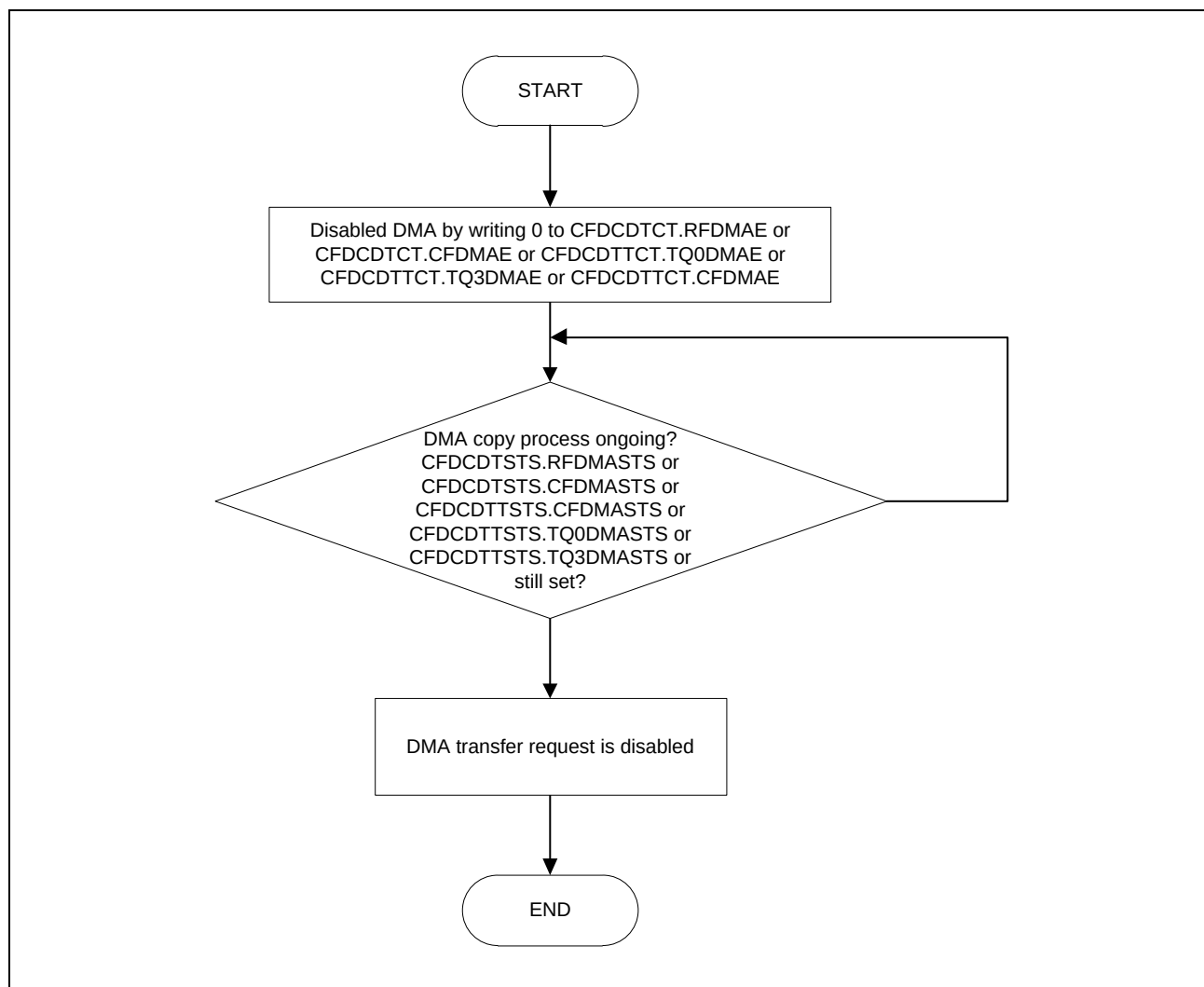


Figure 7.9-34 DMA Disable Flow

7.9.8 Reception and Transmission

7.9.8.1 Reception

FIFO buffers or Common FIFO buffers configured in RX mode or GW mode depending on the Acceptance Filter List entries are as follows:

- Up to 32 RX Message Buffers can be configured
- 8 RX FIFO buffers available
- Up to 6 Common FIFO buffers can be configured in RX mode or GW mode
- Up to 4 TX Queues can be configured in GW mode

7.9.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX Message Buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX Message Buffer.

If a new message is stored into a RX Message Buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX Message Buffer. If such a loss of messages is not acceptable, then RX FIFO should be used to store related messages.

Note: Interrupts are not provided for the RX Message Buffers in the CAN-FD module and so the RX Message Buffer New Data Register should be accessed periodically to check if a new message has been stored in the RX Message Buffers.

Note: Unused data bytes are filled with 00h depending on the DLC value.

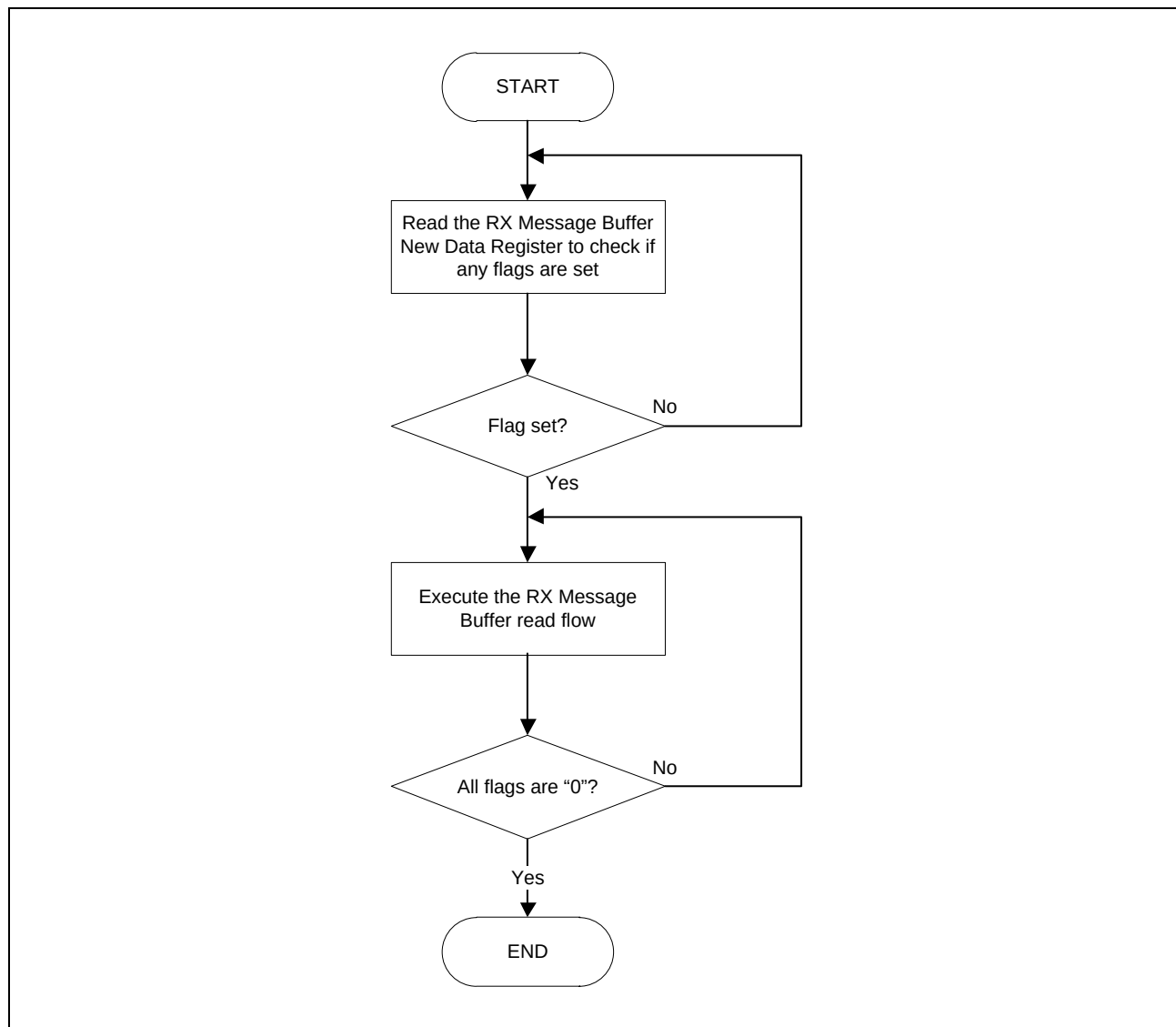


Figure 7.9-35 RX Message Buffer Message Access Flow

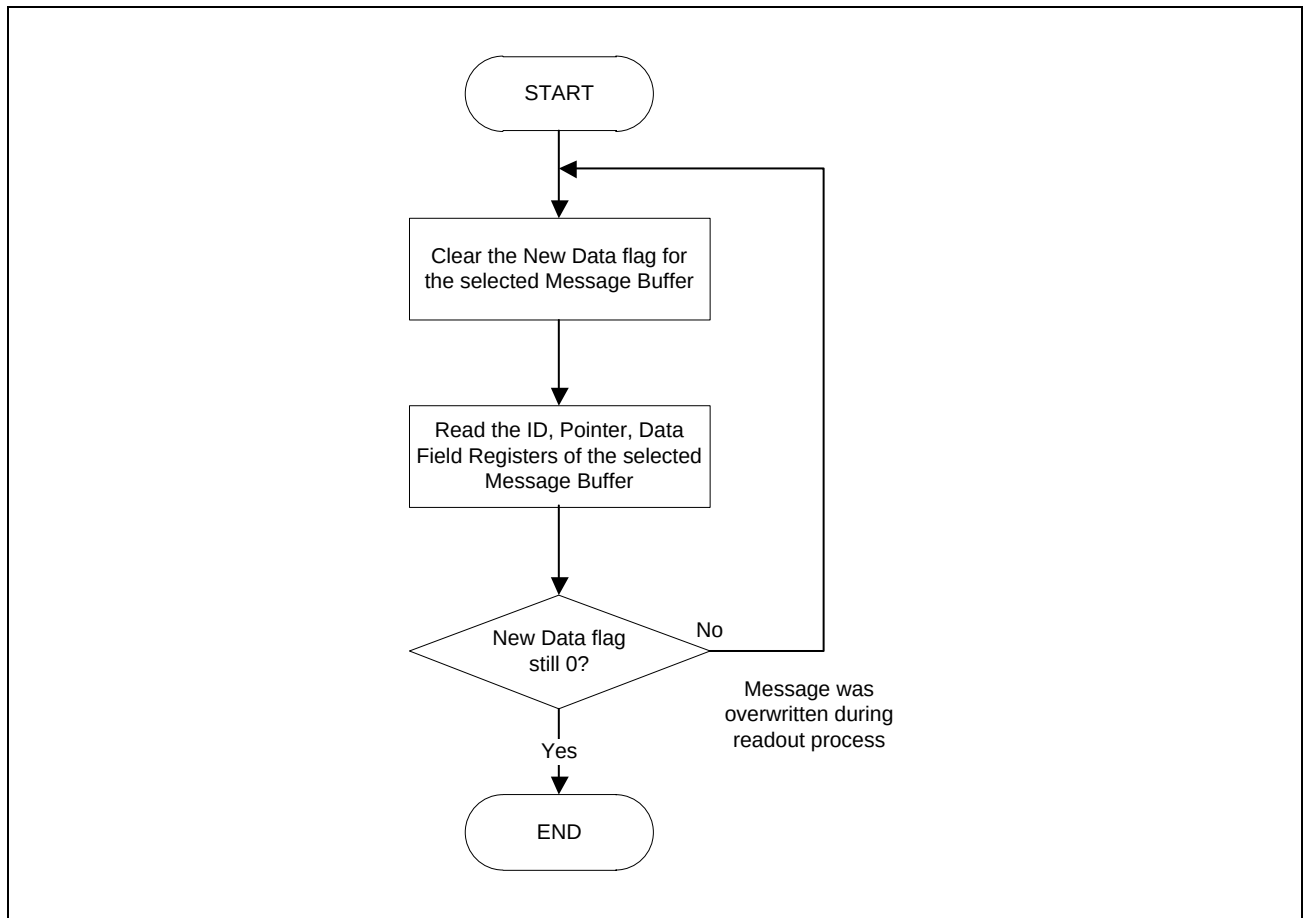


Figure 7.9-36 RX Message Buffer Read Flow

7.9.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffers configured in RX or GW mode should be configured based on system requirements.

The CFDAFLP1n.GAFLFDP[31:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffers configured in RX mode or GW mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Registers.

Depending on the configuration of the FIFO buffers, an interrupt may also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value FFh is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write FFh to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to overrun condition.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCn.CFMOWM bit.

- When CFDCFCCn.CFMOWM = 0b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1b.

- When CFDCFCCn.CFMOWM = 1b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.

The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.

The CFDCFSTSn.CFMOW bit is then set to 1b, which notifies that the oldest message has been overwritten with the received message.

In addition, if a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission of the message is not performed. The read point moves to the next message automatically.

Do not write change for this bit when the CFDCFCCn.CFE bit is 1b. Common FIFO can set interrupt when:

- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

Note: The message lost can be set only in RX or GW mode by CAN, the flag is not set when the CPU is overloading the FIFO buffers.

Note: When CFDGAFLP0n.GAFLSRDi (i = 0 to 2) is set and CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

The RX FIFO Buffers and the Common FIFO Buffers configured in RX or GW mode can be disabled at any time by clearing the CFDRFCCn.RFE or CFDCFCCn.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCn.RFE or CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO Buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO Buffers or Common FIFO Buffers configured in RX mode are assigned as DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write FFh to the FIFO Pointer Control Register (CFDFPCTRn.CFPC or CFDRFPCTRn.RFPC), because this could lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO Buffer and then the FIFO is disabled, the interrupt flag is not automatically cleared. The interrupt flag should be cleared before disabling the FIFO.

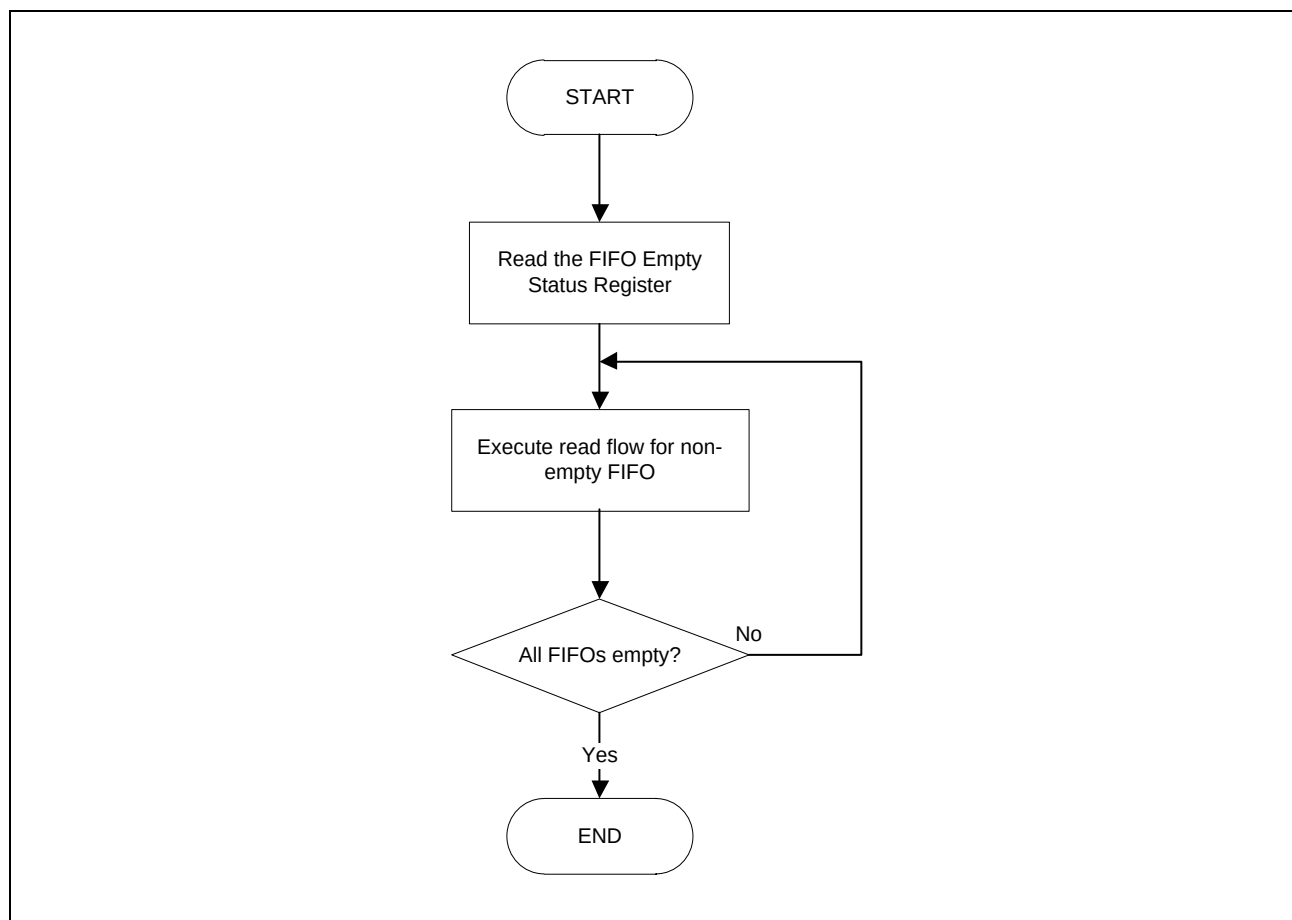


Figure 7.9-37 FIFO Buffer Message Access Flow (Example for Polling Case)

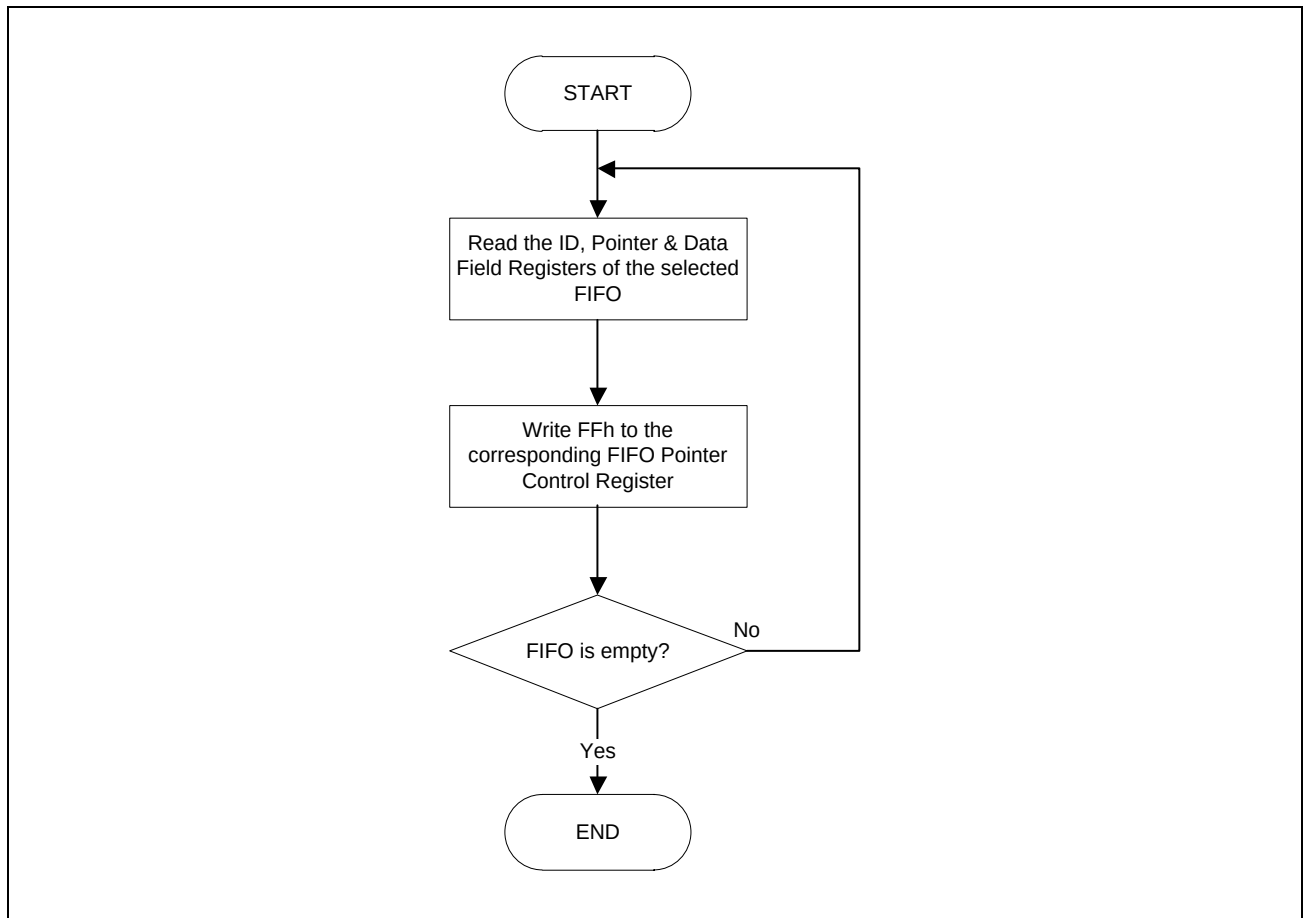


Figure 7.9-38 RX FIFO Buffer Read Flow (Example for Polling Case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even if it clears an interruption flag after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the deadline, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

7.9.8.1.3 Timestamp

The Timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the `CFDGFDCFG.TSCCFG[1:0]` configuration (at the sample point of start of frame, point in time when the frame is valid, or for CAN-FD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and Data into the target RX Message Buffer or RX/GW FIFO.

For transmit message, the Timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the `CFDGCFG.TSSS` bit of the Global Configuration Register. If it is 0b, the peripheral clock is used. If it is 1b, the selected CAN channel bit time clock is used.

The channel selection is done with the `CFDGCFG.TSBTCS` bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset mode for this channel, the Timestamp counter is stopped. Therefore, for other CAN channels, the Timestamp counter value is also not updated.

If peripheral clock is selected as the Timestamp counter clock source, Channel modes do not influence the Timestamp counter function.

The source clock for the Timestamp counter can be divided by a factor defined by the `CFDGCFG.TSP` bits (Timestamp Prescaler) in the Global Configuration Register.

The Timestamp counter can be reset to 0000h with the `CFDGCTR.TSRST` bit (Timestamp Reset).

7.9.8.2 Transmission

There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- TX Queue transmission

A fixed number of transmission message buffers (32 TX message buffers) are dedicated for each channel. These message buffers are only used for transmission and cannot be configured for reception.

Additionally transmission from TX Queue and/or Common FIFO in TX or GW mode can be configured in the following way (see **Figure 7.9-39**):

- TX Queue: Up to 16 transmission message buffers for one channel can be grouped to form a TX Queue with a common access window.

Upper transmission message buffers are used to form the TXQ1 or TXQ3. Lower transmission message buffers are used to form the TXQ0 or TXQ2.

Transmission control and status registers of these transmission message buffers should not be used. One channel has four TX Queues. Each TXQ has an access window:

- TXQ0 is transmission Message Buffer 0 of each channel
- TXQ1 is transmission Message Buffer 31 of each channel
- TXQ2 is transmission Message Buffer 32 of each channel
- TXQ3 is transmission Message Buffer 63 of each channel

When using TXQ1 and TXQ0 simultaneously, the sum of the depths of TXQ1 and TXQ0 should not exceed 16.

When using TXQ3 and TXQ2 simultaneously, the sum of the depths of TXQ3 and TXQ2 should not exceed 16.

- Common FIFO (TX/GW mode): Each Common FIFO in TX or GW mode is linked to a dedicated channel. Each channel has a fixed number of three Common FIFOs assigned to it. Within the channel, a Common FIFO configured in TX or GW mode, can be freely linked (assigned) between 32 and 47 transmission message buffers (only one FIFO to one transmission message buffer).
The Common FIFO buffer then replaces the transmission message buffer linked to it.
Transmission control and status registers of these transmission message buffers should not be used.

See **Figure 7.9-28** for information on Common FIFO buffer assignment to related channels.

Note: Common FIFO buffers should not be linked to TX message buffers that are already part of a TX Queue.

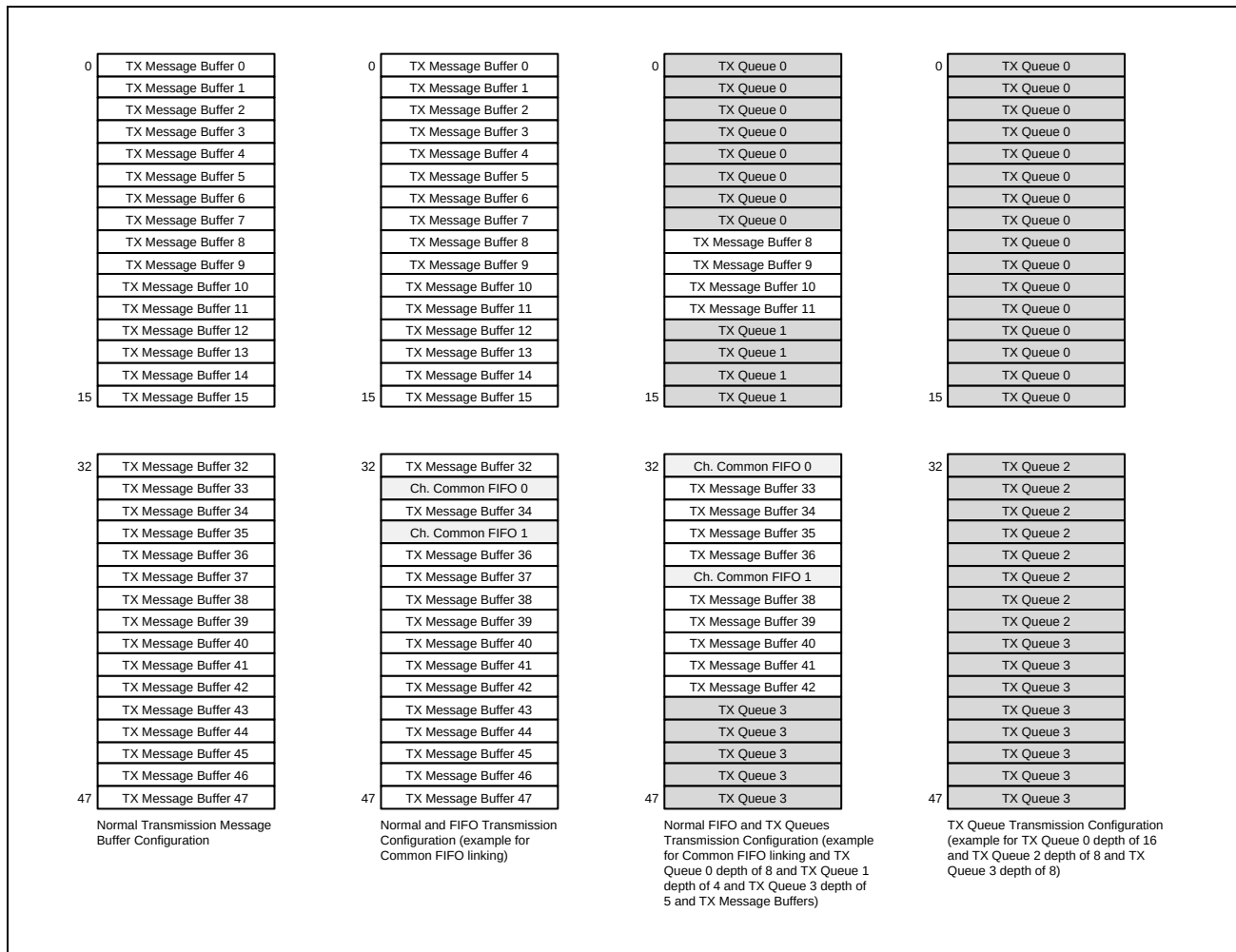


Figure 7.9-39 Channel Transmission Message Buffer Configuration

7.9.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, the transmission priority in the CAN-FD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority

The transmission priority mode is common for all message buffers and all CAN channels. It can be configured with the `CFDGCFG.TPRI` bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffers configured in TX mode or GW mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, the next pending message within the same FIFO will be considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 7.9-40 shows the transmission configuration flow.

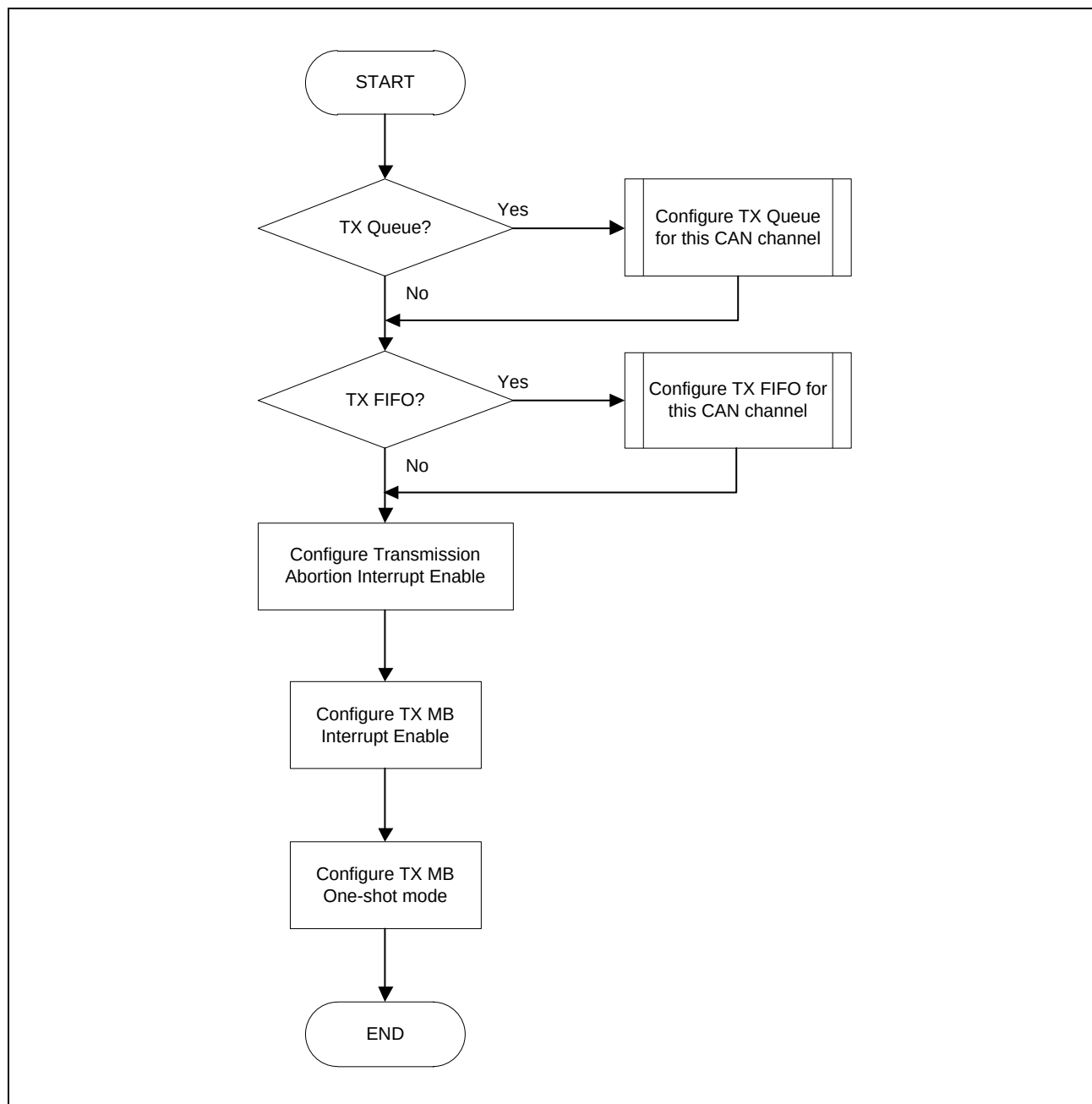


Figure 7.9-40 Transmission Configuration Flow

7.9.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

- Regular transmission mode

If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.

Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission will be attempted further if no transmission abort request is set for this transmission message buffer.

New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

- One-shot transmission mode

When the CFDTMCn.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, then the message buffer is placed in one-shot transmission mode and attempts to transmit a message only once.

Completion of one-shot transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. The CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b when the one-shot transmission is successful.

The CFDTMSTSn.TMTRF[1:0] bits are set to 01b when arbitration is lost or an error occurs during the transmission of the related message buffer.

Additional message transmission will not be attempted in this case.

The regular transmission request procedure after a configuration is shown in **Figure 7.9-41**.

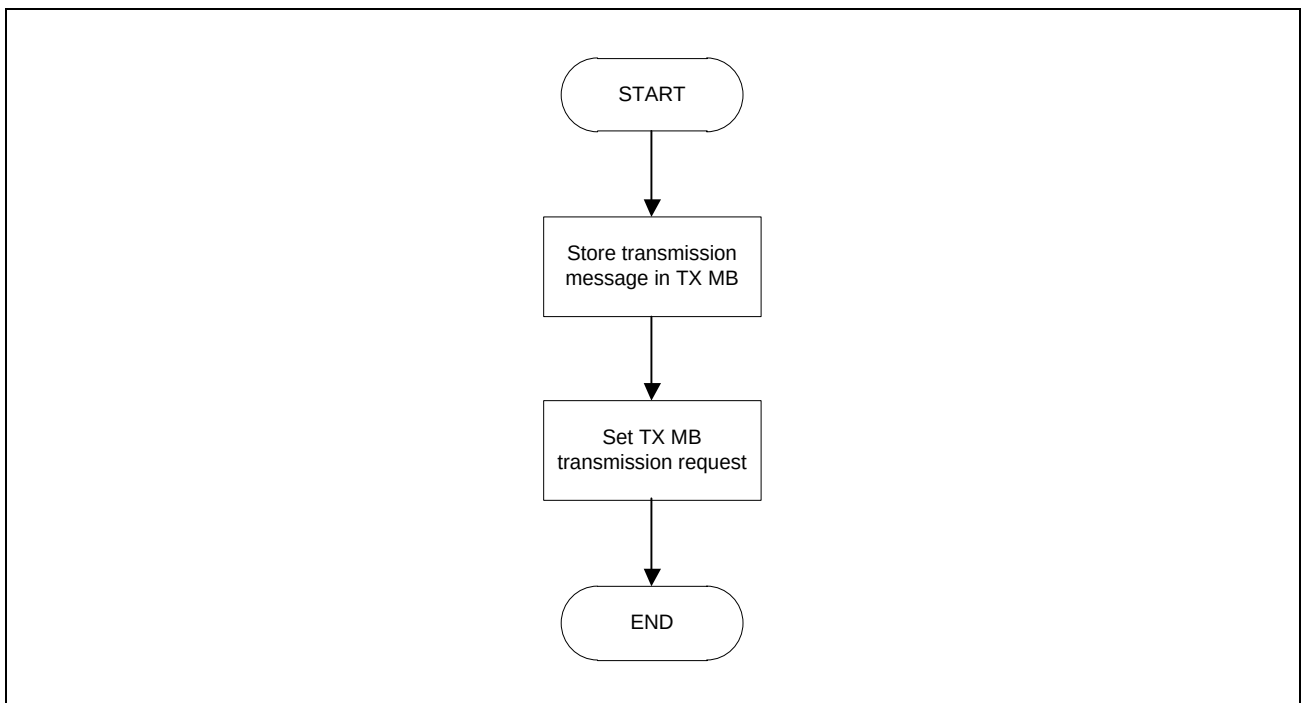


Figure 7.9-41 Transmission Request Procedure Using Normal TX Message Buffer Mode

(1) TX Message Buffer Control Register Setting

Table 7.9-27 shows configuration of the normal CAN transmission mode.

Table 7.9-27 Configuration of CAN Transmission Mode

Transmission Request CFDTMCn.TMTR	Transmission Abortion Request CFDTMCn.TMTAR	One Shot Enable CFDTMCn.TMOM	Communication Activity
0	0	0	Message buffer is disabled
0	0	1	Message buffer is disabled
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abortion is requested
1	1	1	One shot transmission abortion is requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 7.9-42 shows timings for successful transmission for two message buffers of one channel.

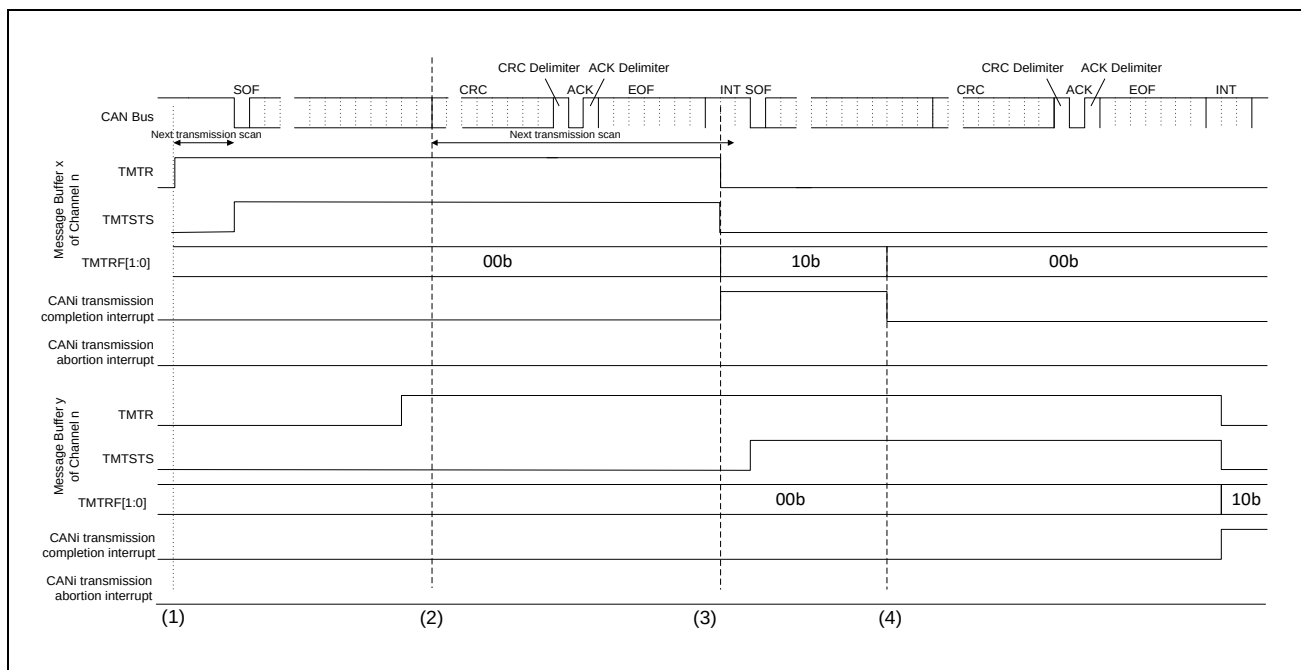


Figure 7.9-42 Timing of Request and Flag Bits for Successful Transmission

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission. When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the related TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission*¹.
2. On the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist. The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.

3. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared. When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated. To clear the related interrupt line the CFDTMSTSn.TMTRF flag bits must be cleared.
4. Before starting the next transmission, clear the CFDTMSTSn.TMTRF[1:0] bits. Load the next message in the transmission message buffer and set the CFDTMCn.TMTR bit again. CFDTMCn.TMTR bit cannot be set again before CFDTMSTSn.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during the transmission or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSn.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 7.9-43 shows timings for transmission abort for two message buffers of one channel.

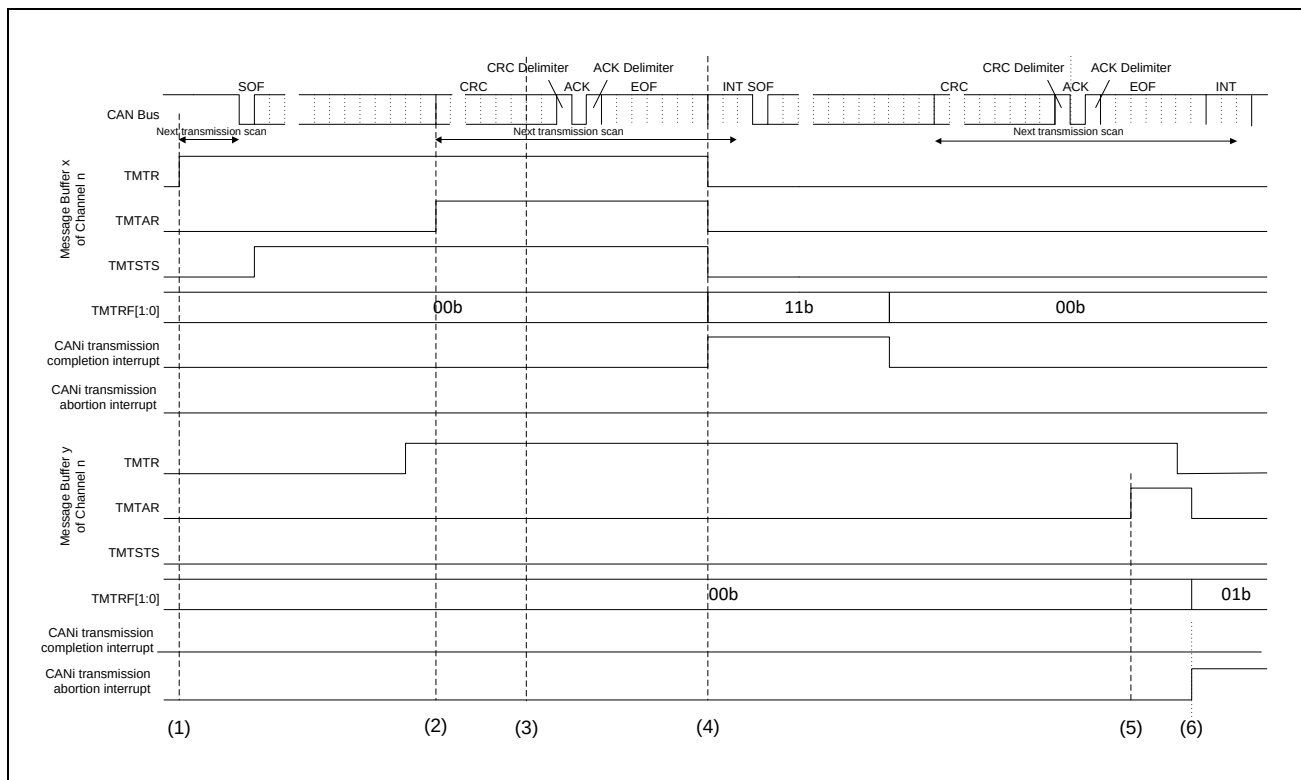


Figure 7.9-43 Timing of Request and Flag Bits for Transmission Abort

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission. When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission*¹.

2. If the CFDTMCn.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. On the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example timing diagram, message buffer y is not selected as the next transmission message buffer.
The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.
4. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSn.TMTSTS not set). If the CFDTMCn.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSn.TMTRF[1:0] bits are set to 01b.
If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
In addition, CFDTMCn.TMTR, and CFDTMCn.TMTAR bits are cleared automatically.
When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set, an interrupt is generated for successful transmission abort.
To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit. If an error occurs, either during the transmission, or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

7.9.8.2.3 TX FIFO or GW FIFO Transmission

Three common FIFO buffers are assigned to each channel. The three FIFO buffers can be linked to any normal TX Message Buffer position for this channel by the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Register if configured in TX or GW mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX Message Buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX Message Buffer linked to a FIFO buffer configured in TX or GW mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value FFh is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers.

If the message count matches the FIFO Depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0b, an interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCCn.CFIM bit is 1b, an interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

Common FIFO can set interrupt, when CAN frame transmitted is completed.

The Common FIFO buffers configured in TX mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Registers. If this bit is cleared to 0b, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCCn.CFE bit only when the Empty flag is set for the corresponding Common FIFO Buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCCn.CFE is set again, ensure that CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the TX FIFO.

When the CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in **Figure 7.9-44**.

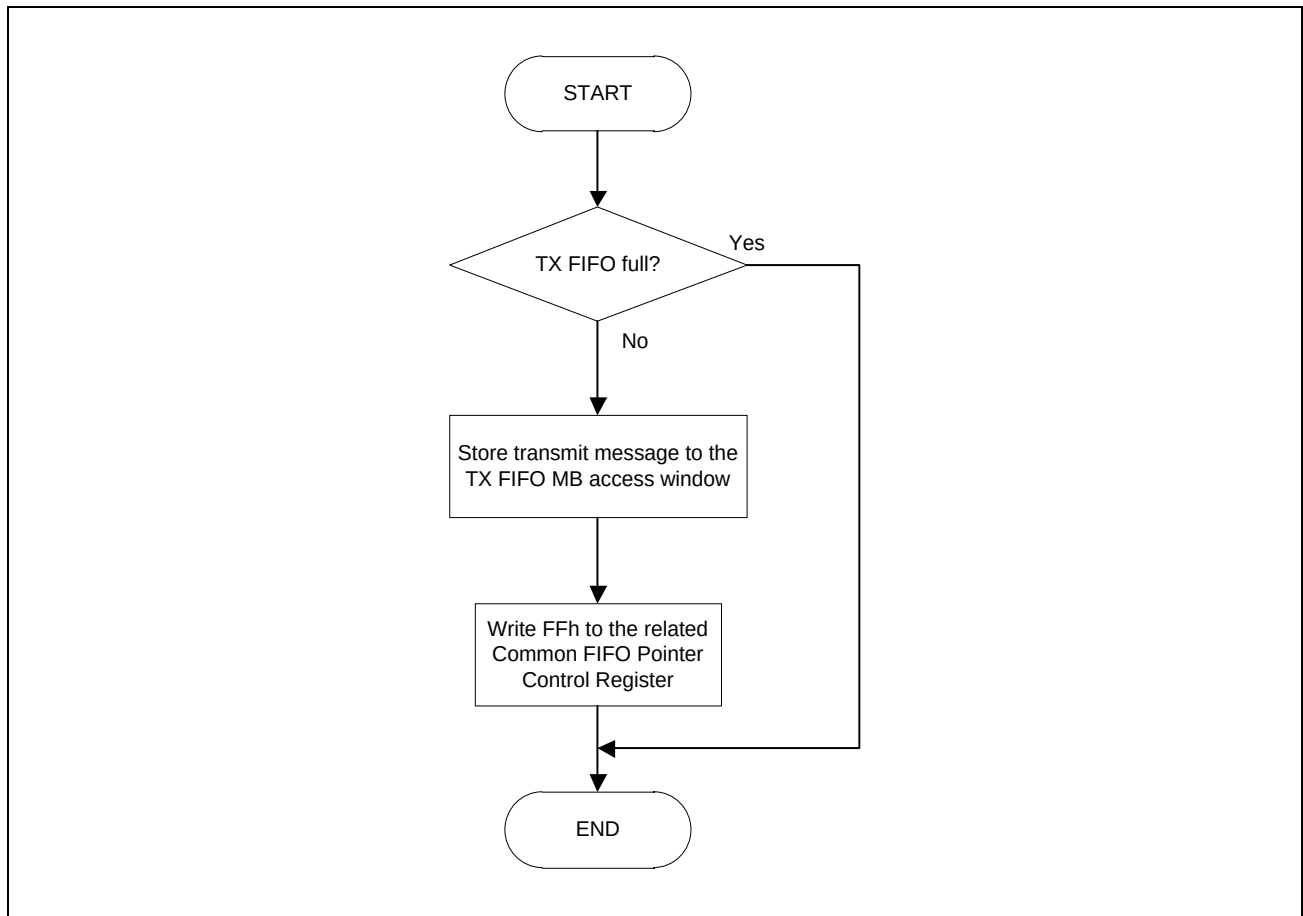


Figure 7.9-44 TX FIFO Transmission Request Procedure

(2) GW FIFO Operation

The AFL entries for routing the received messages to GW FIFO buffers should be configured based on the requirements of the system. The matching AFL entry selects the GW FIFO buffer for storage of a received message on any of the CAN channels.

When a message is successfully received and stored in a GW FIFO buffer, the FIFO message count in the corresponding FIFO Status Register is incremented by 1.

If the message count matches the FIFO depth, then the FIFO Full flag is set.

The oldest message in the GW FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the GW FIFO, the message count value is decremented by 1. When all the messages from the GW FIFO are transmitted, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCn.CFMOWM = 0b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1b.

- When CFDCFCCn.CFMOWM = 1b:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.

The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.

The CFDCFSTSn.CFMOW bit is then set to 1b, which notifies that the oldest message is overwritten with the received message.

In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point moves to the next message automatically.

The interrupt generation conditions for the GW FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0b, then RX interrupt flag is set when FIFO counter increments and reaches value configured by CFDCFCCn.CFIGCV and the TX interrupt flag is set when FIFO transmits the last message successfully.

If CFDCFCCn.CFIM bit is 1b, then RX interrupt flag is set at the end of storage of every received message and TX interrupt flag is set if a message is successfully transmitted from the FIFO.

Common FIFO can set interrupt when:

- CAN frame transmitted is completed
- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

When CFDCFCCn.CFBME = 1b, it becomes FIFO buffering mode, send data is stored in Common FIFO, and transmission is stopped. Transmission will be started if it is set as CFDCFCCn.CFBME = 0b.

The Common FIFO buffers configured in GW mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared, the GW FIFO becomes empty as follows:

- Immediately if the message from the GW FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the GW FIFO is already scheduled for transmission or already in transmission

Other possible messages pending from the GW FIFO are lost.

Before CFDCFCCn.CFE is set again, ensure that the CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the GW FIFO.

When the CFDCFCCn.CFE bit is cleared and the CFDCFSTSn.CFEMP bit is set, the message read and write pointers of the GW FIFO are cleared and are no longer active. Therefore, all messages in the GW FIFO buffers are lost and no further message can be stored into the GW FIFO.

In applications intended to be used as CAN-to-CAN gateways, it is useful if the Error State Indication (ESI) information of the routing messages is not replaced by the sending node Error State Indication. For this, each channel has the control function register CFDCnFDCFG.ESIC to replace their own ESI information by the routing ESI information.

Note: If the sending node is error passive, the ESI bit is sent anyway as error passive (ESI = 1b).

(3) Interval Timer for FIFO Transmission

For each Common FIFO in TX or GW mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCCn.CFE bit is set.

When the Common FIFO in TX or GW mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCCn.CFE bit
- CAN channel is in CH_RESET mode

The interval time is specified by the CFDCFCCn.CFITT value in the Common FIFO Configuration/Control Register and can be specified from 0 to 255 timer units.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, a value of 0 should be selected.

The timer source can be selected by the configuration bit CFITSS in the Common FIFO Configuration/Control Register. For the timer source the CAN bit timing clock of the FIFO related channel or a global reference clock can be selected.

If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT or CH_RESET or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDCCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See **Table 7.9-28** for CFDCCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency/period.

Table 7.9-28 Configuration Example for the FIFO Interval Timer Reference Clock

Peripheral Clock (CANFD_0_pclk)	Reference Clock		
	1 μ s	100 μ s	500 μ s
75 MHz / 13.33 ns	75	7500	37500
100 MHz / 10 ns	100	10000	50000

Additionally, the reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCCn.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value ($\times 1$ or $\times 10$).

The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 μ s to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX/GW FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time.

Figure 7.9-45 shows an example timing of the internal processing.

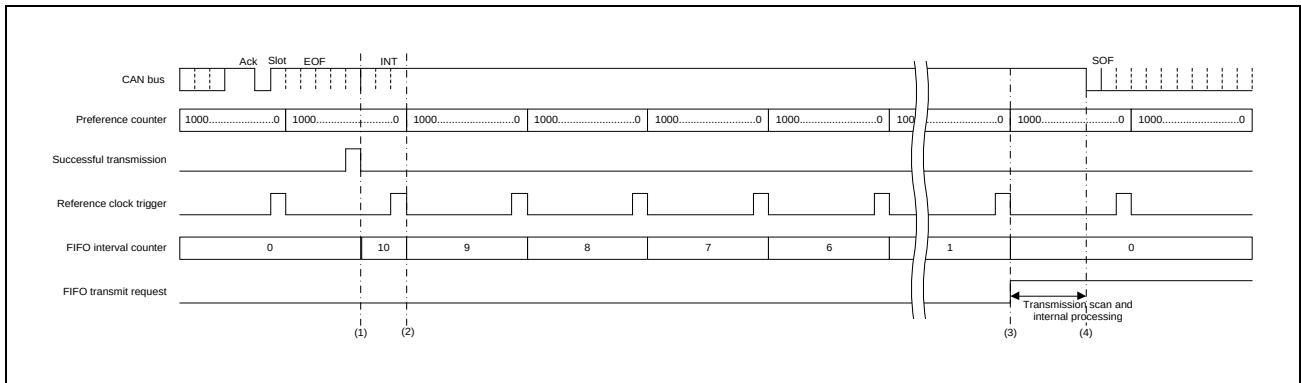


Figure 7.9-45 Example for Interval Processing Time

The configuration for the above timing is as follows:

- Peripheral clock frequency = 100 MHz
 - Interval timer reference clock (CFDGCFG.ITRCP) = 1000 times
 - Reference clock due to the above settings = 10 μ s
 - Common FIFO interval timer source selection (CFDCFCCn.CFITSS) = 0b
 - Common FIFO interval timer resolution (CFDCFCCn.CFITR) = 0b
 - Common FIFO interval transmission time (CFDCFCCn.CFITT) = 10 times
 - Theoretical message separation interval = 100 μ s
1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to one reference clock interval.
 2. With the next reference clock trigger, the FIFO interval timer is decremented.
 3. When the FIFO interval timer reached the value 0 the FIFO transmit request is set.
 4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing this usually takes less than 3 CAN bit time, between internal FIFO transmit request set 3 and actual transmission.

In the worst case when multi events such as reception scan, internal message routing, transmit scan on all channels occur, then it can take up to 432 peripheral clock cycles.

As shown in **Figure 7.9-45**, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure `CFDCCn.CFITT` to the required minimum value + 1.

If additional TX message buffers or TX/GW FIFOs are configured for transmission for the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time due to higher priority message transmission from these TX message buffers or TX/GW FIFOs.

Figure 7.9-46 shows a block diagram of the FIFO interval time generation circuit.

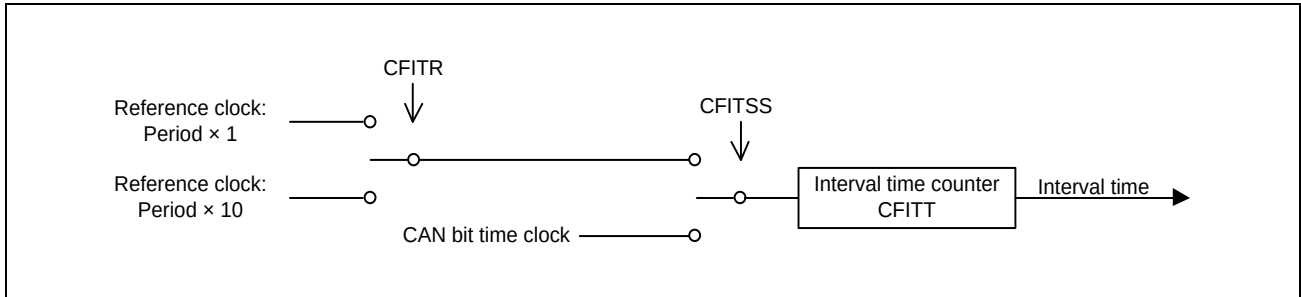


Figure 7.9-46 Block Diagram of FIFO Interval Timer

7.9.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 32 TX message buffers, which are accessed by one access window. One channel has 4 TX Queues. One TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 0 as access window (referred to as TXQ0). The second TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 31 as access window (referred to as TXQ1). The third TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 32 as access window (referred to as TXQ2). The fourth TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 63 as access window (referred to as TXQ3).

All TXQ0, TXQ1, TXQ2, and TXQ3 messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFCFG.TPRI = 0b).

The registers for TXQ0 are CFDTXQCC0n, CFDTXQSTS0n, and CFDTXQPCTR0n.

The registers for TXQ1 are CFDTXQCC1n, CFDTXQSTS1n, and CFDTXQPCTR1n.

The registers for TXQ2 are CFDTXQCC2n, CFDTXQSTS2n, and CFDTXQPCTR2n.

The registers for TXQ3 are CFDTXQCC3n, CFDTXQSTS3n, and CFDTXQPCTR3n.

When access window TX Message Buffer No.63 (TXQ3) or TX Message Buffer No.32 (TXQ2) or TX Message Buffer No.31 (TXQ1) or TX Message Buffer No.0 (TXQ0) is used, refer to the related access registers TX Message Buffer ID Registers (CFDTMIDn), TX Message Buffer Pointer Registers (CFDTMPTRn), TX Message Buffer Data Field 0 Registers (CFDTMDF0_n), and TX Message Buffer Data Field 1 Registers (CFDTMDF1_n).

The depth of each TXQ0 buffer can be configured by writing to the CFDTXQCC0n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ0 can set from TXMB0 to TXMB15 as a queue buffer at the maximum.

The 15 available options for depth configuration are:

- 00h: TX Queue disabled
- 01h: Reserved
- 02h: 3 messages
- ⋮
- 0Dh: 14 messages
- 0Eh: 15 messages
- 0Fh: 16 messages

The depth of each TXQ1 buffer can be configured by writing to the CFDTXQCC1n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ1 can set from TXMB15 to TXMB0 as a queue buffer at the maximum.

The 15 available options for depth configuration are:

- 00h: TX Queue disabled
- 01h: Reserved
- 02h: 3 messages
- ⋮
- 0Dh: 14 messages
- 0Eh: 15 messages
- 0Fh: 16 messages

The depth of each TXQ2 buffer can be configured by writing to the CFDTXQCC2n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ2 can set from TXMB32 to TXMB47 as a queue buffer at the maximum.

The 15 available options for depth configuration are:

- 00h: TX Queue disabled
- 01h: Reserved
- 02h: 3 messages
- ⋮
- 0Dh: 14 messages
- 0Eh: 15 messages
- 0Fh: 16 messages

The depth of each TXQ3 buffer can be configured by writing to the CFDTXQCC3n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ3 can set from TXMB47 to TXMB32 as a queue buffer at the maximum.

The 15 available options for depth configuration are:

- 00h: TX Queue disabled
- 01h: Reserved
- 02h: 3 messages
- ⋮
- 0Dh: 14 messages
- 0Eh: 15 messages
- 0Fh: 16 messages

When using TXQ1 and TXQ0 simultaneously, the depth of TXQ is 16 or less in total.

When using TXQ3 and TXQ2 simultaneously, the depth of TXQ is 16 or less in total.

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 63, TX Message Buffer No. 32, TX Message Buffer No. 31 and TX Message Buffer No. 0, which acts as TX Queue access window).

When CFDGAFLP0n.GAFLSRD i (i = 0 to 2) is set and the CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

When CFDTXQCCn.TXQOWE bit is 1b, the TX Queue is in TX Queue overwrite mode. If the message of the same ID is stored in TX Queue when a frame is received and it is stored in TX Queue, an old message is overwritten by a new message. Therefore, an old message is not transmitted. When the old message of the same ID is transmitting and a CAN bus error and an arbitration-lost occur, the message of old ID is not resent.

When using the function in GW mode and TX Queue overwrite mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3. If it accesses by routing in gateway mode when a TXQ buffer is full, CFDTXQSTS.TXQMLT is set and send data is thrown away.

The function is valid for the standard ID frame and is invalid for the extended ID frame.

Explanation of operation of the TX Queue with same ID over-writing function in GW mode is shown in **Figure 7.9-47**.

TXQ0 depth = 6 buffers, kind of ID is 3, TXQ0 GW mode

1. Three frames and stored.

Now transmitting	ID0	TXMB0
Next transmission	ID1	TXMB1
Waiting for txscan	ID2	TXMB2
		TXMB3
		TXMB4
		TXMB5



2. ID0 is stored in TXMB3 and abort is set as TXMB0.

Now transmitting	ID0	TXMB0	← Set abort request
Next transmission	ID1	TXMB1	
Waiting for txscan	ID2	TXMB2	
Entry New ID →	ID0	TXMB3	
		TXMB4	
		TXMB5	



3. ID1 is stored in TXMB4 and abort is set as TXMB1.

Now transmitting	ID0	TXMB0	← Wait for abort
Next transmission	ID1	TXMB1	← Set abort request
Waiting for txscan	ID2	TXMB2	
Entry New ID →	ID1	TXMB4	
		TXMB5	



4. ID2 is stored in TXMB5 and the transmit request of TXMB2 is cleared.

Now transmitting	ID0	TXMB0	← Wait for abort
Next transmission	ID1	TXMB1	← Wait for abort
Waiting for txscan		TXMB2	← Clear transmission request
Entry New ID →	ID0	TXMB3	
	ID1	TXMB4	
	ID2	TXMB5	



5. Transmission of TXMB0 is completed and transmission of ID1 of TXMB1 is started.

Completion of transmitting		TXMB0	
Transmitting	ID1	TXMB1	← Wait for abort
Waiting for txscan		TXMB2	
	ID0	TXMB3	
	ID1	TXMB4	
	ID2	TXMB5	

Figure 7.9-47 Operation of TXQ0 in Gateway Mode

When a system writes in TXQ, a system should write in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full then no further access should be done to the queue, until it is no longer full. If it accesses by software writing in when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTSn.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored into the TX Queue.

When a message has been stored to the TX Queue, FFh must be written into the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same identifier are stored in the TX Queue, then the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same identifier is stored in the TX Queue. Or if TX queue overwrite mode is used, the frame of the same ID is rewritten on a new frame.

For the TX Queue a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCCn.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in **Figure 7.9-48**.

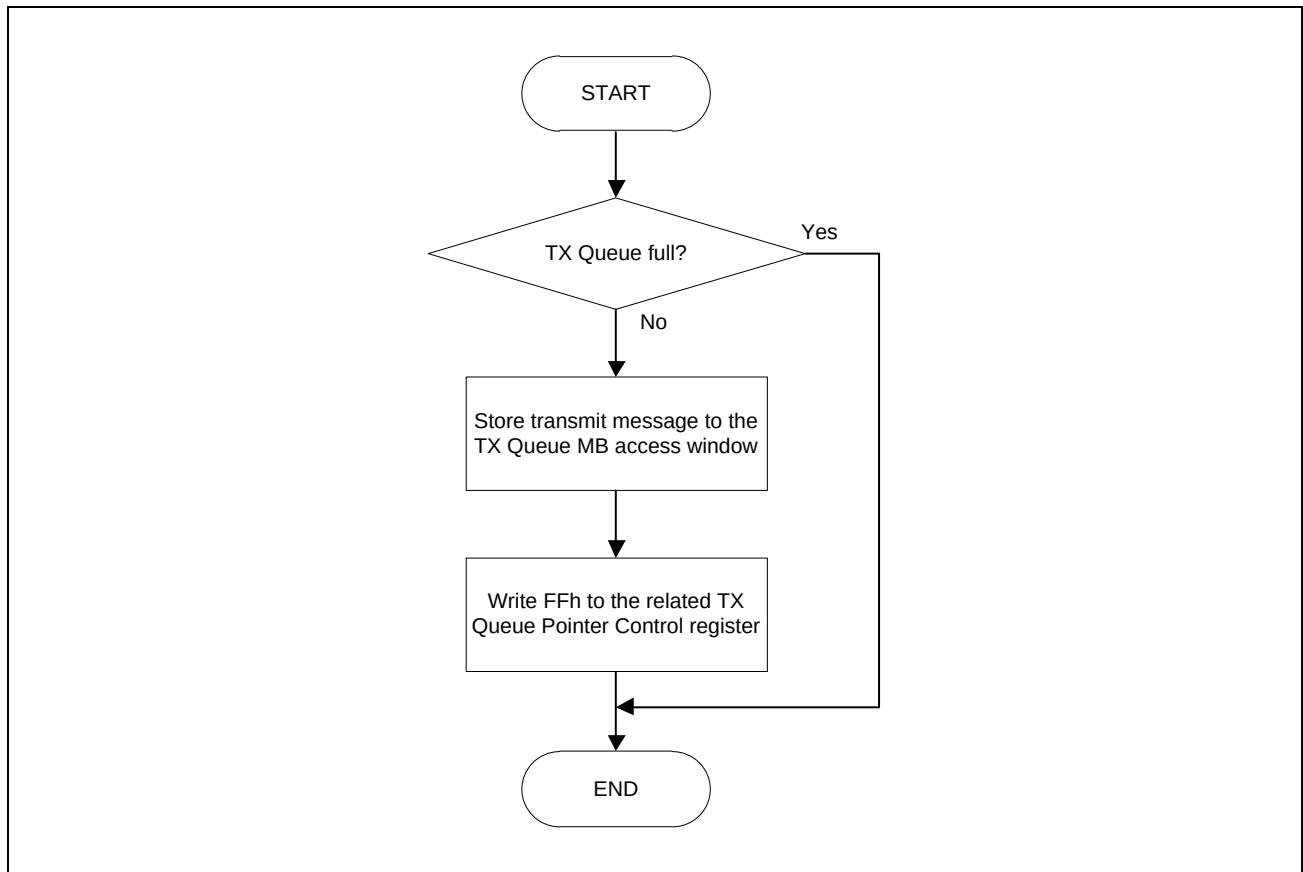


Figure 7.9-48 TX Queue Transmission Request

TXQ Name	Access Window	Range Width	Direction	Hardware Routing Access Point	CPU Access Point	DMA Access Point	Note
TXQ0	TXMB0	0, 3-16	TXMB0 → TXMB15	Yes	Yes	Yes	When using both TXQ0/1, the total number of stages is 16 or less
TXQ1	TXMB31	0, 3-16	TXMB15 → TXMB0	Yes	Yes	No	
TXQ2	TXMB32	0, 3-16	TXMB32 → TXMB47	Yes	Yes	No	When using both TXQ2/3, the total number of stages is 16 or less
TXQ3	TXMB63	0, 3-16	TXMB47 → TXMB32	No	Yes	Yes	

TXQ0 can use hardware routing, CPU access, and DMA access.

Hardware routing access, CPU access, and DMA access should not be used simultaneously. Choose one access method.

7.9.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List buffers for each CAN channel. Two TX History List buffers are provided for each CAN channel and each TX History List buffer can store up to 16 TX History List entries for a CAN channel.

The CFDTHLCCn.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFOs/TX Queue is stored or if all transmit message information from TX Queue, TX FIFO or normal TX message buffers should be stored in the TX History List for a CAN channel.

When a CFDTHLCCn.THLDGE bit is set up, the information on all the frames transmitted in GW mode is stored in TX History List.

Each transmit message can be individually configured for acceptance to the TX History List by the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted on that CAN channel.

Storing to the List is not synchronized with the status of CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the List can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF bit is set to 1b when the THLIE is configured to 1b or when the TX History List counter CFDTHLSTSn.THLMC[5:0] is increased.

The delay time is dependent on the number of channels due to internal processing.

- Maximum delay time from setting the CFDTMSTSn.TMTRF to storing the TX History List data is 224 peripheral bus clock cycles.

The History list records the following information of the transmitted message:

- Buffer type:
 - 001b: TX Message Buffer
 - 010b: TX FIFO
 - 100b: TX Queue
- Buffer number:

TX Message Buffer, TX Queue Message Buffer or TX Message Buffer Link for the Common FIFO Buffer from which the transmission occurred. The number depends on the buffer type, see **Table 7.9-29**.
- Transmission ID:

Transmission pointer stored in the transmission message
- Transmit timestamp:

Message timestamp captured at the capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:

Transmission information label stored in the transmission message.
- Transmit gateway buffer indication:

For data transmitted from the gateway, CFDTHLACC0n.TGW bit is set to 1b.

Table 7.9-29 TX History List Buffer Number Entry

CFDTHLACC0n.BT[2:0] Buffer Type		
001b	101b	100b
TX Message Buffer	TX FIFO	TX Queue
TXMB0	—	Number shown corresponds to the message buffer belonging to the TX Queue for which the frame was transmitted.
TXMB1		
TXMB2		
⋮		
TXMB13		
TXMB14		
TXMB15		
TXMB32	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO Configuration	
TXMB33		
TXMB34		
⋮		
TXMB45		
TXMB46		
TXMB47		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTSn.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, the identification number is then stored together with the other message related information to the TX History List and can be read with the Transmission ID (TID) of the TX History List Access Register.

Also for normal TX message buffers, the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List. Information label is the same.

Figure 7.9-49 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, FFh must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 7.9-50 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCCn.THLIM bit of the corresponding TX History List Configuration/Control Registers and enabled with the CFDTHLCCn.THLIE bit of the same registers, either to generate an interrupt when the history list reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTSn.THLELT bit in the TX History List Status Register.

Status of this bit is also shown by the THLES bit in the Global Error Flag Register.

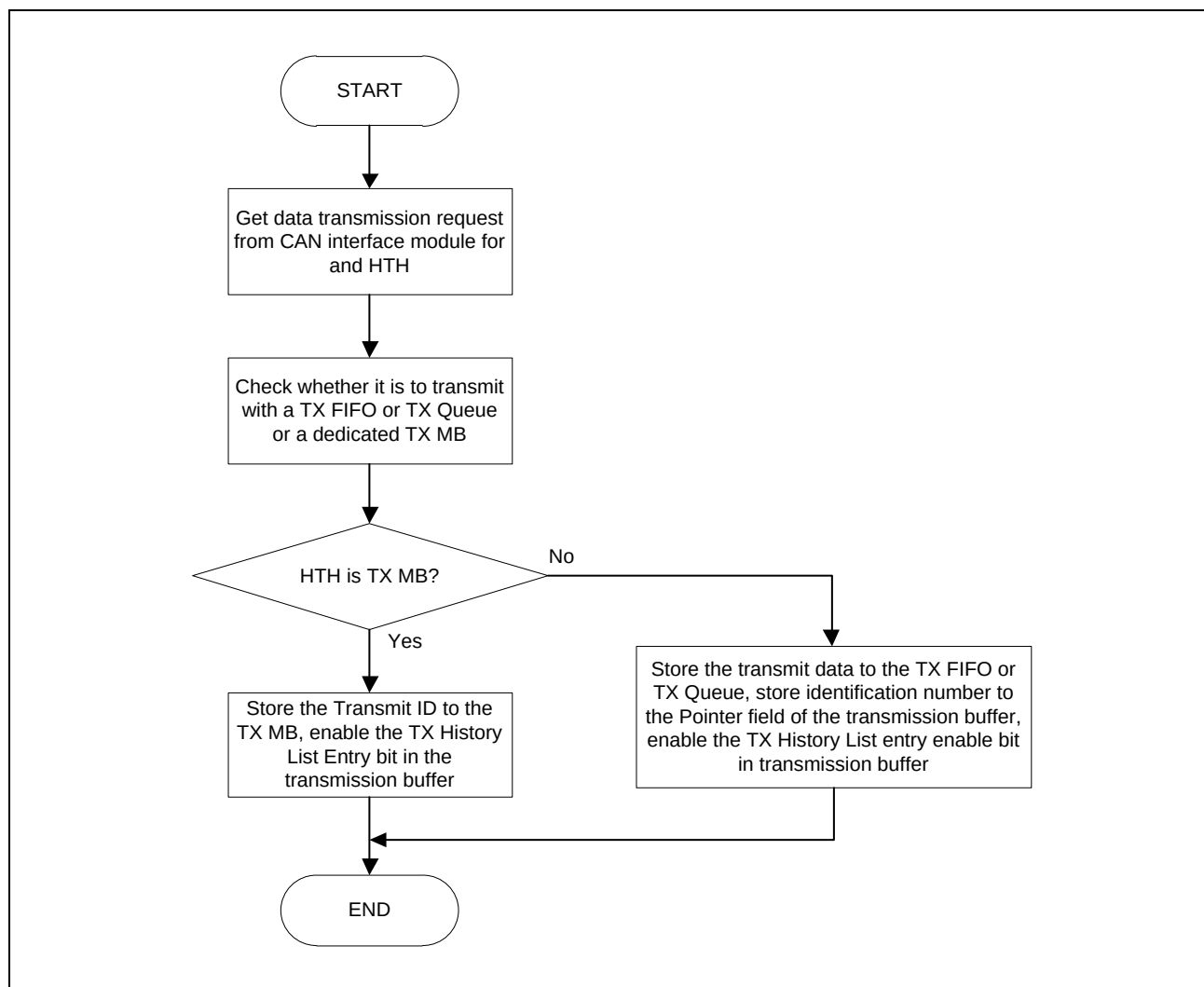


Figure 7.9-49 Transmission Preparation Flow when TX History List is Used

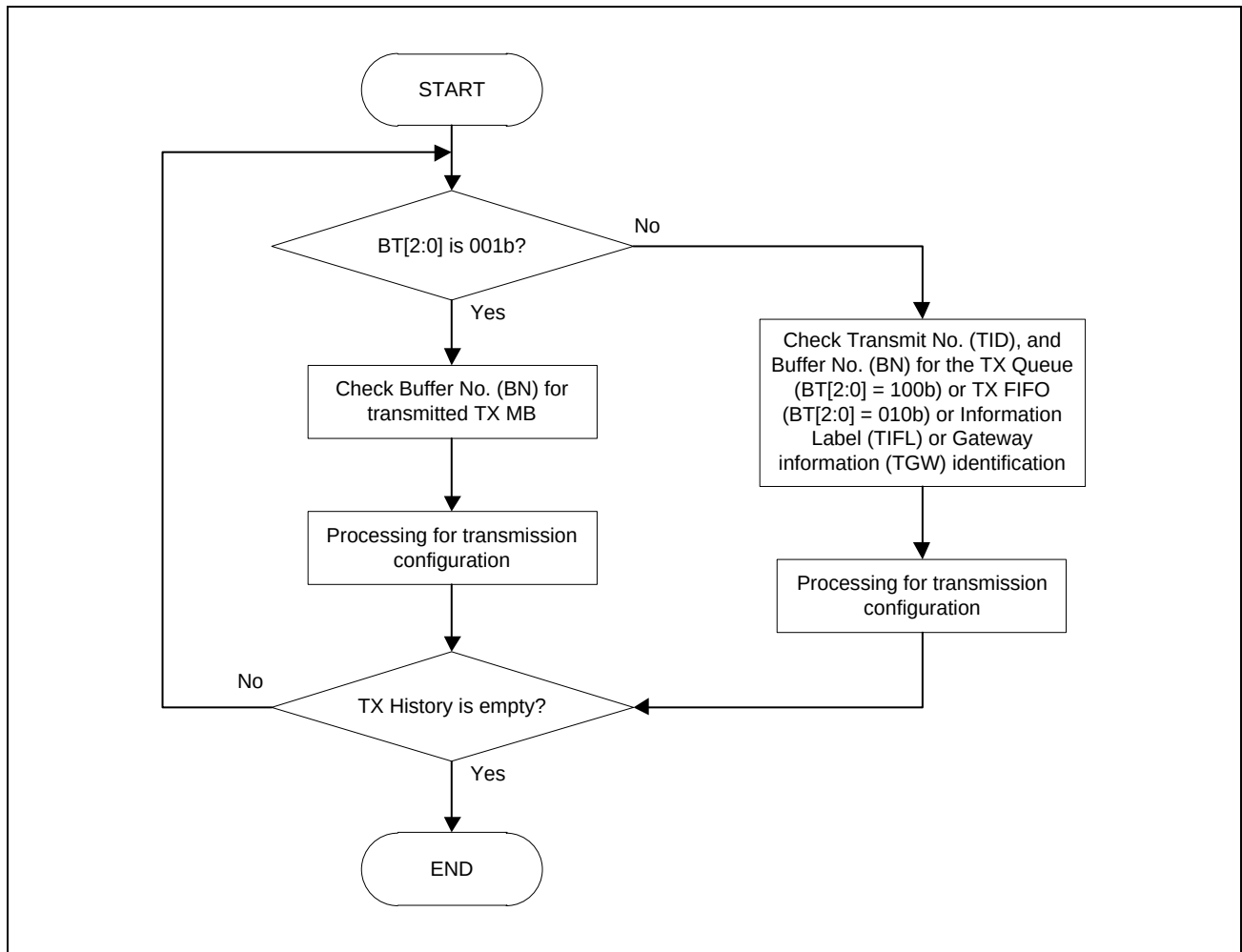


Figure 7.9-50 Example Flow for Processing TX History List Information

7.9.8.2.6 TX Data Padding

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, then the data bytes beyond the restricted range are replaced by bytes with the value of CC HEX.

This can happen for Common FIFOs configured as (TX or GW) when the transmit message DLC is higher than CFDCFCCn.CFPLS.

This can also happen in FD-only mode, if a Classical Frame is configured with a DLC larger than 8.

7.9.9 Test Mode

The CAN-FD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CAN-FD module in test modes.

All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes

7.9.9.1 Channel Specific Test Modes

Each CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External Loop Back mode)
- Self-test mode 1 (Internal Loop Back mode)
- Restricted operation mode

7.9.9.1.1 Basic Test Mode

The Basic test mode should be used when a particular test setting must be enabled other than when in Listen-Only and Self- test modes.

7.9.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional Bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, Overload flag, Active Error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX Message Buffer or TX-/GW-FIFO of this channel.

Note: If a message is stored in GW FIFO or Routing TXQ, ensure that the transmitting channel is not in Listen-only mode so that transmission is not requested for this channel from the GW FIFO or Routing TXQ.

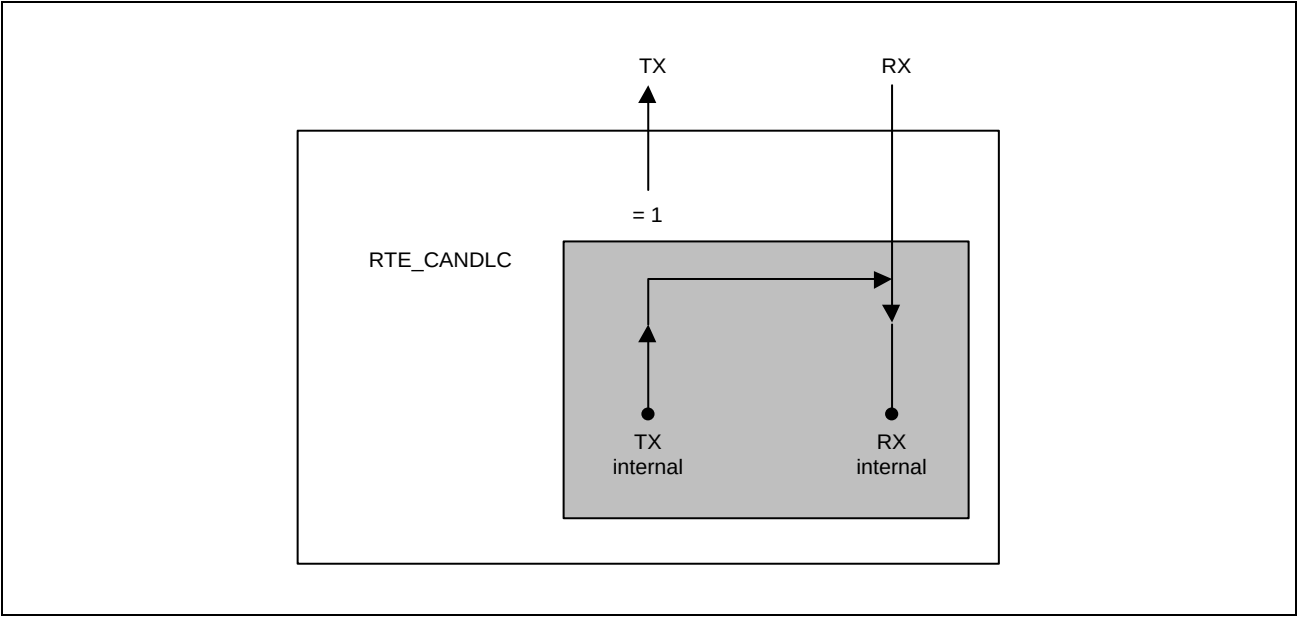


Figure 7.9-51 Listen-Only Mode

7.9.9.1.3 Self-Test Mode 0 (External Loop Back Mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and can store them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests.

The RX/TX pins should be connected to the transceiver.

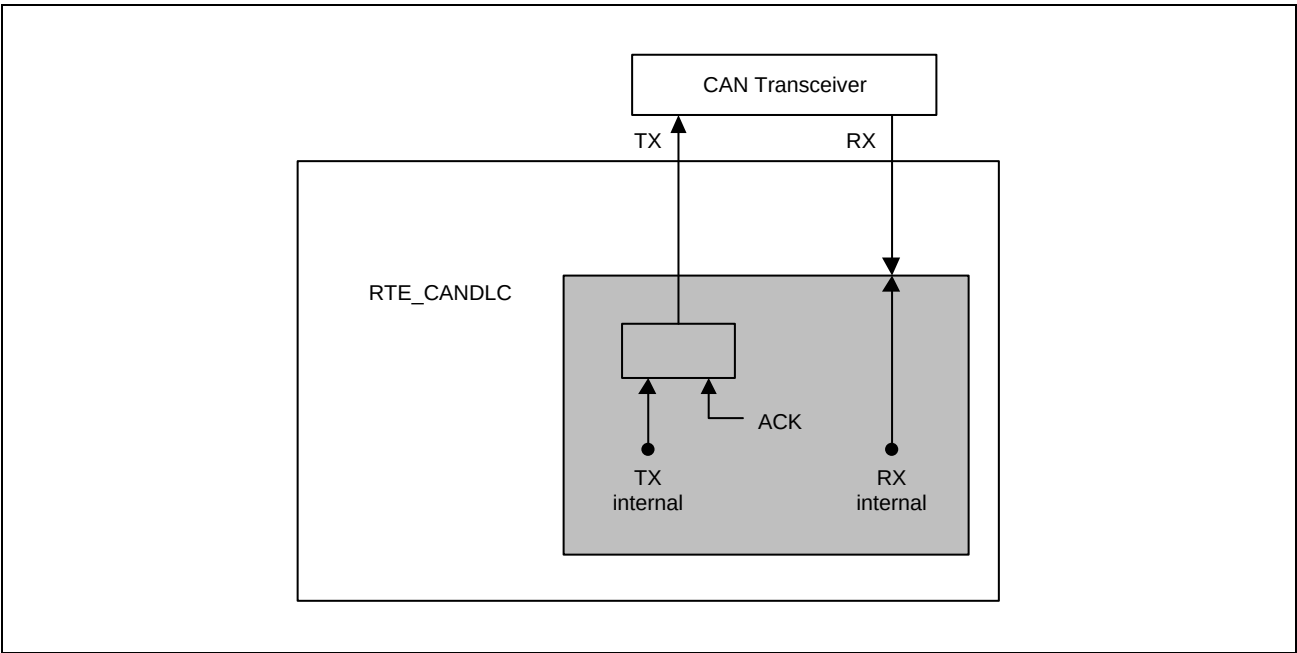


Figure 7.9-52 Self-Test Mode 0 (External Loop Back Mode)

7.9.9.1.4 Self-Test Mode 1 (Internal Loop Back Mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits.

The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.

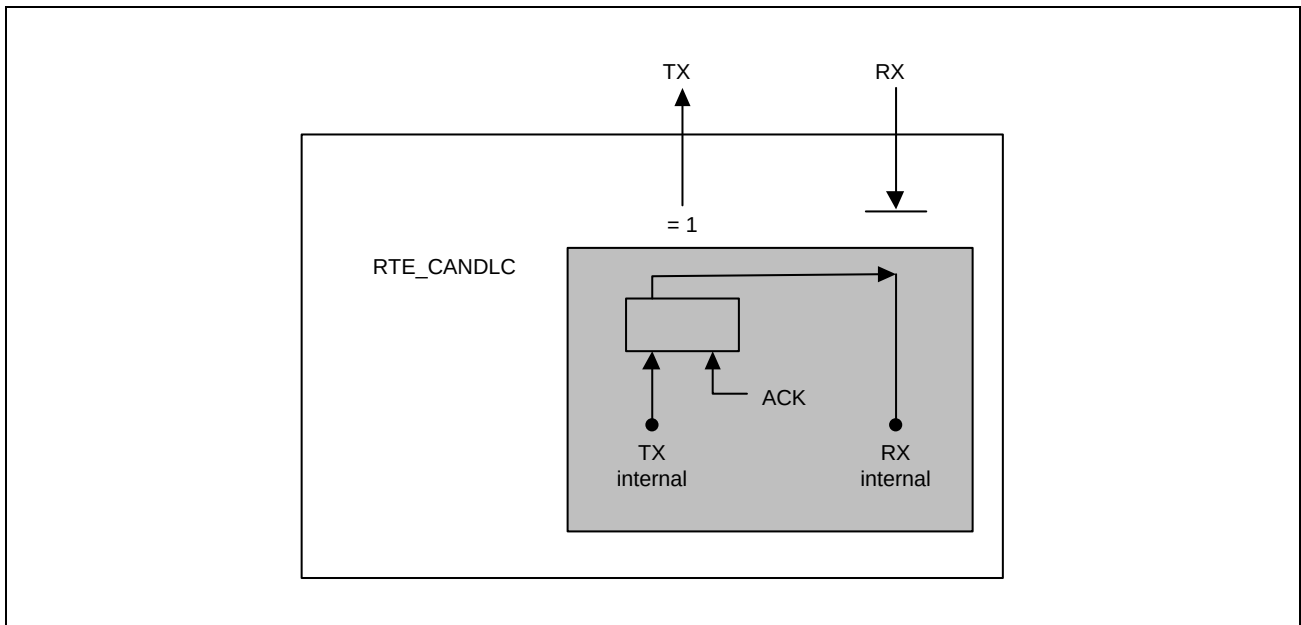


Figure 7.9-53 Self-Test Mode 1 (Internal Loop Back Mode)

7.9.9.1.5 Restricted Operation Mode

In Restricted operation mode, the CAN node is able to receive valid data and remote frames that generates the Acknowledge bit.

Active error and overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors.

The mode is specified as in ISO 11898-1. However, it is permitted to set any requested transmit.

7.9.9.2 Global Test Modes

The CAN-FD module can be configured into the following test modes:

- RAM test mode
- Internal CAN bus communication mode
- CRC error test

These test modes are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key, the related unlock key can be seen in the following table.

Test Mode	Unlock Key 1	Unlock Key 2
RAM test mode	7575h	8A8Ah

If the software sequence of the two consecutive unlock key write accesses (half-word or word accesses) is interrupted by any other write access to the SFR or if incorrect data is written to the Global Unlock Key Register then the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding Test Mode Enable bit. If this is not followed, the unlock mechanism resets and the Test Mode Enable bit cannot be set. At this time, the unlock sequence must be restarted.

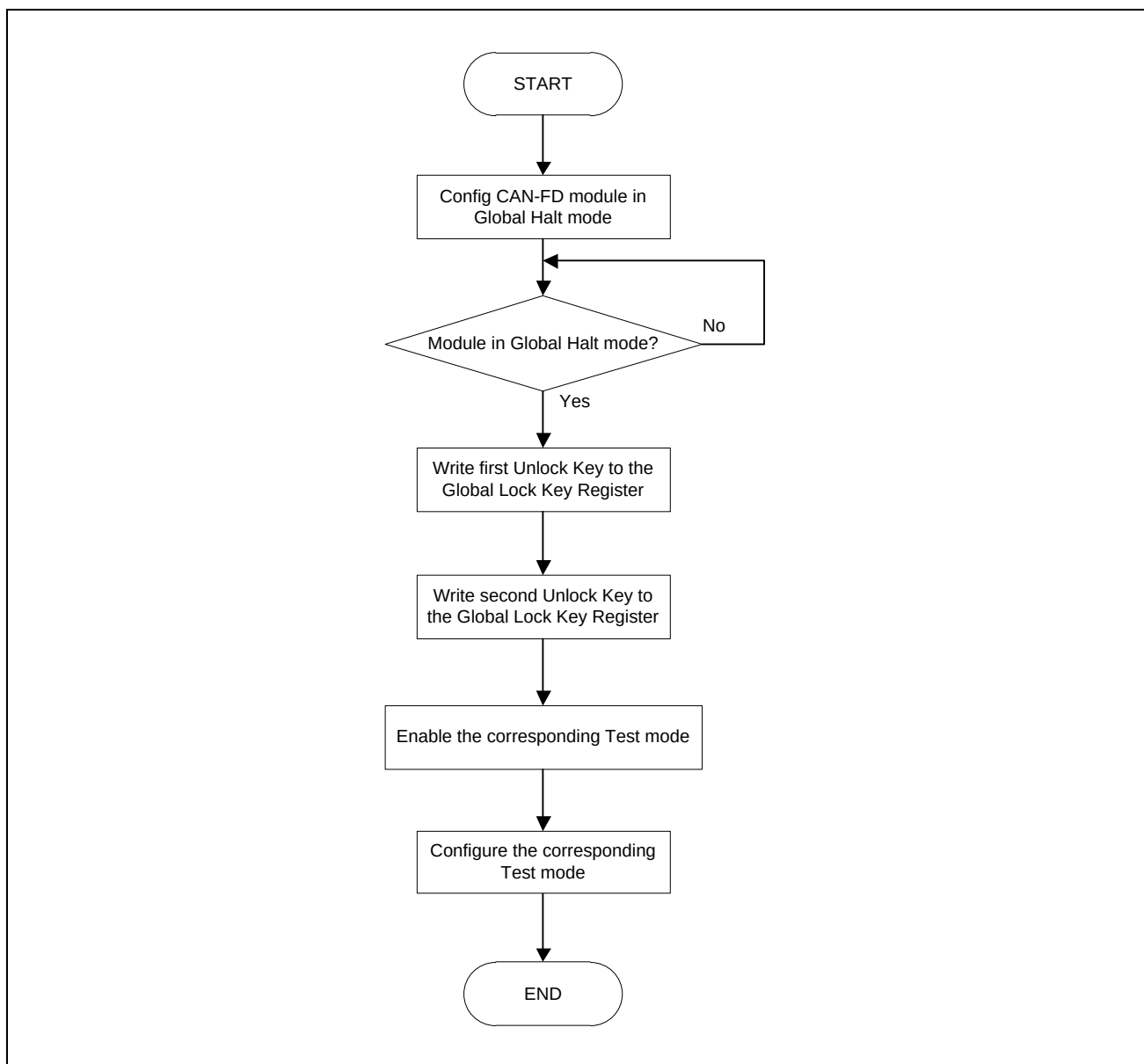


Figure 7.9-54 Unlock Software Protection Routine

7.9.9.2.1 RAM Test Mode

The CAN-FD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes each. Which can be accessed with the CFDRPGACCN register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[9:0] bits in the Global Test Control Register. Data can then be read from or written into the RAM Test Page Access Registers.

Figure 7.9-55 shows the structure of the pages in the RAM when performing a RAM test mode.

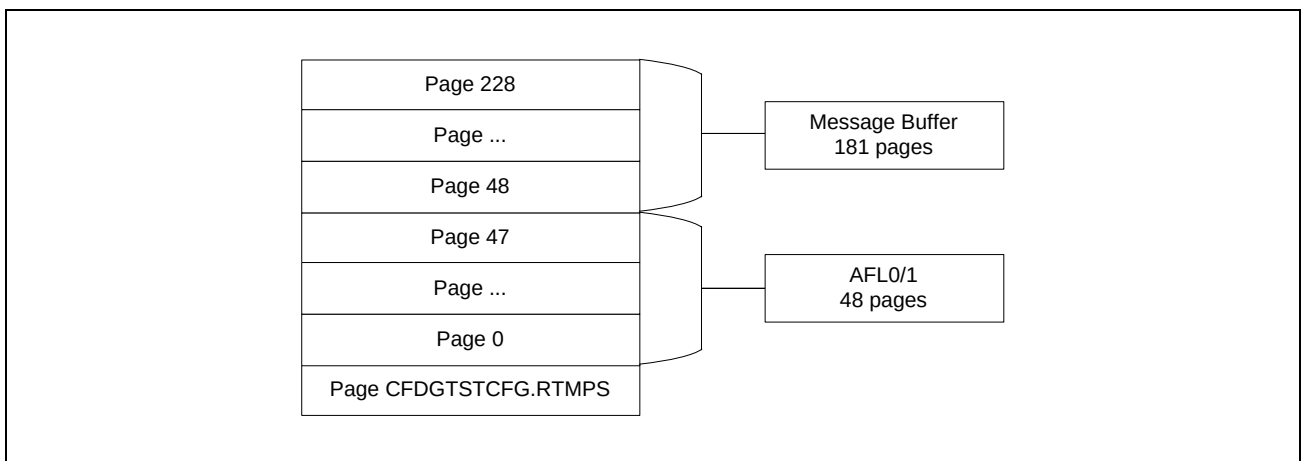


Figure 7.9-55 RAM Page Structure

The total available RAM size for a 6-CAN channel version is 12288 bytes for the AFL RAM and 46272 bytes for the Message Buffer RAM.

AFL RAM0/1 can treat RAM test mode as one RAM.

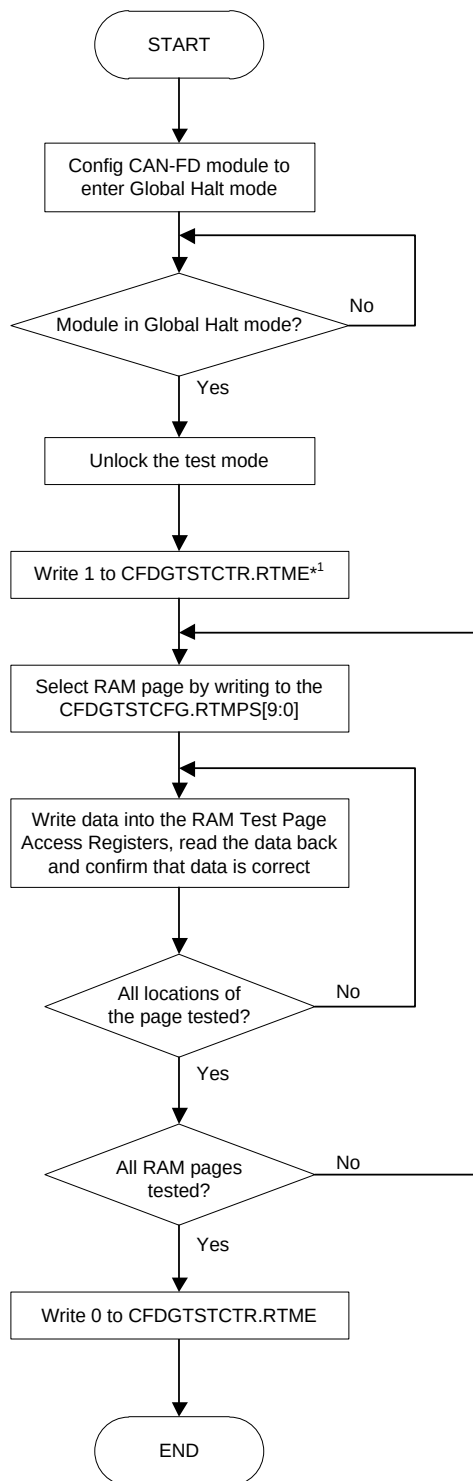
The number of pages (pn) and CFDGTSTCFG.RTMPS[9:0] values for the AFL and MB RAMs are calculated in the following way:

$pn = \text{ceil}(\text{total RAM size in bytes} / \text{number of bytes per page})$

- AFL RAM:
 $pn = \text{ceil}(12288/256) = 48$ pages
 CFDGTSTCFG.RTMPS[9:0] = 0 to 47 (02Fh) inclusive
- MB RAM:
 $pn = \text{ceil}(46272/256) = 181$ pages
 CFDGTSTCFG.RTMPS[9:0] = 48 to 228 (0E4h) inclusive

Do not access more than 192 bytes of RAM on the last page (RTMPS[9:0] = 0E4h).

Figure 7.9-56 shows the software flow for RAM test mode.



Note 1. Change into the following status before changing to RAMTEST.

- Cancel of a request of transmission
- Disable of all the FIFO and TXQ
- Clear of the receiving flag of a receiving buffer

Figure 7.9-56 Software Flow of RAM Test Mode

To exit this test mode, clear the CFDGTSTCTR.RTME bit. The CFDGTSTCTR.RTME bit is cleared by writing 0b to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CAN-FD module enters Global Reset mode from the test mode.

7.9.9.2.2 Internal CAN Bus Communication Mode

The CAN-FD module can be configured in Internal CAN bus communication test mode by setting the CFDGTSTCTR.ICBCTME bit in the Global Test Control Register. This is a special test mode, in which the CAN channels can be connected together internally to generate a CAN cluster within the CAN-FD module.

Only use the following sequence to enter Internal CAN bus communication test mode:

1. Configure all channels in Halt mode and check that all channels have entered Global Halt mode.
2. Write data into the Global Test Configuration Register to select the channels participating in the Internal CAN bus communication test.
3. Set the CFDGTSTCTR.ICBCTME bit of the Global Test Control Register.
4. Check that CFDGTSTCTR.ICBCTME bit is set in the Global Test Control Register.

In this mode, the TXD outputs of the channels participating (configured) in Internal CAN bus communication mode are connected together using AND gate. The output of the AND gate is connected to the RXD inputs of all participating channels to create a CAN cluster within the CAN-FD module. The channels are isolated from the external CAN bus while the CAN-FD module is in this test mode.

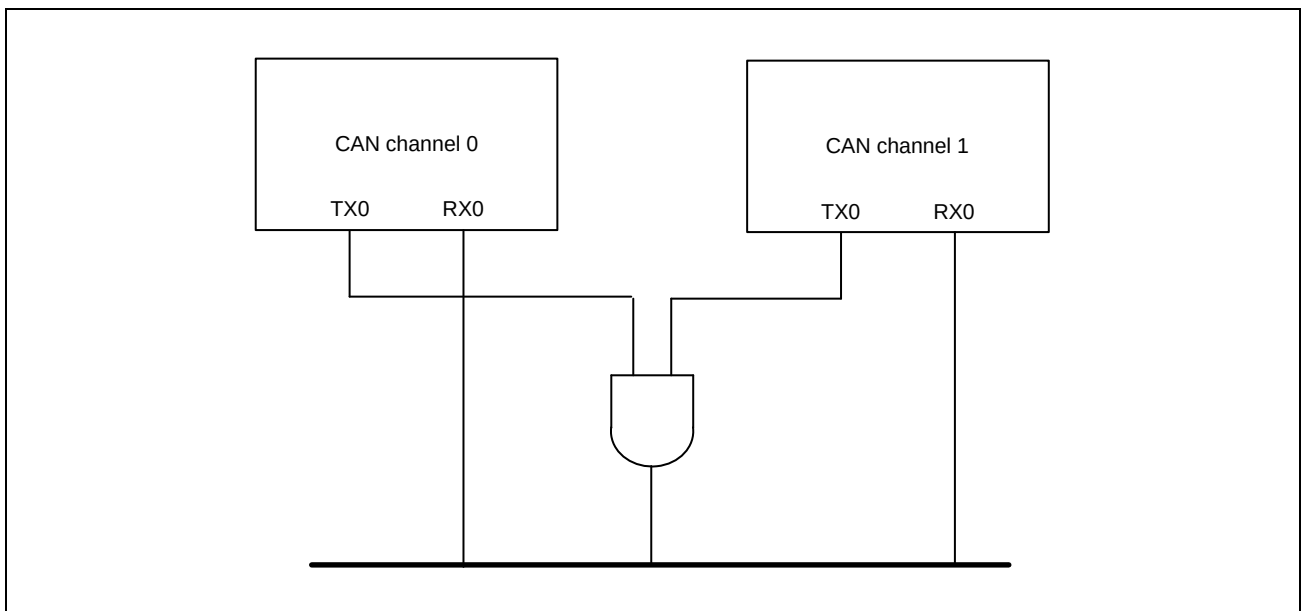


Figure 7.9-57 Internal CAN Bus Connections

The AFL, Flat RX message buffers, FIFO buffers, Flat TX message buffers, and various registers can now be configured as normal to start communication between channels.

The channels not participating in Internal CAN bus should only be configured in Halt mode.

7.9.9.2.3 CRC Error Test

After the CAN-FD module has been configured in Internal CAN bus communication test mode, use the following sequence to perform CRC error testing. In the following sequence, channel x is the reference transmitter CAN-FD module and channel y is the receiver CAN-FD module where $(x, y = [0, \dots, n])$ and $x \neq y$:

1. Configure channel x node to transmit one reference message.
2. Set the CFDCyCTR.CRCT bit to 1b, in order to invert the first bit of the incoming bit stream from channel x.
3. Set the CFDTMCx.TMTR.
4. Read either CFDCyERFL.CRCREG or CFDCyFDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from channel x.
5. Check that CFDCyERFL.CERR is 1b.

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC error test.

7.9.10 Bus Traffic Measurement

The idle time of the CAN bus can be measured using CANFD_0_pclk or CANFD_0_clkc. Bus traffic can be calculated based on the measurement results.

7.9.10.1 How to Count the CAN Bus Idle Time

Figure 7.9-58 shows the concept of measuring the idle time of the CAN bus.

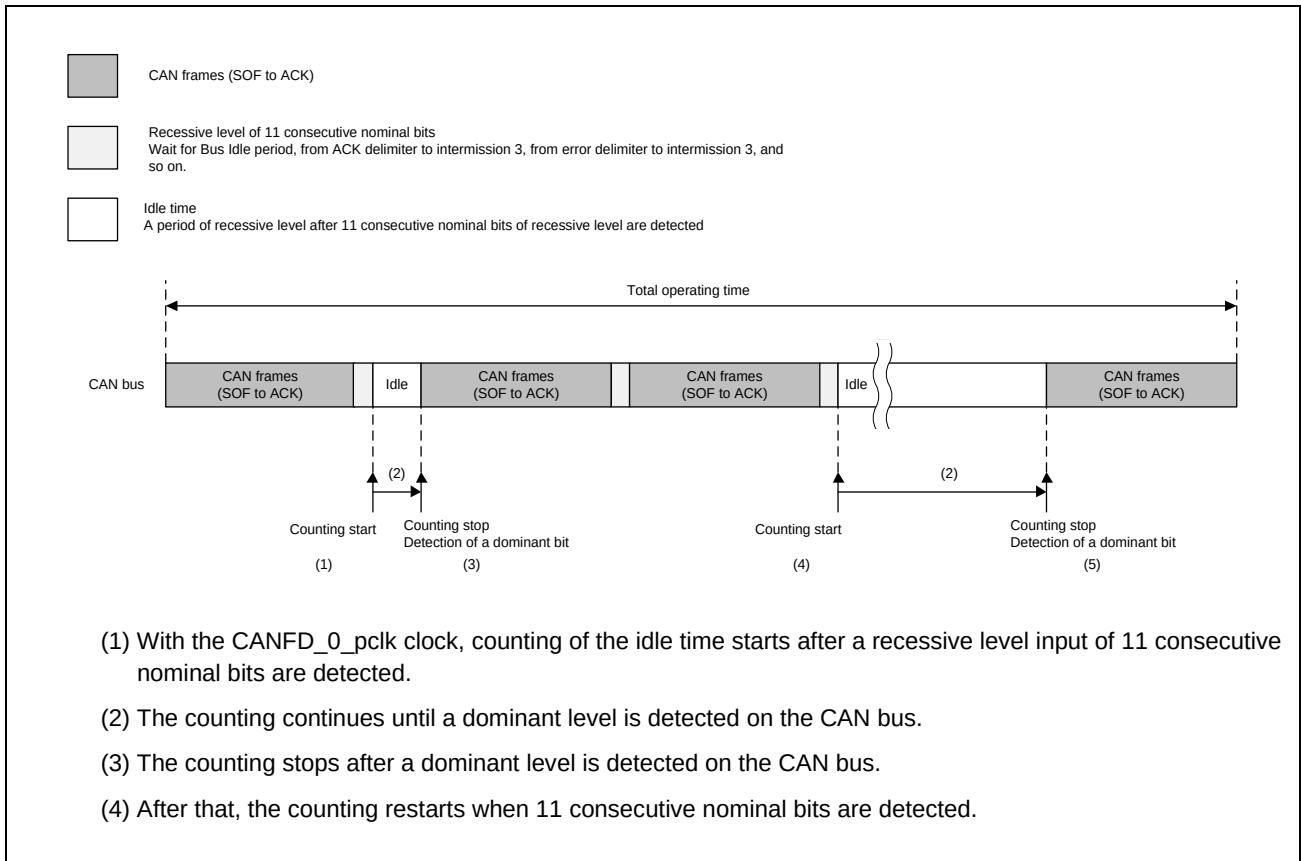


Figure 7.9-58 How to Measure Idle Time of the CAN Bus

7.9.10.2 Operations and Measurement Procedure

The following procedure is for measuring the idle time of the CAN bus.

1. The channel to be measured transits to operation mode.
2. Write 1b to CFDCnBLCT.BLCE bit to set the measuring counter to operating mode.
3. Write 1b to CFDCnBLCT.BLCLD bit to clear the counter register.
4. Detect a recessive level input of 11 consecutive nominal bits.
5. Start counting the bus idle time.
6. Detect a dominant level.
7. The counting stops.
8. Detect a recessive level input of 11 consecutive nominal bits.
9. The counting starts.

10. Write 1b to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the counter value to CFDCnBLSTS.
11. Read the value of CFDCnBLSTS.

To stop the measurement counter, write 0b to CFDCnBLCT.BLCE bit.

To initialize the counter, write 1b to CFDCnBLCT.BLCLD bit.

This measurement is enabled when the channel to be measured is in Operation mode.

When the relevant channels are in reset mode, the counter does not operate.

Also, accurate measurements are not available in test mode.

Write 1b to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the value of the counter to CFDCnBLSTS.

The lower 3 bits of CFDCnBLSTS are fixed to 0b.

Based on the values of the counter, software can calculate the CAN bus traffic according to the following formulas:

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{\text{Bus operating time}}{\text{Total operating time}} = \text{Bus usage ratio}$$

- Total idle time: a value read from CFDCnBLSTS × a clock cycle of CANFD_0_pclk
- Total operating time: a setting interval of CFDCnBLCT.BLCLD bit

Example: Below is a calculation example under the following conditions:

- Conditions: nominal bit rate = 1 Mbps
- CANFD_0_clkc = 80 MHz (= 12.5ns)
- A setting interval of CFDCnBLCT.BLCLD bit = cycle of 1 ms
- A read value of CFDCnBLSTS register = 4E20h

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{(1000000 \text{ ns} - 20000 \times 12.5 \text{ ns})}{1000000 \text{ ns}} = 75\%$$

7.9.11 Flexible CAN mode

In this mode, it is possible to connect 2 channels of the CAN module to a single CAN driver.

The pairs of channels in Flexible CAN mode are as follows.

- When the CFDGFCMC.FLXC0 bit is set, channels 0 and 1 of the RS-CAN-FD module are in Flexible CAN mode.
- When the CFDGFCMC.FLXC1 bit is set, channels 2 and 3 of the RS-CAN-FD module are in Flexible CAN mode.
- When the CFDGFCMC.FLXC2 bit is set, channels 4 and 5 of the RS-CAN-FD module are in Flexible CAN mode.

Channel n+1 uses the TX/RX terminal of channel n.

The TX/RX terminal of channel n+1 cannot be used.

In Flexible CAN mode, each channel performs communication processing independently.

However, when one of the channels transmits, the other channel will not return an acknowledge bit.

Remark When operating in Flexible CAN mode, the error counters (TEC/REC) of the two CAN nodes are not synchronized with each other.

Flexible transmission buffer assignment configured in the CFDGFTBAC register and Flexible CAN mode configured in the CFDGFCMC register should not be used simultaneously.

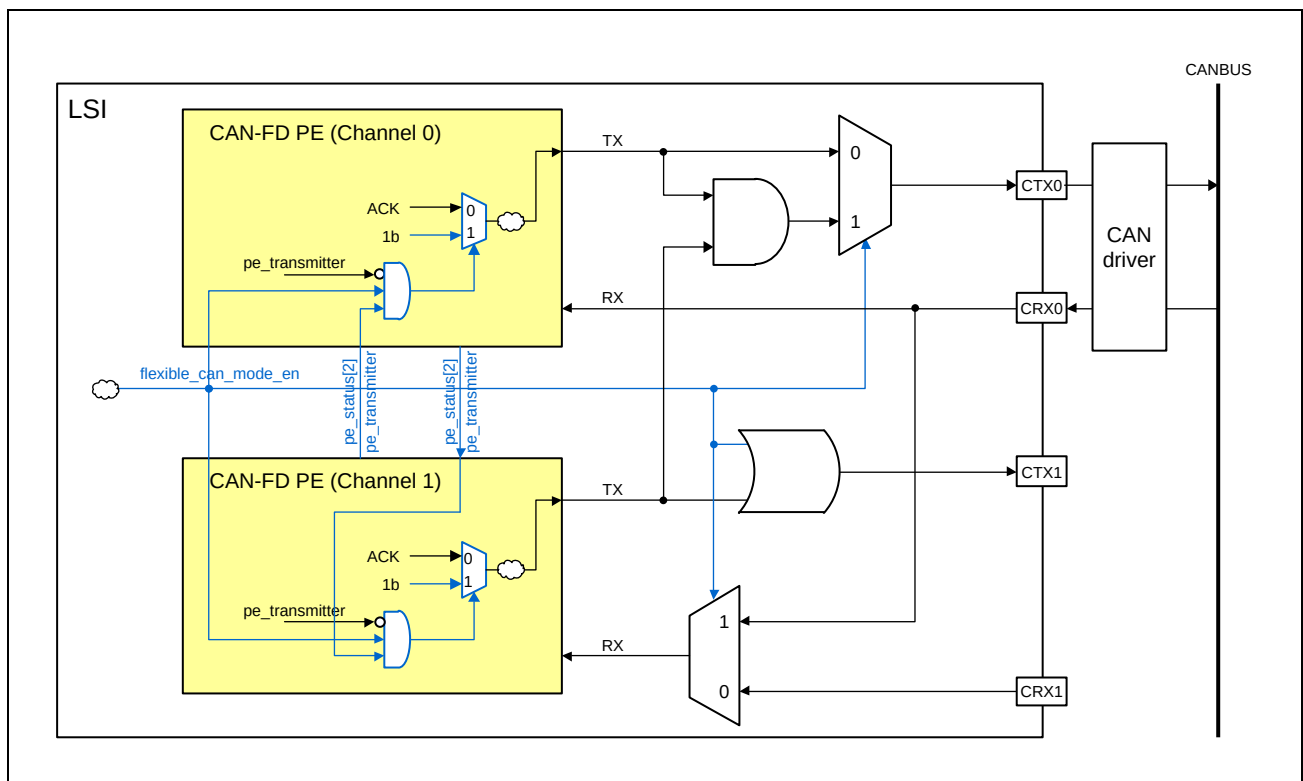


Figure 7.9-59 Diagram of the Flexible CAN for Channel 0 and Channel 1

7.9.12 Flexible Transmission Buffer Assignment

Each channel has 32 transmit buffers for exclusive use.

This channel can borrow up to 16 TXMBs from the next channel in case it runs out of TXMBs.

Buffers that can be rented range from 4 to 16 buffers in increments of 4 units.

The pairs of channels in Flexible CAN mode are as follows.

- When the CFDGFTBAC.FLXMB0 bit is set, flexible transmission buffer assignment between channel 0 and channel 1.
- When the CFDGFTBAC.FLXMB1 bit is set, flexible transmission buffer assignment between channel 2 and channel 3.
- When the CFDGFTBAC.FLXMB2 bit is set, flexible transmission buffer assignment between channel 4 and channel 5.

Flexible transmission buffer assignment configured in the CFDGFTBAC register and Flexible CAN mode configured in the CFDGFCMC register should not be used simultaneously.

The interrupt of the rented buffer is output to the interrupt of the rented channel.

When using TXQ through the rented buffer, TXQ should only be set within the rented range.

The rented TXMB operates in the mode of the channel to be used.

For example, if channel 1 is in reset mode, channel 0 transmission is possible using a rented TXMB.

Also, the rented TXMB is affected by the transmission status of the rented channel, TX SCAN PROCESS, or aborted transmission.

In message buffer number priority mode, TXMB0 has a higher priority and TXMB79 (when 16 buffers are borrowed) has a lower priority. The priority of the rented TXMB is lowered.

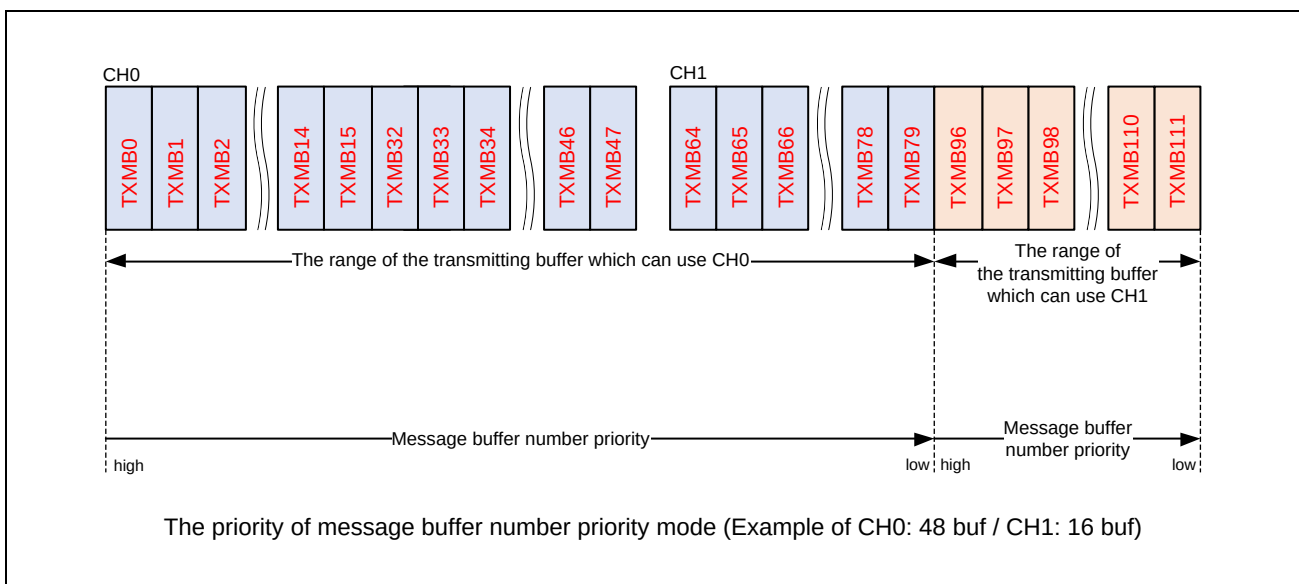


Figure 7.9-60 Example of Message Buffer Number Priority Mode

The example of a rental channel 0 and channel 1 is shown below.

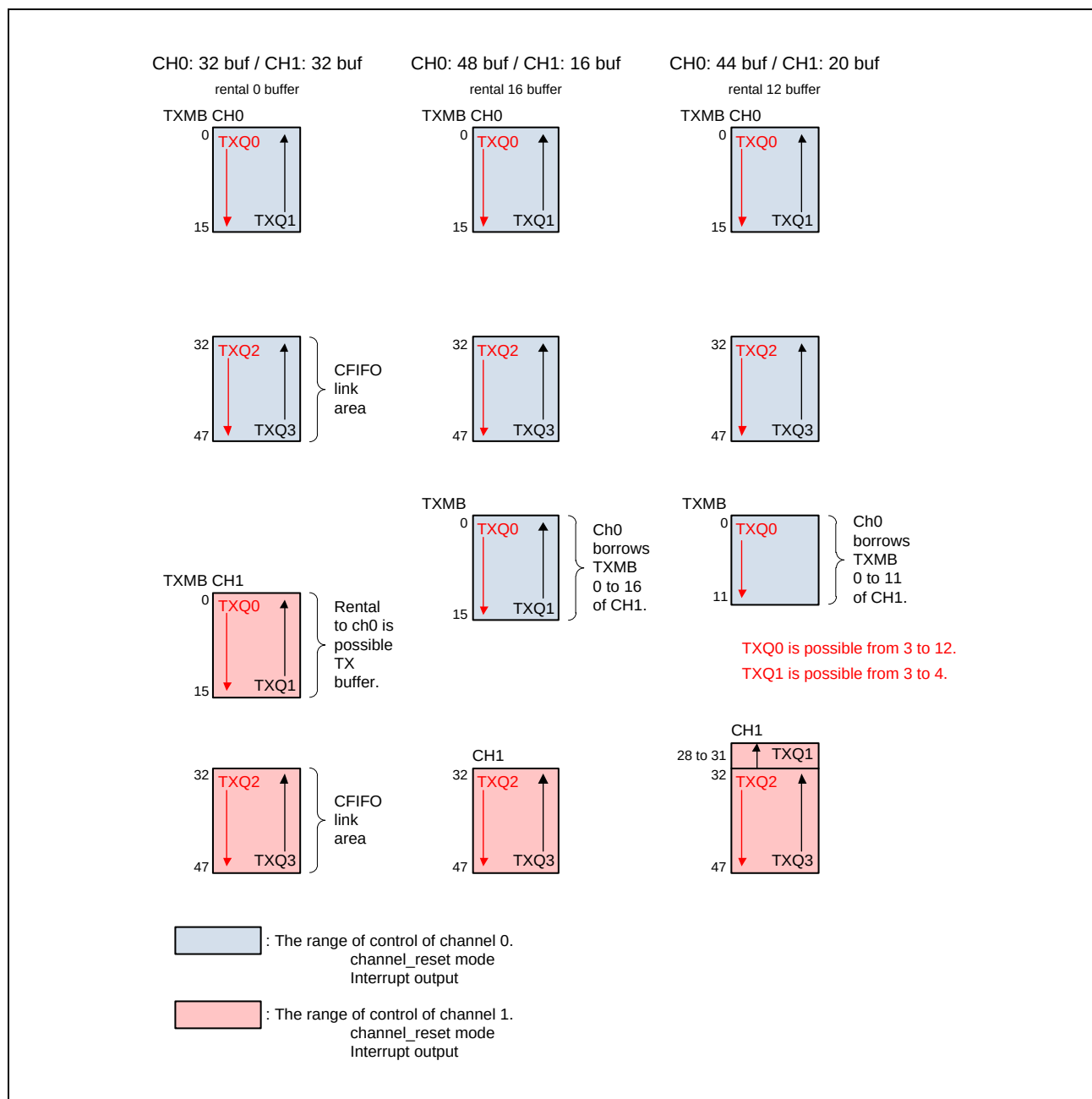


Figure 7.9-61 Example of Message Buffer Number Priority Mode (Channels 0 and 1)

SECTION 7 LOW-SPEED INTERFACE

7.10 12-Bit A/D Converter (ADC)

7.10.1 Overview

This LSI incorporates three units of 12-bit successive approximation A/D converter. Analog inputs of up to 8 channels are selectable per unit and total up to 24 channels.

Each unit of the 12-bit A/D converter converts analog inputs of up to 8 selected channels to a 12-bit digital value through successive approximation.

The 12-bit A/D converter has three operating modes: single scan mode in which analog inputs selected arbitrarily are converted only once in ascending order of channel number; continuous scan mode in which analog inputs selected arbitrarily are continuously converted in ascending order of channel number; and group scan mode in which analog inputs are arbitrarily selected and divided into two groups (group A and group B) or three groups (group A, group B and group C), and then analog inputs of channels selected for each group are converted in ascending order of channel number.

In group scan mode, the conditions for scanning start of each groups (A, B or A, B, C) (internal trigger) can be independently selected, thus allowing A/D conversion of each groups to be started independently. In addition to the above actions, the group priority operation in group scan mode accepts the start of scanning the priority group during the scan of the low priority group, interrupts the scan of the low priority group, and starts the scan of the priority group. The priority of group priority operations is group A > group B > group C. In group priority operation, if the scan start of group B is accepted during the scan of group C, the scan of group C is interrupted and the scan of group B is started. If the scan start of group A is accepted during the scan of group C, the scan of group A is started. The scan of Group C is interrupted and the scan of Group A is started. Similarly, if the scan start of Group A is accepted during the scan of Group B, the scan of Group B is interrupted and the scan of Group A is started. The interrupted group scan is able to resume after the priority group scan is complete also.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second internal triggers are stored into different registers (duplication of A/D conversion data).

Safety features such as data register overwrite check and automatic clearing of data registers are available.

The compare function is a function that outputs an interrupt (ada_compai_n / ada_compbi_n) when the A/D conversion value of the selected channel by specifying the upper and lower reference values for each window A/B matches the comparison conditions. The event (ada_elcondmtch / ada_elcondunmtch) is output according to the event conditions (A or B, A and B, A exor B). In addition, a comparator operation that compares the A/D conversion value with the lower reference value is also possible.

Table 7.10-1 lists the specifications of the 12-bit A/D converter and **Table 7.10-2** indicates the functions of the 12-bit A/D converter. **Figure 7.10-1** shows a block diagram of the 12-bit A/D converter.

Table 7.10-1 ADC Specifications (1/2)

Item	Description
Number of units	3 units
Input channels	- Up to 8 channels per unit - Up to 24 channels in total for 3 units
A/D conversion method	Successive approximation method

Table 7.10-1 ADC Specifications (2/2)

Item	Description
Resolution	Selectable from 12 bits or 8 bits
Sampling rate	• 2.5 Msps (when A/D conversion clock ADC_m_ADCLK is 50 MHz) (m = 0 to 2) • 2.0 Msps (when A/D conversion clock ADC_m_ADCLK is 40 MHz) (m = 0 to 2) • 1.0 Msps (when A/D conversion clock ADC_m_ADCLK is 20 MHz) (m = 0 to 2) • 0.5 Msps (when A/D conversion clock ADC_m_ADCLK is 10 MHz) (m = 0 to 2) • 0.25 Msps (when A/D conversion clock ADC_m_ADCLK is 5 MHz) (m = 0 to 2)
A/D conversion clock	ADC_m_ADCLK (50 MHz / 40 MHz / 20 MHz / 10 MHz / 5 MHz) (m = 0 to 2)
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of arbitrarily selected channels • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels • Group scan mode: Two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used. Only the combination of groups A and B can be selected when the number of the groups is two. Analog inputs that are arbitrarily selected are divided into two groups (group A and B) or three groups (group A, B, and C), and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of groups A, B, and C (internal trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently. • Group scan mode (when Group A is given priority): If a priority-group trigger is input during scanning of the low-priority group, scan of the low-priority group is stopped and scan of the priority group is started. The priority order is group A (highest) > group B > group C (lowest). Whether or not to restart scanning of the low-priority group after processing for the high-priority group completes, is selectable. Rescan can also be set to start either from the beginning of the selected channel or the channel on which A/D conversion is not completed.
Conditions for A/D conversion start	• Software trigger • Internal trigger Trigger by the multi-function timer pulse unit (GPT) and event link controller (ELC)*1 • External trigger A/D conversion can be triggered by the external trigger ADTRG.
Functions	• Selectable A/D-converted value addition mode or average mode • Double-trigger mode (duplication of A/D conversion data) • Automatic clear function for A/D data registers • Digital comparison of values in the comparison and data registers, and between values in the data registers • Overwrite checking function of the A/D data register
Interrupt sources and ELC events	• ada_adireq_n: A/D scan end interrupt, A/D scan end interrupt for Group A) • ada_gbadireq_n: A/D scan end interrupt for Group B • ada_gcadireq_n: A/D scan end interrupt for Group C • ada_compai_n: Window A compare match • ada_compb_i_n: Window B compare match • ada_adelcreq: A/D scan end interrupt for ELC • ada_elcondmitch: compare match • ada_elcondunmitch: compare unmatched • ada_adereq_n: A/D error interrupt request by overwrite check
ELC interface	• An ELC event is generated upon completion of all scans • Able to start scanning by a trigger from the ELC
Power-Down mode	The A/D converter can be set to power-down mode.

Note 1. The ELC is included in the Interrupt Control Unit (ICU).

Table 7.10-2 Function of ADC (1/2)

Parameter			Unit (ADC)
Analog input channel			ANI000 to ANI007
Conditions for A/D conversion start	Software	Software trigger	Enabled
	External trigger	Trigger input pin (Use only internal trigger as the scan start in group scan mode)	ADTRG
	Internal trigger	Compare match with GPT00.GTADTRA	GTADTRA0N
		Compare match with GPT00.GTADTRB	GTADTRB0N
		Compare match with GPT01.GTADTRA	GTADTRA1N
		Compare match with GPT01.GTADTRB	GTADTRB1N
		Compare match with GPT02.GTADTRA	GTADTRA2N
		Compare match with GPT02.GTADTRB	GTADTRB2N
		Compare match with GPT03.GTADTRA	GTADTRA3N
		Compare match with GPT03.GTADTRB	GTADTRB3N
		Compare match with GPT04.GTADTRA	GTADTRA4N
		Compare match with GPT04.GTADTRB	GTADTRB4N
		Compare match with GPT05.GTADTRA	GTADTRA5N
		Compare match with GPT05.GTADTRB	GTADTRB5N
		Compare match with GPT06.GTADTRA	GTADTRA6N
		Compare match with GPT06.GTADTRB	GTADTRB6N
		Compare match with GPT07.GTADTRA	GTADTRA7N
		Compare match with GPT07.GTADTRB	GTADTRB7N
		Compare match with GPT10.GTADTRA	GTADTRA8N
		Compare match with GPT10.GTADTRB	GTADTRB8N
		Compare match with GPT11.GTADTRA	GTADTRA9N
		Compare match with GPT11.GTADTRB	GTADTRB9N
		Compare match with GPT12.GTADTRA	GTADTRA10N
		Compare match with GPT12.GTADTRB	GTADTRB10N
		Compare match with GPT13.GTADTRA	GTADTRA11N
		Compare match with GPT13.GTADTRB	GTADTRB11N
		Compare match with GPT14.GTADTRA	GTADTRA12N
		Compare match with GPT14.GTADTRB	GTADTRB12N
		Compare match with GPT15.GTADTRA	GTADTRA13N
		Compare match with GPT15.GTADTRB	GTADTRB13N
		Compare match with GPT16.GTADTRA	GTADTRA14N
		Compare match with GPT16.GTADTRB	GTADTRB14N
		Compare match with GPT17.GTADTRA	GTADTRA15N
		Compare match with GPT17.GTADTRB	GTADTRB15N
		Compare match with GPT00.GTADTRA or Compare match with GPT00.GTADTRB	GTADTRA0N or GTADTRB0N
		Compare match with GPT01.GTADTRA or Compare match with GPT01.GTADTRB	GTADTRA1N or GTADTRB1N
		Compare match with GPT02.GTADTRA or Compare match with GPT02.GTADTRB	GTADTRA2N or GTADTRB2N
		Compare match with GPT03.GTADTRA or Compare match with GPT03.GTADTRB	GTADTRA3N or GTADTRB3N

Table 7.10-2 Function of ADC (2/2)

Parameter			Unit (ADC)
Conditions for A/D conversion start	Internal trigger	Compare match with GPT04.GTADTRA or Compare match with GPT04.GTADTRB	GTADTRA4N or GTADTRB4N
		Compare match with GPT05.GTADTRA or Compare match with GPT05.GTADTRB	GTADTRA5N or GTADTRB5N
		Compare match with GPT06.GTADTRA or Compare match with GPT06.GTADTRB	GTADTRA6N or GTADTRB6N
		Compare match with GPT07.GTADTRA or Compare match with GPT07.GTADTRB	GTADTRA7N or GTADTRB7N
		Compare match with GPT10.GTADTRA or Compare match with GPT10.GTADTRB	GTADTRA8N or GTADTRB8N
		Compare match with GPT11.GTADTRA or Compare match with GPT11.GTADTRB	GTADTRA9N or GTADTRB9N
		Compare match with GPT12.GTADTRA or Compare match with GPT12.GTADTRB	GTADTRA10N or GTADTRB10N
		Compare match with GPT13.GTADTRA or Compare match with GPT13.GTADTRB	GTADTRA11N or GTADTRB11N
		Compare match with GPT14.GTADTRA or Compare match with GPT14.GTADTRB	GTADTRA12N or GTADTRB12N
		Compare match with GPT15.GTADTRA or Compare match with GPT15.GTADTRB	GTADTRA13N or GTADTRB13N
		Compare match with GPT16.GTADTRA or Compare match with GPT16.GTADTRB	GTADTRA14N or GTADTRB14N
		Compare match with GPT17.GTADTRA or Compare match with GPT17.GTADTRB	GTADTRA15N or GTADTRB15N
		ELC trigger	ELCTRG0
		Interrupt	ada_adireq_n ada_gbadireq_n ada_gcadireq_n ada_compai_n ada_compbi_n
		Output to ELC	ada_adelcreq ada_elcondmtch ada_elcondunmtch
		Power-Down mode settings	Available

Table 7.10-3 ADC Interrupt Sources

Name	Interrupt Sources	CPU request	DMAC Activation	ELC Activation	EMC request
ada_adireq_n	A/D scan end interrupt, A/D scan end interrupt for Group A	Possible	Possible	Possible	Not possible
ada_gbadireq_n	A/D scan end interrupt for Group B	Possible	Possible	Possible	Not possible
ada_gcadireq_n	A/D scan end interrupt for Group C	Possible	Possible	Possible	Not possible
ada_compai_n	Window A compare match	Possible	Not possible	Not possible	Not possible
ada_compbi_n	Window B compare match	Possible	Not possible	Not possible	Not possible
ada_adelcreq	A/D scan end interrupt for ELC	Not possible	Not possible	Possible	Not possible
ada_elcondmtch	Compare match	Not possible	Possible	Possible	Not possible
ada_elcondunmtch	Compare mismatch	Not possible	Possible	Possible	Not possible
ada_adereq_n	A/D error interrupt request by overwrite check	Not possible	Not possible	Not possible	Possible

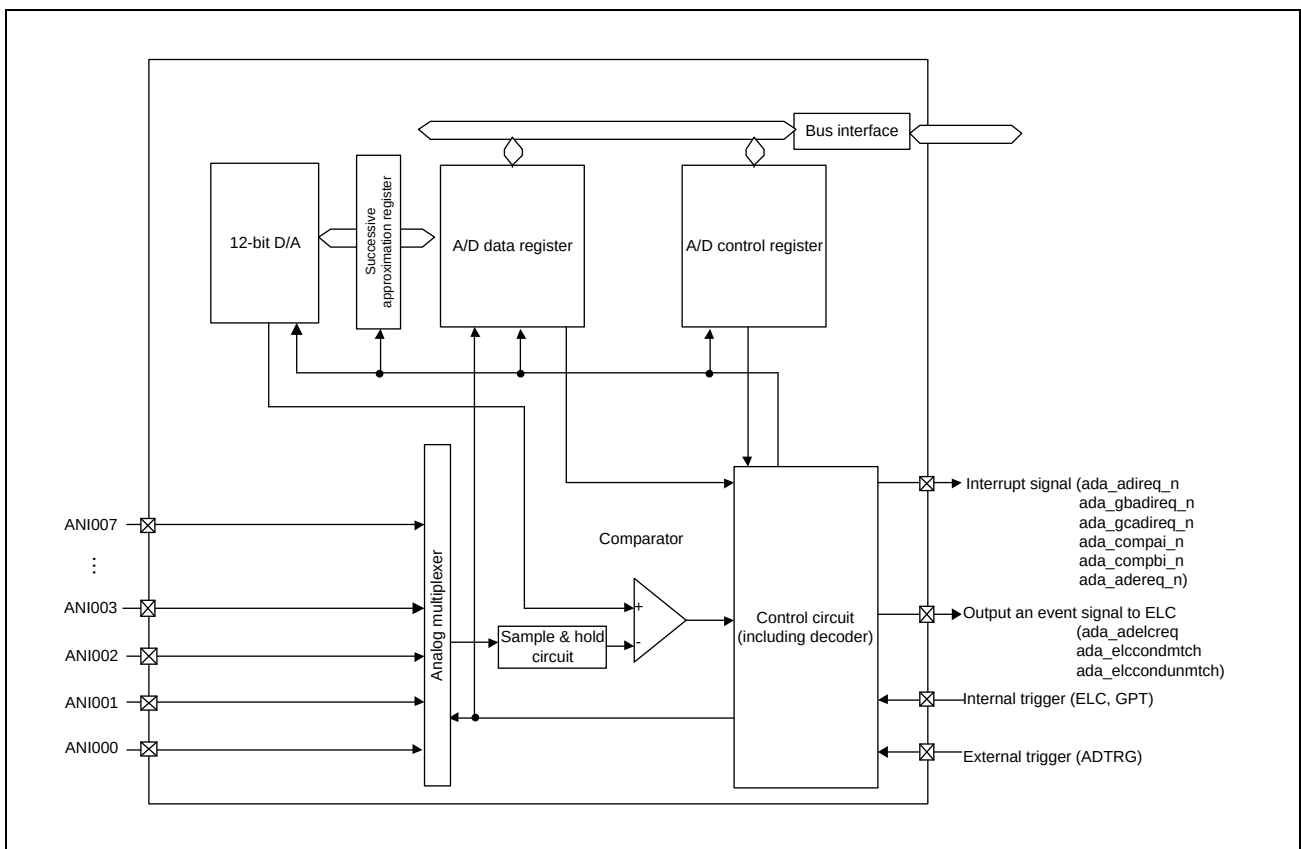


Figure 7.10-1 Block Diagram of 12-Bit A/D Converter (Unit0 to Unit2)

7.10.2 ADC Registers

The base address for the ADC is as follows.

Table 7.10-4 Register Base Address

Base Register Name	ADC ch	Base Address
<ADC0_base>	ADC0	0_11C0_0000h (51C0_0000h* ¹ , 41C0_0000h* ²)
<ADC1_base>	ADC1	0_11C0_2800h (51C0_2800h* ¹ , 41C0_2800h* ²)
<ADC2_base>	ADC2	0_11C0_2C00h (51C0_2C00h* ¹ , 41C0_2C00h* ²)

Note 1. CM33 Address Space Non-Secure

Note 2. CM33 Address Space Secure

7.10.2.1 List of Registers

The registers are listed below. (m = 0 to 2)

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
A/D Control Register	ADCM_ADCSR	0000h	0000h	16
A/D Conversion Status Register	ADCM_ADREF	00h	0002h	8
Reserve	-	-	0003h	-
A/D Channel Select Register A0	ADCM_ADANSA0	0000h	0004h	16
Reserve	-	-	0006h to 0007h	-
A/D-Converted Value Addition/Average Function Channel Select Register 0	ADCM_ADADS0	0000h	0008h	16
Reserve	-	-	000Ah to 000Bh	-
A/D-Converted Value Addition/Average Count Select Register	ADCM_ADADC	00h	000Ch	8
Reserve	-	-	000Dh	-
A/D Control Extended Register	ADCM_ADCER	0000h	000Eh	16
A/D Conversion Start Trigger Select Register	ADCM_ADSTRGR	0000h	0010h	16
Reserve	-	-	0012h to 0013h	-
A/D Channel Select Register B0	ADCM_ADANSB0	0000h	0014h	16
Reserve	-	-	0016h to 0017h	-
A/D Data Duplication Register	ADCM_ADDBLDR	0000h	0018h	16
A/D Data Register 0	ADCM_ADDR0	0000h	0020h	16
A/D Data Register 1	ADCM_ADDR1	0000h	0022h	16
A/D Data Register 2	ADCM_ADDR2	0000h	0024h	16
A/D Data Register 3	ADCM_ADDR3	0000h	0026h	16
A/D Data Register 4	ADCM_ADDR4	0000h	0028h	16
A/D Data Register 5	ADCM_ADDR5	0000h	002Ah	16
A/D Data Register 6	ADCM_ADDR6	0000h	002Ch	16
A/D Data Register 7	ADCM_ADDR7	0000h	002Eh	16
Reserve	-	-	0030h to 007Ch	-
A/D Event Link Control Register	ADCM_ADELCCR	00h	007Dh	8
Reserve	-	-	007Eh to 007Fh	-
A/D Group Scan Priority Control Register	ADCM_ADGSPCR	0000h	0080h	16

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Reserve	-	-	0082h to 0083h	-
A/D Data Duplication Register A	ADCm_ADDBLDRA	0000h	0084h	16
A/D Data Duplication Register B	ADCm_ADDBLDRB	0000h	0086h	16
Reserve	-	-	0088h to 008Bh	-
A/D Compare Function Window A/B Status Monitoring Register	ADCm_ADWINMON	00h	008Ch	8
Reserve	-	-	008Dh to 008Fh	-
A/D Compare Function Control Register	ADCm_ADCMPCR	0000h	0090h	16
Reserve	-	-	0092h to 0093h	-
A/D Compare Function Window A Channel Select Register 0	ADCm_ADCMPANSR0	0000h	0094h	16
Reserve	-	-	0096h to 0097h	-
A/D Compare Function Window A Comparison Condition Setting Register 0	ADCm_ADCMPLR0	0000h	0098h	16
Reserve	-	-	009Ah to 009Bh	-
A/D Comparison Function Window A Lower Level Setting Register	ADCm_ADCMPDR0	0000h	009Ch	16
A/D Comparison Function Window A Upper Level Setting Register	ADCm_ADCMPDR1	0000h	009Eh	16
A/D Comparison Function Window A Channel Status Register 0	ADCm_ADCMPSR0	0000h	00A0h	16
Reserve	-	-	00A2h to 00A5h	-
A/D Compare Function Window B Channel Select Register	ADCm_ADCMPBNSR	00h	00A6h	8
Reserve	-	-	00A7h	-
A/D Compare Function Window B Lower-Side Level Setting Register	ADCm_ADWINLLB	0000h	00A8h	16
A/D Compare Function Window B Upper-Side Level Setting Register	ADCm_ADWINULB	0000h	00AAh	16
A/D Compare Function Window B Status Register	ADCm_ADCMPBSR	00h	00ACh	8
Reserve	-	-	00ADh to 00D3h	-
A/D Channel Select Register C0	ADCm_ADANSC0	0000h	00D4h	16
Reserve	-	-	00D6h to 00D8h	-
A/D Group C Trigger Select Register	ADCm_ADGCTRGR	00h	00D9h	8
Reserve	-	-	00DAh to 01E1h	-
A/D Error Control Register	ADCm_ADERCR	00h	1E2h	8
A/D Error Clear Register	ADCm_ADERCLR	00h	1E3h	8
Reserve	-	-	01E4h to 01E7h	-
A/D Overwrite Error Register	ADCm_ADOWER	0000h	1E8h	16
A/D Overwrite Error Extended Register	ADCm_ADOWEER	0000h	1EAh	16

7.10.2.2 ADC Register Descriptions

The prefix (ADCM_) of the register names is omitted in this and subsequent sections.

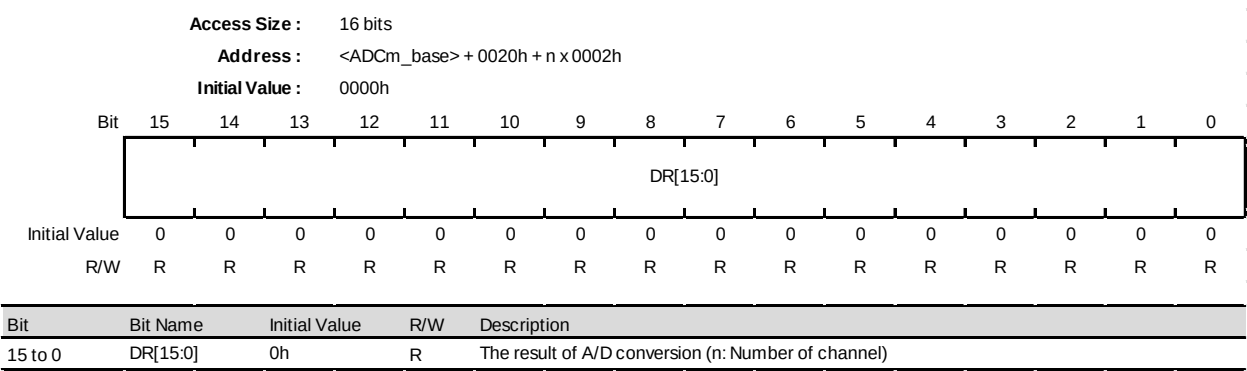
7.10.2.2.1 A/D Data Registers

The A/D-converted value has a 12-bit/8-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the A/D conversion accuracy specify bits (ADCER.ADPRC[1:0]) (12-bit or 8-bit accuracy is selectable.)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set from one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

(1) A/D Data Register n (ADCM_ADDRn) (n = 0 to 7)



(2) A/D Data Duplication Register (ADCM_ADDBLDR)

Access Size :		16 bits														
Address :		<ADCM_base> + 0018h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBLDR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W		Description									
15 to 0	DBLDR[15:0]		0h		R		The result of A/D conversion in response to the second trigger in double trigger mode.									

(3) A/D Data Duplication Register A (ADCm_ADDBLDRA)

Access Size :		16 bits														
Address :		<ADCm_base> + 0084h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBLDRA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DBLDRA[15:0]	0h	R	The result of A/D conversion during extended operation in double trigger mode

(4) A/D Data Duplication Register B (ADCM_ADDBLDRB)

Access Size :		16 bits														
Address :		<ADCM_base> + 0086h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBLDRB[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DBLDRB[15:0]	0h	R	The result of A/D conversion during extended operation in double trigger mode

7.10.2.2.2 A/D Control Register (ADCm_ADCSR)

Access Size : 16 bits

Address : <ADCm_base> + 0000h

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADST	ADCS[1:0]	ADIE	-	-	TRGE	EXTRG	DBLE	GBADIE	-	DBLANS[4:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	R	R	RW	RW	RW	RW	R	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15	ADST	0h	RW	A/D conversion Start 0b: Stops A/D conversion process 1b: Starts A/D conversion process
14, 13	ADCS[1:0]	0h	RW	Scan Mode Select 00b: Single scan mode 01b: Group scan mode 10b: Continuous scan mode 11b: Setting prohibited
12	ADIE	0h	RW	Scan End Interrupt Enable 0b: Disables ada_adireq_n interrupt generation upon scan completion 1b: Enables ada_adireq_n interrupt generation upon scan completion
11, 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	TRGE	0h	RW	Trigger Start Enable 0b: Disables A/D conversion to be started by the internal or external trigger 1b: Enables A/D conversion to be started by the internal or external trigger
8	EXTRG	0h	RW	Trigger Select 0b: A/D conversion is started by the internal trigger (GPT, ELC) 1b: A/D conversion is started by the external trigger (ADTRG)
7	DBLE	0h	RW	Double Trigger Mode Select 0b: Deselects double trigger mode 1b: Selects double trigger mode
6	GBADIE	0h	RW	Group B Scan End Interrupt Enable 0b: Disables ada_gbadireq_n interrupt generation upon group B scan completion 1b: Enables ada_gbadireq_n interrupt generation upon group B scan completion
5	-	0h	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
4 to 0	DBLANS[4:0]	0h	RW	Double Trigger Channel Select These bits select one analog input channel for double triggered operation. The setting is only valid while double trigger mode is selected.

7.10.2.2.3 A/D Conversion Status Register (ADCM_ADREF)

The ADREF register indicates the status of the A/D conversion.

Access Size :		8 bits						
Address :		<ADCM_base> + 0002h						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	ADSCACT	-	-	-	-	-	-	ADF
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ADSCACT	0h	R	0b: A/D conversion is not in progress. 1b: A/D conversion is in progress.
6 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read.
0	ADF	0h	R	Reserved The read value is undefined.

7.10.2.2.4 A/D Channel Select Register (ADCm_ADANSA0)

Access Size :		16 bits														
Address :		<ADCm_base> + 0004h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ANSA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

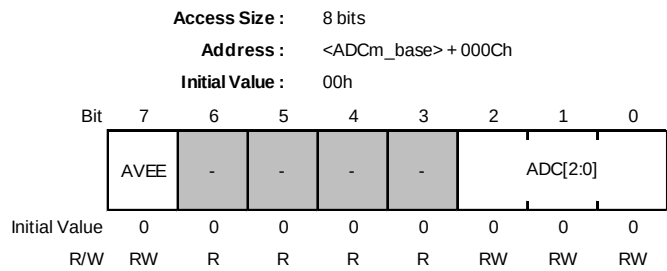
Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	ANSA0[7:0]	0h	RW	A/D conversion Analog input Channel Select ANSA0[0] bit corresponds to ch0 and ANSA0[7] corresponds to ch7. 0b: ch0-ch7 are not subjected to conversion. 1b: ch0-ch7 are subjected to conversion.

7.10.2.2.5 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADCM_ADADS0)

Access Size :		16 bits														
Address :		<ADCM_base> + 0008h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ADS0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	ADS0[7:0]	0h	RW	A/D-Converted Value Addition/Average Channel Select ADS0[0] bit corresponds to ch0 and ADS0[7] corresponds to ch7. 0b: A/D-converted value addition/average mode for ch0 to ch7 is not selected 1b: A/D-converted value addition/average mode for ch0 to ch7 is selected

7.10.2.2.6 A/D-Converted Value Addition/Average Count Select Register (ADCm_ADADC)



Bit	Bit Name	Initial Value	R/W	Description
7	AVEE	0h	RW	Average Mode Enable 0b: Addition mode is selected 1b: Average mode is selected
6 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2 to 0	ADC[2:0]	0h	RW	Addition Count Select 000b: 1-time conversion (no addition; same as normal conversion) 001b: 2-time conversion (addition once) 010b: 3-time conversion (addition twice)* ¹ 011b: 4-time conversion (addition three times) 101b: 16-time conversion (addition 15 times)* ² Others: Setting prohibited

- Note 1. When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the Addition count to 3 times (ADADC.ADC[2:0] = 010b).
- Note 2. For the 12-bit conversion accuracy, the Addition count can only be set to select 16 times (ADADC.ADC[2:0] = 101b).

7.10.2.2.7 A/D Control Extended Register (ADCM_ADCER)

Access Size : 16 bits

Address : <ADCM_base> + 000Eh

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADRFMT	-	-	-	-	-	-	-	-	-	ACE	-	-	ADPRC[1:0]	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	RW	R	R	RW	RW	R

Bit	Bit Name	Initial Value	R/W	Description
15	ADRFMT	0h	RW	A/D Data Register Format Select 0b: Right-alignment is selected for the A/D data register format 1b: Left-alignment is selected for the A/D data register format
14 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5	ACE	0h	RW	A/D Data Register Automatic Clearing Enable 0b: Disables automatic clearing 1b: Enables automatic clearing
4, 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2, 1	ADPRC[1:0]	0h	RW	A/D Conversion Accuracy Specify 00b: A/D conversion is performed with 12-bit accuracy 10b: A/D conversion is performed with 8-bit accuracy Others: Setting prohibited
0	-	0h	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

7.10.2.2.8 A/D Conversion Start Trigger Select Register (ADCM_ADSTRGR)

Access Size : 16 bits

Address : <ADCM_base> + 0010h

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TRSA[5:0]						-	-	TRSB[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
13 to 8	TRSA[5:0]	0h	RW	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected. See Table 7.10-6 for details.
7, 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5 to 0	TRSB[5:0]	0h	RW	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode. See Table 7.10-5 for details.

Table 7.10-5 Selection of A/D Activation Sources by the TRSB[5:0] Bits (for Group B Only) (1/3)

Module	Source	Remarks	TRSB					
			[5]	[4]	[3]	[2]	[1]	[0]
Trigger source de-selection state			1	1	1	1	1	1
GPT	GTADTRA0N	Compare match with GPT00.GTADTRA	0	0	0	0	0	1
	GTADTRB0N	Compare match with GPT00.GTADTRB	0	0	0	0	1	0
	GTADTRA1N	Compare match with GPT01.GTADTRA	0	0	0	0	1	1
	GTADTRB1N	Compare match with GPT01.GTADTRB	0	0	0	1	0	0
	GTADTRA2N	Compare match with GPT02.GTADTRA	0	0	0	1	0	1
	GTADTRB2N	Compare match with GPT02.GTADTRB	0	0	0	1	1	0
	GTADTRA3N	Compare match with GPT03.GTADTRA	0	0	0	1	1	1
	GTADTRB3N	Compare match with GPT03.GTADTRB	0	0	1	0	0	0
	GTADTRA4N	Compare match with GPT04.GTADTRA	0	0	1	0	0	1
	GTADTRB4N	Compare match with GPT04.GTADTRB	0	0	1	0	1	0
	GTADTRA5N	Compare match with GPT05.GTADTRA	0	0	1	0	1	1
	GTADTRB5N	Compare match with GPT05.GTADTRB	0	0	1	1	0	0
	GTADTRA6N	Compare match with GPT06.GTADTRA	0	0	1	1	0	1
	GTADTRB6N	Compare match with GPT06.GTADTRB	0	0	1	1	1	0
	GTADTRA7N	Compare match with GPT07.GTADTRA	0	0	1	1	1	1
	GTADTRB7N	Compare match with GPT07.GTADTRB	0	1	0	0	0	0
	GTADTRA8N	Compare match with GPT10.GTADTRA	0	1	0	0	0	1
	GTADTRB8N	Compare match with GPT10.GTADTRB	0	1	0	0	1	0
	GTADTRA9N	Compare match with GPT11.GTADTRA	0	1	0	0	1	1
	GTADTRB9N	Compare match with GPT11.GTADTRB	0	1	0	1	0	0
	GTADTRA10N	Compare match with GPT12.GTADTRA	0	1	0	1	0	1
	GTADTRB10N	Compare match with GPT12.GTADTRB	0	1	0	1	1	0
	GTADTRA11N	Compare match with GPT13.GTADTRA	0	1	0	1	1	1

Table 7.10-5 Selection of A/D Activation Sources by the TRSB[5:0] Bits (for Group B Only) (2/3)

Module	Source	Remarks	TRSB					
			[5]	[4]	[3]	[2]	[1]	[0]
	GTADTRB11N	Compare match with GPT13.GTADTRB	0	1	1	0	0	0
	GTADTRA12N	Compare match with GPT14.GTADTRA	0	1	1	0	0	1
	GTADTRB12N	Compare match with GPT14.GTADTRB	0	1	1	0	1	0
	GTADTRA13N	Compare match with GPT15.GTADTRA	0	1	1	0	1	1
	GTADTRB13N	Compare match with GPT15.GTADTRB	0	1	1	1	0	0
	GTADTRA14N	Compare match with GPT16.GTADTRA	0	1	1	1	0	1
	GTADTRB14N	Compare match with GPT16.GTADTRB	0	1	1	1	1	0
	GTADTRA15N	Compare match with GPT17.GTADTRA	0	1	1	1	1	1
	GTADTRB15N	Compare match with GPT17.GTADTRB	1	0	0	0	0	0
	GTADTRA0N or GTADTRB0N	Compare match with GPT00.GTADTRA or Compare match with GPT00.GTADTRB	1	0	0	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPT01.GTADTRA or Compare match with GPT01.GTADTRB	1	0	0	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPT02.GTADTRA or Compare match with GPT02.GTADTRB	1	0	0	0	1	1
	GTADTRA3N or GTADTRB3N	Compare match with GPT03.GTADTRA or Compare match with GPT03.GTADTRB	1	0	0	1	0	0
	GTADTRA4N or GTADTRB4N	Compare match with GPT04.GTADTRA or Compare match with GPT04.GTADTRB	1	0	0	1	0	1
	GTADTRA5N or GTADTRB5N	Compare match with GPT05.GTADTRA or Compare match with GPT05.GTADTRB	1	0	0	1	1	0
	GTADTRA6N or GTADTRB6N	Compare match with GPT06.GTADTRA or Compare match with GPT06.GTADTRB	1	0	0	1	1	1
	GTADTRA7N or GTADTRB7N	Compare match with GPT07.GTADTRA or Compare match with GPT07.GTADTRB	1	0	1	0	0	0
	GTADTRA8N or GTADTRB8N	Compare match with GPT10.GTADTRA or Compare match with GPT10.GTADTRB	1	0	1	0	0	1
	GTADTRA9N or GTADTRB9N	Compare match with GPT11.GTADTRA or Compare match with GPT11.GTADTRB	1	0	1	0	1	0
	GTADTRA10N or GTADTRB10N	Compare match with GPT12.GTADTRA or Compare match with GPT12.GTADTRB	1	0	1	0	1	1
	GTADTRA11N or GTADTRB11N	Compare match with GPT13.GTADTRA or Compare match with GPT13.GTADTRB	1	0	1	1	0	0
	GTADTRA12N or GTADTRB12N	Compare match with GPT14.GTADTRA or Compare match with GPT14.GTADTRB	1	0	1	1	0	1
	GTADTRA13N or GTADTRB13N	Compare match with GPT15.GTADTRA or Compare match with GPT15.GTADTRB	1	0	1	1	1	0
	GTADTRA14N or GTADTRB14N	Compare match with GPT16.GTADTRA or Compare match with GPT16.GTADTRB	1	0	1	1	1	1

Table 7.10-5 Selection of A/D Activation Sources by the TRSB[5:0] Bits (for Group B Only) (3/3)

Module	Source	Remarks	TRSB					
			[5]	[4]	[3]	[2]	[1]	[0]
	GTADTRA15N or GTADTRB15N	Compare match with GPT17.GTADTRA or Compare match with GPT17.GTADTRB	1	1	0	0	0	0
ELC	ELCTRG0		1	1	0	0	0	1

Table 7.10-6 Selection of A/D Activation Sources by the TRSA[5:0] Bits (1/2)

Module	Source	Remarks	TRSA					
			[5]	[4]	[3]	[2]	[1]	[0]
Trigger source de-selection state			1	1	1	1	1	1
External	ADTRG	Input pin for the trigger	0	0	0	0	0	0
GPT	GTADTRA0N	Compare match with GPT00.GTADTRA	0	0	0	0	0	1
	GTADTRB0N	Compare match with GPT00.GTADTRB	0	0	0	0	1	0
	GTADTRA1N	Compare match with GPT01.GTADTRA	0	0	0	0	1	1
	GTADTRB1N	Compare match with GPT01.GTADTRB	0	0	0	1	0	0
	GTADTRA2N	Compare match with GPT02.GTADTRA	0	0	0	1	0	1
	GTADTRB2N	Compare match with GPT02.GTADTRB	0	0	0	1	1	0
	GTADTRA3N	Compare match with GPT03.GTADTRA	0	0	0	1	1	1
	GTADTRB3N	Compare match with GPT03.GTADTRB	0	0	1	0	0	0
	GTADTRA4N	Compare match with GPT04.GTADTRA	0	0	1	0	0	1
	GTADTRB4N	Compare match with GPT04.GTADTRB	0	0	1	0	1	0
	GTADTRA5N	Compare match with GPT05.GTADTRA	0	0	1	0	1	1
	GTADTRB5N	Compare match with GPT05.GTADTRB	0	0	1	1	0	0
	GTADTRA6N	Compare match with GPT06.GTADTRA	0	0	1	1	0	1
	GTADTRB6N	Compare match with GPT06.GTADTRB	0	0	1	1	1	0
	GTADTRA7N	Compare match with GPT07.GTADTRA	0	0	1	1	1	1
	GTADTRB7N	Compare match with GPT07.GTADTRB	0	1	0	0	0	0
	GTADTRA8N	Compare match with GPT10.GTADTRA	0	1	0	0	0	1
	GTADTRB8N	Compare match with GPT10.GTADTRB	0	1	0	0	1	0
	GTADTRA9N	Compare match with GPT11.GTADTRA	0	1	0	0	1	1
	GTADTRB9N	Compare match with GPT11.GTADTRB	0	1	0	1	0	0
	GTADTRA10N	Compare match with GPT12.GTADTRA	0	1	0	1	0	1
	GTADTRB10N	Compare match with GPT12.GTADTRB	0	1	0	1	1	0
	GTADTRA11N	Compare match with GPT13.GTADTRA	0	1	0	1	1	1
	GTADTRB11N	Compare match with GPT13.GTADTRB	0	1	1	0	0	0
	GTADTRA12N	Compare match with GPT14.GTADTRA	0	1	1	0	0	1
	GTADTRB12N	Compare match with GPT14.GTADTRB	0	1	1	0	1	0
	GTADTRA13N	Compare match with GPT15.GTADTRA	0	1	1	0	1	1
	GTADTRB13N	Compare match with GPT15.GTADTRB	0	1	1	1	0	0
	GTADTRA14N	Compare match with GPT16.GTADTRA	0	1	1	1	0	1
	GTADTRB14N	Compare match with GPT16.GTADTRB	0	1	1	1	1	0
	GTADTRA15N	Compare match with GPT17.GTADTRA	0	1	1	1	1	1
	GTADTRB15N	Compare match with GPT17.GTADTRB	1	0	0	0	0	0

Table 7.10-6 Selection of A/D Activation Sources by the TRSA[5:0] Bits (2/2)

Module	Source	Remarks	TRSA					
			[5]	[4]	[3]	[2]	[1]	[0]
	GTADTRA0N or GTADTRB0N	Compare match with GPT00.GTADTRA or Compare match with GPT00.GTADTRB	1	0	0	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPT01.GTADTRA or Compare match with GPT01.GTADTRB	1	0	0	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPT02.GTADTRA or Compare match with GPT02.GTADTRB	1	0	0	0	1	1
	GTADTRA3N or GTADTRB3N	Compare match with GPT03.GTADTRA or Compare match with GPT03.GTADTRB	1	0	0	1	0	0
	GTADTRA4N or GTADTRB4N	Compare match with GPT04.GTADTRA or Compare match with GPT04.GTADTRB	1	0	0	1	0	1
	GTADTRA5N or GTADTRB5N	Compare match with GPT05.GTADTRA or Compare match with GPT05.GTADTRB	1	0	0	1	1	0
	GTADTRA6N or GTADTRB6N	Compare match with GPT06.GTADTRA or Compare match with GPT06.GTADTRB	1	0	0	1	1	1
	GTADTRA7N or GTADTRB7N	Compare match with GPT07.GTADTRA or Compare match with GPT07.GTADTRB	1	0	1	0	0	0
	GTADTRA8N or GTADTRB8N	Compare match with GPT10.GTADTRA or Compare match with GPT10.GTADTRB	1	0	1	0	0	1
	GTADTRA9N or GTADTRB9N	Compare match with GPT11.GTADTRA or Compare match with GPT11.GTADTRB	1	0	1	0	1	0
	GTADTRA10N or GTADTRB10N	Compare match with GPT12.GTADTRA or Compare match with GPT12.GTADTRB	1	0	1	0	1	1
	GTADTRA11N or GTADTRB11N	Compare match with GPT13.GTADTRA or Compare match with GPT13.GTADTRB	1	0	1	1	0	0
	GTADTRA12N or GTADTRB12N	Compare match with GPT14.GTADTRA or Compare match with GPT14.GTADTRB	1	0	1	1	0	1
	GTADTRA13N or GTADTRB13N	Compare match with GPT15.GTADTRA or Compare match with GPT15.GTADTRB	1	0	1	1	1	0
	GTADTRA14N or GTADTRB14N	Compare match with GPT16.GTADTRA or Compare match with GPT16.GTADTRB	1	0	1	1	1	1
	GTADTRA15N or GTADTRB15N	Compare match with GPT17.GTADTRA or Compare match with GPT17.GTADTRB	1	1	0	0	0	0
ELC	ELCTRG0		1	1	0	0	0	1

7.10.2.2.9 A/D Channel Select Register B0 (ADCm_ADANSB0)

Access Size :		16 bits														
Address :		<ADCm_base> + 0014h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ANSB0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	ANSB0[7:0]	0h	RW	A/D Conversion Analog Input Channel Select ANSB0[0] bit corresponds to ch0 and ANSB0[7] corresponds to ch7. 0b: ch0-ch7 are not subjected to conversion 1b: ch0-ch7 are subjected to conversion

7.10.2.2.10 A/D Error Control Register (ADCm_ADERCR)

ADERCR controls the error detection function. For details about errors, see **7.10.3.10 Error Detection Function (Overwrite Check Function)**.

Access Size :		8 bits						
Address :		<ADCm_base> + 01E2h						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OWEIE	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	OWEIE	0h	RW	Overwrite Error Interrupt Enable 0b: Disables interrupt generation when an overwrite error is detected. 1b: Enables interrupt generation when an overwrite error is detected.
1, 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

7.10.2.2.11 A/D Error Clear Register (ADCM_ADERCLR)

The A/D error clear register is a write-only register for clearing errors.

Access Size :		8 bits						
Address :		<ADCM_base> + 01E3h						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OWEC	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	OWEC	0h	RW	Overwrite Error Clear Writing 0b: Disables clearing overwrite errors. Writing 1b: Enables clearing overwrite errors.
1, 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.

7.10.2.2.12 A/D Overwrite Error Register (ADCM_ADOWER)

The ADOWER register is a status register that indicates that the results of A/D conversion in the ADDRn register were not read and an overwrite occurred.

Access Size :		16 bits														
Address :		<ADCM_base> + 01E8h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	OWE[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read.
7 to 0	OWE[7:0]	0h	R	Overwrite Error OWE[0] bit corresponds to ch0 and OWE[7] corresponds to ch7. 0b: No overwrite error occurred in ADDR0 to ADDR7. 1b: An overwrite error occurred in ADDR0 to ADDR7.

7.10.2.2.13 A/D Overwrite Error Extended Register (ADCm_ADOWEER)

The ADOWEER register is a status register that indicates that the results of A/D conversion in the ADDBLDRB, ADDBLDRA, ADDBLDR registers were not read and an overwrite occurred.

Access Size :		16 bits														
Address :		<ADCm_base> + 01EAh														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DOWE	DAOWE	DBOWE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read.
2	DOWE	0h	R	A/D Data Duplication Register overwrite Error 0b: No overwrite error occurred in ADDBLDR. 1b: An overwrite error occurred in ADDBLDR.
1	DAOWE	0h	R	A/D Data Duplication Register A overwrite Error 0b: No overwrite error occurred in ADDBLDRA. 1b: An overwrite error occurred in ADDBLDRA.
0	DBOWE	0h	R	A/D Data Duplication Register B overwrite Error 0b: No overwrite error occurred in ADDBLDRB. 1b: An overwrite error occurred in ADDBLDRB.

7.10.2.2.14 A/D Event Link Control Register (ADCm_ADELCCR)

Access Size :		8 bits						
Address :		<ADCm_base> + 007Dh						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	GCELC	ELCC[1:0]	
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
2	GCELC	0h	RW	Event control bit for Group C 0b: Not output event signal at the end of group scan of group C. 1b: Output event signal at the end of group scan of group C.
1, 0	ELCC[1:0]	0h	RW	Event link control bits Control the end of scan event (ada_adelcreq). To enable the ELCC[1:0] setting, set GCELC to 0. 00b: Output event signal at the end of scan except the end of group scan of group B, C 01b: Output event signal at the end of group scan of group B 1xb: Output event signal at the end of scan

7.10.2.2.15 A/D Group Scan Priority Control Register (ADCM_ADGSPCR)

Access Size : 16 bits

Address : <ADCM_base> + 0080h

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GBRP	LGRRS	-	-	-	-	-	-	-	-	-	-	-	-	GBRS CN	PGS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15	GBRP	0h	RW	Low priority Group Single Scan Continuous Start*2 (Enabled only when PGS = 1b. Set 0 when PGS = 0b.) 0b: Single scan for Low priority group is not continuously activated 1b: Single scan for Low priority group is continuously activated
14	LGRRS	0h	RW	Restart Channel Select (Enabled only when PGS = 1b and GBRSCN = 1b, Set 0b when PGS = 0b.) 0b: Restart scan with first channel 1b: Restart scan with suspended channel
13 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1	GBRSCN	0h	RW	Low priority Group Restart Setting (Enabled only when PGS = 1b. Set 0 when PGS = 0b.) 0b: Scanning for group is not restarted when priority control is performed 1b: Scanning for group is restarted when priority control is performed
0	PGS	0h	RW	Group Priority Control Setting*1 0b: Operation is without group priority control 1b: Operation is with group priority control

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
If the bits are set to any other values, proper operation cannot be guaranteed.

Note 2. When the GBRP bit has been set to 1, single scan is performed continuously for low priority group regardless of the setting of the GBRSCN bit.

7.10.2.2.16 A/D Compare Function Window A/B Status Monitoring Register
(ADCm_ADWINMON)

Access Size :		8 bits						
Offset Address :		<ADCm_base> + 008Ch						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	MONC MPB	MONC MPA	-	-	-	MONC OMB
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	-	All 0	R	Reserved Whenever it is read, 0b is read.
5	MONCMPB	0h	R	Comparing result monitor for window B 0b: Window B comparing condition does not match. 1b: Window B comparing condition match.
4	MONCMPA	0h	R	Comparing result monitor for window A 0b: Window A comparing condition does not match. 1b: Window A comparing condition match.
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read.
0	MONCOMB	0h	R	Combination result monitor 0b: Window A/B combination condition does not match. 1b: Window A/B combination condition match.

7.10.2.2.17 A/D Compare Function Control Register (ADCM_ADCMPCR)

Access Size : 16 bits

Offset Address : <ADCM_base> + 0090h

Initial Value : 0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMPAIE	WCMP E	CMPBIE	-	CMPA E	-	CMPB E	-	-	-	-	-	-	-	CMPAB[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	R	RW	R	RW	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15	CMPAIE	0h	RW	Compare window A Interrupt Enable 0b: Generation of an ada_compai_n interrupt in response to matches with a condition for comparison is disabled. 1b: Generation of an ada_compai_n interrupt in response to matches with a condition for comparison is enabled.
14	WCMP E	0h	RW	Window Function enable 0b: Window function disabled 1b: Window function enabled
13	CMPBIE	0h	RW	Compare window B Interrupt Enable 0b: Generation of an ada_compb_i_n interrupt in response to matches with a condition for comparison is disabled. 1b: Generation of an ada_compb_i_n interrupt in response to matches with a condition for comparison is enabled.
12	-	0h	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
11	CMPA E	0h	RW	Window A operation permission 0b: Window A is not in operation. Output of ada_elcondmtch/ada_elcondummtch is not permitted. 1b: Window A is in operation.
10	-	0h	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
9	CMPB E	0h	RW	Window B operation permission 0b: Window B is not in operation. Output of ada_elcondmtch/ada_elcondummtch is not permitted. 1b: Window B is in operation.
8 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
1, 0	CMPAB[1:0]	0h	RW	Window A/B combination condition setting This bit is valid when both windows A/B are valid (CMPA E = 1b and CMPB E = 1b). 00b: Output ada_elcondmtch when window A comparison condition match OR window B comparison condition match, otherwise output ada_elcondummtch 01b: Output ada_elcondmtch when window A comparison condition match EXOR window B comparison condition match, otherwise output ada_elcondummtch 10b: Output ada_elcondmtch when window A comparison condition match AND window B comparison condition match, otherwise output ada_elcondummtch 11b: Setting prohibited.

7.10.2.2.18 A/D Compare Function Window A Channel Select Register 0
(ADCm_ADCMPANSR0)

Access Size :		16 bits														
Offset Address :		<ADCm_base> + 0094h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMPCHA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	CMPCHA0[7:0]	0h	RW	Window A Channel Select Set the CMPCHA0[7:0] bits when ADCSR.ADST = 0b. CMPCHA0[0] bit corresponds to ch0 and CMPCHA0[7] corresponds to ch7. 0b: Channel is not target 1b: Channel is target

7.10.2.2.19 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCm_ADCMPLR0)

Access Size :		16 bits														
Offset Address :		<ADCm_base> + 0098h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMPLCHA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	CMPLCHA0 [7:0]	0h	RW	Window A comparison condition for target channel (ch0-7) setting CMPLCHA0[7] corresponds to channel 7 and CMPLCHA0[0] corresponds to channel 0. When the comparison result of each channel input matches the set condition, the ADCMPSR0.CMPSTCHA0n bit is set to 1b and a compare interrupt (ada_compai_n) is generated. - When window function is disabled (ADCMPCR.WCMPE = 0:- 0b: ADCMPDR0 register value > A/D-converted value 1b: ADCMPDR0 register value < A/D-converted value - When window function is enabled (ADCMPCR.WCMPE = 1)- 0b: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1b: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value

7.10.2.2.20 A/D Comparison Function Window A Lower Level Setting Register (ADCm_ADCMPDR0)

Access Size :		16 bits														
Offset Address :		<ADCm_base> + 009Ch														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPLLA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CMPLLA[15:0]	0h	RW	Reference Lower data setting when using the compare function window A

7.10.2.2.21 A/D Comparison Function Window A Upper Level Setting Register (ADCm_ADCMPDR1)

Access Size :		16 bits														
Offset Address :		<ADCm_base> + 009Eh														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPULA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

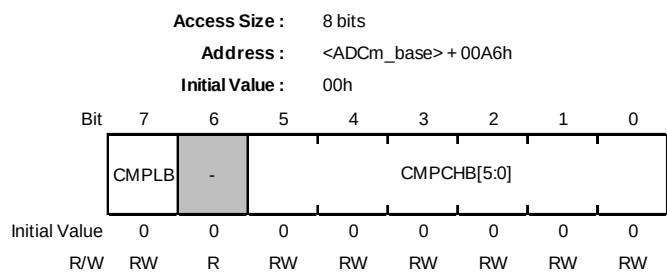
Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CMPULA[15:0]	0h	RW	Reference Upper data setting when using the compare function window A

7.10.2.2.22 A/D Comparison Function Window A Channel Status Register 0 (ADCM_ADCMPSR0)

Access Size :		16 bits														
Address :		<ADCM_base> + 00A0h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMPSTCHA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	CMPSTCHA0 [7:0]	0h	RW	Window A Status Flag CMPSTCHA0[0] bit corresponds to ch0 and CMPSTCHA0[7] corresponds to ch7. This flag shows the comparison result of each targeted channel, when window A is operating (ADCMPCR.CMPAE = 1b). If this flag is set while ADCMPCR.CMPAIE = 1b, a compare interrupt (ada_compai_n) request is generated. 0b: Condition does not match 1b: Condition matches

7.10.2.2.23 A/D Compare Function Window B Channel Select Register
(ADCm_ADCMPBNSR)



Bit	Bit Name	Initial Value	R/W	Description
7	CMPLB	0h	RW	Window B Comparison Condition Setting When the comparison result of target input matches the set condition, the ADCMPBSR0.CMPSTB bit is set to 1b and a compare interrupt (ada_compbi_n) is generated. - When window function is disabled (ADCMPCR.WCMPE = 0b)- 0b: ADWINLLB register value > A/D-converted value. 1b: ADWINLLB register value < A/D-converted value. - When window function is enabled (ADCMPCR.WCMPE = 1b)- 0b: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value. 1b: ADWINLLB register value < A/D-converted value < ADWINULB register value.
6	-	0h	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
5 to 0	CMPCHB[5:0]	0h	RW	Window B Channel Select Select channels for comparing with window B condition. Set the CMPCHB[5:0] bits when the ADCSR.ADST bit is 0b. 00h: ch0 01h: ch1 02h: ch2 : 07h: ch7 3Fh: No channel is selected Others: Setting prohibited

7.10.2.2.24 A/D Compare Function Window B Lower-Side Level Setting Register
(ADCm_ADWINLLB)

Access Size :		16 bits														
Address :		<ADCm_base> + 00A8h														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPLLB[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CMPLLB[15:0]	0h	RW	Reference lower data setting when using the compare function window B

7.10.2.2.25 A/D Compare Function Window B Upper-Side Level Setting Register (ADCm_ADWINULB)

Access Size :		16 bits														
Address :		<ADCm_base> + 00AAh														
Initial Value :		0000h														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPULB[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	CMPULB[15:0]	0h	RW	Reference upper data setting when using the compare function window B

7.10.2.2.26 A/D Compare Function Window B Status Register (ADCm_ADCMPBSR)

Access Size :		8 bits						
Address :		<ADCm_base> + 00ACh						
Initial Value :		00h						
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CMPS TB
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW0

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
0	CMPSTB	0h	RW0	Window B Flag When window B operation is enabled (ADCMPCR.CMPBE = 1b), this bit indicates the comparison result of channels to which window B comparison conditions are applied. 0b: Comparison conditions are not met 1b: Comparison conditions are met

7.10.2.2.27 A/D Channel Select Register C0 (ADCm_ADANSC0)

Access Size :

16 bits

Address :

<ADCm_base> + 00D4h

Initial Value :

0000h

Bit

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

-

-

-

-

-

-

-

ANSC0[7:0]

Initial Value

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

R/W

R

R

R

R

R

R

R

RW

RW

RW

RW

RW

RW

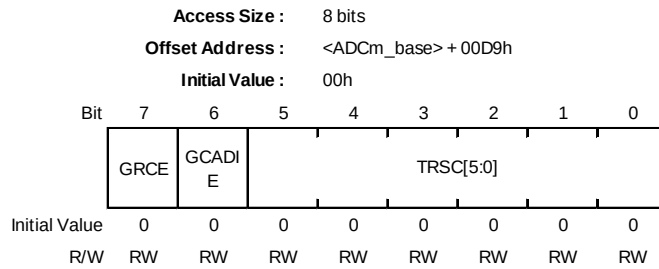
RW

RW

RW

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 0	ANSC0[7:0]	0h	RW	A/D-Converted Channel Select for Group C in Group Scan Mode ANSC0[7] corresponds to ch7, and ANSC0[0] corresponds to ch0. 0b: Channel is not target 1b: Channel is target

7.10.2.2.28 A/D Group C Trigger Select Register (ADCm_ADGCTRGR)



Bit	Bit Name	Initial Value	R/W	Description
7	GRCE	0h	RW	Group C A/D Conversion Enable 0b: Group C A/D conversion is disabled 1b: Group C A/D conversion is enabled
6	GCADIE	0h	RW	Group C Scan Completion Interrupt Enable 0b: Group C scan completed interrupt (ada_gcadireq_n) is disabled 1b: Group C scan completed interrupt (ada_gcadireq_n) is enabled
5 to 0	TRSC[5:0]	0h	RW	Group C A/D Conversion Start Trigger Select Select starting trigger for A/D conversion of group C in group scan mode. See Table 7.10-7 for more details.

Table 7.10-7 A/D Conversion Start Trigger for Group C in Group Scan Mode (1/3)

			TRSC					
Module	Trigger	Description	[5]	[4]	[3]	[2]	[1]	[0]
No trigger is selected			1	1	1	1	1	1
GPT	GTADTRA0N	Compare match with GPT00.GTADTRA	0	0	0	0	0	1
	GTADTRB0N	Compare match with GPT00.GTADTRB	0	0	0	0	1	0
	GTADTRA1N	Compare match with GPT01.GTADTRA	0	0	0	0	1	1
	GTADTRB1N	Compare match with GPT01.GTADTRB	0	0	0	1	0	0
	GTADTRA2N	Compare match with GPT02.GTADTRA	0	0	0	1	0	1
	GTADTRB2N	Compare match with GPT02.GTADTRB	0	0	0	1	1	0
	GTADTRA3N	Compare match with GPT03.GTADTRA	0	0	0	1	1	1
	GTADTRB3N	Compare match with GPT03.GTADTRB	0	0	1	0	0	0
	GTADTRA4N	Compare match with GPT04.GTADTRA	0	0	1	0	0	1
	GTADTRB4N	Compare match with GPT04.GTADTRB	0	0	1	0	1	0
	GTADTRA5N	Compare match with GPT05.GTADTRA	0	0	1	0	1	1
	GTADTRB5N	Compare match with GPT05.GTADTRB	0	0	1	1	0	0
	GTADTRA6N	Compare match with GPT06.GTADTRA	0	0	1	1	0	1
	GTADTRB6N	Compare match with GPT06.GTADTRB	0	0	1	1	1	0
	GTADTRA7N	Compare match with GPT07.GTADTRA	0	0	1	1	1	1
	GTADTRB7N	Compare match with GPT07.GTADTRB	0	1	0	0	0	0
	GTADTRA8N	Compare match with GPT10.GTADTRA	0	1	0	0	0	1
	GTADTRB8N	Compare match with GPT10.GTADTRB	0	1	0	0	1	0
	GTADTRA9N	Compare match with GPT11.GTADTRA	0	1	0	0	1	1
	GTADTRB9N	Compare match with GPT11.GTADTRB	0	1	0	1	0	0
	GTADTRA10N	Compare match with GPT12.GTADTRA	0	1	0	1	0	1
	GTADTRB10N	Compare match with GPT12.GTADTRB	0	1	0	1	1	0
	GTADTRA11N	Compare match with GPT13.GTADTRA	0	1	0	1	1	1
	GTADTRB11N	Compare match with GPT13.GTADTRB	0	1	1	0	0	0
	GTADTRA12N	Compare match with GPT14.GTADTRA	0	1	1	0	0	1

Table 7.10-7 A/D Conversion Start Trigger for Group C in Group Scan Mode (2/2)

Module	Trigger	Description	TRSC					
			[5]	[4]	[3]	[2]	[1]	[0]
	GTADTRB12N	Compare match with GPT14.GTADTRB	0	1	1	0	1	0
	GTADTRA13N	Compare match with GPT15.GTADTRA	0	1	1	0	1	1
	GTADTRB13N	Compare match with GPT15.GTADTRB	0	1	1	1	0	0
	GTADTRA14N	Compare match with GPT16.GTADTRA	0	1	1	1	0	1
	GTADTRB14N	Compare match with GPT16.GTADTRB	0	1	1	1	1	0
	GTADTRA15N	Compare match with GPT17.GTADTRA	0	1	1	1	1	1
	GTADTRB15N	Compare match with GPT17.GTADTRB	1	0	0	0	0	0
	GTADTRA0N or GTADTRB0N	Compare match with GPT00.GTADTRA or Compare match with GPT00.GTADTRB	1	0	0	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPT01.GTADTRA or Compare match with GPT01.GTADTRB	1	0	0	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPT02.GTADTRA or Compare match with GPT02.GTADTRB	1	0	0	0	1	1
	GTADTRA3N or GTADTRB3N	Compare match with GPT03.GTADTRA or Compare match with GPT03.GTADTRB	1	0	0	1	0	0
	GTADTRA4N or GTADTRB4N	Compare match with GPT04.GTADTRA or Compare match with GPT04.GTADTRB	1	0	0	1	0	1
	GTADTRA5N or GTADTRB5N	Compare match with GPT05.GTADTRA or Compare match with GPT05.GTADTRB	1	0	0	1	1	0
	GTADTRA6N or GTADTRB6N	Compare match with GPT06.GTADTRA or Compare match with GPT06.GTADTRB	1	0	0	1	1	1
	GTADTRA7N or GTADTRB7N	Compare match with GPT07.GTADTRA or Compare match with GPT07.GTADTRB	1	0	1	0	0	0
	GTADTRA8N or GTADTRB8N	Compare match with GPT10.GTADTRA or Compare match with GPT10.GTADTRB	1	0	1	0	0	1
	GTADTRA9N or GTADTRB9N	Compare match with GPT11.GTADTRA or Compare match with GPT11.GTADTRB	1	0	1	0	1	0
	GTADTRA10N or GTADTRB10N	Compare match with GPT12.GTADTRA or Compare match with GPT12.GTADTRB	1	0	1	0	1	1
	GTADTRA11N or GTADTRB11N	Compare match with GPT13.GTADTRA or Compare match with GPT13.GTADTRB	1	0	1	1	0	0
	GTADTRA12N or GTADTRB12N	Compare match with GPT14.GTADTRA or Compare match with GPT14.GTADTRB	1	0	1	1	0	1
	GTADTRA13N or GTADTRB13N	Compare match with GPT15.GTADTRA or Compare match with GPT15.GTADTRB	1	0	1	1	1	0
	GTADTRA14N or GTADTRB14N	Compare match with GPT16.GTADTRA or Compare match with GPT16.GTADTRB	1	0	1	1	1	1

Table 7.10-7 A/D Conversion Start Trigger for Group C in Group Scan Mode (3/3)

Module	Trigger	Description	TRSC					
			[5]	[4]	[3]	[2]	[1]	[0]
	GTADTRA15N or GTADTRB15N	Compare match with GPT17.GTADTRA or Compare match with GPT17.GTADTRB	1	1	0	0	0	0
ELC	ELCTRG0		1	1	0	0	0	1

7.10.3 Operation

7.10.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

There are three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0b from 1b. In group scan mode, the selected channels of group A and the selected channels of group B (and the selected channels of group C) are scanned once after starting to be scanned according to the respective internal triggers (GPT, ELC).

In single scan mode and continuous scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register, starting from the pin with the smallest number n. In group scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register for group A, and performed for ANn pins selected by the ADANSB0 register for group B and performed for ANn pins selected by the ADANSC0 register for group C, respectively, starting from the pin with the smallest number n.

Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by any of the internal triggers (GPT, ELC) selected by the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use the double trigger.

To enable double trigger extended mode, select double trigger mode, and then select the internal trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 100101b
GTADTRA4N or GTAGTRB4N
- ADSTRGR.TRSA[5:0] = 101000b
GTADTRA7N or GTADTRB7N

In the double trigger extended mode, in addition to the normal double trigger mode operation, the A/D conversion data activated by the odd number trigger (GTADTRA4N , GTADTRA7N) is stored in the A/D data duplex register A (ADDBLDRA), and the A/D conversion data activated by the even number trigger (GTADTRB4N , GTADTRB7N) is stored in the A/D Data Duplication register B (ADDBLDRB).

If two types of trigger source occur at the same time, the data is not sorted by the trigger factor and the A/D conversion data is stored in the Data Duplication register B (ADDBLDRB). If one trigger is performing A/D conversion and the other trigger is input, the other trigger will be ignored.

7.10.3.2 Single Scan Mode

7.10.3.2.1 Basic operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

1. When the ADCSR.ADST bit becomes 1b (A/D conversion start) due to software or internal trigger (GPT, ELC) input, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. If the ADCSR.ADIE bit is set to 1b (ada_adireq_n interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ada_adireq_n interrupt request is generated.
4. ADCSR.ADST bit holds 1b (A/D conversion start) during A/D conversion and is automatically cleared when A/D conversion of all selected channels is completed, and the 12-bit A/D converter goes into a standby state.

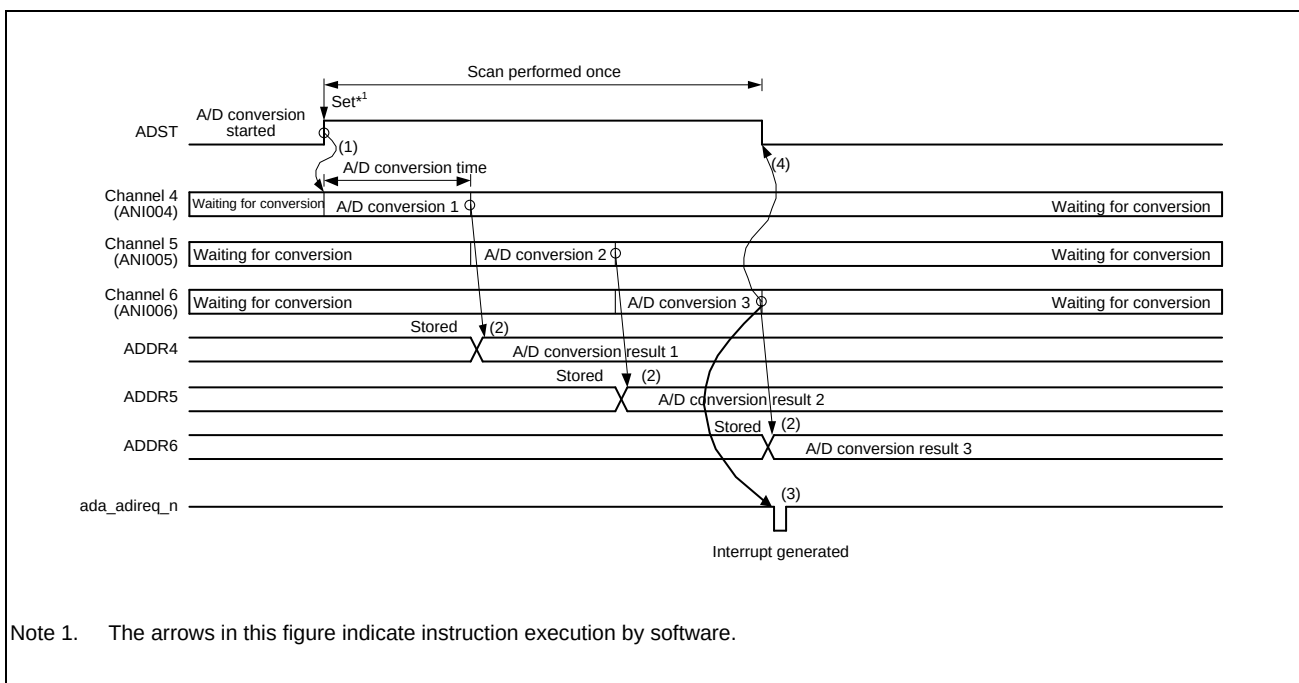


Figure 7.10-2 Example of Operation in Single Scan Mode (Basic Operation ANI004 to ANI006 Selected)

7.10.3.2.2 A/D conversion in double trigger mode

When double trigger mode is selected in single scan mode, A/D conversion is performed for two rounds of single scan operation started by a internal trigger (GPT, ELC).

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1b. When ADCSR.DBLE is set to 1b, the channel selection of the ADANSA0 register is disabled.

To select the double trigger mode, use ADSTRGR.TRSA[5:0] to select the internal trigger (GPT, ELC), set the ADCSR.EXTRG bit to 0b, and set the ADCSR.TRGE bit to 1b. Also, do not use software trigger.

1. When the ADCSR.ADST bit becomes 1b (A/D conversion start) due to the internal trigger (GPT, ELC) input, the A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ada_adireq_n interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ada_adireq_n interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1b (A/D conversion start) by the second internal trigger input, A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) dedicated to the double trigger mode.
6. If the ADCSR.ADIE bit is set to 1 (ada_adireq_n interrupt enabled at the end of scanning), an ada_adireq_n interrupt request is generated.
7. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

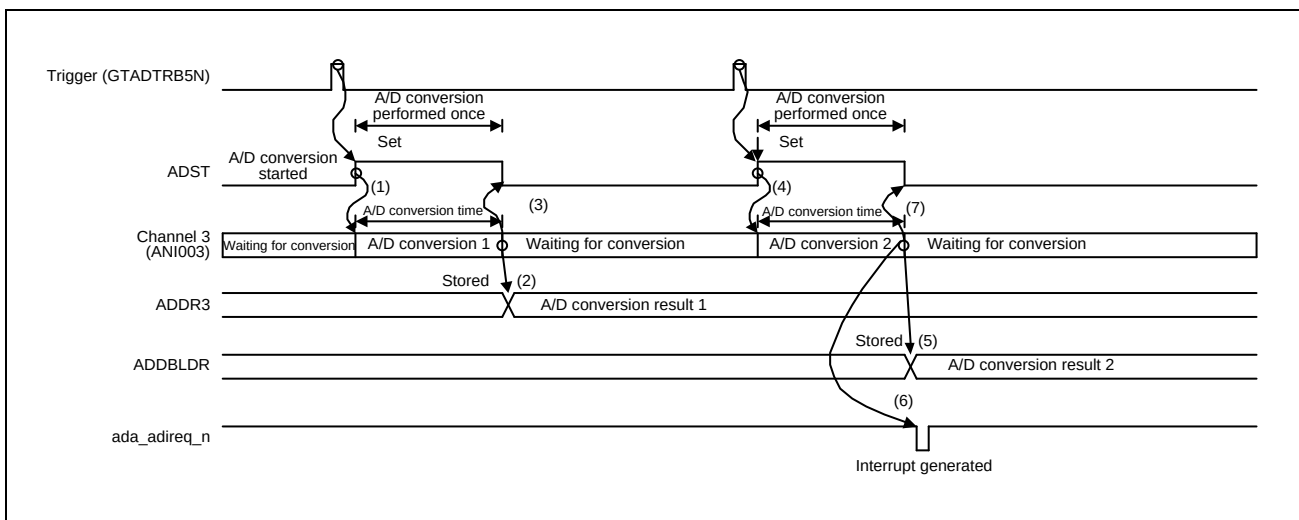


Figure 7.10-3 Example of Operation in Single Scan Mode (Double Trigger Mode, ANI003 Duplicated, GTADTRB5N is Selected as the Trigger)

7.10.3.2.3 Extended operations when double trigger mode is selected

When double trigger mode is selected in single scan mode, single scan is performed twice.

To enable double trigger extended mode, and then select the following internal trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 100001b
GTADTRA0N or GTADTRB0N
- ADSTRGR.TRSA[5:0] = 100110b
GTADTRA5N or GTADTRB5N

And set the ADCSR.EXTRG bit to 0b and the ADCSR.TRGE bit to 1b. Also, do not use software trigger.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1b. When ADCSR.DBLE is set to 1b, the channel selection of the ADANSA0 register is disabled.

1. When the ADCSR.ADST bit becomes 1b (A/D conversion start) by GTADTRA0N, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy) and A/D Data Duplication register A (ADDBLDRA).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ada_adireq_n interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ada_adireq_n interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1b (A/D conversion start) by GTADTRB0N input, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) and the A/D Data Duplication register B (ADDBLDRB).
6. If the ADCSR.ADIE bit is set to 1b (ada_adireq_n interrupt enabled at the end of scanning), an ada_adireq_n interrupt request is generated.
7. The ADCSR.ADST bit holds 1b (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

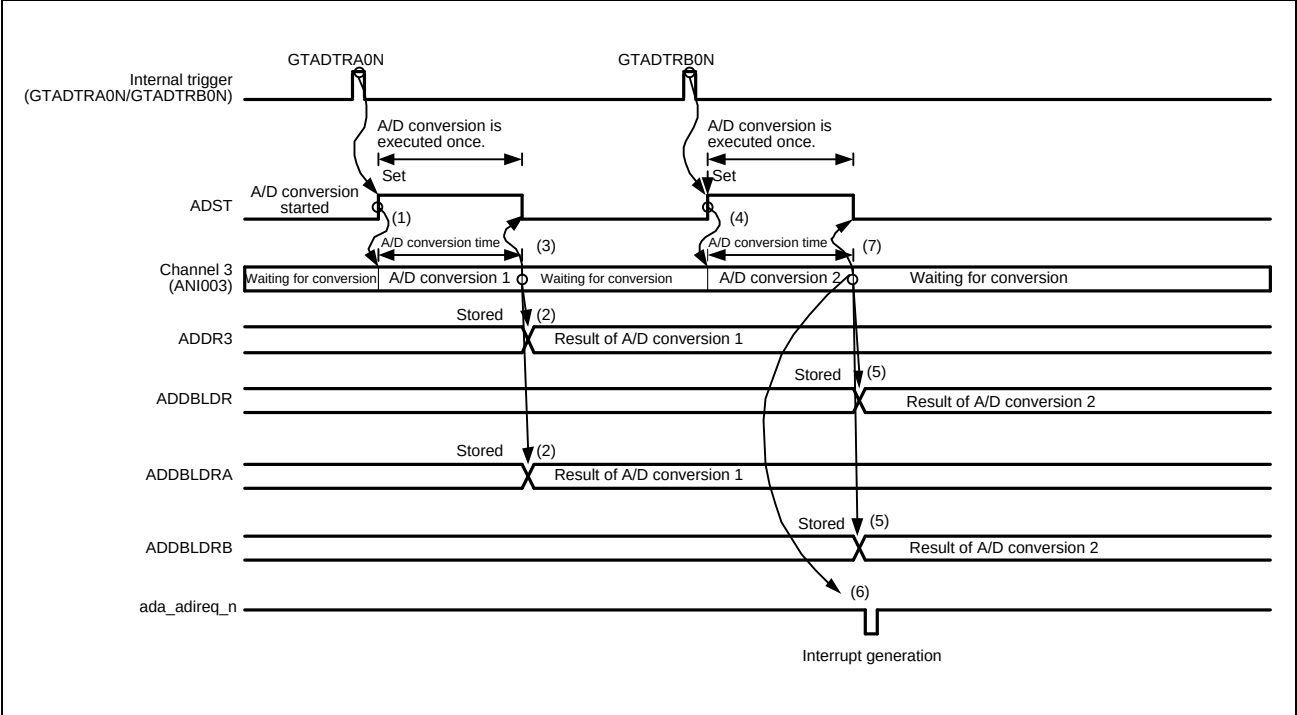


Figure 7.10-4 Example of Extended Operation in Double Trigger Mode (Duplication Selected for ANI003, GTADTRA0N , GTADTRB0N Selected)

7.10.3.3 Continuous Scan Mode

7.10.3.3.1 Basic operation (not use S/H circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the channels selected by the ADANSA0 register.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, internal trigger (GPT, ELC), or external trigger, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. If the ADCSR.ADIE bit is set to 1b (ada_adireq_n interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ada_adireq_n interrupt request is generated. In addition, the 12-bit A/D converter continuously starts A/D conversion in ascending order of n of ANn selected in the ADANSA0 register.
4. The ADCSR.ADST bit is not cleared automatically, and (2) to (3) are repeated during 1b (A/D conversion start). When the ADCSR.ADST bit is set to 0b (A/D conversion stopped), A/D conversion is stopped and the 12-bit A/D converter is in the standby state.
5. After that, when the ADCSR.ADST bit becomes 1b (A/D conversion starts), A/D conversion starts again in ascending order of n of ANn selected in the ADANSA0 register.

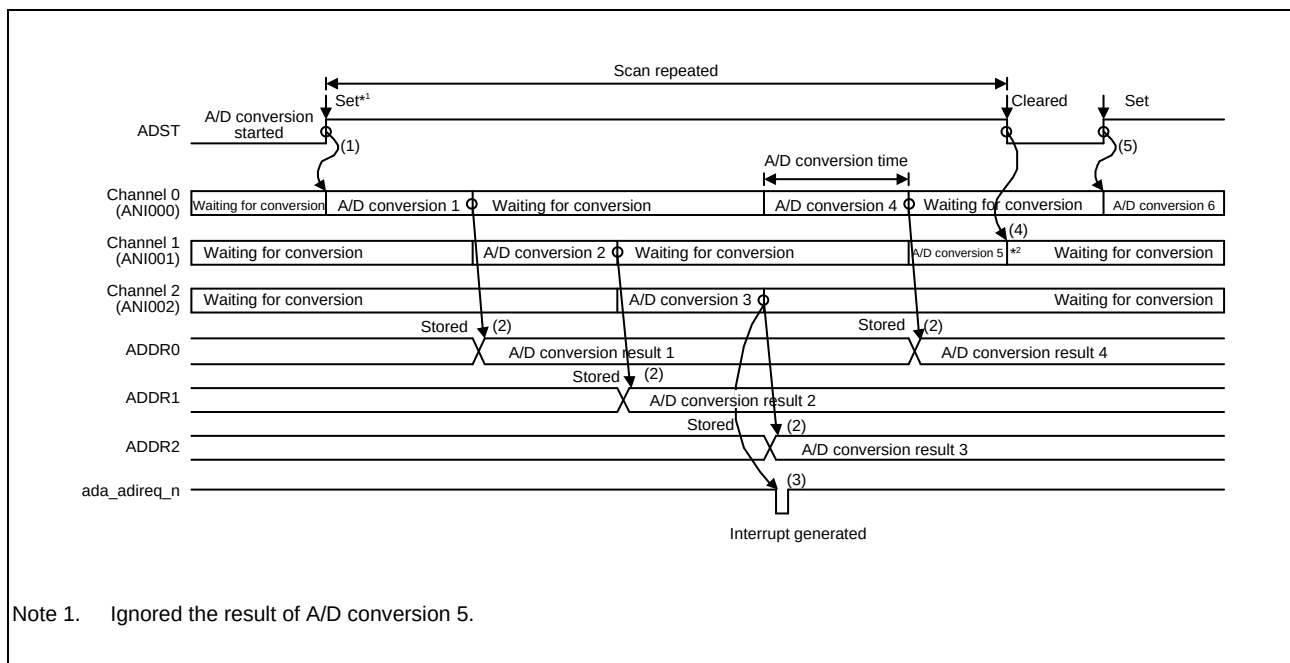


Figure 7.10-5 Example of Operation in Continuous Scan Mode (Basic Operation ANI000 to ANI002 Selected)

7.10.3.4 Group Scan Mode

7.10.3.4.1 Basic operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in each group (A and B or A, B and C) after scanning is started by an internal trigger (GPT, ELC). Scan operation of each group is similar to the scan operation in single scan mode.

To set the trigger for group scan mode, select the trigger for group A with the ADSTRGR.TRSA[5:0] bits, select the trigger for group B with the ADSTRGR.TRSB[5:0] bits, and select the trigger for group C with the ADGCTRGR.TRSC[5:0] bits. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers. For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

The following is an example of the operation of the group scan mode by the internal trigger from the GPT. Group A is set to start conversion with a GTADTRA0N trigger from GPT, Group B is set to start conversion with a GTADTRA3N trigger from GPT, and Group C is set to start conversion with a GTADTRA4N trigger from GPT.

1. Start scanning of group A with GTADTRA0N trigger from GPT.
2. If the ADCSR.ADIE bit is set to 1b (ada_adireq_n interrupt enabled) at the end of group A scanning, the ada_adireq_n interrupt is output.
3. Start scanning Group B with GTADTRA3N trigger from GPT.
4. If the ADCSR.GBADIE bit is set to 1b (ada_gbadireq_n interrupt enabled) at the end of Group B scanning, an ada_gbadireq_n interrupt is output.
5. Trigger GTADTRA4N from GPT to start scanning for Group C.
6. If the ADGCTRGR.GCADIE bit is set to 1b (ada_gcadireq_n interrupt enabled) at the end of Group C scanning, an ada_gcadireq_n interrupt is output.

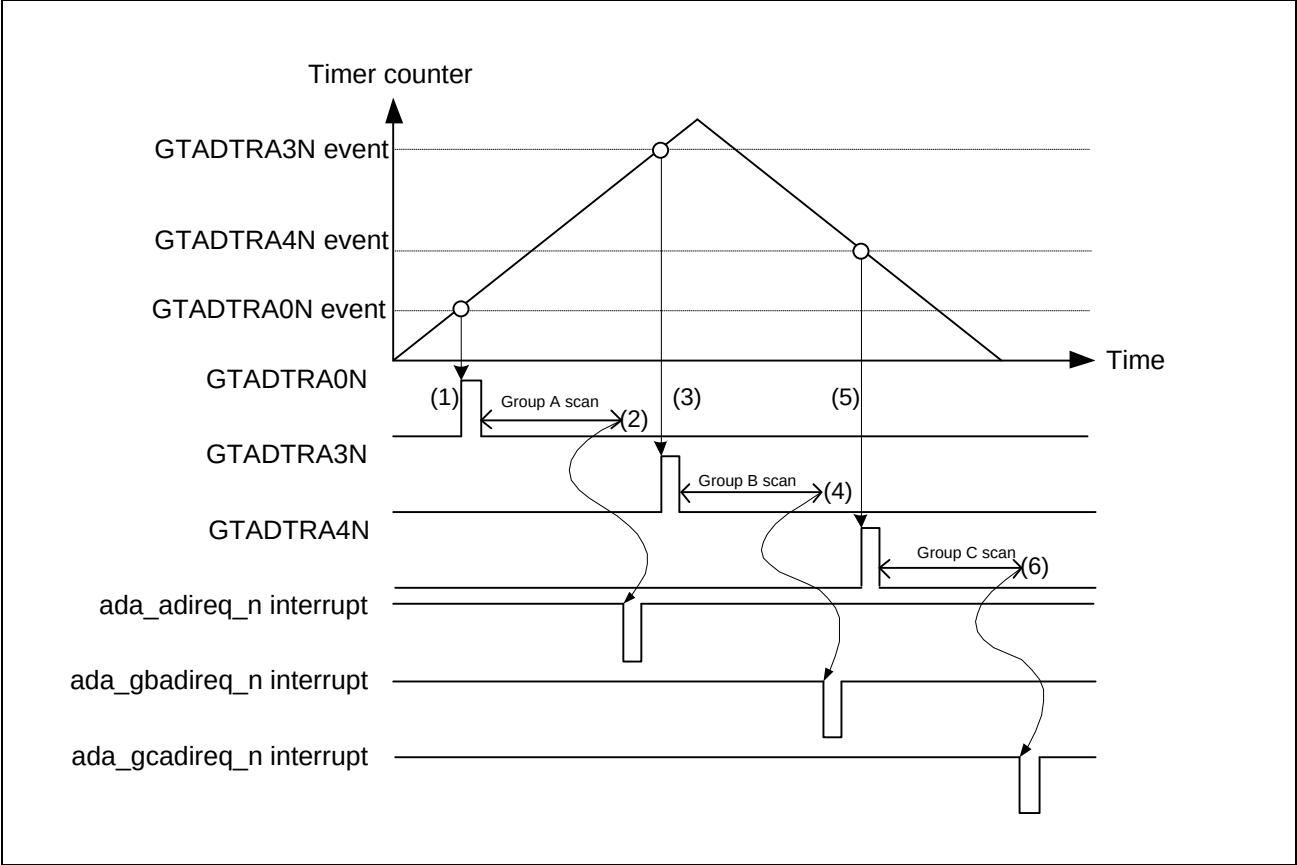


Figure 7.10-6 Example of Operation in Group Scan Mode (Basic Operation Trigger from GPT)

7.10.3.4.2 A/D conversion in double trigger mode

When double trigger mode is selected in group scan mode, group A controls two single scan mode executions starting with an internal trigger (GPT, ELC) as a series of operations. Group B and Group C behave in the same way as single scan mode started with an internal trigger (GPT, ELC). For group scan mode trigger settings, the ADSTRGR.TRSA[5:0] bits select the trigger for group A, the ADSTRGR.TRSB[5:0] bits select the trigger for group B, and ADGCTRGR.TRSC[5:0] bits selects the trigger for Group C. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers.

If you select “GTADTRA0N or GTADTRB0N” to “GTADTRA15N or GTADTRB15N” as the trigger for group A in the ADSTRGR.TRSA[5:0] bits, it operates in double trigger extended mode.

For the channels to be scanned, select the group A channel with the ADCSR.DBLANS[4:0] bits, select the group B channel with the ADANSB0 register, and select the group C channel with the ADANSC0 register.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following shows an operation example when the group scan mode and double trigger mode are set by the internal trigger from the GPT. Group A is set to start conversion with a GTADTRA1N trigger from GPT, Group B is set to start conversion with a GTADTRB7N trigger from GPT, and Group C is set to start conversion with a GTADTRA5N trigger from GPT.

1. Start scanning Group C with the GTADTRA5N trigger from the GPT.
2. If the ADGCTRGR.GCADIE bit is set to 1 (ada_gcadireq_n interrupt enabled) at the end of Group C scanning, the ada_gcadireq_n interrupt is output.
3. The GTADTRB7N trigger from the GPT starts scanning for group B.
4. If the ADCSR.GBADIE bit is set to 1b (ada_gbadireq_n interrupt enabled) at the end of Group B scanning, an ada_gbadireq_n interrupt is output.
5. The first GTADTRA1N trigger from the GPT will start the first scan of Group A.
6. At the end of the first scan of Group A, the converted data is stored in ADDRy, and the ada_adireq_n interrupt request does not occur regardless of the setting of the ADCSR.ADIE bit.
7. The second GTADTRA1N trigger from the GPT will start the second scan of Group A.
8. At the end of the second scan of group A, the conversion data is stored in ADDBLDR, and if the ADCSR.ADIE bit is set to 1b (ada_adireq_n interrupt enabled), the ada_adireq_n interrupt is output.
9. The second GTADTRB7N trigger from the GPT starts the second scan of group B.
10. At the end of the second scan of Group B, if the ADCSR.GBADIE bit is set to 1b (ada_gcadireq_n interrupt enabled), an ada_gbadireq_n interrupt is output.
11. The second GTADTRA5N trigger from the GPT starts the second scan of Group C.
12. At the end of the second scan of Group C, if the ADGCTRGR.GCADIE bit is set to 1b (ada_gcadireq_n interrupt enabled), an ada_gcadireq_n interrupt is output.

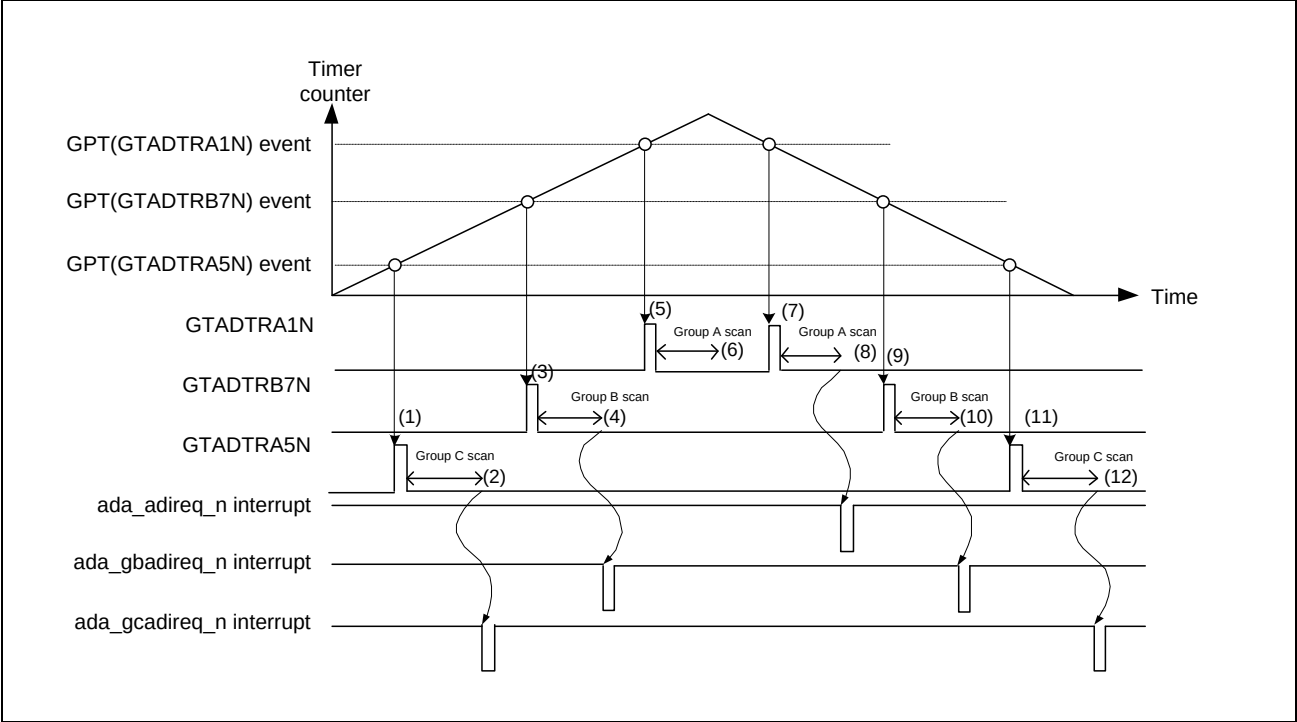


Figure 7.10-7 Example of Operation in Group Scan Mode with Double Trigger (Trigger from GPT)

7.10.3.4.3 Operation under Group Priority Control

When the ADGSPCR.PGS bit is set to 1b in group scan mode, group priority operation is performed. Group priority is in the order of group A > group B > group C. The number of groups used in group scan mode can be selected from either 2 (Groups A, B) or 3 (Groups A, B, C) depending on the setting of ADGCTRGR.GRCE. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure in **Figure 7.10-8** to set the relevant registers. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In the basic operation of group scan mode, the trigger input generated during scanning of groups A, B, and C is ignored, and the scanning operation of groups A, B, and C is the same as that of single scan mode.

In the group priority operation, if there is a trigger input of the priority group during the scan of the low priority group, the scan of the low priority group is interrupted and the priority group is scanned.

When the ADGSPCR.GBRSCN bit is 0b, the low priority group is in the standby state after the priority group scan is completed. Also, the low priority group trigger input that occurs during the scan is ignored.

When the ADGSPCR.GBRSCN bit is 1b, the scan of the low priority group is automatically re-executed after the scan of the priority group is completed. In addition, the trigger input of the low priority group that occurred during the scan of the priority group is valid, and the scan of the low priority group is automatically executed after the scan of the priority group is completed.

When the ADGSPCR.GBRSCN bit is 1b and the ADGSPCR.LGRRS is 0b, the scan of the low priority group is restarted from the beginning. Also, when ADGSPCR.LGRRS is 1b, the scan of the low priority group is re-executed from the interrupted channel.

Table 7.10-8 shows the setting of the ADGSPCR.GBRSCN bit and the operation when a trigger is input during scanning.

When the ADGSPCR.GBRP bit is set to 1, the scan operation of the lowest priority group is the operation of executing a single scan continuously.

For the trigger setting in group scan mode, select the group A internal trigger with the ADSTRGR.TRSA[5:0] bits, and set the group B internal trigger different from the group A trigger with the ADSTRGR.TRSB[5:0] bits. Select a Group C internal trigger that is different from the Group A and B triggers on the ADGCTRGR.TRSC[5:0] bits.

When the group scan mode is set to 2 groups (ADGCTRGR.GRCE bit is set to 0b) and the ADGSPCR.GBRP bit is set to 1, the ADSTRGR.TRSB[5:0] bits is set to 3Fh.

If the group scan mode is set to 3 groups (ADGCTRGR.GRCE bit is set to 1) and the ADGSPCR.GBRP bit is set to 1, set the ADGCTRGR.TRSC[5:0] bits to 3Fh.

For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

Table 7.10-9 shows Group priority operation setting and operation mode of 2 groups, and **Table 7.10-10** shows Group priority operation setting and operation mode of 3 groups.

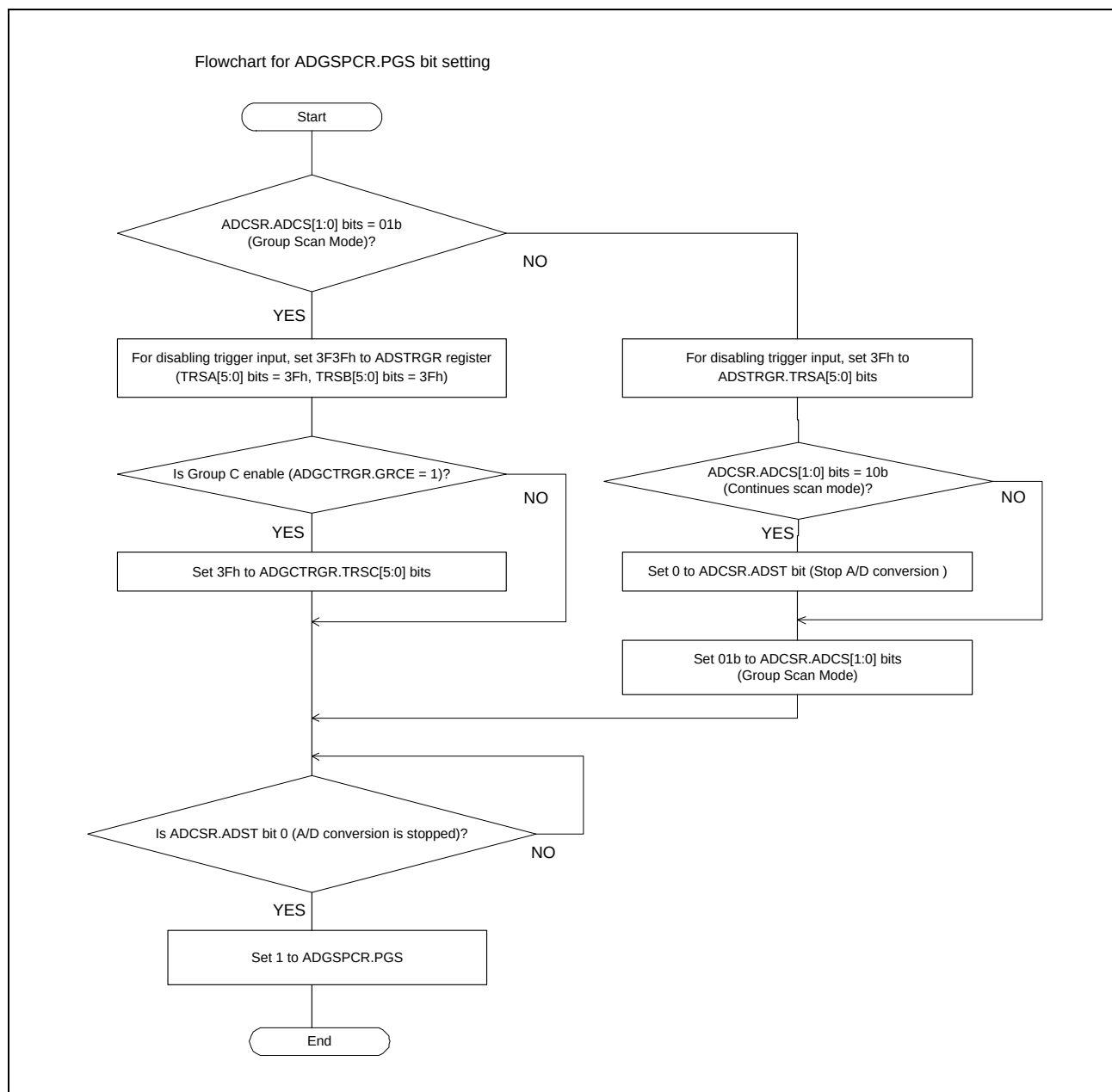


Figure 7.10-8 Flowchart for ADGSPCR.PGS Bit Setting

Table 7.10-8 Control of A/D Conversion Operations according to the Settings of the ADGSPCR.GBRSCN Bit

A/D Conversion Operation	Trigger Input	ADGSPCR.GBRSCN = 0b	ADGSPCR.GBRSCN = 1b
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	- Conversion in progress for group B is discontinued and conversion for group A starts. - Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group B is completed.
When A/D conversion for group C is in progress	Input of trigger for group A	Conversion for group C that is in progress is discontinued and conversion for group A starts.	- Conversion in progress for group C is discontinued and conversion for group A starts. - Conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	Conversion for group C that is in progress is discontinued and conversion for group B starts.	- Conversion in progress for group C is discontinued and conversion for group B starts. - Conversion for group C starts after conversion for group B is completed.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

Table 7.10-9 Group Priority Operation Setting and Operation Mode for 2 Groups (ADGSPCR.PGS = 1b, ADGCTRGR.GRCE = 0b)

ADGSPCR			Operation Mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 2 groups (groups A and B). When the trigger of group A is input, the scan of group B ends (does not re-execute).
1	0	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B restarts the scan from the beginning of the channel specified by ADANSB0 after the scan of group A is completed.
1	1	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B resumes the scan from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.
—	0	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the group B scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSB0 after the group A scan is completed.
1	1	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the scan of group B is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.

Table 7.10-10 Group Priority Operation Setting and Operation Mode for 3 Groups (ADGSPCR.PGS = 1b, ADGCTRGR.GRCE = 1b)

ADGSPCR			Operation Mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). When the trigger of group A or group B is input, the scan of group C ends (does not re-execute).
0	—	1	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	0	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the top channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the top channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	1	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	0	1	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is interrupted, the scan is restarted from the beginning of the channel specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	1	1	Group priority operation of 3 groups (groups A, B, C). After the A/D conversion process of group B is interrupted, the interrupted channel is restarted from the channels specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSC0 after the group A/B scan is completed.

7.10.3.5 Compare Function (Window A, Window B)

7.10.3.5.1 Compare function window A/B

The compare function is a function that compares the reference value set in the register with the A/D conversion result, and the reference value can be set for each window (A/B). Double trigger mode cannot be used when using the compare function. The major difference between window A and window B is that window B has one selectable channel and different interrupt output signals.

Operation using compare function in combination with continuous scan mode is described below.

1. When the ADCSR.ADST bit becomes 1b (A/D conversion start) by software, internal trigger (GPT, ELC), or external trigger, A/D conversion of the selected channel is started.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy). When ADCMPCR.CMPAE = 1b, if it is set for window A in the ADCMPANSR0 register, it is compared with the ADCMPDR0 register setting value. When ADCMPCR.CMPBE = 1b, if the ADCMPBNSR register is set for window B, it will be compared with the ADWINULB/ADWINLLB register setting value.
3. As a result of comparison, when the condition set in ADCMPLR0 is matched in window A, the flag (ADCMPSR0.CMPSTCHA0n) bit of the compare function window A is set to 1b. At this time, if the ADCMPCR.CMPAIE bit is set to 1b, an ada_compai_n interrupt request (level) is generated. Similarly, when window B matches the conditions set in ADCMPBNSR.CMPLB, the compare function window B flag (ADCMPBSR.CMPSTB) bit is set to 1. At this time, if the ADCMPCR.CMPBIE bit is set to 1b, an ada_compbi_n interrupt request (level) is generated.
4. When all the selected A/D conversions and comparisons are completed, scanning is performed again.
5. After accepting the ada_compai_n/ ada_compbi_n interrupt, set the ADCSR.ADST bit to 0b (A/D conversion stopped) and execute processing for the channel for which the compare flag is set.
6. When all the compare flags in window A are cleared, the ada_compai_n interrupt request is canceled. Similarly, clearing the compare window B flag cancels the ada_compbi_n interrupt request. If you want to perform the compare again, start the A/D conversion again.

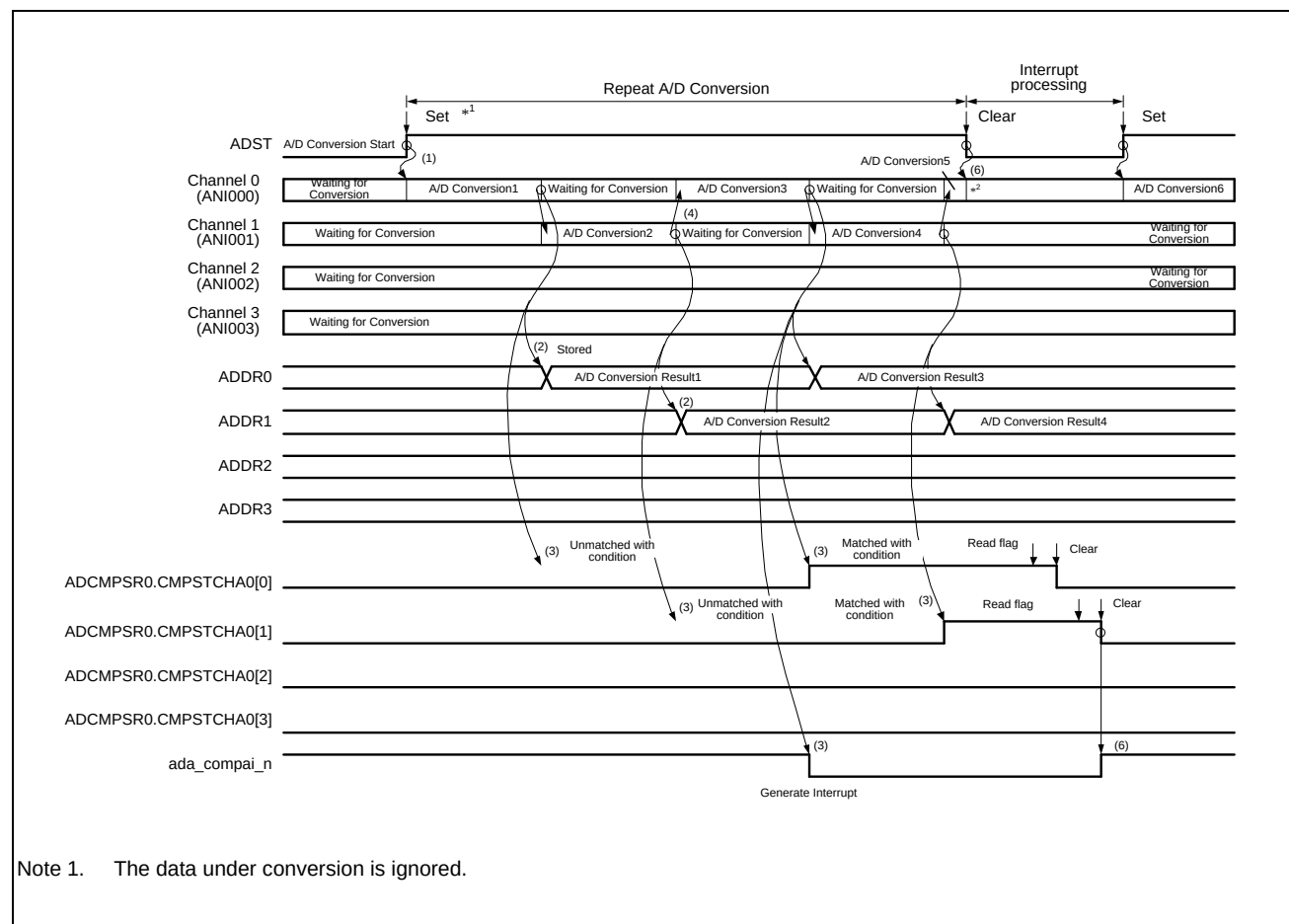


Figure 7.10-9 Example of Operation of Compare Function (Compare Target: ANI000-ANI003)

7.10.3.5.2 Event output in compare function

For the event output of the compare function, specify the upper and lower reference values for each of windows A/B, compare the A/D conversion value of the selected channel with the upper/lower reference values, and then the event (ada_elccondmtch / ada_elccondunmtch) is output according to the event condition (A or B, A and B, A exor B) from the result of satisfying/failing the comparison condition of windows A and B. When using this function, perform A/D conversion in single scan mode.

In window A, any channel of ANn00-ANn07 can be selected. When multiple channels are selected in window A and the comparison condition is satisfied in one of the channels, the comparison condition is satisfied in window A. In window B, one channel of ANn00-ANn15 can be selected.

The setting procedure and setting example when using the event output of the compare function are shown below.

1. ADCSR.ADCS bit must be 00b (single scan mode).
2. Select the channel to be used for window A in the ADCMPANSR0 register, set the comparison condition of the compare function window in the ADCMPLR0 register, and set the upper/lower reference value in the ADCMPDR0/1 register.
3. Select the channel to be used for window B and the comparison condition in the ADCMPBNSR register, and set the upper/lower reference values in the ADWINULB/ADWINLLB registers.
4. In the ADCMPCR register, set the window A/B compound condition, window A/B operation permission, and interrupt output permission.

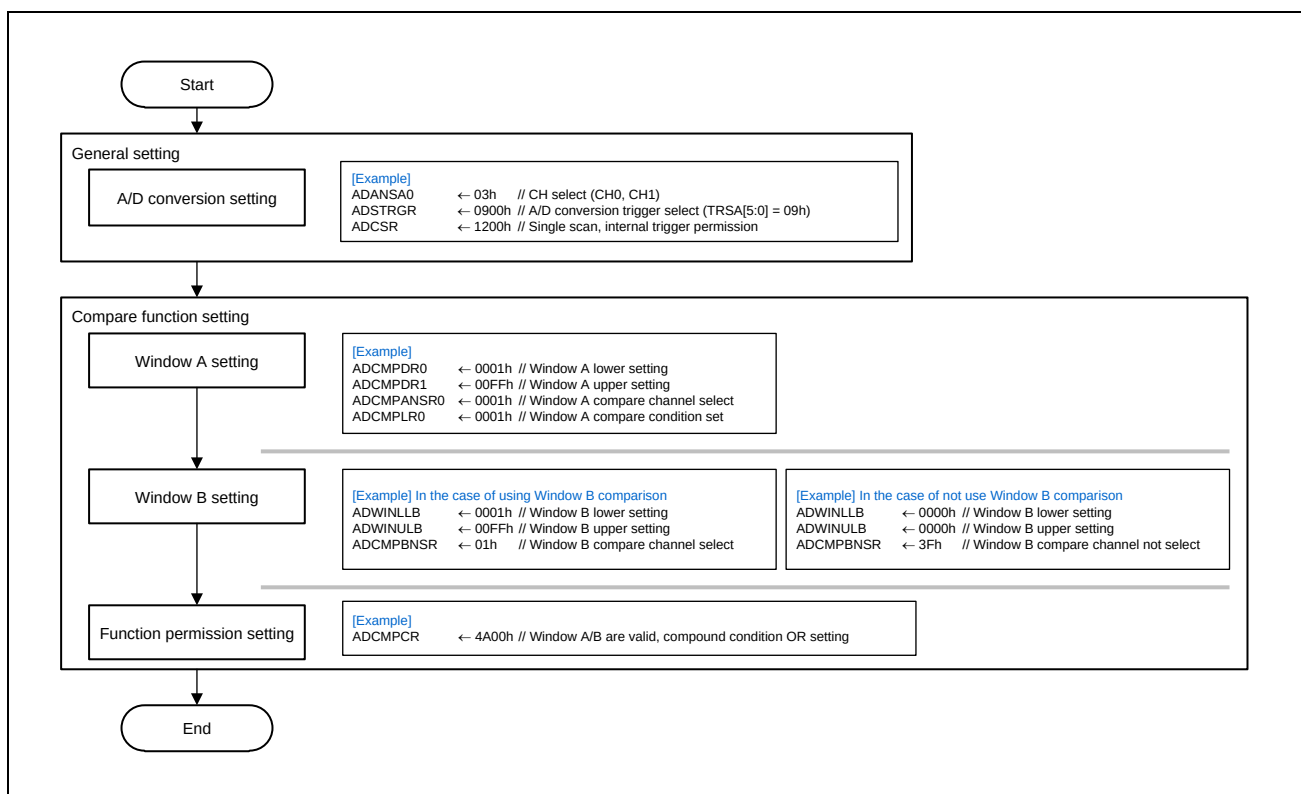


Figure 7.10-10 Example of Setting for Using Event Output of Compare Function

Precautions when using event output using only window A of the compare function

- Enable both windows A and B. (ADCMPCR.CMPAE = 1b, ADCMPCR.CMPBE = 1b)
- Set the compound condition setting for windows A and B to “OR condition”. (ADCMPCR.CMPAB[1:0] = 00b)

- Set the comparison target channel in window B to “Unselected”. (ADCMPCBNSR.CMPCHB[5:0] = 111111b).
- Set the comparison condition of window B to $0 < \text{conversion result} < 0$, which is “always inconsistent”. (ADCMPCR.WCMPE = 1b, ADWINLLB.CMPLLB[15:0] = ADWINULB.CMPULB[15:0] = 0000h, ADCMPBNSR.CMPLB = 1b)

Figure 7.10-11 shows example of event output operation of compare function.

The scan end event is output when one single scan ends. Also, depending on the setting of ADCMPPCR.CMPAB[1:0], the match/unmatch event (ada_elcondmitch / ada_elcondunmitch) is output with a delay of 1.

NOTE

Match/unmatch events are exclusive outputs, and both events are not output at the same time.

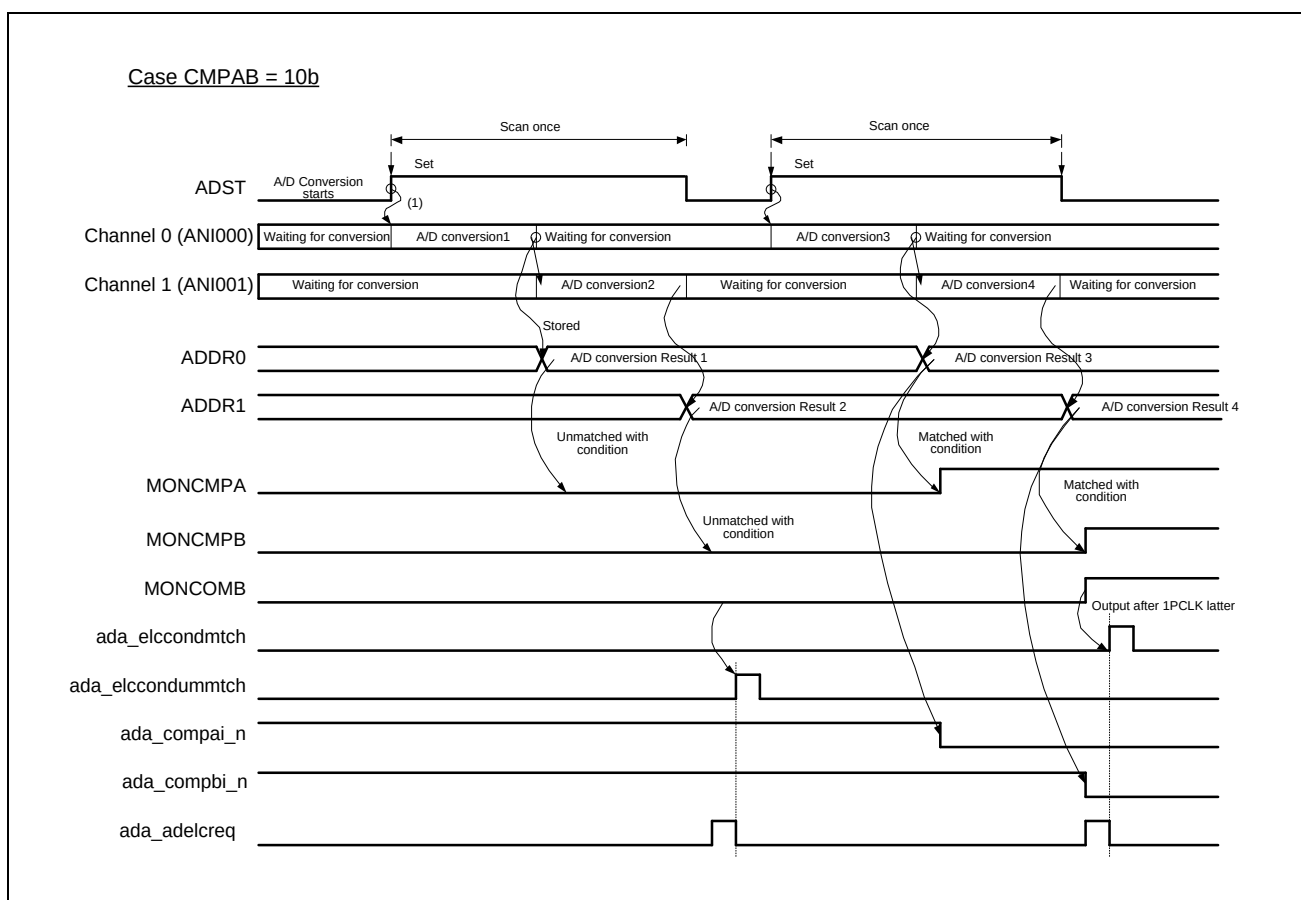


Figure 7.10-11 Example of Operation for Event Output of Compare Function

Precautions when using event output of the compare function

The event output of the compare function outputs match/unmatch according to the setting of ADCMPPCR.CMPAB[1:0] from the comparison result of window A and the comparison result of window B. The comparison result of window A is the logical sum of the comparison result of the channel to be compared in window A.

The comparison result of window A and the comparison result of window B are updated every A/D conversion of the channel to be compared, and the comparison result is retained even after the single scan is completed. To clear the comparison result (0), set ADCMPPCR.CMPAE and ADCMPPCR.CMPBE to 0b.

7.10.3.5.3 Compare function constraint

The compare function has the following restrictions.

- Double trigger mode cannot be used together.
(ADDBLDR, ADDBLDRA, ADDBLDRB are not subject to the compare function.)
- If you want to use match/unmatch event output, set the single scan mode.
- The same CH cannot be set in window A and window B.
- Set so that the upper reference value $\geq$ the lower reference value.

7.10.3.6 A/D Data Register Automatic Clearing Function

By setting the ADCER.ACE bit to 1b, these are used when the CPU and DMAC read the A/D data registers (ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB). The register can be cleared automatically to 0000h.

By using this function, it is possible to detect an un-updated failure of the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers. The following is an example when the automatic clear function of the ADDRy register is disabled/ enabled.

When the ADCER.ACE bit is 0b (automatic clearing prohibited), the old data (0111h) becomes the value of the ADDRy register when the A/D conversion result (0222h) is not written to the ADDRy register for some reason. Furthermore, when the value of this ADDRy register is read to the general-purpose register by using the A/D conversion end interrupt, the old data (0111h) can be saved in the general-purpose register. However, when checking for un-updated data, it is necessary to check while holding old data in RAM and general-purpose registers one by one.

When the ADCER.ACE bit is 1 (automatic clearing enabled), the ADDRy register is automatically cleared to 0000h when ADDRy = 0111h is read by the CPU and DMAC. After that, when 0222h of the A/D conversion result cannot be transferred to the ADDRy register for some reason, the cleared data (0000h) remains as the ADDRy register value. If the value of this ADDRy register is read to a general-purpose register by using the A/D conversion end interrupt, 0000h is held in the general-purpose register. By simply checking that the read data value is 0000h, it can be determined that there was an un-updated failure of the ADDRy register.

7.10.3.7 A/D-Converted Value Addition/Average Function

In A/D-converted value addition function, the same channel is A/D-converted two, three, four or sixteen consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average function, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that affect A/D conversion. This function, however, cannot always guarantee an improvement in A/D conversion accuracy. The A/D-converted value addition/average mode can be specified when A/D conversion of the analog input of the selected channels is selected.

7.10.3.8 Starting A/D Conversion with External Trigger

The A/D conversion can be started by the input of an external trigger. To start up the A/D converter by an external trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 00_0000b and a low-level signal should be input to the external trigger (ADTRG). Both the ADCSR.TRGE and ADCSR.EXTRG bits then should be set to 1b.

Internal triggers cannot be selected for groups B and C used in group scan mode.

7.10.3.9 Starting A/D Conversion with Internal Trigger from Peripheral Module

A/D conversion can be started by the internal trigger from GPT and ELC. When starting A/D conversion with an internal trigger, the ADCSR.TRGE bit is set to 1b, the ADCSR.EXTRG bit is set to 0b, the ADSTRGR.TRSA[5:0], ADSTRGR.TRSB[5:0], and ADGCTRGR.TRSC[5:0] bits are set to the appropriate trigger factor.

7.10.3.10 Error Detection Function (Overwrite Check Function)

The overwrite error detection function detects update of the A/D conversion results held in various A/D data registers, without a need of reading those results. The 12-bit A/D converter can generate an `ada_adereq_n` interrupt, which is an overwrite error interrupt request to the error control module (ECM).

Setting the ADERCR.OWEIE to 1b enables generation of an `ada_adereq_n` interrupt and clearing the bit to 0b disables generation of an `ada_adereq_n` interrupt. Reading the ADOWER and ADOWEER registers can determine the A/D data register in which the overwrite error occurred. Writing 1b to the ADERCLR.OWEC clears all the overwrite error flags held in the ADOWER and ADOWEER registers.

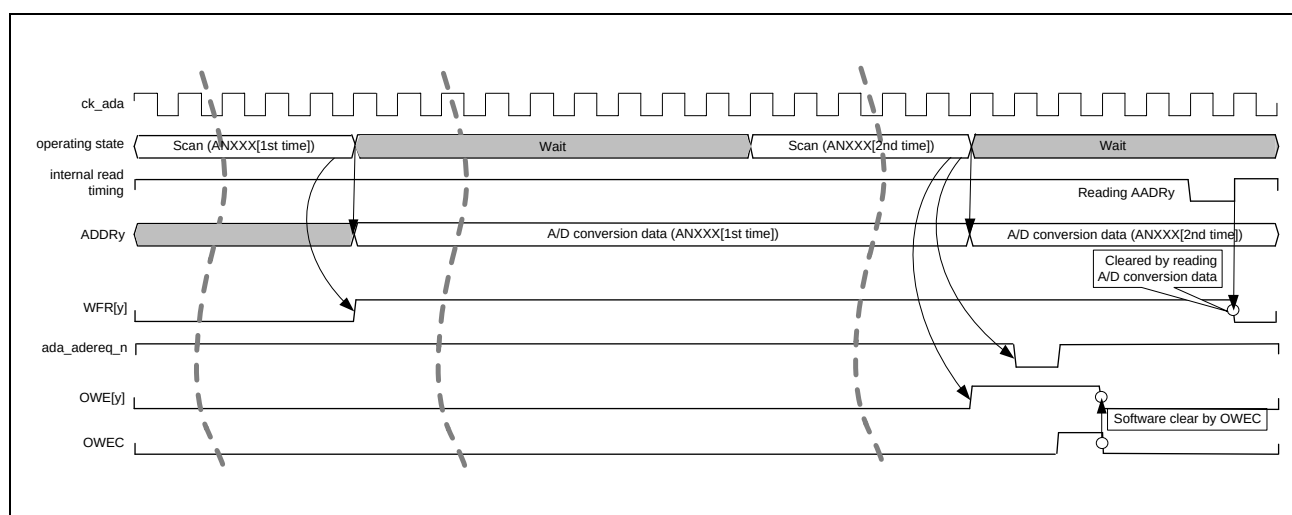


Figure 7.10-12 Example of Generation of AD Overwrite Error Interrupt

7.10.3.11 Event Link Function

7.10.3.11.1 Event output to ELC

ELC can use the `ada_adireq_n` interrupt request signal as an event signal (`ada_adelcreq`) to link to a preset module (`ada_gbadireq_n` / `ada_gcadireq_n` interrupt, `ada_compai_n` / `ada_compb_i_n` interrupt cannot be used as an event signal). The event signal can be output regardless of the setting of the corresponding interrupt request permission bit.

The 12-bit A/D converter outputs the scan end event (`ada_adelcreq`), compare function match event (`ada_elcondmitch`), and unmatched event (`ada_elcondunmitch`). The scan end event (`ada_adelcreq`) outputs a “High pulse” for 1 cycle of `ADC_m_PCLK` ($m = 0$ to 2) at the same output timing as the interrupt output (`ada_adireq_n`) regardless of the setting of `ADCSR.ADIE`.

The compare match/unmatch event (`ada_elcondmitch` / `ada_elcondunmitch`) to ELC outputs a “High pulse” for 1 cycle of `ADC_m_PCLK` ($m = 0$ to 2) at a timing delayed by 1 cycle of `ADC_m_PCLK` ($m = 0$ to 2) from the interrupt output (`ada_adireq_n`) regardless of the setting of `ADCSR.ADIE`. When using the compare match/unmatch event to ELC (`ada_elcondmitch` / `ada_elcondunmitch`), set it to single scan mode.

7.10.3.11.2 Starting A/D conversion with ELC event

The 12-bit A/D converter can start A/D conversion by a preset event by setting `EVTSELx` in `ICU`.

7.10.3.11.3 Notes on ELC trigger

The trigger from the ELC (`ELCTRG0`) is invalid during A/D conversion.

7.10.4 Usage Notes

7.10.4.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, and A/D data duplication register B must be read in 16-bit word units, to avoid unmatched with the upper value and lower value by reading in byte units separately.

7.10.4.2 Notes on Stopping A/D Conversion

7.10.4.2.1 A/D conversion stop procedure

When using an external or internal trigger as the A/D conversion start condition, follow the procedure in the flowchart in **Figure 7.10-13** to stop the A/D conversion.

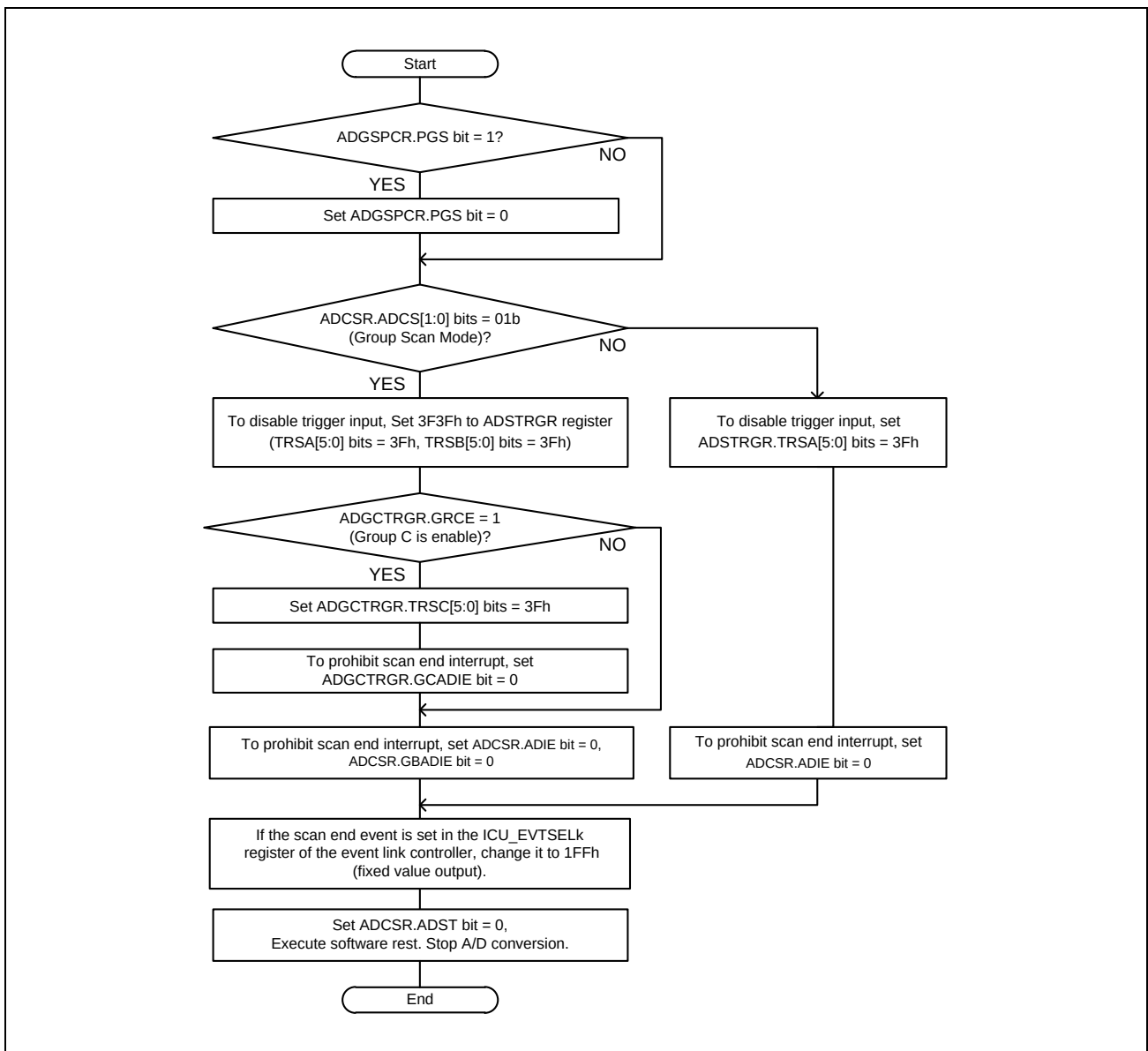


Figure 7.10-13 A/D Conversion Stop Procedure

It takes 2 cycles of ADC_m_ADCLK (m = 0 to 2) from a software clear until the soft macro stops scanning. If you want to set the following after executing a software clear, enter a wait of 2 cycles of ADC_m_ADCLK (m = 0 to 2) or more.

- Scan end interrupt enabled
- Event link controller scan end event valid setting
- Start of A/D conversion by software
- Trigger input valid setting

7.10.4.2.2 Notes on mode/status bits

Initialize or reset the odd/even judgment and monitor bits of the compare function in the double trigger mode as necessary.

- Double trigger mode starts from the first scan operation when ADCSR.DBLE is set from 0b to 1b.
- To initialize the monitor bits (MONCMPA, MONCMPB) of the compare function, set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0b.

7.10.4.3 A/D conversion Forced Stop and Operation Timing at Starts

Up to 6 cycles of the ADC_m_ADCLK (m = 0 to 2) are required to start the A/D converter by setting the ADCSR.ADST bit = 1b from the stopped status.

When the ADCSR.ADST bit is set to 0b and the A/D conversion is forcibly stopped, the analog part of the 12-bit A/D converter requires a maximum of 2 ADC_m_ADCLK (m = 0 to 2) cycles to stop operating.

7.10.4.4 Notes on Scan End Interrupt Processing

When scanning the same analog input twice using any trigger, if the CPU does not complete reading out the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated, the first A/D-converted data is overwritten with the second A/D-converted data and an overwrite error occurs. For details on how to detect overwrite errors in A/D conversion results, please see **7.10.3.10 Error Detection Function (Overwrite Check Function)**.

7.10.4.5 Notes on Using Hardware and Software Triggers Simultaneously.

If the next A/D conversion start trigger is input during A/D conversion, the trigger will be ignored. Therefore, when using hardware trigger (i.e., internal or external trigger) and software trigger at the same time, ensure that the hardware trigger is not input during the execution of A/D conversion started by the software trigger. Also, ensure that the software trigger is not input during A/D conversion started by the hardware trigger. Software can check whether A/D conversion is in progress by ADREF. ADSCACT bit.

7.10.4.6 Power-down Mode Settings

The 12-bit A/D converter can be set to power-down mode by SYS_ADC_CFG.SY_MSTP_ADA bit. By default, the A/D converter is in power-down mode (SYS_ADC_CFG.SY_MSTP_ADA = 1b). After leaving power-down mode by setting the SYS_ADC_CFG.SY_MSTP_ADA bit to 0b, it is necessary to wait at least 20 μ s with ADC_m_ADCLK (m = 0 to 2) running before starting A/D conversion. When entering power-down mode, the SYS_ADC_CFG.SY_MSTP_ADA bit must be set to 1b before stopping ADC_m_ADCLK (m = 0 to 2).

a) Procedure for leaving power-down mode

1. Set the SYS_ADC_CFG.SY_MSTP_ADA bit to 0b.
2. Wait at least 20 μ s with ADC_m_ADCLK (m = 0 to 2) running (CPG_CLKON_16.CLK8_ON = 1b) before starting A/D conversion.

b) Procedure for entering power-down mode

1. If A/D conversion is in operation, stop A/D conversion according to **7.10.4.2.1 A/D conversion stop procedure**.
2. Set the SYS_ADC_CFG.SY_MSTP_ADA bit to 1b.
3. Stop ADC_m_ADCLK (m = 0 to 2).

7.10.4.7 Notes on Entering Low Power Consumption States

When transitioning to module standby mode or power-down mode, stop A/D conversion. When stopping the A/D conversion, it is necessary to secure the time until the analog part of the 12-bit A/D converter stops after setting the ADCSR.ADST bit to 0b. For details, see **7.10.4.2.1 A/D conversion stop procedure**.

SECTION 7 LOW-SPEED INTERFACE

7.11 Temperature Sensor Unit (TSU)

This section describes the functions of the temperature sensor unit (TSU).

7.11.1 Functional Overview

The temperature sensor unit (TSU) includes a temperature sensor IP to acquire a code corresponding to the temperature. The temperature sensor IP is composed of a main sensor and an ADC that converts analog outputs of the sensor to digital code.

The acquired temperature code is readable via the APB. Additionally, an interrupt can be output in response to the result of comparison with set upper and lower limits of the temperature code.

There are two ways to start conversion: by software through a register setting and by an ELC trigger. The conversion mode is single scan mode in which conversion is only performed once with averaging of the set number of samples.

Table 7.11-1 TSU Functions in Outline

Classification	Function	Description
Temperature sensor unit	Temperature code read	• Temperature sensor IP control function • Temperature code averaging function • Comparison with the upper and lower limit settings • Interrupt function • Conversion start trigger (by software and by an ELC trigger) • Single scan
AMBA interface	APB slave interface	• Register access

7.11.2 Connection Configuration

Figure 7.11-1 shows the connection configuration of the TSU.

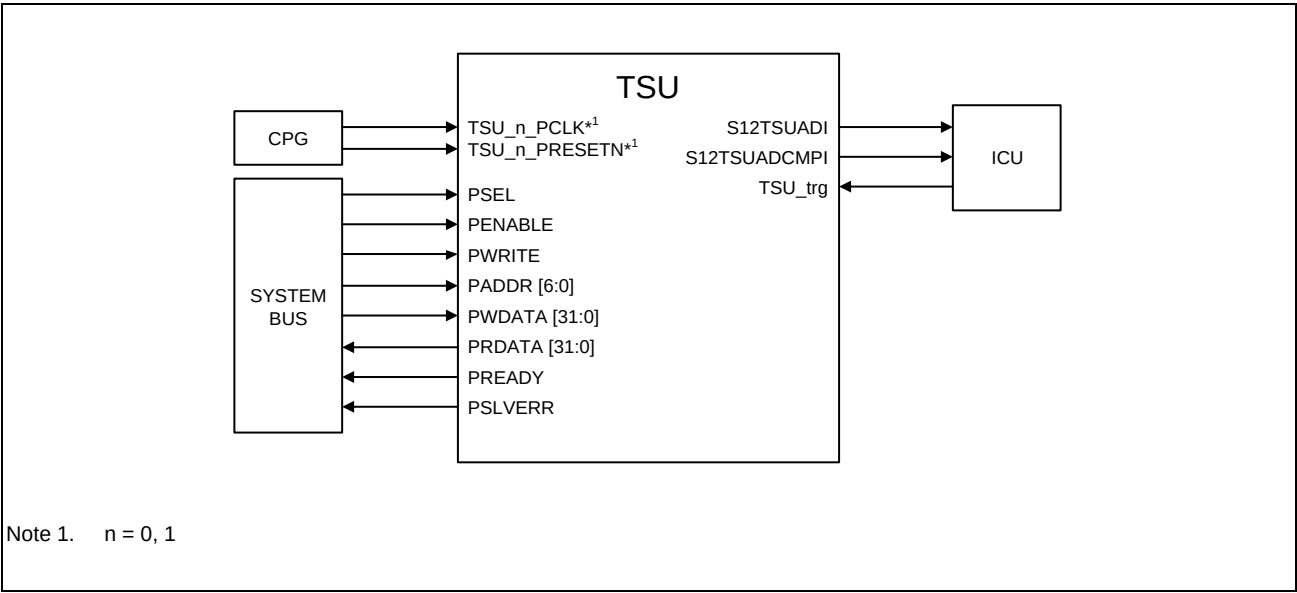


Figure 7.11-1 TSU Connection Configuration Diagram

Table 7.11-2 lists the connected units.

Table 7.11-2 List of Connected Units

Unit Name	Connected Unit Name	Function of Connected Unit
TSU	CPG	Clock pulse generator
	SYSTEM BUS	System bus
	ICU	Interrupt control unit

7.11.3 Pin Description

7.11.3.1 List of Internal Pins

Table 7.11-3 lists the internal pins of the TSU.

Table 7.11-3 List of Internal I/O Pins

Pin Name	Input/ Output	Description
Clock & Reset		
TSU_0_PCLK	Input	Ch0 APB port clock
TSU_1_PCLK	Input	Ch1 APB port clock
TSU_0_PRESETN	Input	Ch0 APB port reset
TSU_1_PRESETN	Input	Ch1 APB port reset
IP Control Interface		
TSU_trg	Input	Conversion start trigger by the ELC (via the ICU) (pulse)
S12TSUADI	Output	Conversion complete interrupt signal (pulse)
S12TSUADCMP1	Output	Comparison result interrupt signal (level)

7.11.3.2 Internal Pin Functions

7.11.3.2.1 Clock and Reset

Clock and reset inputs of the TSU:

■ TSU_n_PCLK (n = 0, 1)

A clock input of the TSU as a whole. The frequency is fixed to 24 MHz.

■ TSU_n_PRESETN (n = 0, 1)

A reset input of the TSU as a whole. This signal resets the TSU when low and releases the reset when high.

7.11.3.2.2 Control Interface

The signals to control the TSU:

■ TSU_trg

This conversion start trigger signal is input to the TSU from the ELC unit via the ICU. This signal is active low and asynchronous relative to TSU_n_PCLK (n = 0, 1), and is effective when a low pulse with a length of eight cycles at 100 MHz is input. After synchronization within the TSU, conversion starts in response to the detection of a falling edge.

■ S12TSUADI

This active-low interrupt output (pulse) signal is generated after the completion of conversion.
It can be masked by a register setting.

■ S12TSUADCMP1

This active-low interrupt output (level) signal is generated when the conditions are met in temperature code comparison.
It can be masked by a register setting.

7.11.4 Clock

7.11.4.1 Clock Signals

Table 7.11-4 Clock Pins

Pin Name	Input/ Output	Frequency (MHz)	Duty Cycle (%)	Function	Value after Reset	Handling of Pins when Not in Use
TSU_0_PCLK	Input	24	—	APB port clock for ch0	—	Low
TSU_1_PCLK	Input	24	—	APB port clock for ch1	—	Low

CAUTION

Fix the TSU_n_PCLK (n = 0, 1) frequency of this unit to 24 MHz. When it is lower than 24 MHz, the temperature sensor IP cannot operate.

7.11.5 Interrupts

The TSU has two interrupt signals, a conversion complete interrupt (S12TSUADI) and a comparison result interrupt (S12TSUADCMPPI).

An interrupt enable/disable switching bit, status confirmation bit, and status clearing bit are in place for each of these signals.

7.11.5.1 Conversion Complete Interrupt

A conversion complete interrupt is generated on completion of conversion.

- When the interrupt is enabled (the ADIE bit of the sensor interrupt enable register = 1)
 - The S12TSUADI signal is low during the period of one TSU_n_PCLK (n = 0, 1) clock cycle and the conversion complete interrupt is generated on completion of conversion by the TSU.
 - The ADF bit of the sensor interrupt status register is set to 1 and the conversion complete status flag is set on completion of conversion by the TSU.
 - Writing 1 to the ADCLR bit of the sensor interrupt clear register sets the ADF bit of the sensor interrupt status register to 0 to clear the status flag (the ADF bit retains the value 1 unless 1 is written to the ADCLR bit).
- When the interrupt is disabled (the ADIE bit of the sensor interrupt enable register = 0)
 - The S12TSUADI signal is not driven to the low level even when conversion by the TSU is completed, and the conversion complete interrupt is not generated.
 - The ADF bit of the sensor interrupt status register is set to 1 on completion of conversion by the TSU and the conversion complete status flag is set, but the interrupt is masked.
 - Writing 1 to the ADCLR bit of the sensor interrupt clear register sets the ADF bit of the sensor interrupt status register to 0 to clear the status flag (the ADF bit retains the value 1 unless 1 is written to the ADCLR bit).

Remark The conversion complete status flag (the ADF bit) is always set to 1 on completion of conversion regardless of the value of the interrupt enable bit ADIE. After the completion of conversion, write 1 to the ADCLR bit to clear the status flag.

7.11.5.2 Comparison Result Interrupt

A comparison result interrupt is generated according to the selected condition for comparison when the result of comparison meets the condition.

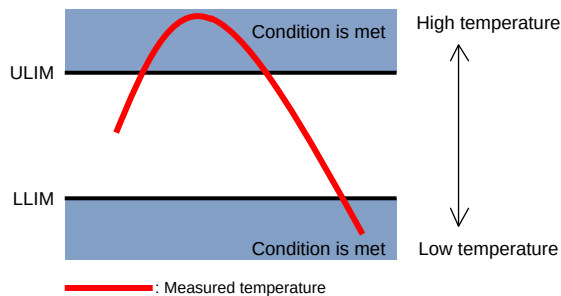
- When the interrupt is enabled (the CMPIE bit of the sensor interrupt enable register = 1)
 - The S12TSUADCMPPI signal is driven to the low level when the result of comparison meets the condition (**Figure 7.11-2**), and the comparison result interrupt is generated.
 - At that time, the CMPF bit of the sensor interrupt status register is set to 1 and the comparison result interrupt status flag is set.
 - Writing 1 to the CMPCLR bit of the sensor interrupt clear register sets the CMPF bit of the sensor interrupt status register to 0 to clear the status flag (the CMPF bit retains the value 1 unless 1 is written to the CMPCLR bit).

- When the interrupt is disabled (when the CMPIE bit of the sensor interrupt enable register = 0)
 - The S12TSUADCMPPI signal is not driven to the low level and the comparison result interrupt is not generated.
 - The CMPF bit of the sensor interrupt status register is set to 1 when the result of comparison meets the condition (**Figure 7.11-2**) and the comparison result interrupt status flag is set, but the interrupt is masked.
 - Writing 1 to the CMPCLR bit of the sensor interrupt clear register sets the CMPF bit of the sensor interrupt status register to 0 to clear the status flag (the CMPF bit retains the value 1 unless 1 is written to the CMPCLR bit).

• **The CMPCOND bit of the compare mode setting register = 0: [Outside the range of the setting values]**

[Comparison condition]

Acquired data > ULIM setting value of the upper limit setting register
or
LLIM setting value of the lower limit setting register > Acquired data



• **The CMPCOND bit of the compare mode setting register = 1: [Within the range of the setting values]**

[Comparison condition]

ULIM setting value of the upper limit setting register > Acquired data
and
Acquired data > LLIM setting value of the lower limit setting register

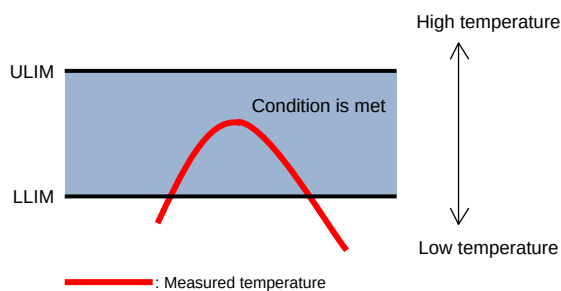


Figure 7.11-2 Comparison Conditions

7.11.6 Registers

The register addresses of TSU are given as offsets from the individual base addresses <TSUm_base>. The register base addresses of each TSU are listed in the following table.

Table 7.11-5 Register Base Addresses

Base Address Name	Unit Name	Base Address
<TSU0_base>	TSU0	0_1100_0000h (5100_0000h* ¹ , 4100_0000h* ²)
<TSU1_base>	TSU1	0_1400_2000h (5400_2000h* ¹ , 4400_2000h* ²)

Note 1. CM33 Address Space Non-Secure

Note 2. CM33 Address Space Secure

7.11.6.1 List of Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Sensor Suspend Register	TSUm_SSUSR	0000_0006h	0000h	32
Sensor Trigger Register	TSUm_STRGR	0000_0000h	0004h	32
Sensor Operation Setting 1 Register	TSUm_SOSR1	0000_000Bh	0008h	32
Reserve	-	-	000Ch to 000Fh	-
Sensor Code Read Register	TSUm_SCRR	0000_0000h	0010h	32
Sensor Status Register	TSUm_SSR	0000_0000h	0014h	32
Compare Mode Setting Register	TSUm_CMSR	0000_0003h	0018h	32
Lower Limit Setting Register	TSUm_LLSR	0000_0000h	001Ch	32
Upper Limit Setting Register	TSUm_ULSR	0000_0000h	0020h	32
Reserve	-	-	0024h to 002Fh	-
Sensor Interrupt Status Register	TSUm_SISR	0000_0000h	0030h	32
Sensor Interrupt Enable Register	TSUm_SIER	0000_0000h	0034h	32
Sensor Interrupt Clear Register	TSUm_SICR	0000_0000h	0038h	32

7.11.6.2 Register Descriptions

The function description of each register is described below.

7.11.6.2.1 Sensor Suspend Register (TSUm_SSUSR)

This register controls settings related to suspension of the temperature sensor IP.

Access Size :		32 bits														
Address :		<TSUm_base> + 0000h														
Initial Value :		0000_0006h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SOC_TS_EN	ADC_PD_TS	EN_TS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	SOC_TS_EN	1h	RW	SOC_TS Enable 0b: SOC_TS disabled (SOC_TS fixed low) 1b: SOC_TS enabled
1	ADC_PD_TS	1h	RW	ADC Power Down 0b: Normal ADC operation 1b: ADC power down
0	EN_TS	0h	RW	Temperature Sensor Enable 0b: Temperature sensor power down 1b: Normal temperature sensor operation

7.11.6.2.2 Sensor Trigger Register (TSUm_STRGR)

This register controls the start trigger for the ADC in the temperature sensor IP.

Access Size :		32 bits														
Address :		<TSUm_base> + 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ADST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W1	W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read.
1	-	0h	W1	Reserved This bit is read as 0b. The write value should always be 0b.
0	ADST	0h	W1	TSU ADC Start Trigger 0b: Invalid 1b: Conversion starts

7.11.6.2.3 Sensor Operation Setting 1 Register (TSUm_SOSR1)

This register controls settings related to operation of the ADC.

Access Size :		32 bits														
Address :		<TSUm_base> + 0008h														
Initial Value :		0000_000Bh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	OUTSEL	-	-	-	-	ADCS	TRGE	-	ADCT[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
R/W	R	R	R	R	R	R	RW	R	R	R	R	RW	RW	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9	OUTSEL	0h	RW	Select data read from Sensor Code Read register 0b: Latest read data 1b: Average data
8 to 5	-	All 0	R	Reserved Whenever it is read, an undefined value is read. The written value will be ignored.
4	ADCS	0h	RW	ADC Scan Mode Select 0b: Single scan 1b: Setting prohibited
3	TRGE	1h	RW	TSU_trg Enable 0b: TSU_trg pin disabled 1b: TSU_trg pin enabled
2	-	0h	R	Reserved Whenever it is read, an undefined value is read. The written value will be ignored.
1, 0	ADCT[1:0]	3h	RW	AD Count Setting 00b: 1 01b: 2 10b: 4 11b: 8

7.11.6.2.4 Sensor Code Read Register (TSUm_SCRR)

This register is a read-only register to read acquired code data. Any written values are ignored.

Access Size :		32 bits														
Address :		<TSUm_base> + 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	OUT12BIT_TS[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	31 to 12		Bit Name	Initial Value	R/W	Description										
			-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
	11 to 0		OUT12BIT_TS [11:0]	0h	R	Temperature sensor code data										

7.11.6.2.5 Sensor Status Register (TSUm_SSR)

This register is a read-only register to read the TSU operating status. Any written values are ignored.

Access Size :		32 bits														
Address :		<TSUm_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CONV
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	CONV	0h	R	TSU operating status 0b: Conversion is finished. 1b: Conversion is in progress. Set: Setting of the ADST bit of the Sensor Trigger Register or assertion of the signal on the TSU_trg pin. Reset: Completion of AD conversion the number of times required for averaging.

7.11.6.2.6 Compare Mode Setting Register (TSUm_CMSR)

This register controls settings related to compare operation.

Access Size :		32 bits														
Address :		<TSUm_base> + 0018h														
Initial Value :		0000_0003h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPC OND	CMPE N
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 2	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
1	CMPCOND		1h	RW	Compare Mode Setting 0b: CMP judges that lower limit (LLILM) > code data or code data > upper limit (ULIM). 1b: CMP judges that lower limit (LLILM) < code data < upper limit (ULIM).											
0	CMPEN		1h	RW	Compare Enable 0b: Comparison disabled 1b: Comparison enabled											

7.11.6.2.7 Lower Limit Setting Register (TSUm_LLSR)

This register controls the lower limit value to be compared with the acquired code data.

Access Size :		32 bits														
Address :		<TSUm_base> + 001Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	LLIM[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 12	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 0	LLIM[11:0]		0h		RW	Lower limit setting value										

7.11.6.2.8 Upper Limit Setting Register (TSUm_ULSR)

This register controls the upper limit value to be compared with the acquired code data.

Access Size :		32 bits														
Address :		<TSUm_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	ULIM[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 12	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 0	ULIM[11:0]		0h		RW	Upper limit setting value										

7.11.6.2.9 Sensor Interrupt Status Register (TSUm_SISR)

This register a read-only register which monitors the interrupt status.

Access Size :

32 bits

Address :

<TSUm_base> + 0030h

Initial Value :

0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPF	ADF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	CMPF	0h	R	Compare result detect flag 0b: Undetected 1b: Detected Set condition: The comparison conditions are met. Reset condition: The CMPCLR bit of the sensor interrupt clear register is set to 1b.
0	ADF	0h	R	AD conversion complete flag 0b: Not completed 1b: Completed Set condition: Completion of AD conversion the number of times required for averaging. Reset condition: The ADCLR bit of the sensor interrupt clear register is set to 1b.

7.11.6.2.10 Sensor Interrupt Enable Register (TSUm_SIER)

This register enables or disables output of the interrupt request output signals (S12TSUADI and S12TSUADCMPI).

Access Size :		32 bits														
Address :		<TSUm_base> + 0034h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPIE	ADIE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 2	-		All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.											
1	CMPIE		0h	RW	Compare Result Detect Interrupt Enable 0b: Interrupt disabled 1b: Interrupt enabled											
0	ADIE		0h	RW	AD Conversion Complete Interrupt Enable 0b: Interrupt disabled 1b: Interrupt enabled											

7.11.6.2.11 Sensor Interrupt Clear Register (TSUm_SICR)

This register clears the interrupt status.

Access Size :		32 bits														
Address :		<TSUm_base> + 0038h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPC LR	ADCLR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W1	W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, an undefined value is read. The written value will be ignored.
1	CMPCLR	0h	W1	The read value is undefined Compare Result Detect Interrupt Clear Write only. This bit is read as 0b. 0b: No operation 1b: Interrupt cleared
0	ADCLR	0h	W1	The read value is undefined AD Conversion Complete Interrupt Clear Write only. This bit is read as 0b. 0b: No operation 1b: Interrupt cleared

7.11.7 Functional Description

This section describes the functions of the TSU.

The prefix (TSUm_) of the register names is omitted in this and subsequent sections.

7.11.7.1 Acquiring Temperature Code

Conversion starts on reception of the conversion start trigger (when the ADST bit of the sensor trigger register is set to 1b or the TSU_trg signal is 0). When the OUT12BIT_TS bit of the sensor code read register is read after the completion of conversion, a 12-bit temperature code can be acquired.

If the conversion start trigger comes within 120T (CLK_TS period) after release of the temperature sensor and ADC from the power-down state (setting the sensor suspend register so that the ADC_PD_TS bit is 0b and the EN_TS bit is 1b), conversion only starts after 120T has elapsed following release.

7.11.7.2 Temperature Code Averaging Function

The temperature codes acquired within the TSU are averaged the number of times set by the ADCT[1:0] bits of the sensor operation setting 1 register.

7.11.7.3 Temperature Code Reading Function

The temperature code can be read from the sensor code read register via the APB.

When the OUTSEL bit of the sensor operation setting 1 register is 0b, the latest converted value is read. When it is 1b, the average value is read.

7.11.7.4 Temperature Code Comparison Functions

This function compares the averaged temperature code and the values set by the LLIM[11:0] bits of the lower limit setting register and the ULIM[11:0] bits of the upper limit setting register when the CMPEN bit of the compare mode setting register is 1b.

The temperature code for use in the comparison function is the averaged temperature code, so the OUTSEL bit of the sensor operation setting 1 register must be set to 1b (average data output) when reading the temperature code for use in comparison. When the averaging count setting is 1, however, the latest value is the same as the average value, so the temperature code for use in comparison will be read regardless of the setting of the OUTSEL bit (see **Figure 7.11-3**).

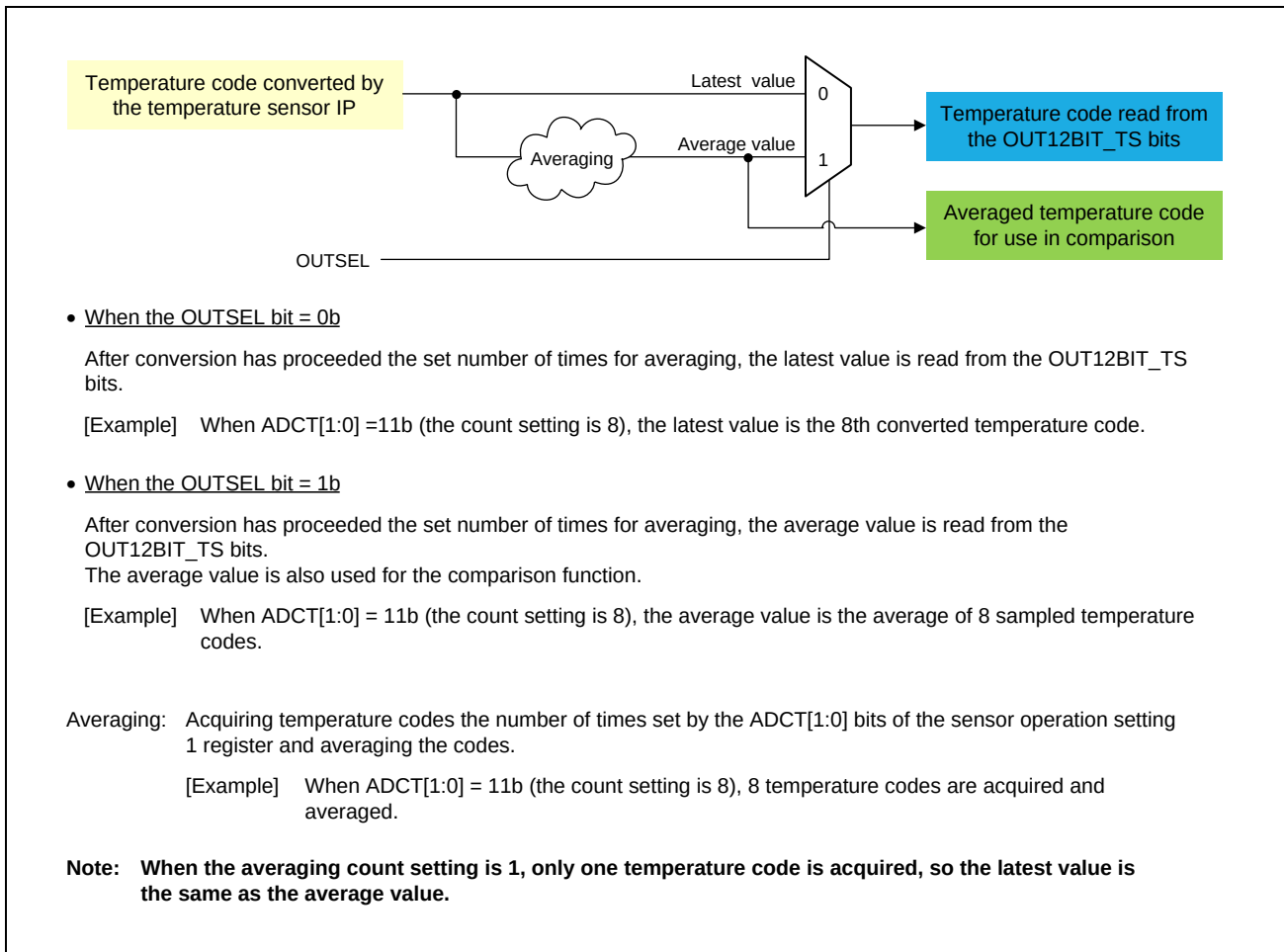


Figure 7.11-3 OUTSEL Bit Setting when Temperature Code Comparison is Used

The CMPF bit of the sensor interrupt status register is set to 1b when the value meets the selected condition for comparison as shown in **Figure 7.11-2**.

The values set in the LLIM[11:0] and ULIM[11:0] bits can be calculated from the formula shown in **7.11.7.9 Comparison Setting Value Calculation**.

7.11.7.5 Interrupt Output Function

An interrupt is output when conversion is completed and the result of temperature code comparison meets the condition.

For details of the interrupt output function, see **7.11.5 Interrupts**.

7.11.7.6 Conversion Start Trigger

There are two ways to start conversion by the TSU: by software through a register setting and by an ELC trigger (TSU_trg).

The TRGE bit of the sensor operation setting 1 register is used to set the ELC trigger.

(TRGE bit = 0b: TSU_trg disabled; TRGE bit = 1b: TSU_trg enabled)

7.11.7.7 Single Scan

In single scan mode, conversion is only performed once with averaging of the number of samples set by the ADCT[1:0] bits of the sensor operation setting 1 register.

Single scan mode is set by default in the ADCS bit of the sensor operation setting 1 register.

(ADCS bit = 0b: Single scan mode; ADCS bit = 1b: Setting prohibited)

7.11.7.8 Temperature Compensation Calculation

After acquiring the temperature, the compensated temperature T (°C) must be calculated from the value (a) of the OUT12BIT_TS bits of the sensor code read register and the values (b) to (c) read from the SYS register.

Follow the procedure below to calculate the compensated temperature. (Maximum error is $\pm 5^{\circ}\text{C}$)

$$T(^{\circ}\text{C}) = \left(\frac{e - d}{c - b} \right) \times (a - b) + d$$

Remarks: $b = \text{SYS: OTPTSUnTRMVAL0}[11:0]$
 $c = \text{SYS: OTPTSUnTRMVAL1}[11:0]$
 $d = -41$
 $e = 126$

7.11.7.9 Comparison Setting Value Calculation

The comparison setting values (a) can be calculated from each temperature T (°C) you want to set and the values from (b) to (c) which are read from the SYS register.

$$a = \frac{(T - d) \times (c - b)}{(e - d)} + b$$

Remarks: $b = \text{SYS: OTPTSUnTRMVAL0}[11:0]$
 $c = \text{SYS: OTPTSUnTRMVAL1}[11:0]$
 $d = -41$
 $e = 126$

Set the calculated values (a) in the LLIM[11:0] bits of the low limit setting register and the ULIM[11:0] bits of the upper limit setting register.

Calculation of the comparison setting values is a reverse calculation of the **(7.11.7.8 Temperature Compensation Calculation)**.

- Temperature compensation calculation: Calculate the temperature T (°C) from the acquired temperature code (a).
- Comparison setting value calculation: Calculate the temperature code (a) from the temperature T (°C) you want to set.

7.11.8 Operation Procedure

The TSU is mainly used in the following two ways. This section describes the procedures of operations for usage in each of these ways.

- Using an interrupt for each completion of conversion (**7.11.8.1 Using Interrupt for Each Completion of Conversion**)

Conversion is started by software or an ELC trigger and the interrupt is generated on completion of conversion.

After generation of the interrupt, the temperature code is read.

The comparison function is enabled by default. The upper and lower limit settings for use in comparison are both all 0s by default and the conditions for comparison are not met even while the comparison function is enabled, but the conditions may inadvertently be met if operation proceeds after the upper and lower limits have been set.

When the comparison function is not to be used, set the CMPEN bit of the compare mode setting register to 0b to disable the comparison function.

- Using the comparison result interrupt (**7.11.8.2 Using Comparison Result Interrupt**)

Conversion is started by software or an ELC trigger after the conditions for comparison (within or outside the range, and the upper and lower limits) have been set and the interrupt is generated when the result of comparison with the settings meets the conditions.

After generation of the interrupt, the temperature code is read.

7.11.8.1 Using Interrupt for Each Completion of Conversion

7.11.8.1.1 Single Scan, Start by Software

Table 7.11-6 shows an example of the procedure for operations when an interrupt is to be used to indicate each completion of conversion (single scan, start by software). This is an example of reading the average value when the number of samples for averaging is 8.

Table 7.11-6 Operation Procedure Example when Interrupt is to be Used for Each Completion of Conversion (Single Scan, Start by Software) (1/2)

Step		Register Name	Bit Name	Processing Details	Outline
(1)	Supply TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKO N_16	CLK9_ONWEN, CLK9_ON(TSU0)/ CLK10_ONWEN, CLK10_ON(TSU1)	Write "1b" ($\_ONWEN$ is write enable)	Supply TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(2)	Check TSU_n_PCLK (n = 0, 1) supply.	CPG:CPG_CLKM ON_8	CLK9_MON(TSU0)/ CLK10_MON(TSU1)	Read "1b"	Check that TSU_0_PCLK/TSU_1_PCLK was supplied to TSU0/TSU1.
(3)	De-assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_ 15	RSTB7_WEN, RSTB7(TSU0)/ RSTB8_WEN, RSTB8(TSU1)	Write "1b" ($\_WEN$ is write enable)	De-assert TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1.
(4)	Check TSU_n_PRESETN (n = 0, 1) de-assertion.	CPG:CPG_RSTM ON_7	RST8_MON(TSU0)/ RST9_MON(TSU1)	Read "0b"	Check that TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1 was de-asserted.

Table 7.11-6 Operation Procedure Example when Interrupt is to be Used for Each Completion of Conversion (Single Scan, Start by Software) (2/2)

Step		Register Name	Bit Name	Processing Details	Outline
(5)	Release the temperature sensor IP from the power-down state.	Sensor Suspend Register	SOC_TS_EN	Write "1b"	Enable the SOC_TS signal.
			ADC_PD_TS	Write "0b"	Normal ADC operation
			EN_TS	Write "1b"	Normal temperature sensor operation
(6)	Set the operating mode.	Sensor Operation Setting 1 Register	OUTSEL	Write "1b"	Average data output
			ADCS	Write "0b"	Single scan mode
			TRGE	Write "0b"	Disable the ELC trigger.
			ADCT[1:0]	Write "11b"	Set the number of samples for averaging to 8.
		Compare Mode Setting Register	CMPCOND	Write "0b"	The setting remains default.
			CMPEN	Write "0b"	Disable the comparison function.
		Sensor Interrupt Enable Register	CMPIE	Write "0b"	Mask the comparison result interrupt.
			ADIE	Write "1b"	Enable the conversion complete interrupt.
(7)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0b (conversion is not in progress).
(8)	Start conversion.	Sensor Trigger Register	ADST	Write "1b"	Start conversion by software.
(9)	The conversion complete interrupt is generated. → Check the interrupt status.	Sensor Interrupt Status Register	ADF	Read "1b"	Read the conversion complete status flag.
(10)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0b (conversion is not in progress).
(11)	Read the converted code.	Sensor Code Read Register	OUT12BIT_TS[11:0]	12-bit read	Read the 12-bit converted temperature code.
(12)	Calculate the temperature compensation.	See 7.11.7.8 Temperature Compensation Calculation.			
(13)	Clear the interrupt status.	Sensor Interrupt Clear Register	ADCLR	Write "1b"	Clear the conversion complete status flag.
(14)	Check the interrupt status.	Sensor Interrupt Status Register	ADF	Read "0b"	Check that the conversion complete status flag was cleared.
(15)	Power down the temperature sensor IP.	Sensor Suspend Register	SOC_TS_EN	Write "0b"	Disable the SOC_TS signal.
			ADC_PD_TS	Write "1b"	Power down the ADC.
			EN_TS	Write "0b"	Power down the temperature sensor.
(16)	Assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "0b" (_WEN is write enable)	Assert TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1.
(17)	Check TSU_n_PRESETN (n = 0, 1) assertion.	CPG:CPG_RSTM_ON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "1b"	Check that TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1 was asserted.
(18)	Stop TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKO N_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "0b" (_ONWEN is write enable)	Stop TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(19)	Check stopping of TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKM_ON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "0b"	Check that TSU_0_PCLK/TSU_1_PCLK supply to TSU0/TSU1 was stopped.

7.11.8.1.2 Single Scan, Start by an ELC Trigger

Table 7.11-7 shows an example of the procedure for operations when an interrupt is to be used to indicate each completion of conversion (single scan, start by an ELC trigger). This is an example of reading the average value when the number of samples for averaging is 8.

Table 7.11-7 Operation Procedure Example when Interrupt is to be Used for Each Completion of Conversion (Single Scan, Start by ELC Trigger) (1/2)

Step		Register Name	Bit Name	Processing Details	Outline
(1)	Supply TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKO N_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "1b" ($\_ONWEN$ is write enable)	Supply TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(2)	Check TSU_n_PCLK (n = 0, 1) supply.	CPG:CPG_CLKM ON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "1b"	Check that TSU_0_PCLK/TSU_1_PCLK was supplied to TSU0/TSU1.
(3)	De-assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "1b" ($\_WEN$ is write enable)	De-assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(4)	Check TSU_n_PRESETN (n = 0, 1) de-assertion.	CPG:CPG_RSTM ON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "0b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was de-asserted.
(5)	Release the temperature sensor IP from power down.	Sensor Suspend Register	SOC_TS_EN	Write "1b"	Enable the SOC_TS signal.
			ADC_PD_TS	Write "0b"	Normal ADC operation
			EN_TS	Write "1b"	Normal temperature sensor operation
(6)	Set the operating mode.	Sensor Operation Setting 1 Register	OUTSEL	Write "1b"	Average data output
			ADCS	Write "0b"	Single scan mode
			TRGE	Write "1b"	Enable the ELC trigger.
			ADCT[1:0]	Write "11b"	Set the number of samples for averaging to 8.
		Compare Mode Setting Register	CMPCOND	Write "0b"	The setting remains default.
			CMPEN	Write "0b"	Disable the comparison function.
		Sensor Interrupt Enable Register	CMPIE	Write "0b"	Mask the comparison result interrupt.
			ADIE	Write "1b"	Enable the conversion complete interrupt.
(7)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0 (conversion is not in progress).
(8)	Set the ELC trigger.	ICU:EVTSEL5(TSU0)/EVTSEL6(TSU1)	TSU0: Bit position [29:20] TSU1: Bit position [9:0]	Write the ELC no.*1 of the event to be used.	Select the ELC trigger to be used from the ICU register.
(9)	Start conversion.	—	—	—	Receive the ELC trigger from the ICU.
(10)	The conversion complete interrupt is generated. → Check the interrupt status.	Sensor Interrupt Status Register	ADF	Read "1b"	Read the conversion complete status flag.
(11)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0 (conversion is not in progress).
(12)	Read the converted code.	Sensor Code Read Register	OUT12BIT_TS[11:0]	12-bit read	Read the 12-bit converted temperature code.-

Table 7.11-7 Operation Procedure Example when Interrupt is to be Used for Each Completion of Conversion (Single Scan, Start by ELC Trigger) (2/2)

Step		Register Name	Bit Name	Processing Details	Outline
(13)	Calculate the temperature compensation.	See 7.11.7.8 Temperature Compensation Calculation.			
(14)	Clear the interrupt status.	Sensor Interrupt Clear Register	ADCLR	Write "1b"	Clear the conversion complete status flag.
(15)	Check the interrupt status.	Sensor Interrupt Status Register	ADF	Read "0b"	Check that the conversion complete status flag was cleared.
(16)	Power down the temperature sensor IP.	Sensor Suspend Register	SOC_TS_EN	Write "0b"	Disable the SOC_TS signal.
			ADC_PD_TS	Write "1b"	Power down the ADC.
			EN_TS	Write "0b"	Power down the temperature sensor.
(17)	Assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "0b" (_WEN is write enable)	Assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(18)	Check TSU_n_PRESETN (n = 0, 1) assertion.	CPG:CPG_RSTMON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "1b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was asserted.
(19)	Stop TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKON_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "0b" (_ONWEN is write enable)	Stop TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(20)	Check stopping of TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKMON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "0b"	Check that TSU_0_PCLK/TSU_1_PCLK supply to TSU0/TSU1 was stopped.

Note 1. For the ELC no., refer to the ICU document "Event_list file".

7.11.8.2 Using Comparison Result Interrupt

7.11.8.2.1 Single Scan, Start by Software

Table 7.11-8 shows an example of the procedure for operations when the comparison result interrupt is to be used (single scan, start by software). This is an example of reading the average value when the number of samples for averaging is 8.

Table 7.11-8 Operation Procedure Example when Comparison Result Interrupt is to be Used (Single Scan, Start by Software) (1/2)

Step		Register Name	Bit Name	Processing Details	Outline
(1)	Supply TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKO N_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write “1b” (_ONWEN is write enable)	Supply TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(2)	Check TSU_n_PCLK (n = 0, 1) supply.	CPG:CPG_CLKM ON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read “1b”	Check that TSU_0_PCLK/TSU_1_PCLK was supplied to TSU0/TSU1.
(3)	De-assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write “1b” (_WEN is write enable)	De-assert TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1.
(4)	Check TSU_n_PRESETN (n = 0, 1) de-assertion.	CPG:CPG_RSTM ON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read “0b”	Check that TSU_0_PRESETN/TSU_1_PRESET N of TSU0/TSU1 was de-asserted.
(5)	Release the temperature sensor IP from the power-down state.	Sensor Suspend Register	SOC_TS_EN	Write “1b”	Enable the SOC_TS signal.
			ADC_PD_TS	Write “0b”	Normal ADC operation
			EN_TS	Write “1b”	Normal temperature sensor operation
(6)	Set the operating mode.	Sensor Operation Setting 1 Register	OUTSEL	Write “1b”	Average data output
			ADCS	Write “0b”	Single scan mode
			TRGE	Write “0b”	Disable the ELC trigger.
			ADCT[1:0]	Write “11b”	Set the number of samples for averaging to 8.
		Compare Mode Setting Register	CMPCOND	Write “1b”	Compare mode (within the range setting)
			CMPEN	Write “1b”	Enable the comparison function.
		Sensor Interrupt Enable Register	CMPIE	Write “1b”	Enable the comparison result interrupt.
			ADIE	Write “0b”	Enable the conversion complete interrupt.
(7)	Set the comparison setting values.	Lower Limit Setting Register	LLIM[11:0]	12-bit write	For how to calculate the comparison setting values, see 7.11.7.9 Comparison Setting Value Calculation .
		Upper Limit Setting Register	ULIM[11:0]	12-bit write	For how to calculate the comparison setting values, see 7.11.7.9 Comparison Setting Value Calculation .
(8)	Check the conversion status.	Sensor Status Register	CONV	Read “0b”	Read the CONV bit as 0b (conversion is not in progress).
(9)	Start conversion.	Sensor Trigger Register	ADST	Write “1b”	Start conversion by software.

Table 7.11-8 Operation Procedure Example when Comparison Result Interrupt is to be Used (Single Scan, Start by Software) (2/2)

Step		Register Name	Bit Name	Processing Details	Outline
(10)	The comparison result interrupt is generated. → Check the interrupt status.	Sensor Interrupt Status Register	CMPF, ADF	Read "11b"	Read the comparison result status and conversion complete status flags.
(11)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0b (conversion is not in progress).
(12)	Read the converted code.	Sensor Code Read Register	OUT12BIT_TS[11:0]	12-bit read	Read the 12-bit converted temperature code.
(13)	Calculate the temperature compensation.	See 7.11.7.8 Temperature Compensation Calculation .			
(14)	Clear the interrupt status.	Sensor Interrupt Clear Register	CMPCLR, ADCLR	Write "1b"	Clear the comparison result status and conversion complete status flags.
(15)	Check the interrupt status.	Sensor Interrupt Status Register	CMPF, ADF	Read "00b"	Check that the comparison result status and conversion complete status flags were cleared.
(16)	Power down the temperature sensor IP.	Sensor Suspend Register	SOC_TS_EN	Write "0b"	Disable the SOC_TS signal.
			ADC_PD_TS	Write "1b"	Power down the ADC.
			EN_TS	Write "0b"	Power down the temperature sensor.
(17)	Assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "0b" (_WEN is write enable)	Assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(18)	Check TSU_n_PRESETN (n = 0, 1) assertion.	CPG:CPG_RSTMON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "1b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was asserted.
(19)	Stop TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKON_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "0b" (_ONWEN is write enable)	Stop TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(20)	Check stopping of TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKMON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "0b"	Check that TSU_0_PCLK/TSU_1_PCLK supply to TSU0/TSU1 was stopped.

7.11.8.2.2 Single Scan, Start by an ELC Trigger

Table 7.11-9 shows an example of the procedure for operations when the comparison result interrupt is to be used (single scan, start by an ELC trigger). This is an example of reading the average value when the number of samples for averaging is 8.

Table 7.11-9 Operation Procedure Example when Comparison Result Interrupt is to be Used (Single Scan, Start by ELC Trigger) (1/2)

Step		Register Name	Bit Name	Processing Details	Outline
(1)	Supply TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKO N_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "1b" (_ONWEN is write enable)	Supply TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(2)	Check TSU_n_PCLK (n = 0, 1) supply.	CPG:CPG_CLKM ON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "1b"	Check that TSU_0_PCLK/TSU_1_PCLK was supplied to TSU0/TSU1.
(3)	De-assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "1b" (_WEN is write enable)	De-assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(4)	Check TSU_n_PRESETN (n = 0, 1) de-assertion.	CPG:CPG_RSTM ON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "0b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was de-asserted.
(5)	Release the temperature sensor IP from the power-down state.	Sensor Suspend Register	SOC_TS_EN	Write "1b"	Enable the SOC_TS signal.
			ADC_PD_TS	Write "0b"	Normal ADC operation
			EN_TS	Write "1b"	Normal temperature sensor operation
(6)	Set the operating mode.	Sensor Operation Setting 1 Register	OUTSEL	Write "1b"	Average data output
			ADCS	Write "0b"	Single scan mode
			TRGE	Write "1b"	Enable the ELC trigger.
			ADCT[1:0]	Write "11b"	Set the number of samples for averaging to 8.
		Compare Mode Setting Register	CMPCOND	Write "1b"	Compare mode (within the range setting)
			CMPEN	Write "1b"	Enable the comparison function.
		Sensor Interrupt Enable Register	CMPIE	Write "1b"	Enable the comparison result interrupt.
			ADIE	Write "0b"	Enable the conversion complete interrupt.
(7)	Set the comparison setting values.	Lower Limit Setting Register	LLIM[11:0]	12-bit write	For how to calculate the comparison setting values, see 7.11.7.9 Comparison Setting Value Calculation .
		Upper Limit Setting Register	ULIM[11:0]	12-bit write	For how to calculate the comparison setting values, see 7.11.7.9 Comparison Setting Value Calculation .
(8)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0 (conversion is not in progress).
(9)	Set the ELC trigger.	ICU:EVTSEL5(TSU0)/EVTSEL6(TSU1)	TSU0: Bit position [29:20] TSU1: Bit position [9:0]	Write the ELC no.*1 of the event to be used.	Select the ELC trigger to be used from the ICU register.
(10)	Start conversion.	—	—	—	Receive the ELC trigger from the ICU.

Table 7.11-9 Operation Procedure Example when Comparison Result Interrupt is to be Used (Single Scan, Start by ELC Trigger) (2/2)

Step		Register Name	Bit Name	Processing Details	Outline
(11)	The comparison result interrupt is generated. → Check the interrupt status.	Sensor Interrupt Status Register	CMPF, ADF	Read "11b"	Read the comparison result status and conversion complete status flags.
(12)	Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0b (conversion is not in progress).
(13)	Read the converted code.	Sensor Code Read Register	OUT12BIT_TS[11:0]	12-bit read	Read the 12-bit converted temperature code.
(14)	Calculate the temperature compensation.	See 7.11.7.8 Temperature Compensation Calculation .			
(15)	Clear the interrupt status.	Sensor Interrupt Clear Register	CMPCLR, ADCLR	Write "1b"	Clear the comparison result status and conversion complete status flags.
(16)	Check the interrupt status.	Sensor Interrupt Status Register	CMPF, ADF	Read "00b"	Check the comparison result status and conversion complete status flags were cleared.
(17)	Power down the temperature sensor IP.	Sensor Suspend Register	SOC_TS_EN	Write "0b"	Disable the SOC_TS signal.
			ADC_PD_TS	Write "1b"	Power down the ADC.
			EN_TS	Write "0b"	Power down the temperature sensor.
(18)	Assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "0b" (_WEN is write enable)	Assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(19)	Check TSU_n_PRESETN (n = 0, 1) assertion.	CPG:CPG_RSTMON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "1b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was asserted.
(20)	Stop TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKON_16	CLK9_ONWEN, CLK9_ON(TSU0)/CLK10_ONWEN, CLK10_ON(TSU1)	Write "0b" (_ONWEN is write enable)	Stop TSU_0_PCLK/TSU_1_PCLK to TSU0/TSU1.
(21)	Check stopping of TSU_n_PCLK (n = 0, 1).	CPG:CPG_CLKMON_8	CLK9_MON(TSU0)/CLK10_MON(TSU1)	Read "0b"	Check that TSU_0_PCLK/TSU_1_PCLK supply to TSU0/TSU1 was stopped.

Note 1. For the ELC no., refer to the ICU document "Event_list file".

7.11.8.3 Initialization Procedure

Returning the TSU to its initial state requires asserting the reset signal from the CPG register and then de-asserting the signal.

Follow the procedure in **Table 7.11-10** to perform initialization.

Table 7.11-10 TSU Initialization Procedure

Step	Register Name	Bit Name	Processing Details	Outline
(1) Assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "0b" (_WEN is write enable)	Assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(2) Check TSU_n_PRESETN (n = 0, 1) assertion.	CPG:CPG_RSTMON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "1b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was asserted.
(3) De-assert TSU_n_PRESETN (n = 0, 1).	CPG:CPG_RST_15	RSTB7_WEN, RSTB7(TSU0)/RSTB8_WEN, RSTB8(TSU1)	Write "1b" (_WEN is write enable)	De-assert TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1.
(4) Check TSU_n_PRESETN (n = 0, 1) de-assertion.	CPG:CPG_RSTMON_7	RST8_MON(TSU0)/RST9_MON(TSU1)	Read "0b"	Check that TSU_0_PRESETN/TSU_1_PRESETN of TSU0/TSU1 was de-asserted.

7.11.8.4 Procedure for Changing Operating Mode

The operating mode can be changed while conversion is stopped. **Table 7.11-11** shows the procedure for changing the operating mode after checking the CONV bit of the sensor status register = 0.

Table 7.11-11 Procedure for Changing Operating Mode

Step	Register Name	Bit Name	Processing Details	Outline
(1) Check the conversion status.	Sensor Status Register	CONV	Read "0b"	Read the CONV bit as 0b (conversion is not in progress).
(2) Set the operating mode.	Set the operating mode to change.			

7.11.9 Usage Notes

7.11.9.1 Release from Power-down

Set the SOC_TS_EN bit of the Sensor Suspend Register to 1b at the same time or before setting the EN_TS bit of the same register to 1b and the ADC_PD_TS bit to 0b to release the temperature sensor IP from the power-down state.

7.11.9.2 Conversion Start Trigger

Before making the setting to start conversion by software or an ELC trigger, check that the CONV bit of the sensor status register is 0b.

7.11.9.3 Restarting the Stopped Clock

Restarting the clock after having stopped it requires placing the module in the power-down state once and then releasing it from the power-down state.

- Power down: Sensor Suspend Register

SOC_TS_EN bit = 0b

ADC_PD_TS bit = 1b

EN_TS bit = 0b

- Release from power-down: Sensor Suspend Register

SOC_TS_EN bit = 1b

ADC_PD_TS bit = 0b

EN_TS bit = 1b

SECTION 8 AUDIO

8.1 Audio Overview

8.1.1 Overview

This section describes the data path between audio modules, list of routings, and transfer flow.

8.1.1.1 Features

The audio module is a module connecting ADMAC, SCU, SSIU, SPDIF, PDM, and ADG.

Refer to the section of each module for the detailed function.

8.1.1.2 Block Diagram

Figure 8.1-1 is a block diagram of the audio module. Figure 8.1-2 shows the data paths between SSIU and SCU and ADMAC.

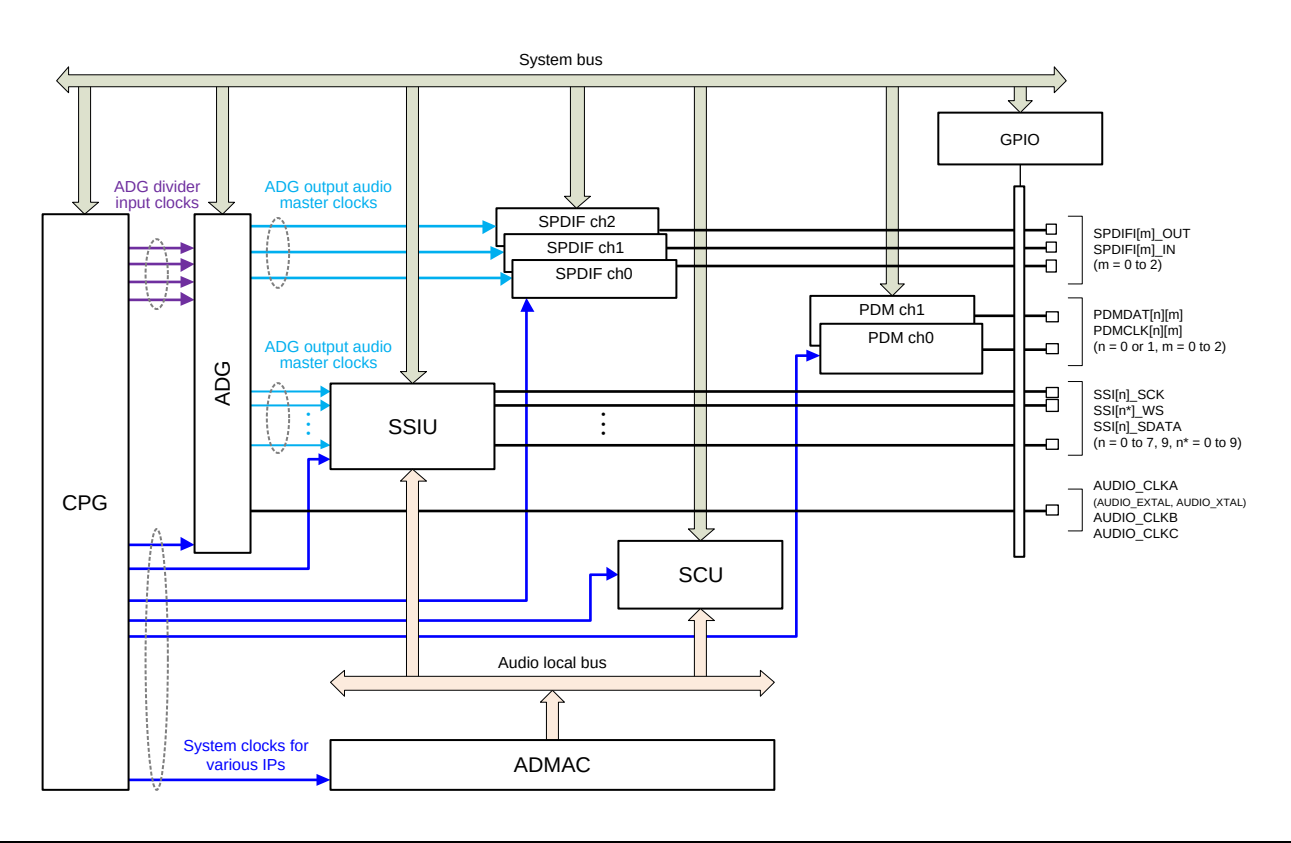


Figure 8.1-1 Audio Module Block Diagram

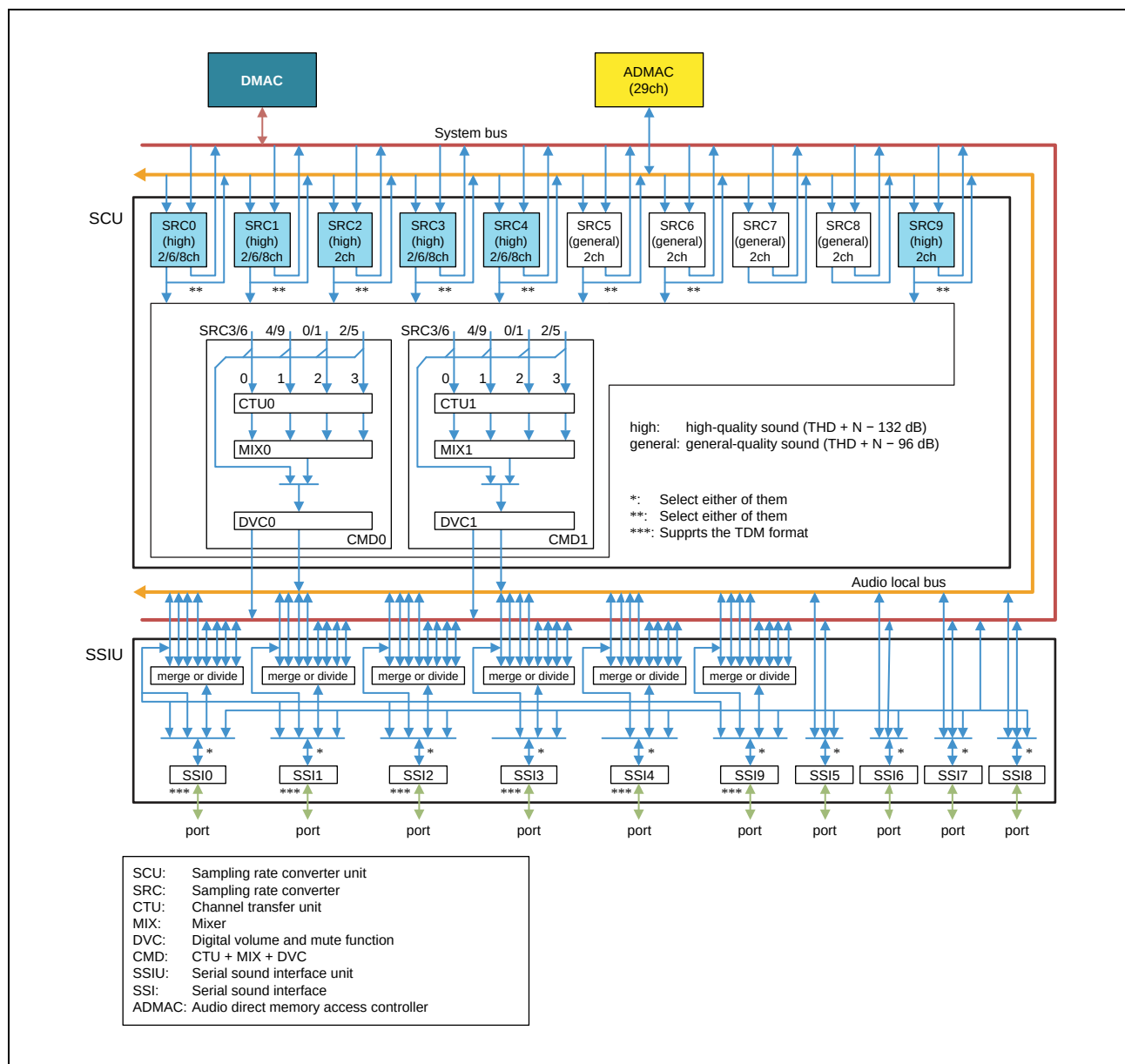


Figure 8.1-2 Data Paths between Audio Modules

8.1.1.3 External Pins

Refer to the following sections for more information about terminal information.

- Section ADG
- Section SSIU
- Section SPDIF
- Section PDM

8.1.1.4 Register Configuration

The audio module does not have a register.

8.1.1.5 Connected Module

Table 8.1-1 Connected Modules

Module Name		Connected Module Name	Function of Connected Module
Audio	Related	SYSTEMBUS	System bus (access the register)
		GPIO	General-purpose I/O, pin function controller
		CPG	Clock pulse generator (Clock ON/OFF, software reset ON/OFF)
		ICU	Interrupt control unit
	Including	SSIU	Serial sound interface unit
		SSI	Serial sound interface
		ADG	Audio clock generator (Output clocks for audio module)
		SCU	Sampling rate converter unit
		SPDIF	SPDIF interface
		PDM	Pulse density modulation
		DMAC	Direct memory access controller
		ADMAC	Audio-DMAC-Peripheral-Peripheral (for audio modules connected to the audio local bus)

8.1.2 Register Description

The audio module does not have a register.

8.1.3 Operation

8.1.3.1 Audio Module Description

1. Audio-DMAC-Peripheral-Peripheral (ADMAC)

The ADMAC controls data transfer between the audio modules (SSIU, SCU) connected to the audio local bus.

- Number of channels: 29 channels
- Data transfer size: Longword (4 bytes)
- Addressing mode: Dual addressing; fixed access size
- Transfer count: Not programmable
- Interrupt processing: None

2. Sampling Rate Converter Unit (SCU)

The SCU has ten SRC modules (six for high-sound-quality type; four for general-sound-quality type) that are useful for synchronization of asynchronous data, which is necessary for data transfer with external memory or external devices. It also provides the functions to change the number of channels, perform mixing, and control the volume.

2-1) Sampling Rate Converter (SRC)

- Asynchronous sampling rate conversion is available
- Supports resolutions up to 24 bits
- High-sound-quality type (THD + N*¹ is – 132 dB) and general-sound-quality type (THD + N*¹ is -96 dB)
- Automatically generates antialiasing filter coefficients
- Four modules support one, two, four, six, or eight channels, and six modules support one or two channels.

Note 1. Total harmonic distortion plus noise

2-2) Channel Transfer Unit (CTU)

- Downmixing and splitter functions
 - Conversion of eight input channels into two output channels
 - Conversion of six input channels into two output channels
 - Conversion of two input channels into four sets of two output channels
 - Conversion of one input channel into eight sets of one output channel
 - No conversion

2-3) Mixer (MIX)

- Mixing (adds) two to four sources into one
- Ratio for adding sources is selectable
- Ratio is dynamically changeable
- Mixing with volume ramp is available (ramp period is selectable)

2-4) Digital Volume and Mute Function (DVC)

- Volume control function including digital volume, volume ramp, and zero-crossing mute
- The digital volume function is specified by a 24-bit fixed-point value within the range from 0 to 8 times (mute or -120 to 18 dB)
- The volume ramp function can be used for soft mute, fade-in, fade-out, or desired volume adjustment
- The volume ramp period can be changed within the sampling range from the 0th to 23rd power of 2
- The zero-crossing mute function silences the sound at the zero-crossing point of the audio data

Remark The CTU, MIX, and DVC functional blocks are collectively called "CMD".

3. Serial Sound Interface Unit (SSIU)

The SSIU, incorporating ten SSI modules, allows independent operation of SSI modules, operation of multiple SSI modules sharing the same serial clock, and connection or split of multichannel data.

- Incorporates ten SSI modules.
- Supports 6 channels/1 sound source with three SSI modules (SSI0, SSI1, and SSI2).
- Supports 8 channels/1 sound source with four SSI modules (SSI0, SSI1, SSI2, and SSI9).
- Operation of multiple modules on the same serial clock (SSI0/SSI1/SSI2/SSI3/SSI9, SSI3/SSI4/SSI9, SSI5/SSI6, or SSI7/SSI8).
- TDM format (Basic Configuration) corresponds to 4, 6, or 8-channel data.
- Handles 8-channel data on the serial bus and 6-channel data in the SoC (TDM extend mode).
- Handles 6-channel data on the serial bus and 8-channel data in the SoC (TDM extend mode).
- Connects monaural or stereo data to output TDM format data (TDM split mode).
- Splits TDM format input data into monaural or stereo data (TDM split mode).
- Connects stereo or multi data to output TDM format data (TDM ex-split mode).
- Splits TDM format input data into stereo or multi data (TDM ex-split mode).
- The frequency range of SCK signal is from 297.3 kHz to 12.5 MHz in master mode, and from 297.3 kHz to 12.5 MHz in slave mode.

4. SPDIF Interface (SPDIF)

The SPDIF is an interface that complies with the IEC60958 standard. Three SPDIF IPs are incorporated in the chip.

- Compliant with the IEC60958 standard (stereo and consumer use modes only)
- Sampling frequency: 32 kHz, 44.1 kHz, 48 kHz
- Audio word size: 16 to 24 bits/sample
- Biphase mark encoding
- Double buffer for data
- Serial data with parity
- Capable of simultaneous transmission and reception

- The receiver autodetects the IEC61937 compressed mode data.

5. Pulse Density Modulation (PDM)

The PDM has 3 PDM function channels and allows independent operation of each channel. 2 PDM IPs are incorporated in the chip.

- Capable of filtering 1-bit digital input data PDMDAT_{nm} (n = 0, 1; m = 0 to 2) and converting them into 20-bit or 16-bit digital data. The bit order is little-endian.
- IP supports stereo microphone (L/R sampling by rising/falling clock edge).
- IP supports sound activity detector.
- Each channel of IP includes programmable filters: 4th order sinc filter, high-pass filter (for suppression of DC bias), correction filter (for sinc passband distortion), half-band decimation filter (for aliasing distortion).
- IP supports programmable and flexible decimation ratios.
- The sinc filter is selectable as first-, second-, third-, fourth-order.
- IP supports DMA operation through APB.
- Each channel of IP has an internal buffer.
- 64 stages
- Capable of storing voice data during low power mode
- Error detection functions can be used for debugging.

6. Audio Clock Generator (ADG)

The ADG selects and supplies the necessary clock for the SSIU, SCU, or SPDIF module. It also divides the frequency of the selected clock and sends it outside the chip.

- Selects the clock signal from the AUDIO_CLKA, AUDIO_CLKB, or AUDIO_CLKC pin or the internal clock and supplies it to the SSIU, SCU, or SPDIF module.
- The frequency of the clock signals from the AUDIO_CLKA, AUDIO_CLKB, and AUDIO_CLKC pins and the internal clock can be divided before use.
- The divided clock can be output through the AUDIO_CLKOUT pin.

8.1.3.2 List of Routings

Data transmission paths in the audio module are shown in **Figure 8.1-3**.

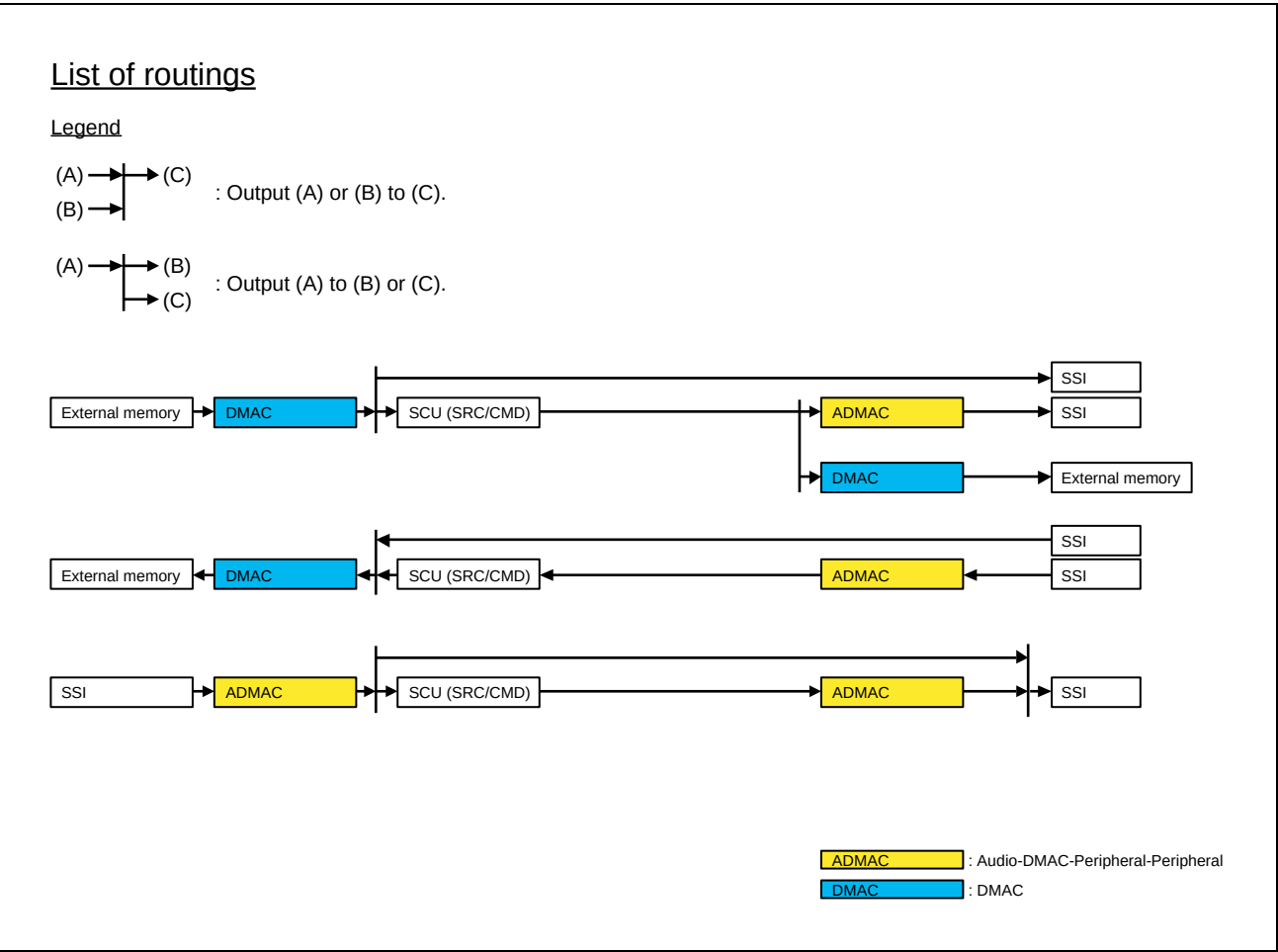


Figure 8.1-3 List of Routings

8.1.3.3 DMAC

The DMAC is used to transfer data between the internal/external memory of the chip and the audio module. **Table 8.1-2** lists the settings of the DMAC source/destination addresses.

SSI_BASE and SCU_BASE represent the base addresses of the SSI and SCU, respectively. For details, refer to section Address Map.

Table 8.1-2 DMAC Source/Destination Address Settings (1/2)

Name of Register	DMASAR/DMADAR	
SSI0-0_BUSIF	SSI_BASE	0000h
SSI0-1_BUSIF		0400h
SSI0-2_BUSIF		0800h
SSI0-3_BUSIF		0C00h
SSI1-0_BUSIF		1000h
SSI1-1_BUSIF		1400h
SSI1-2_BUSIF		1800h
SSI1-3_BUSIF		1C00h
SSI2-0_BUSIF		2000h
SSI2-1_BUSIF		2400h
SSI2-2_BUSIF		2800h
SSI2-3_BUSIF		2C00h
SSI3-0_BUSIF		3000h
SSI3-1_BUSIF		3400h
SSI3-2_BUSIF		3800h
SSI3-3_BUSIF		3C00h
SSI4-0_BUSIF		4000h
SSI4-1_BUSIF		4400h
SSI4-2_BUSIF		4800h
SSI4-3_BUSIF		4C00h
SSI5_BUSIF		5000h
SSI6_BUSIF		6000h
SSI7_BUSIF		7000h
SSI8_BUSIF		8000h
SSI9-0_BUSIF		9000h
SSI9-1_BUSIF		9400h
SSI9-2_BUSIF		9800h
SSI9-3_BUSIF		9C00h

Table 8.1-2 DMAC Source/Destination Address Settings (2/2)

Name of Register	DMASAR/DMADAR	
SRC0in_BUSIF*1	SCU_BASE	0000h
SRC1in_BUSIF*1		0400h
SRC2in_BUSIF*1		0800h
SRC3in_BUSIF*1		0C00h
SRC4in_BUSIF*1		1000h
SRC5in_BUSIF*1		1400h
SRC6in_BUSIF*1		1800h
SRC7in_BUSIF*1		1C00h
SRC8in_BUSIF*1		2000h
SRC9in_BUSIF*1		2400h
SRC0out_BUSIF*2		4000h
SRC1out_BUSIF*2		4400h
SRC2out_BUSIF*2		4800h
SRC3out_BUSIF*2		4C00h
SRC4out_BUSIF*2		5000h
SRC5out_BUSIF*2		5400h
SRC6out_BUSIF*2		5800h
SRC7out_BUSIF*2		5C00h
SRC8out_BUSIF*2		6000h
SRC9out_BUSIF*2		6400h
CMD0out_BUSIF*2		8000h
CMD1out_BUSIF*2		8400h

Note 1. Used only by DMASAR.

Note 2. Used only by DMADAR.

8.1.3.4 Data Format on Audio Local Bus

Figure 8.1-4 shows data format processing on the audio local bus. When write/read data passes through the DMAC, the order of data is rearranged to the appropriate data format in Figure 8.1-4.

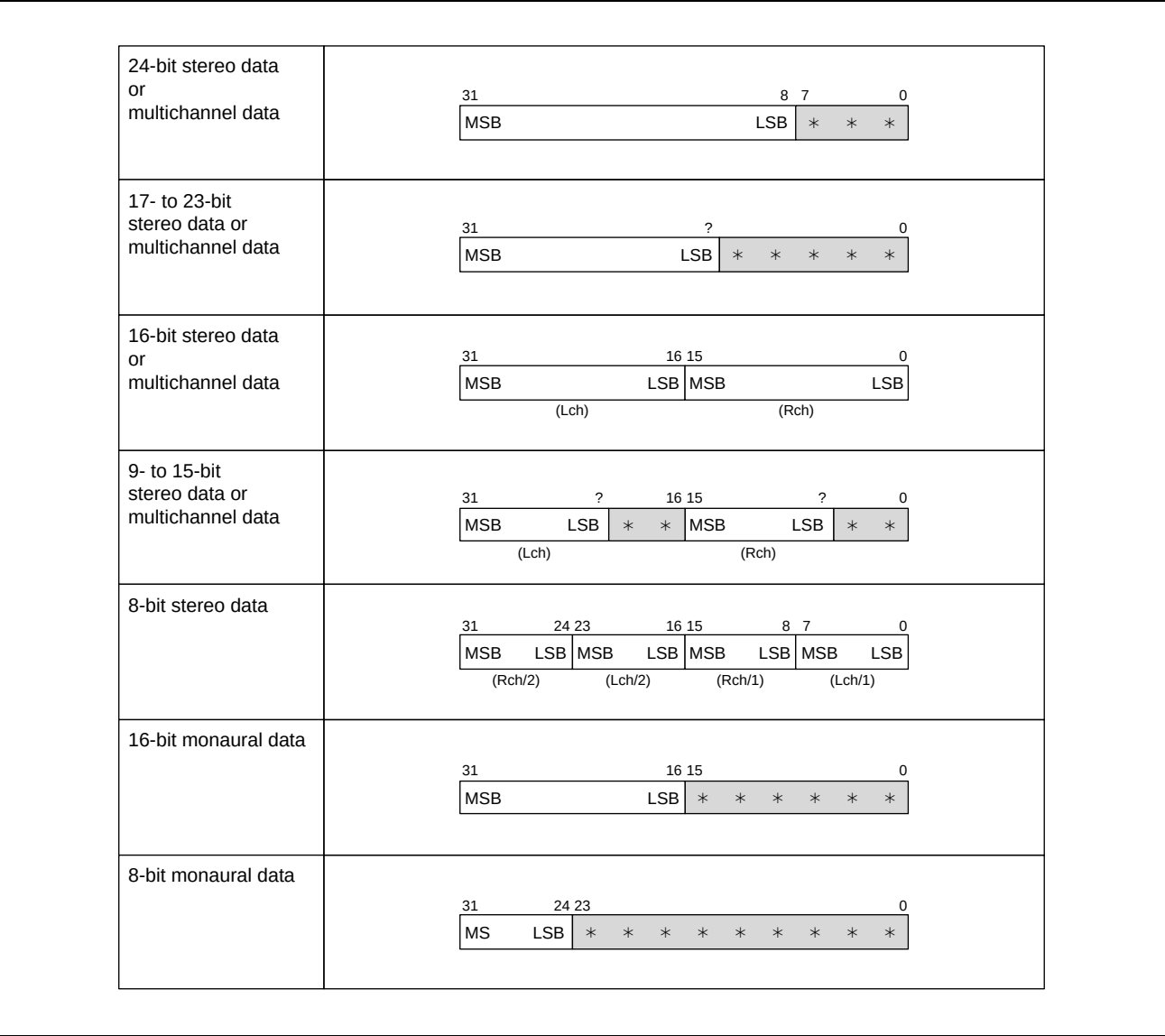


Figure 8.1-4 Data Format

- Remark 1.** Write 0 to the * bits in the figure when writing data. When reading data, ignore the values read from these bits.
- Remark 2.** In the 8-bit stereo data format, (Lch/1) and (Rch/1) indicate the data pair to be processed first and (Lch/2) and (Rch/2) indicate the next data pair to be processed.
- Remark 3.** Only the MSB-first data formats can be used.

8.1.3.5 Rearranging the Order of Data

For the audio modules (SSI, SCU), rearranging the order of data is possible in the channel unit. Places of data in each data format are defined as in **Table 8.1-3**.

Table 8.1-3 Definition of Data Places in Each Data Format (1)

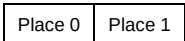
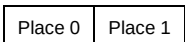
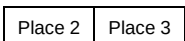
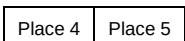
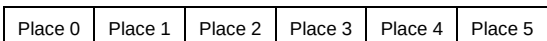
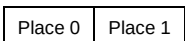
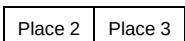
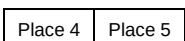
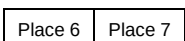
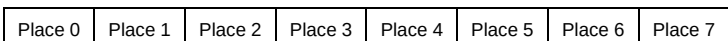
SSI • Stereo (2 channels)	SSI0_WS  SSI_SDAT00 
SSI • Stereo × 3 (6 channels)	SSI0_WS  SSI_SDAT00  SSI_SDAT01  SSI_SDAT02 
SSI • TDM (6 channels)	SSI0_WS  SSI_SDAT00 
SSI • Stereo × 4 (8 channels)	SSI0_WS  SSI_SDAT00  SSI_SDAT01  SSI_SDAT02  SSI_SDAT09 
SSI • TDM (8 channels)	SSI0_WS  SSI_SDAT00 

Table 8.1-3 Definition of Data Places in Each Data Format (2)

BUSIF - Stereo (2 channels) 24 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>*</td></tr><tr><td>04h</td><td>Place 1</td><td>*</td></tr><tr><td>08h</td><td>Place 0</td><td>*</td></tr><tr><td>0Ch</td><td>Place 1</td><td>*</td></tr><tr><td>...</td><td>...</td><td>*</td></tr></table>	00h	Place 0	*	04h	Place 1	*	08h	Place 0	*	0Ch	Place 1	*	...	...	*												
00h	Place 0	*																										
04h	Place 1	*																										
08h	Place 0	*																										
0Ch	Place 1	*																										
...	...	*																										
BUSIF - Stereo (2 channels) 16 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>04h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>08h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>0Ch</td><td>Place 0</td><td>Place 1</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table>	00h	Place 0	Place 1	04h	Place 0	Place 1	08h	Place 0	Place 1	0Ch	Place 0	Place 1	...	...	...												
00h	Place 0	Place 1																										
04h	Place 0	Place 1																										
08h	Place 0	Place 1																										
0Ch	Place 0	Place 1																										
...	...	...																										
BUSIF - Stereo (2 channels) 8 bits	External memory image 310 <table><tr><td>00h</td><td>Place 1</td><td>Place 0</td><td>Place 1</td><td>Place 0</td></tr><tr><td>04h</td><td>Place 1</td><td>Place 0</td><td>Place 1</td><td>Place 0</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr></table>	00h	Place 1	Place 0	Place 1	Place 0	04h	Place 1	Place 0	Place 1	Place 0	...	...	...	...	...												
00h	Place 1	Place 0	Place 1	Place 0																								
04h	Place 1	Place 0	Place 1	Place 0																								
...	...	...	...	...																								
BUSIF - Monaural (1 channel) 8/16 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>*</td></tr><tr><td>04h</td><td>Place 0</td><td>*</td></tr><tr><td>08h</td><td>Place 0</td><td>*</td></tr><tr><td>0Ch</td><td>Place 0</td><td>*</td></tr><tr><td>...</td><td>...</td><td>*</td></tr></table>	00h	Place 0	*	04h	Place 0	*	08h	Place 0	*	0Ch	Place 0	*	...	...	*												
00h	Place 0	*																										
04h	Place 0	*																										
08h	Place 0	*																										
0Ch	Place 0	*																										
...	...	*																										
BUSIF - Multichannel (4 channels) 24 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>*</td></tr><tr><td>04h</td><td>Place 1</td><td>*</td></tr><tr><td>08h</td><td>Place 2</td><td>*</td></tr><tr><td>0Ch</td><td>Place 3</td><td>*</td></tr><tr><td>10h</td><td>Place 0</td><td>*</td></tr><tr><td>14h</td><td>Place 1</td><td>*</td></tr><tr><td>18h</td><td>Place 2</td><td>*</td></tr><tr><td>1Ch</td><td>Place 3</td><td>*</td></tr><tr><td>...</td><td>...</td><td>*</td></tr></table>	00h	Place 0	*	04h	Place 1	*	08h	Place 2	*	0Ch	Place 3	*	10h	Place 0	*	14h	Place 1	*	18h	Place 2	*	1Ch	Place 3	*	...	...	*
00h	Place 0	*																										
04h	Place 1	*																										
08h	Place 2	*																										
0Ch	Place 3	*																										
10h	Place 0	*																										
14h	Place 1	*																										
18h	Place 2	*																										
1Ch	Place 3	*																										
...	...	*																										
BUSIF - Multichannel (4 channels) 16 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>04h</td><td>Place 2</td><td>Place 3</td></tr><tr><td>08h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>0Ch</td><td>Place 2</td><td>Place 3</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table>	00h	Place 0	Place 1	04h	Place 2	Place 3	08h	Place 0	Place 1	0Ch	Place 2	Place 3	...	...	...												
00h	Place 0	Place 1																										
04h	Place 2	Place 3																										
08h	Place 0	Place 1																										
0Ch	Place 2	Place 3																										
...	...	...																										

Table 8.1-3 Definition of Data Places in Each Data Format (3)

BUSIF • Multichannel (6 channels) • 24 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>*</td></tr><tr><td>04h</td><td>Place 1</td><td>*</td></tr><tr><td>08h</td><td>Place 2</td><td>*</td></tr><tr><td>0Ch</td><td>Place 3</td><td>*</td></tr><tr><td>10h</td><td>Place 4</td><td>*</td></tr><tr><td>14h</td><td>Place 5</td><td>*</td></tr><tr><td>18h</td><td>Place 0</td><td>*</td></tr><tr><td>1Ch</td><td>Place 1</td><td>*</td></tr><tr><td>...</td><td>...</td><td>*</td></tr></table>	00h	Place 0	*	04h	Place 1	*	08h	Place 2	*	0Ch	Place 3	*	10h	Place 4	*	14h	Place 5	*	18h	Place 0	*	1Ch	Place 1	*	...	...	*			
00h	Place 0	*																													
04h	Place 1	*																													
08h	Place 2	*																													
0Ch	Place 3	*																													
10h	Place 4	*																													
14h	Place 5	*																													
18h	Place 0	*																													
1Ch	Place 1	*																													
...	...	*																													
BUSIF • Multichannel (6 channels) • 16 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>04h</td><td>Place 2</td><td>Place 3</td></tr><tr><td>08h</td><td>Place 4</td><td>Place 5</td></tr><tr><td>0Ch</td><td>Place 0</td><td>Place 1</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table>	00h	Place 0	Place 1	04h	Place 2	Place 3	08h	Place 4	Place 5	0Ch	Place 0	Place 1	...	...	...															
00h	Place 0	Place 1																													
04h	Place 2	Place 3																													
08h	Place 4	Place 5																													
0Ch	Place 0	Place 1																													
...	...	...																													
BUSIF • Multichannel (8 channels) • 24 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>*</td></tr><tr><td>04h</td><td>Place 1</td><td>*</td></tr><tr><td>08h</td><td>Place 2</td><td>*</td></tr><tr><td>0Ch</td><td>Place 3</td><td>*</td></tr><tr><td>10h</td><td>Place 4</td><td>*</td></tr><tr><td>14h</td><td>Place 5</td><td>*</td></tr><tr><td>18h</td><td>Place 6</td><td>*</td></tr><tr><td>1Ch</td><td>Place 7</td><td>*</td></tr><tr><td>20h</td><td>Place 0</td><td>*</td></tr><tr><td>...</td><td>...</td><td>*</td></tr></table>	00h	Place 0	*	04h	Place 1	*	08h	Place 2	*	0Ch	Place 3	*	10h	Place 4	*	14h	Place 5	*	18h	Place 6	*	1Ch	Place 7	*	20h	Place 0	*	...	...	*
00h	Place 0	*																													
04h	Place 1	*																													
08h	Place 2	*																													
0Ch	Place 3	*																													
10h	Place 4	*																													
14h	Place 5	*																													
18h	Place 6	*																													
1Ch	Place 7	*																													
20h	Place 0	*																													
...	...	*																													
BUSIF • Multichannel (8 channels) • 16 bits	External memory image 310 <table><tr><td>00h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>04h</td><td>Place 2</td><td>Place 3</td></tr><tr><td>08h</td><td>Place 4</td><td>Place 5</td></tr><tr><td>0Ch</td><td>Place 6</td><td>Place 7</td></tr><tr><td>10h</td><td>Place 0</td><td>Place 1</td></tr><tr><td>...</td><td>...</td><td>...</td></tr></table>	00h	Place 0	Place 1	04h	Place 2	Place 3	08h	Place 4	Place 5	0Ch	Place 6	Place 7	10h	Place 0	Place 1	...	...	...												
00h	Place 0	Place 1																													
04h	Place 2	Place 3																													
08h	Place 4	Place 5																													
0Ch	Place 6	Place 7																													
10h	Place 0	Place 1																													
...	...	...																													

8.1.3.6 Clock and Reset

The clock and reset signals for the audio modules (ADG, ADMAC, SCU, SSIU, SPDIF, and PDM) can be switched on or off by setting the registers of the CPG.

8.1.3.6.1 Clock and Reset Configuration

Figure 8.1-5 shows the clock and reset configuration of the audio modules (ADG, ADMAC, SCU, SSIU, SPDIF, and PDM).

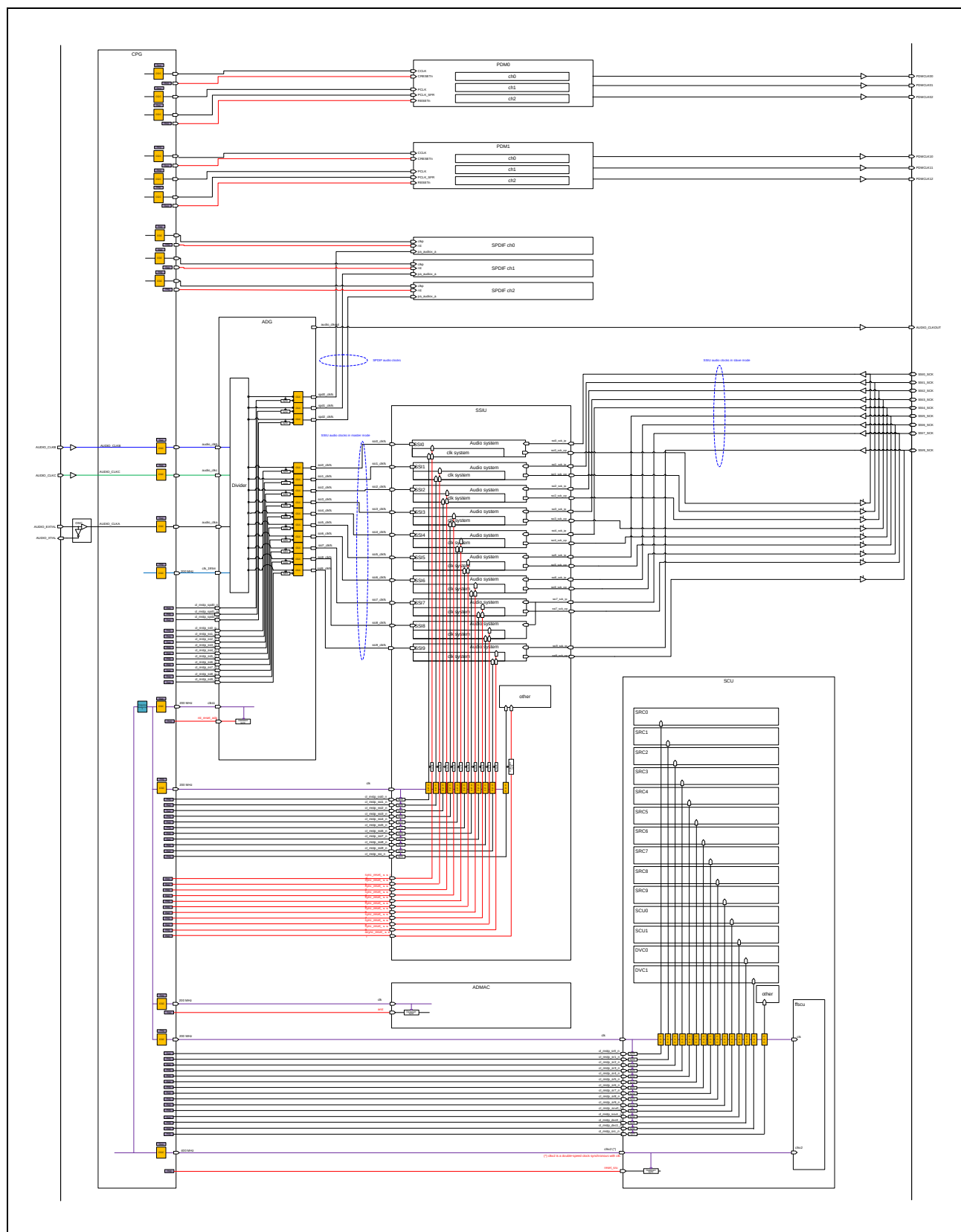


Figure 8.1-5 Clock and Reset Configuration of Audio Module

8.1.3.6.2 System Clock for IP

The system clock for each IP can be supplied or stopped by using the corresponding register of the CPG. The system clock for each IP is also used as a bus clock. Supply the system clock to the target IP before access to the register.

(1) System Clock for ADG

The system clock clks1 input to the ADG can be supplied or stopped by using the CLK9_ON bit of the CPG_CLKON_15 register. The clock is stopped at its initial state after power is on.

Table 8.1-4 ADG System Clock Control

Register Name	Bit	Function
CPG_CLKON_15	CLK9_ON	Supplying or stopping the system clock clks1 input to ADG (0: Stopped; 1: Supplied)

(2) System Clock for ADMAC

The system clock clk input to the ADMAC can be supplied or stopped by using the CLK8_ON bit of the CPG_CLKON_15 register. The clock is stopped at its initial state after power is on.

Table 8.1-5 ADMAC System Clock Control

Register Name	Bit	Function
CPG_CLKON_15	CLK8_ON	Supplying or stopping the system clock clk input to ADMAC (0: Stopped; 1: Supplied)

(3) System Clock for SCU

The system clocks clk and clkx2 input to the SCU can be supplied or stopped by using the CLK6_ON and CLK7_ON bits of the CPG_CLKON_15 register. The clocks are stopped at their initial states after power is on.

Table 8.1-6 SCU System Clock Control

Register Name	Bit	Function
CPG_CLKON_15	CLK6_ON	Supplying or stopping the system clock clk input to SCU (0: Stopped; 1: Supplied)
	CLK7_ON	Supplying or stopping the system clock clkx2 input to SCU (0: Stopped; 1: Supplied)

The system clocks clk and clkx2 input to the SCU are distributed to the given function modules within the SCU via the CGC cells. The system clock for the function modules can be supplied or stopped individually by using the CLK0_ON to CLK14_ON bits of the CPG_CLKON_23 register.

Table 8.1-7 System Clock Control for Internal Modules of SCU

Register Name	Bit	Function
CPG_CLKON_23	CLK0_ON	Supplying or stopping the system clock supplied to the DVC ch0 module of SCU (0: Stopped; 1: Supplied)
	CLK1_ON	Supplying or stopping the system clock supplied to the DVC ch1 module of SCU (0: Stopped; 1: Supplied)
	CLK2_ON	Supplying or stopping the system clock supplied to the CTU/MIX ch0 module of SCU (0: Stopped; 1: Supplied)
	CLK3_ON	Supplying or stopping the system clock supplied to the CTU/MIX ch1 module of SCU (0: Stopped; 1: Supplied)
	CLK4_ON	Supplying or stopping the system clock supplied to the SRC ch0 module of SCU (0: Stopped; 1: Supplied)
	CLK5_ON	Supplying or stopping the system clock supplied to the SRC ch1 module of SCU (0: Stopped; 1: Supplied)
	CLK6_ON	Supplying or stopping the system clock supplied to the SRC ch2 module of SCU (0: Stopped; 1: Supplied)
	CLK7_ON	Supplying or stopping the system clock supplied to the SRC ch3 module of SCU (0: Stopped; 1: Supplied)
	CLK8_ON	Supplying or stopping the system clock supplied to the SRC ch4 module of SCU (0: Stopped; 1: Supplied)
	CLK9_ON	Supplying or stopping the system clock supplied to the SRC ch5 module of SCU (0: Stopped; 1: Supplied)
	CLK10_ON	Supplying or stopping the system clock supplied to the SRC ch6 module of SCU (0: Stopped; 1: Supplied)
	CLK11_ON	Supplying or stopping the system clock supplied to the SRC ch7 module of SCU (0: Stopped; 1: Supplied)
	CLK12_ON	Supplying or stopping the system clock supplied to the SRC ch8 module of SCU (0: Stopped; 1: Supplied)
	CLK13_ON	Supplying or stopping the system clock supplied to the SRC ch9 module of SCU (0: Stopped; 1: Supplied)
	CLK14_ON	Supplying or stopping the system clock supplied to the modules other than SCU (0: Stopped; 1: Supplied)

Table 8.1-8 shows the relationship between the CPG registers (CPG_CLKON_15 and CPG_CLKON_23 registers) and the target function modules for clock control.

Table 8.1-8 SCU System Clock Control and Target Modules

CPG_CLKON_15 register		CPG_CLKON_23 register														Supply or stop clock for module	
CLK6_ON bit	CLK7_ON bit	CLK0_ON bit	CLK1_ON bit	CLK2_ON bit	CLK3_ON bit	CLK4_ON bit	CLK5_ON bit	CLK6_ON bit	CLK7_ON bit	CLK8_ON bit	CLK9_ON bit	CLK10_ON bit	CLK11_ON bit	CLK12_ON bit	CLK13_ON bit		CLK14_ON bit
1	1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Supply the clock for DVC0
1	1	—	1	—	—	—	—	—	—	—	—	—	—	—	—	—	Supply the clock for DVC1
1	1	—	—	1	—	—	—	—	—	—	—	—	—	—	—	—	Supply the clock for CTU0/MIX0
1	1	—	—	—	1	—	—	—	—	—	—	—	—	—	—	—	Supply the clock for CTU1/MIX1
1	1	—	—	—	—	1	—	—	—	—	—	—	—	—	—	—	Supply the clock for SRC0
1	1	—	—	—	—	—	1	—	—	—	—	—	—	—	—	—	Supply the clock for SRC1
1	1	—	—	—	—	—	—	1	—	—	—	—	—	—	—	—	Supply the clock for SRC2
1	1	—	—	—	—	—	—	—	1	—	—	—	—	—	—	—	Supply the clock for SRC3
1	1	—	—	—	—	—	—	—	—	1	—	—	—	—	—	—	Supply the clock for SRC4
1	1	—	—	—	—	—	—	—	—	—	1	—	—	—	—	—	Supply the clock for SRC5
1	1	—	—	—	—	—	—	—	—	—	—	1	—	—	—	—	Supply the clock for SRC6
1	1	—	—	—	—	—	—	—	—	—	—	—	1	—	—	—	Supply the clock for SRC7
1	1	—	—	—	—	—	—	—	—	—	—	—	—	1	—	—	Supply the clock for SRC8
1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	Supply the clock for SRC9
1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1	Supply the clock for other modules*1
1	1	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Stop the clock for DVC0
1	1	—	0	—	—	—	—	—	—	—	—	—	—	—	—	—	Stop the clock for DVC1
1	1	—	—	0	—	—	—	—	—	—	—	—	—	—	—	—	Stop the clock for CTU0/MIX0
1	1	—	—	—	0	—	—	—	—	—	—	—	—	—	—	—	Stop the clock for CTU1/MIX1
1	1	—	—	—	—	0	—	—	—	—	—	—	—	—	—	—	Stop the clock for SRC0
1	1	—	—	—	—	—	0	—	—	—	—	—	—	—	—	—	Stop the clock for SRC1
1	1	—	—	—	—	—	—	0	—	—	—	—	—	—	—	—	Stop the clock for SRC2
1	1	—	—	—	—	—	—	—	0	—	—	—	—	—	—	—	Stop the clock for SRC3
1	1	—	—	—	—	—	—	—	—	0	—	—	—	—	—	—	Stop the clock for SRC4
1	1	—	—	—	—	—	—	—	—	—	0	—	—	—	—	—	Stop the clock for SRC5
1	1	—	—	—	—	—	—	—	—	—	—	0	—	—	—	—	Stop the clock for SRC6
1	1	—	—	—	—	—	—	—	—	—	—	—	0	—	—	—	Stop the clock for SRC7
1	1	—	—	—	—	—	—	—	—	—	—	—	—	0	—	—	Stop the clock for SRC8
1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	0	—	Stop the clock for SRC9
1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	Stop the clock for other modules*1
0	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Stop all clocks

—: Don't care

Note 1. Function modules excluding SRC0 to SRC9, DVC0 and DVC1, CTU0 and CTU1, and MIX0 and MIX1 (including the setting registers, etc.)

(4) System clock for SSIU

The system clock clk input to the SSIU can be supplied or stopped by using the CLK5_ON bit of the CPG_CLKON_15 register. The clock is stopped at its initial state after power is on.

Table 8.1-9 SSIU System Clock Control

Register Name	Bit	Function
CPG_CLKON_15	CLK5_ON	Supplying or stopping the system clock clk input to SSIU (0: Stopped; 1: Supplied)

The system clock clk input to the SSIU is distributed to the given function modules within the SSIU via the CGC cells. The system clock for the function modules can be supplied or stopped individually by using the CLK0_ON to CLK10_ON bits of the CPG_CLKON_24 register.

Table 8.1-10 System Clock Control for Internal Modules of SSIU

Register Name	Bit	Function
CPG_CLKON_24	CLK0_ON	Supplying or stopping the system clock supplied to other modules (0: Stopped; 1: Supplied)
	CLK1_ON	Supplying or stopping the system clock supplied to the SSI ch0 module of SSIU (0: Stopped; 1: Supplied)
	CLK2_ON	Supplying or stopping the system clock supplied to the SSI ch1 module of SSIU (0: Stopped; 1: Supplied)
	CLK3_ON	Supplying or stopping the system clock supplied to the SSI ch2 module of SSIU (0: Stopped; 1: Supplied)
	CLK4_ON	Supplying or stopping the system clock supplied to the SSI ch3 module of SSIU (0: Stopped; 1: Supplied)
	CLK5_ON	Supplying or stopping the system clock supplied to the SSI ch4 module of SSIU (0: Stopped; 1: Supplied)
	CLK6_ON	Supplying or stopping the system clock supplied to the SSI ch5 module of SSIU (0: Stopped; 1: Supplied)
	CLK7_ON	Supplying or stopping the system clock supplied to the SSI ch6 module of SSIU (0: Stopped; 1: Supplied)
	CLK8_ON	Supplying or stopping the system clock supplied to the SSI ch7 module of SSIU (0: Stopped; 1: Supplied)
	CLK9_ON	Supplying or stopping the system clock supplied to the SSI ch8 module of SSIU (0: Stopped; 1: Supplied)
	CLK10_ON	Supplying or stopping the system clock supplied to the SSI ch9 module of SSIU (0: Stopped; 1: Supplied)

Table 8.1-11 shows the relationship between the CPG registers (CPG_CLKON_15 and CPG_CLKON_24 registers) and the target function modules for clock control.

Table 8.1-11 SSIU System Clock Control and Target Modules

CPG_CLKON_15 register	CPG_CLKON_24 register										Supply or stop clock for module
CLK5_ON bit	CLK0_ON bit	CLK1_ON bit	CLK2_ON bit	CLK3_ON bit	CLK4_ON bit	CLK5_ON bit	CLK6_ON bit	CLK7_ON bit	CLK8_ON bit	CLK9_ON bit	
1	1	—	—	—	—	—	—	—	—	—	Supply the system clock for other modules* ¹
1	—	1	—	—	—	—	—	—	—	—	Supply the system clock for SSI ch0 of SSIU
1	—	—	1	—	—	—	—	—	—	—	Supply the system clock for SSI ch1 of SSIU
1	—	—	—	1	—	—	—	—	—	—	Supply the system clock for SSI ch2 of SSIU
1	—	—	—	—	1	—	—	—	—	—	Supply the system clock for SSI ch3 of SSIU
1	—	—	—	—	—	1	—	—	—	—	Supply the system clock for SSI ch4 of SSIU
1	—	—	—	—	—	—	1	—	—	—	Supply the system clock for SSI ch5 of SSIU
1	—	—	—	—	—	—	—	1	—	—	Supply the system clock for SSI ch6 of SSIU
1	—	—	—	—	—	—	—	—	1	—	Supply the system clock for SSI ch7 of SSIU
1	—	—	—	—	—	—	—	—	—	1	Supply the system clock for SSI ch8 of SSIU
1	—	—	—	—	—	—	—	—	—	1	Supply the system clock for SSI ch9 of SSIU
1	0	—	—	—	—	—	—	—	—	—	Stop the system clock for other modules* ¹
1	—	0	—	—	—	—	—	—	—	—	Stop the system clock for SSI ch0 of SSIU
1	—	—	0	—	—	—	—	—	—	—	Stop the system clock for SSI ch1 of SSIU
1	—	—	—	0	—	—	—	—	—	—	Stop the system clock for SSI ch2 of SSIU
1	—	—	—	—	0	—	—	—	—	—	Stop the system clock for SSI ch3 of SSIU
1	—	—	—	—	—	0	—	—	—	—	Stop the system clock for SSI ch4 of SSIU
1	—	—	—	—	—	—	0	—	—	—	Stop the system clock for SSI ch5 of SSIU
1	—	—	—	—	—	—	—	0	—	—	Stop the system clock for SSI ch6 of SSIU
1	—	—	—	—	—	—	—	—	0	—	Stop the system clock for SSI ch7 of SSIU
1	—	—	—	—	—	—	—	—	—	0	Stop the system clock for SSI ch8 of SSIU
1	—	—	—	—	—	—	—	—	—	0	Stop the system clock for SSI ch9 of SSIU
0	—	—	—	—	—	—	—	—	—	—	Stop all system clocks

—: Don't care

Note 1. Function modules excluding SSI ch0 to SSI ch9 (including the setting registers, etc.)

(5) System Clock for SPDIF

The system clock input to ch0 and ch1 of the SPDIF and the system clock input to ch2 of the SPDIF can be supplied or stopped by using the CLK14_ON and CLK15_ON bits of the CPG_CLKON_15 register and the CLK0_ON bit of the CPG_CLKON_16 register, respectively. The clocks are stopped at their initial states after power is on.

Table 8.1-12 System Clock Control for SPDIF Ch0 and Ch1

Register Name	Bit	Function
CPG_CLKON_15	CLK14_ON	Supplying or stopping the system clock input to SPDIF ch0 (0: Stopped; 1: Supplied)
CPG_CLKON_15	CLK15_ON	Supplying or stopping the system clock input to SPDIF ch1 (0: Stopped; 1: Supplied)

Table 8.1-13 System Clock Control for SPDIF Ch2

Register Name	Bit	Function
CPG_CLKON_16	CLK0_ON	Supplying or stopping the system clock input to SPDIF ch2 (0: Stopped; 1: Supplied)

(6) System Clock for PDM

The system clock input to ch0 of the PDM and the system clock input to ch1 of the PDM can be supplied or stopped by using the CLK1_ON to CLK3_ON bits of the CPG_CLKON_16 register and the CLK4_ON to CLK6_ON bits of the CPG_CLKON_16 register, respectively. The clocks are stopped at their initial states after power is on.

Table 8.1-14 System Clock Control for PDM Ch0

Register Name	Bit	Function
CPG_CLKON_16	CLK1_ON	Supplying or stopping the system clock PCLK input to PDM ch0 (0: Stopped; 1: Supplied)
	CLK2_ON	Supplying or stopping the system clock PCLK_SFR input to PDM ch0 (0: Stopped; 1: Supplied)

Table 8.1-15 External I/F Reference Clock Control for PDM Ch0

Register Name	Bit	Function
CPG_CLKON_16	CLK3_ON	Supplying or stopping the external I/F reference clock CCLK for PDM ch0 (0: Stopped; 1: Supplied)

Table 8.1-16 System Clock Control for PDM Ch1

Register Name	Bit	Function
CPG_CLKON_16	CLK4_ON	Supplying or stopping the system clock PCLK input to PDM ch1 (0: Stopped; 1: Supplied)
	CLK5_ON	Supplying or stopping the system clock PCLK_SFR input to PDM ch1 (0: Stopped; 1: Supplied)

Table 8.1-17 External I/F Reference Clock Control for PDM Ch1

Register Name	Bit	Function
CPG_CLKON_16	CLK6_ON	Supplying or stopping the external I/F reference clock CCLK for PDM ch1 (0: Stopped; 1: Supplied)

8.1.3.6.3 ADG Divider Input Clocks

The clocks input to the divider within the ADG can be supplied or stopped by using the CLK10_ON to CLK13_ON bits of the CPG_CLKON_15 register. The clocks are stopped at their initial states after power is on.

After releasing the ADG from the reset state, set the operating mode by the corresponding register of the ADG before supplying the target clock input to the divider.

Table 8.1-18 ADG Internal Divider Input Clock Control

Register Name	Bit	Function
CPG_CLKON_15	CLK10_ON	Supplying or stopping the internal clock (200 MHz) input to ADG (0: Stopped; 1: Supplied)
	CLK11_ON	Supplying or stopping the audio_clka clock input to ADG (0: Stopped; 1: Supplied)
	CLK12_ON	Supplying or stopping the audio_clkb clock input to ADG (0: Stopped; 1: Supplied)
	CLK13_ON	Supplying or stopping the audio_clkc clock input to ADG (0: Stopped; 1: Supplied)

8.1.3.6.4 ADG Output Audio Master Clocks

(1) ADG Output Audio Master Clocks (for SSIU)

Audio master clocks input to the SSIU (ssi[0-9]_clkfs) are generated via the divider within the ADG. The audio master clocks generated by the ADG can be supplied or stopped by using the CLK0_ON to CLK9_ON bits of the CPG_CLKON_22 register. The clocks are stopped at their initial states after power is on.

Table 8.1-19 ADG Output Audio Master Clock Control (for SSIU)

Register Name	Bit	Function
CPG_CLKON_22	CLK0_ON	Supplying or stopping the audio master clock ssi0_clkfs output by ADG for SSI ch0 (0: Stopped; 1: Supplied)
	CLK1_ON	Supplying or stopping the audio master clock ssi1_clkfs output by ADG for SSI ch1 (0: Stopped; 1: Supplied)
	CLK2_ON	Supplying or stopping the audio master clock ssi2_clkfs output by ADG for SSI ch2 (0: Stopped; 1: Supplied)
	CLK3_ON	Supplying or stopping the audio master clock ssi3_clkfs output by ADG for SSI ch3 (0: Stopped; 1: Supplied)
	CLK4_ON	Supplying or stopping the audio master clock ssi4_clkfs output by ADG for SSI ch4 (0: Stopped; 1: Supplied)
	CLK5_ON	Supplying or stopping the audio master clock ssi5_clkfs output by ADG for SSI ch5 (0: Stopped; 1: Supplied)
	CLK6_ON	Supplying or stopping the audio master clock ssi6_clkfs output by ADG for SSI ch6 (0: Stopped; 1: Supplied)
	CLK7_ON	Supplying or stopping the audio master clock ssi7_clkfs output by ADG for SSI ch7 (0: Stopped; 1: Supplied)
	CLK8_ON	Supplying or stopping the audio master clock ssi8_clkfs output by ADG for SSI ch8 (0: Stopped; 1: Supplied)
	CLK9_ON	Supplying or stopping the audio master clock ssi9_clkfs output by ADG for SSI ch9 (0: Stopped; 1: Supplied)

When stopping the audio master clocks, only do so after stopping operation of the target SSIU channels.

The audio master clocks should be supplied after setting the ADG registers related to the selection of the frequency and clock source of the audio master clock and setting the registers related to the operating mode of the target SSIU channels.

(2) ADG Output Audio Master Clocks (for SPDIF)

Audio master clocks input to the SPDIF (spd[0-2]_clkfs) are generated via the divider within the ADG. The audio master clocks generated by the ADG can be supplied or stopped by using the CLK10_ON to CLK12_ON bits of the CPG_CLKON_22 register. The clocks are stopped at their initial states after power is on.

Table 8.1-20 ADG Output Audio Master Clock Control (for SPDIF)

Register Name	Bit	Function
CPG_CLKON_22	CLK10_ON	Supplying or stopping the audio master clock spd0_clkfs output by ADG for SPDIF ch0 (0: Stopped; 1: Supplied)
	CLK11_ON	Supplying or stopping the audio master clock spd1_clkfs output by ADG for SPDIF ch1 (0: Stopped; 1: Supplied)
	CLK12_ON	Supplying or stopping the audio master clock spd2_clkfs output by ADG for SPDIF ch2 (0: Stopped; 1: Supplied)

When stopping the audio master clocks, only do so after stopping operation of the target SPDIF channels.

The audio master clocks should be supplied after setting the ADG registers related to the selection of the frequency and clock source of the audio master clock and setting the registers related to the operating mode of the target SPDIF channels.

8.1.3.6.5 Reset for IP

Initialization of each IP can be controlled by using the corresponding register of the CPG. Be sure to release the target IP from the reset state before access to the register for the IP.

(1) Reset for ADG

For the ADG, initialization of the whole IP can be controlled by using the RSTB14 bit of the CPG_RST_14 register.

Note that the ADG should be reset after stopping operation of the SSIU, SCU, and SPDIF. Otherwise, correct operation is not guaranteed. When releasing the ADG from the reset state, only do so after supplying the system clock for the ADG.

Table 8.1-21 ADG Reset Control

Register Name	Bit	Function
CPG_RST_14	RSTB14	Resetting the whole ADG or releasing the reset (0: Reset; 1: Release from reset)

(2) Reset for ADMAC

For the ADMAC, initialization of the whole IP can be controlled by using the RSTB13 bit of the CPG_RST_14 register.

Note that the ADMAC should be reset after stopping operation of the target channels of the IP involved in the data transfer path. Otherwise, correct operation is not guaranteed. When releasing the ADMAC from the reset state, only do so after supplying the system clock for the ADMAC.

Table 8.1-22 ADMAC Reset Control

Register Name	Bit	Function
CPG_RST_14	RSTB13	Resetting the whole ADMAC or releasing the reset (0: Reset; 1: Release from reset)

(3) Reset for SCU

For the SCU, initialization of the whole IP can be controlled by using the RSTB12 bit of the CPG_RST_14 register.

Note that the SCU should be reset after stopping operation of the target channels of the IP involved in the data transfer path. Otherwise, correct operation is not guaranteed. When releasing the SCU from the reset state, only do so after supplying the system clock for the SCU.

Table 8.1-23 SCU Reset Control

Register Name	Bit	Function
CPG_RST_14	RSTB12	Resetting the whole SCU or releasing the reset (0: Reset; 1: Release from reset)

(4) Reset for SSIU

For the SSIU, initialization of the whole SSIU or initialization of the individual modules within the SSIU can be controlled by using the RSTB1 to RSTB11 bits of the CPG_RST_14 register.

Note that the SSIU should be reset after stopping operation of the target channels of the IP involved in the data transfer path. Otherwise, correct operation is not guaranteed. When releasing the SSIU from the reset state, only do so after supplying the system clock for the SSIU.

Table 8.1-24 SSIU Reset Control

Register Name	Bit	Function
CPG_RST_14	RSTB1	Resetting the modules other than those below (the setting register modules, etc.) or releasing the reset (0: Reset; 1: Release from reset)
	RSTB2	Resetting the SSI ch0 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB3	Resetting the SSI ch1 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB4	Resetting the SSI ch2 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB5	Resetting the SSI ch3 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB6	Resetting the SSI ch4 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB7	Resetting the SSI ch5 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB8	Resetting the SSI ch6 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB9	Resetting the SSI ch7 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB10	Resetting the SSI ch8 module or releasing the reset (0: Reset; 1: Release from reset)
	RSTB11	Resetting the SSI ch9 module or releasing the reset (0: Reset; 1: Release from reset)

Table 8.1-25 lists the RSTB1 to RSTB11 bits of the CPG_RST_14 register and the system clock settings required for resetting the target channels or releasing them from the reset state.

Table 8.1-25 SSIU Reset Control Registers and Required System Clock Settings

Clock setting required for reset or reset release operation by RSTB1 to RSTB11 of CPG_RST_14 register	CPG_CLKON_15 register	CPG_CLKON_24 register										
	CLK5_ON bit	CLK0_ON bit	CLK1_ON bit	CLK2_ON bit	CLK3_ON bit	CLK4_ON bit	CLK5_ON bit	CLK6_ON bit	CLK7_ON bit	CLK8_ON bit	CLK9_ON bit	CLK10_ON bit
RSTB1	1	1	—	—	—	—	—	—	—	—	—	—
RSTB2	1	—	1	—	—	—	—	—	—	—	—	—
RSTB3	1	—	—	1	—	—	—	—	—	—	—	—
RSTB4	1	—	—	—	1	—	—	—	—	—	—	—
RSTB5	1	—	—	—	—	1	—	—	—	—	—	—
RSTB6	1	—	—	—	—	—	1	—	—	—	—	—
RSTB7	1	—	—	—	—	—	—	1	—	—	—	—
RSTB8	1	—	—	—	—	—	—	—	1	—	—	—
RSTB9	1	—	—	—	—	—	—	—	—	1	—	—
RSTB10	1	—	—	—	—	—	—	—	—	—	1	—
RSTB11	1	—	—	—	—	—	—	—	—	—	—	1

(5) Reset for SPDIF

For ch0 of the SPDIF, initialization of the whole IP can be controlled by using the RSTB15 bit of the CPG_RST_14 register. For ch1 and ch2 of the SPDIF, initialization of the whole IP can be controlled by using the RSTB0 and RSTB1 bits of the CPG_RST_15 register.

Note that the SPDIF should be reset after stopping operation of the target channels of the IP involved in the data transfer path. Otherwise, correct operation is not guaranteed. When releasing the SPDIF from the reset state, only do so after supplying the system clock for the SPDIF.

Table 8.1-26 SPDIF Reset Control (1)

Register Name	Bit	Function
CPG_RST_14	RSTB15	Resetting SPDIF ch0 or releasing the reset (0: Reset; 1: Release from reset)

Table 8.1-26 SPDIF Reset Control (2)

Register Name	Bit	Function
CPG_CLKON_15	RSTB0	Resetting SPDIF ch1 or releasing the reset (0: Reset; 1: Release from reset)
	RSTB1	Resetting SPDIF ch2 or releasing the reset (0: Reset; 1: Release from reset)

(6) Reset for PDM

For ch0 of the PDM, initialization of the whole IP can be controlled by using the RSTB2 and RSTB3 bits of the CPG_RST_15 register. For ch1 of the PDM, initialization of the whole IP can be controlled by using the RSTB4 and RSTB5 bits of the CPG_RST_15 register.

Note that the PDM should be reset after stopping operation of the target channels of the IP involved in the data transfer path. Otherwise, correct operation is not guaranteed. When releasing the PDM from the reset state, only do so after supplying the system clock for the PDM.

Table 8.1-27 PDM Reset Control (1)

Register Name	Bit	Function
CPG_RST_15	RSTB2	Resetting the PCK/PCLK_SFR clock circuit for PDM ch0 or releasing the reset (0: Reset; 1: Release from reset)
	RSTB3	Resetting the CCLK clock circuit for PDM ch0 or releasing the reset (0: Reset; 1: Release from reset)

Table 8.1-27 PDM Reset Control (2)

Register Name	Bit	Function
CPG_CLKON_15	RSTB4	Resetting the PCK/PCLK_SFR clock circuit for PDM ch1 or releasing the reset (0: Reset; 1: Release from reset)
	RSTB5	Resetting the CCLK clock circuit for PDM ch1 or releasing the reset (0: Reset; 1: Release from reset)

8.1.3.6.6 Module Reset for SCU

Table 8.1-28 lists the software reset registers of the SCU. By using these registers, the SCU can control initialization of the target function modules and the associated registers (excluding the software reset registers).

Table 8.1-28 SCU Internal Software Reset Control

Register Name	Bit	Function
SRCm_SWRSR (m = 0 to 9)	SWRST	Resetting the circuit of the corresponding SRC channel within SCU and the associated register (excluding SRC[m]_SWRSR) or releasing the reset (0: Reset; 1: Release from reset)
CTUn_SWRSR (n = 00, 01, 02, 03, 10, 11, 12, or 13)	SWRST	Resetting the circuit of the corresponding CTU channel within SCU and the associated register (excluding CTU[n]_SWRSR) or releasing the reset (0: Reset; 1: Release from reset)
MIXp_SWRSR (p = 0 or 1)	SWRST	Resetting the circuit of the corresponding MIX channel within SCU and the associated register (excluding MIX[p]_SWRSR) or releasing the reset (0: Reset; 1: Release from reset)
DVCp_SWRSR (p = 0 or 1)	SWRST	Resetting the circuit of the corresponding DVC channel within SCU and the associated register (excluding DVC[p]_SWRSR) or releasing the reset (0: Reset; 1: Release from reset)

For reset or reset release operations using the software reset registers, the system clock must be supplied to the target function modules beforehand. **Table 8.1-29** lists the relationship between the software reset registers of the SCU and the clock settings required for initialization control.

Table 8.1-29 SCU Internal Software Reset Control Registers and Required Clock Settings

Software reset register within SCU	CPG_CLKON_15 register		CPG_CLKON_23 register														
	CLK6_ON bit	CLK7_ON bit	CLK0_ON bit	CLK1_ON bit	CLK2_ON bit	CLK3_ON bit	CLK4_ON bit	CLK5_ON bit	CLK6_ON bit	CLK7_ON bit	CLK8_ON bit	CLK9_ON bit	CLK10_ON bit	CLK11_ON bit	CLK12_ON bit	CLK13_ON bit	CLK14_ON bit
DVC0_SWRSR	1	1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	1
DVC1_SWRSR	1	1	—	1	—	—	—	—	—	—	—	—	—	—	—	—	1
CTU00_SWRSR	1	1	—	—	1	—	—	—	—	—	—	—	—	—	—	—	1
CTU01_SWRSR																	
CTU02_SWRSR																	
CTU03_SWRSR																	
MIX0_SWRSR																	
CTU10_SWRSR	1	1	—	—	—	1	—	—	—	—	—	—	—	—	—	—	1
CTU11_SWRSR																	
CTU12_SWRSR																	
CTU13_SWRSR																	
MIX1_SWRSR																	
SRC0_SWRSR	1	1	—	—	—	—	1	—	—	—	—	—	—	—	—	—	1
SRC1_SWRSR	1	1	—	—	—	—	—	1	—	—	—	—	—	—	—	—	1
SRC2_SWRSR	1	1	—	—	—	—	—	—	1	—	—	—	—	—	—	—	1
SRC3_SWRSR	1	1	—	—	—	—	—	—	—	1	—	—	—	—	—	—	1
SRC4_SWRSR	1	1	—	—	—	—	—	—	—	—	1	—	—	—	—	—	1
SRC5_SWRSR	1	1	—	—	—	—	—	—	—	—	—	1	—	—	—	—	1
SRC6_SWRSR	1	1	—	—	—	—	—	—	—	—	—	—	1	—	—	—	1
SRC7_SWRSR	1	1	—	—	—	—	—	—	—	—	—	—	—	1	—	—	1
SRC8_SWRSR	1	1	—	—	—	—	—	—	—	—	—	—	—	—	1	—	1
SRC9_SWRSR	1	1	—	—	—	—	—	—	—	—	—	—	—	—	—	1	1

8.1.3.7 Procedure for Releasing IP from Module Standby

The following describes the procedure for releasing each IP from module standby.

8.1.3.7.1 Procedure for Releasing ADG from Module Standby

Figure 8.1-6 shows the procedure for releasing the ADG from module standby.

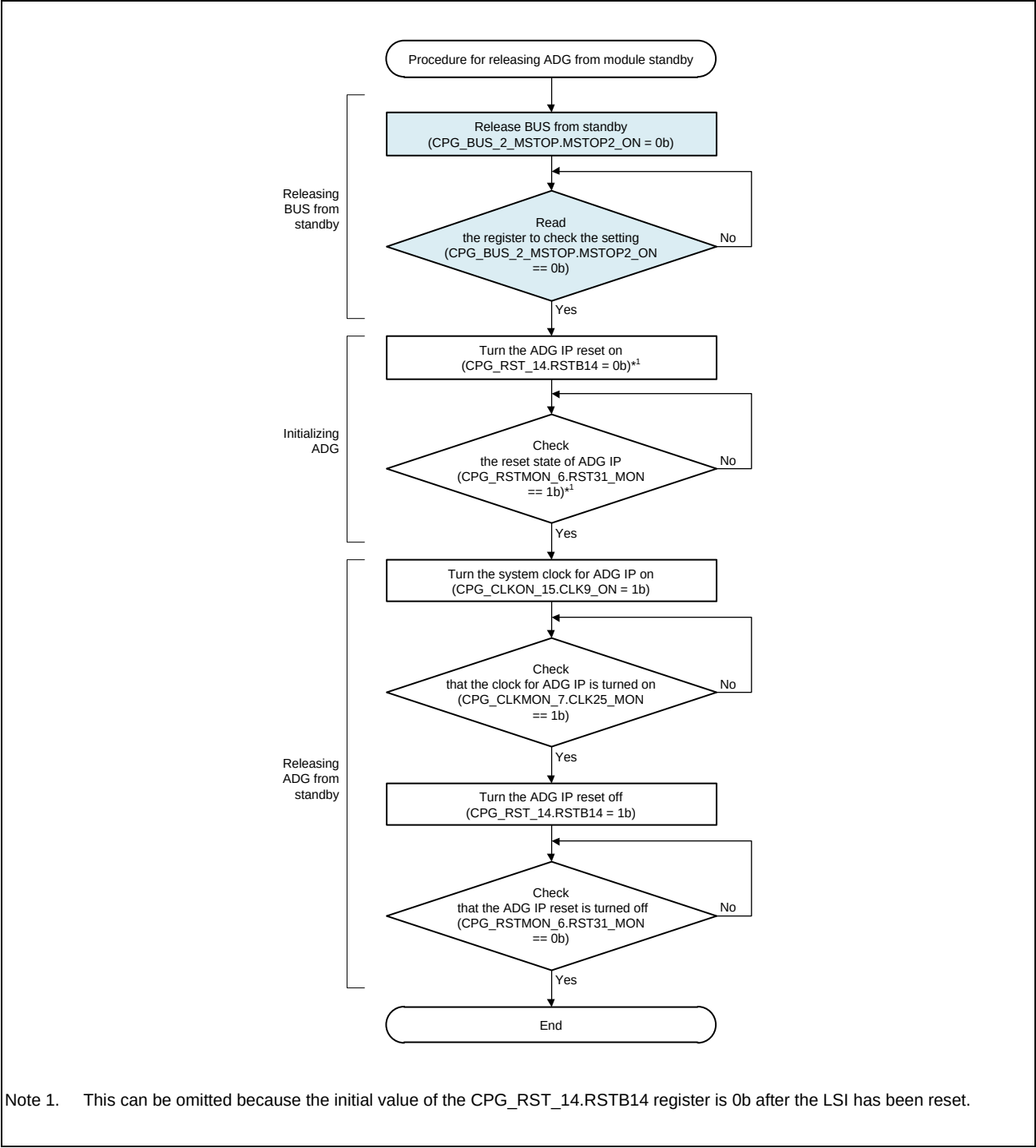


Figure 8.1-6 Procedure for Releasing ADG from Module Standby

8.1.3.7.2 Procedure for Releasing ADMAC from Module Standby

Figure 8.1-7 shows the procedure for releasing the ADMAC from module standby.

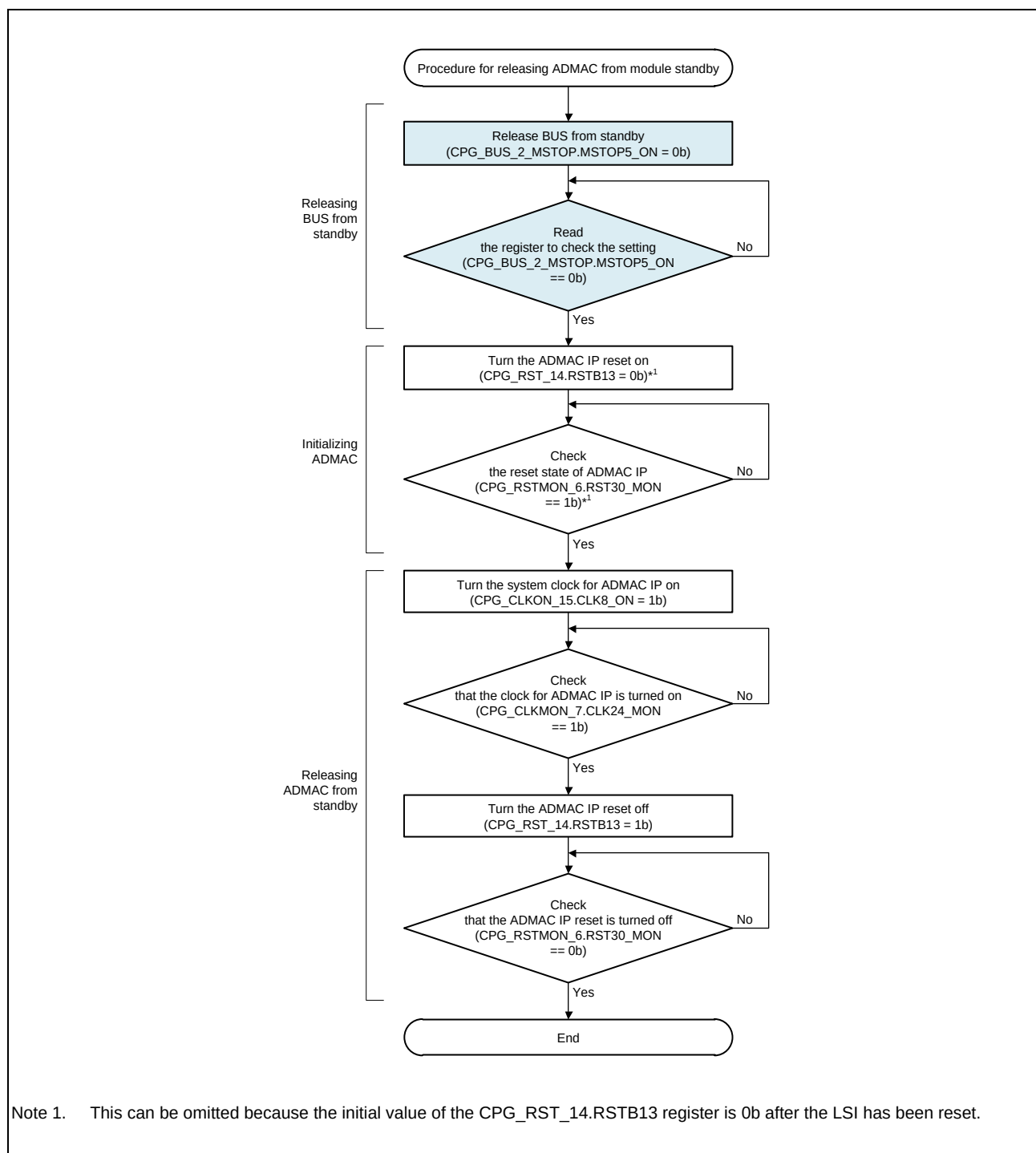


Figure 8.1-7 Procedure for Releasing ADMAC from Module Standby

8.1.3.7.3 Procedure for Releasing SCU from Module Standby

Figure 8.1-8 shows the procedure for releasing the SCU from module standby.

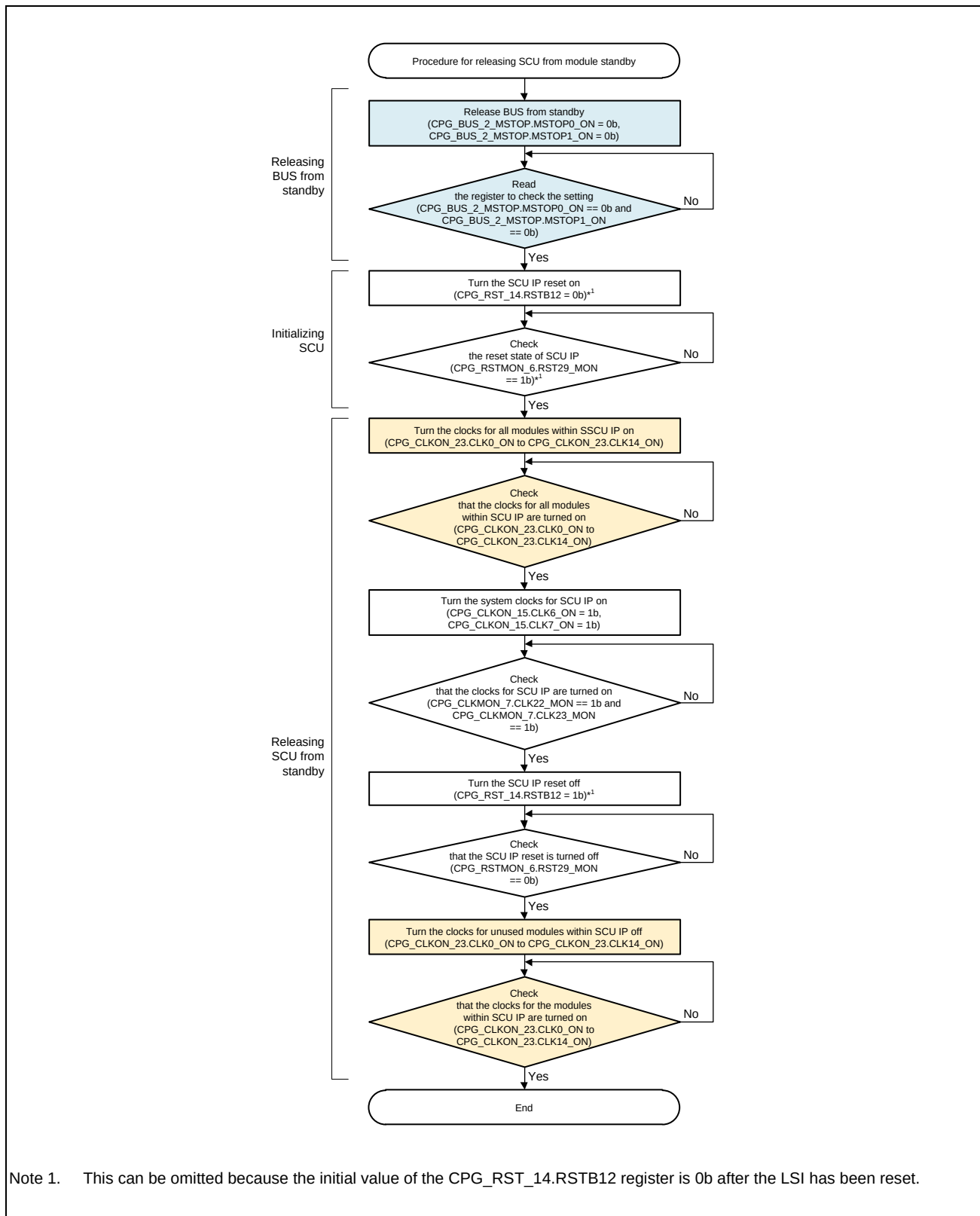
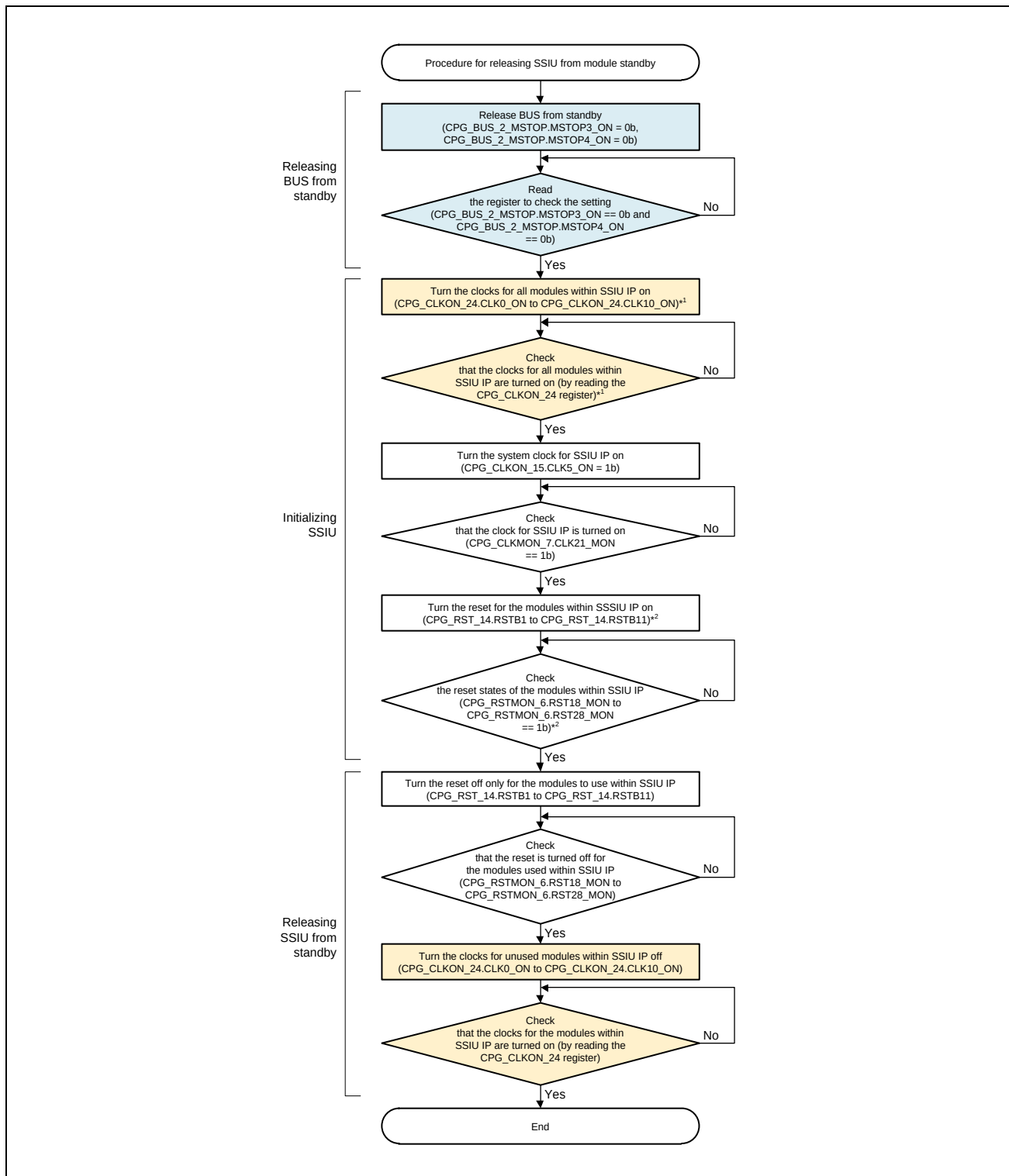


Figure 8.1-8 Procedure for Releasing SCU from Module Standby

8.1.3.7.4 Procedure for Releasing SSIU from Module Standby

Figure 8.1-9 shows the procedure for releasing the SSIU from module standby.



(Continued on next page)

(Continued from previous page)

Note 1.	This can be omitted because the initial value of the CPG_CLKON_24.CLK0_ON to CPG_CLKON_24.CLK10_ON registers is 1b after the LSI has been reset.
Note 2.	This can be omitted because the initial value of the CPG_RST_14.RSTB1 to CPG_RST_14.RSTB11 registers is 0b after the LSI has been reset.

Figure 8.1-9 Procedure for Releasing SSIU from Module Standby

8.1.3.7.5 Procedure for Releasing SPDIF from Module Standby

Figure 8.1-10 shows the procedure for releasing the SPDIF from module standby, taking SPDIF IP ch0 as an example.

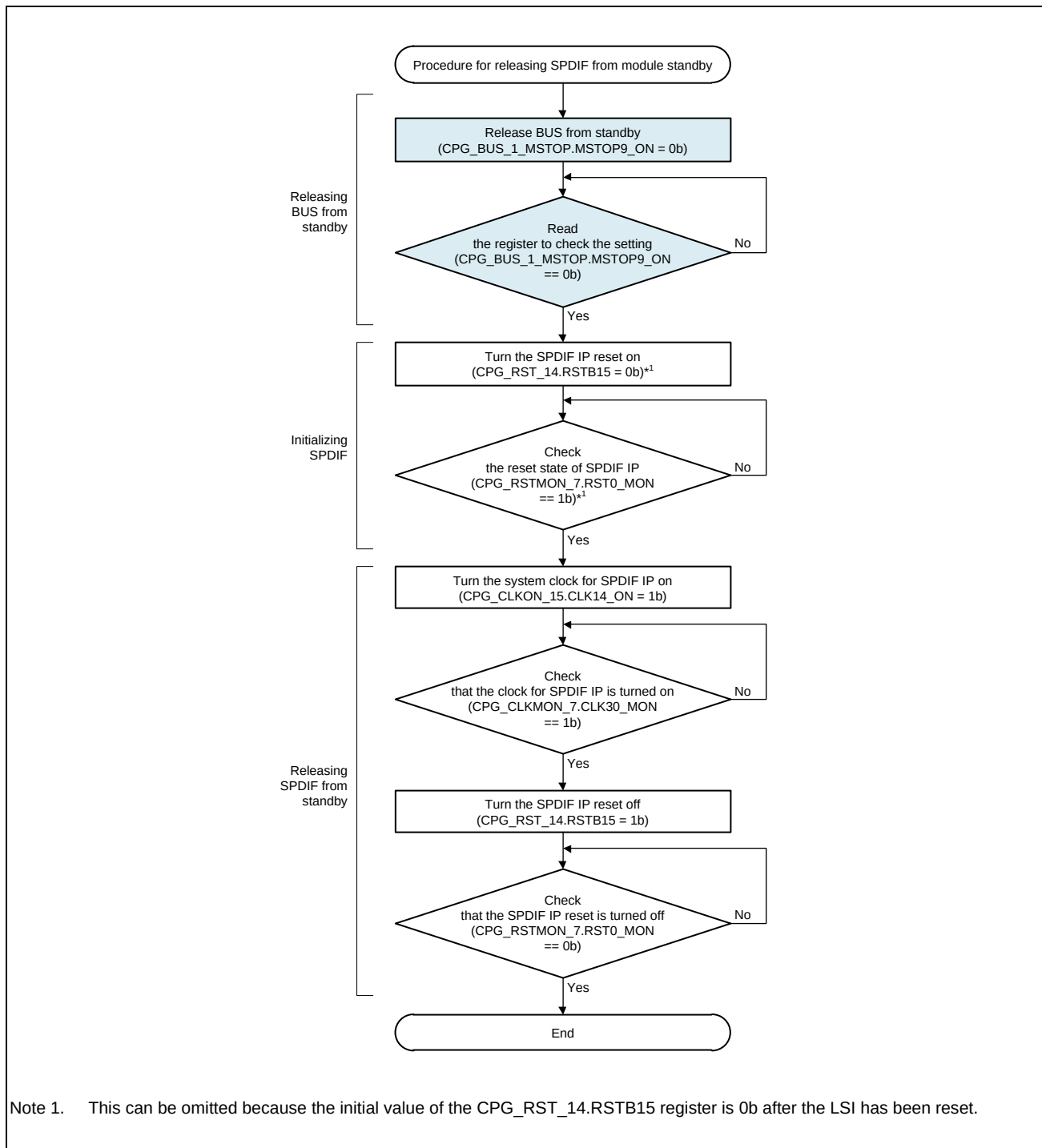


Figure 8.1-10 Procedure for Releasing SPDIF from Module Standby

8.1.3.7.6 Procedure for Releasing PDM from Module Standby

Figure 8.1-11 shows the procedure for releasing the PDM from module standby, taking PDM IP ch0 as an example.

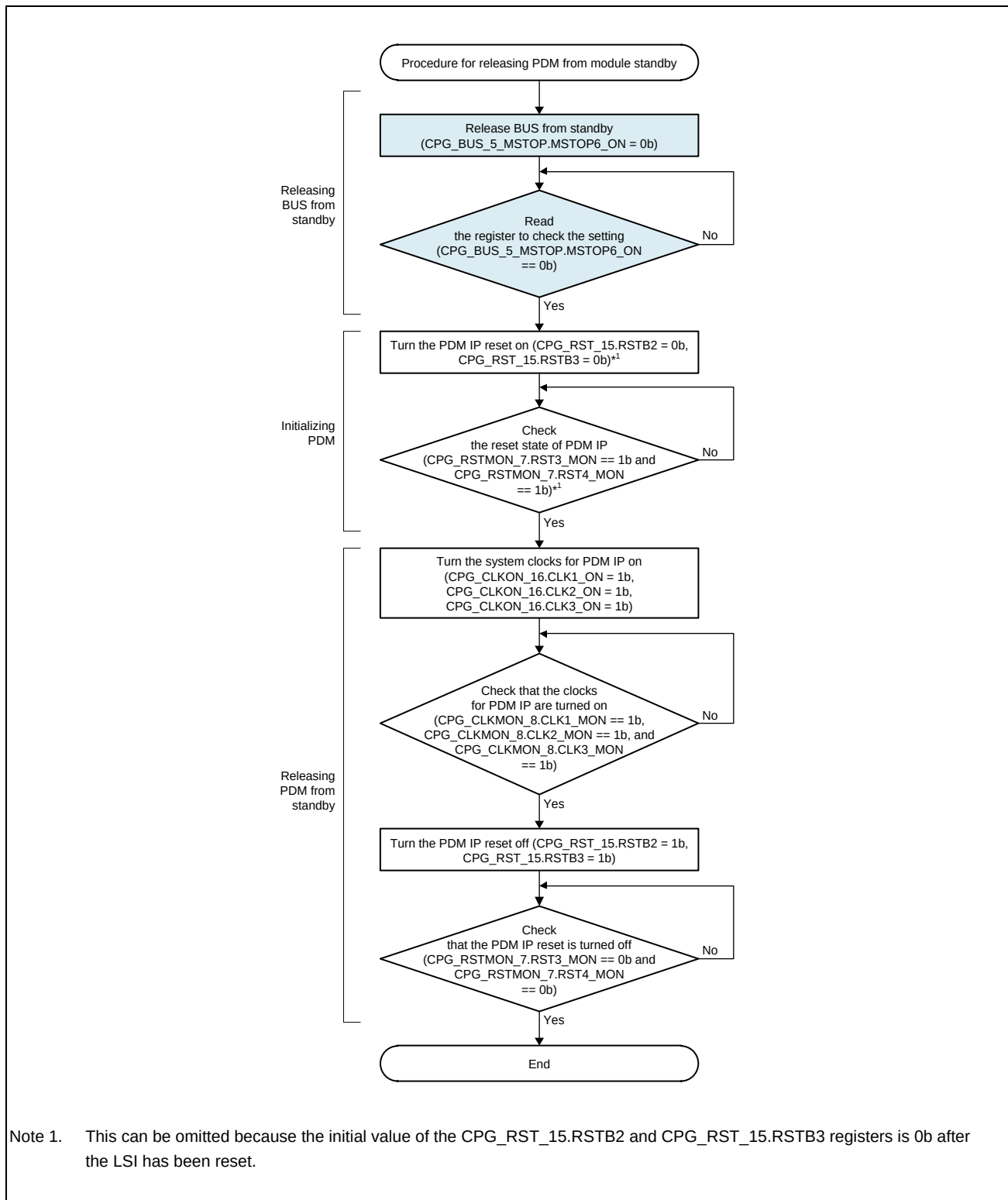


Figure 8.1-11 Procedure for Releasing PDM from Module Standby

8.1.3.8 SSIU/SCU Setup Procedure Overview

Before setting up the audio related modules, you must set the pin functions and release the IPs to use from module standby.

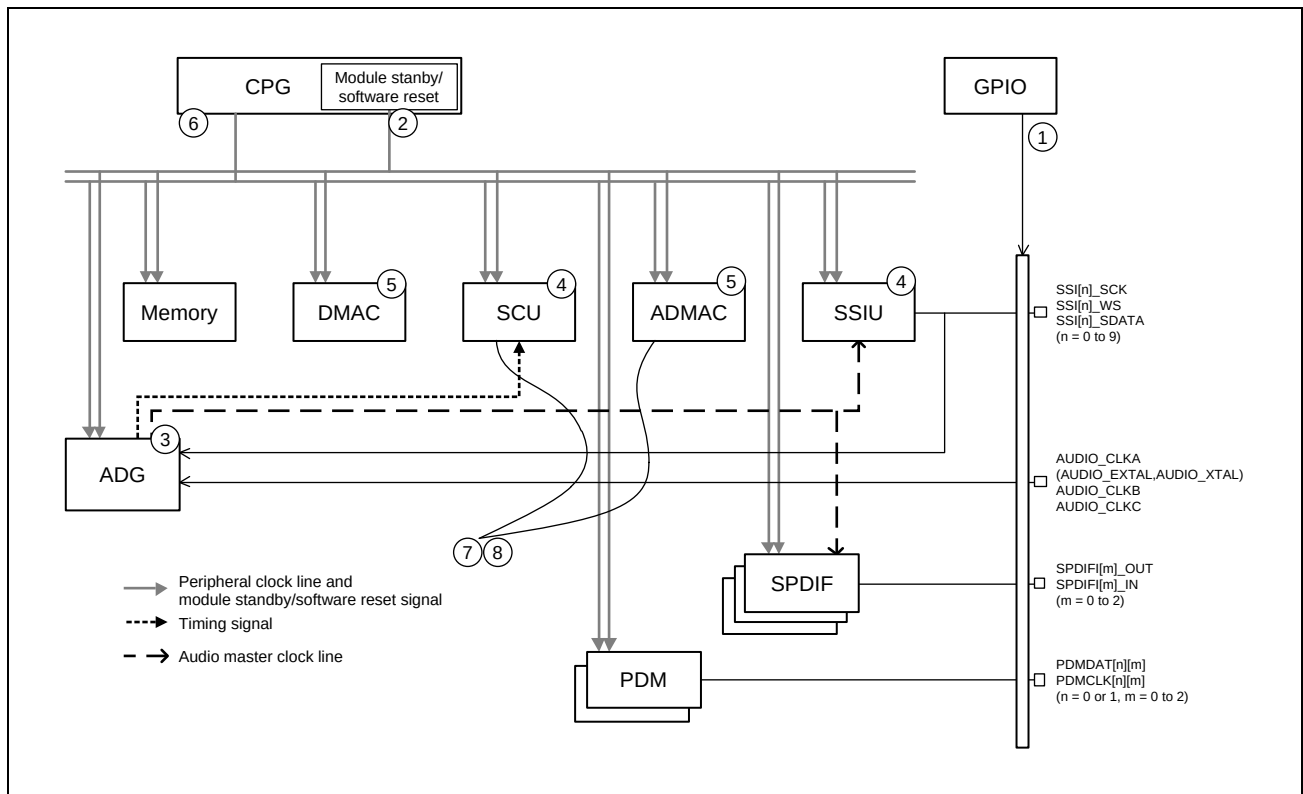


Figure 8.1-12 SSIU/SCU Setup Procedure Diagram

The following is the procedure in outline.

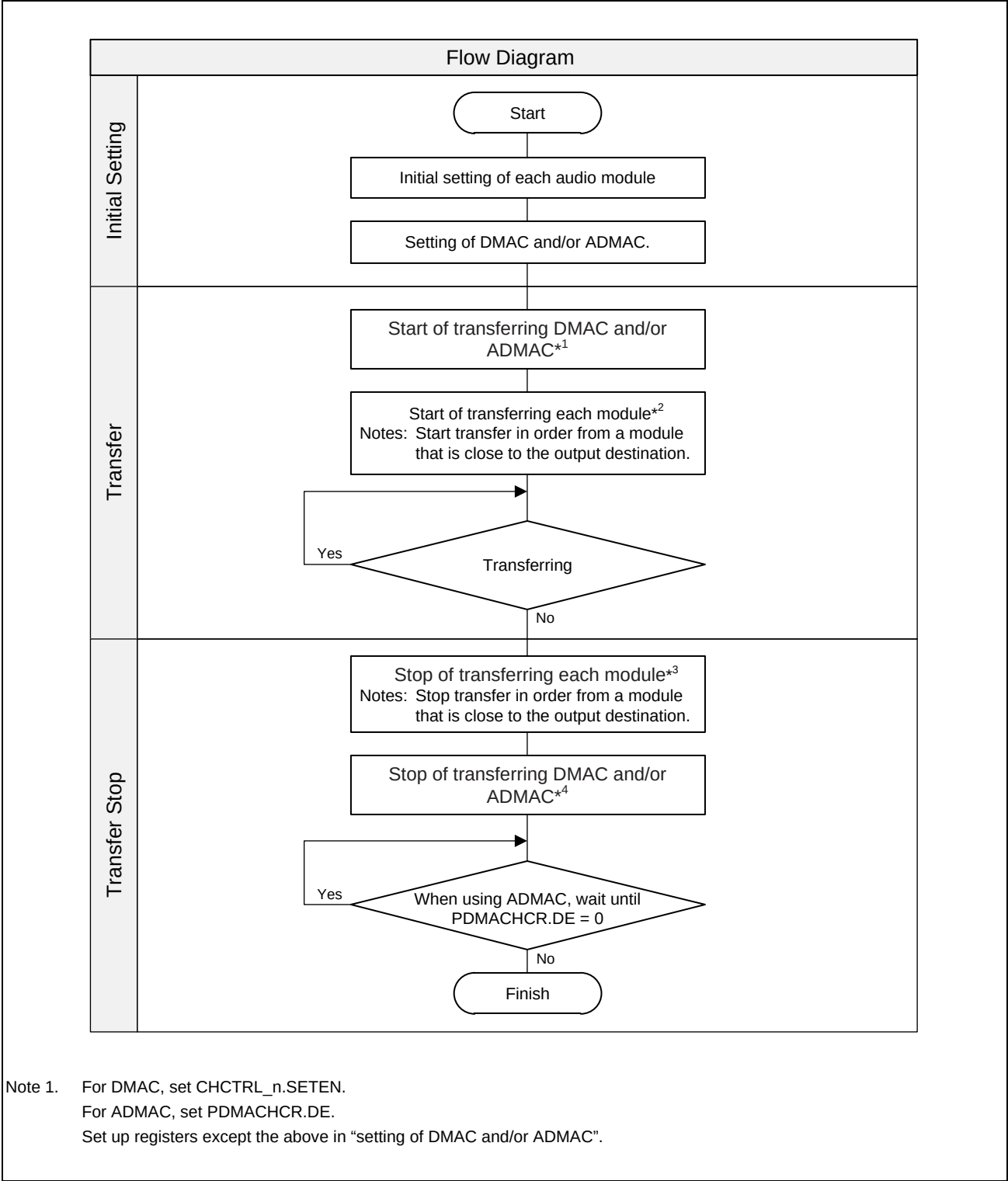
1. Set the pin functions (GPIO).
2. Release module standby (software reset).
3. Set ADG mode.
4. Set SCU/SSIU mode.
5. Set DMAC/ADMAC mode.
6. Turn the audio clocks on (CPG).
Supply the ADG frequency-divided input audio clocks (on*¹) and supply the ADG output audio master clocks for the SSIU (on*²).
7. Initiate transfer between the DMAC/ADMAC and SCU and SSIU.
8. Stop transfer between the DMAC/ADMAC and SCU and SSIU.
9. Restart transfer between the DMAC/ADMAC and SCU and SSIU.

When restarting transfer at step 9, follow the procedure from step 7. You do not need to follow the order of steps 3 to 5 to make initial settings for the audio related modules. Regarding the mode setting registers of the individual IPs, refer to the functional specifications of these IPs.

- Note 1.** CLK9_ON to CLK13_ON of the CPG_CLKON_15 register. Only the clocks required for operation are turned on.
- Note 2.** CLK0_ON to CLK9_ON of the CPG_CLKON_22 register. Only the clocks required for operation are turned on.

8.1.3.8.1 Transfer Flow

When data transfer is performed between the audio modules, use the DMAC and ADMAC to perform settings in accord with the transfer flow shown in **Figure 8.1-13**.



(Continued on next page)

(Continued from previous page)

- Note 2. For SSI (except for the case when TDM split mode, SSI0, SSI1, SSI2, or SSI9, and SSI3 or SSI4 are used at the same time), when transmitting (from SSI to the external device), set SSIsq/r_CONTROL.start after setting SSICR.EN/DMEN). When receiving (from the external device to SSI), set SSICR.EN/DMEN after setting SSIsq/r_CONTROL.start.
For SSI (in the case of TDM split mode) regardless of transmission and reception, set SSIsq/r_CONTROL.start after setting SSICR.EN/DMEN.
- For SSI (when using SSI0, SSI1, SSI2, or SSI9 and SSI3 or SSI4 at the same time), when transmitting (from SSI to the external device), set SSIsq/r_CONTROL.start after setting SSI_CONTROL. When receiving (from the external device to SSI), set SSI_CONTROL after setting SSIsq/r_CONTROL.start. (Set SSICR.DMEN at the time of the initial setting.)
For SCU, Set SRCm_CONTROL and CMDn_CONTROL.
- Set up registers except the above in "initial setting of each audio module".
- Note 3. For SSI (except for the case when TDM split mode, SSI0, SSI1, SSI2, or SSI9, and SSI3 or SSI4 are used at the same time), when transmitting (from SSI to the external device), clear SSIsq/r_CONTROL.start after clearing SSICR.EN/DMEN. When receiving (from the external device to SSI), clear SSICR.EN/DMEN after clearing SSIsq/r_CONTROL.start.
- For SSI (in the case of TDM split mode), regardless of transmission and reception, clear all SSIsq/r_CONTROL.start of the relevant channels to clear SSICR.EN/DMEN.
- For SSI (when using SSI0, SSI1, SSI2, or SSI9 and SSI3 or SSI4 at the same time), when transmitting (from SSI to the external device), clear SSIsq/r_CONTROL.start after clearing SSICR.DMEN of each module and SSI_CONTROL. When receiving (from the external device to SSI), clear SSICR.DMEN of each module and SSI_CONTROL after clearing SSIsq/r_CONTROL.start.
- For SCU, clear SRCm_CONTROL and CMDn_CONTROL.
- Because transfer is stopped in order from a module that is close to the output destination, overflow may occur in a preceding module. Be careful.
- Note 4. For ADMAC, clear PDMACHCR.DE. At this time, other bits must not be changed.
For DMAC, clear CHCTRL_n.SETEN. Clear processing of DMAC can be cleared anywhere in a TransferStop sequence.

Figure 8.1-13 Transfer Flow

8.1.3.9 SPDIF Setup Procedure Overview

Before setting up the audio related modules, you must set the pin functions and release the IPs to use from module standby.

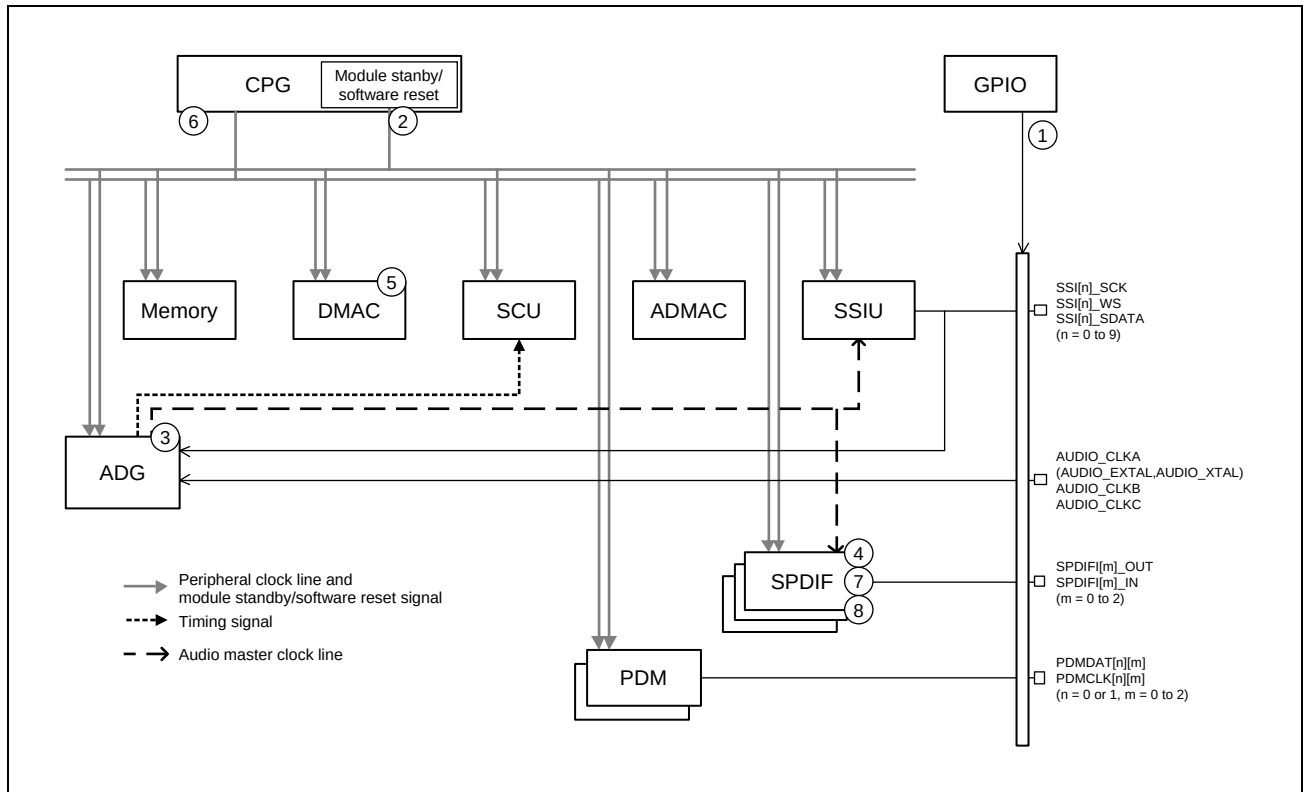


Figure 8.1-14 SPDIF Setup Procedure Diagram

The following is the procedure in outline.

1. Set the pin functions (GPIO).
2. Release module standby (software reset).
3. Set the audio clocks (ADG).
4. Set SPDIF mode.
5. Set DMAC mode.
6. Turn the audio clocks on (CPG).
Supply the ADG frequency-divided input audio clocks (on*¹) and supply the ADG output audio master clocks for the SPDIF (on*²).
7. Initiate transfer between the DMAC and SPDIF.
8. Stop transfer between the DMAC and SPDIF.
9. Restart transfer between the DMAC and SPDIF.

When restarting transfer at step 9, follow the procedure from step 7. You do not need to follow the order of steps 3 to 5 to make initial settings for the audio related modules. Regarding the mode setting registers of the individual IPs, refer to the functional specifications of these IPs.

Note 1. CLK9_ON to CLK13_ON of the CPG_CLKON_15 register. Only the clocks required for operation are turned on.

Note 2. CLK10_ON to CLK12_ON of the CPG_CLKON_22 register. Only the clocks required for operation are turned on.

8.1.3.10 PDM Setup Procedure Overview

Before setting up the audio related modules, you must set the pin functions and release the IPs to use from module standby.

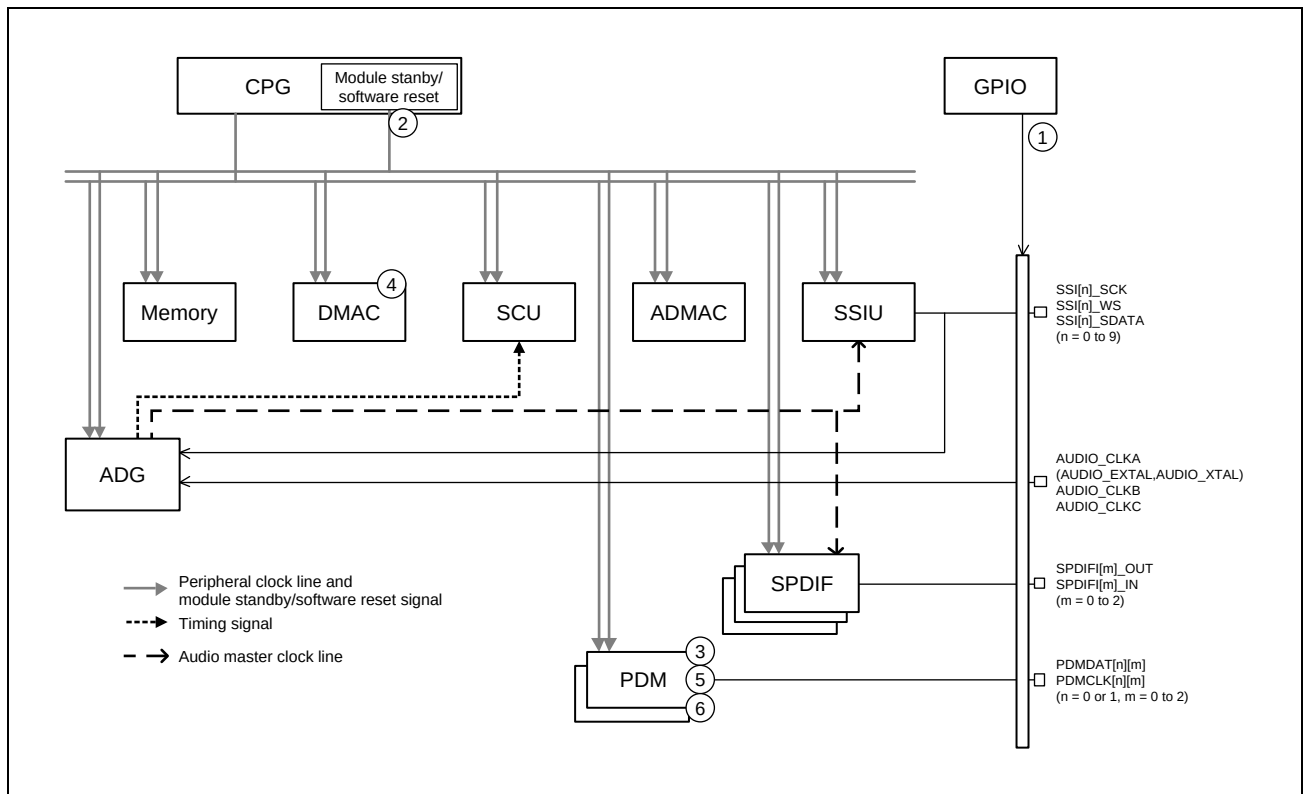


Figure 8.1-15 PDM Setup Procedure Diagram

The following is the procedure in outline.

1. Set the pin functions (GPIO).
2. Release module standby (software reset).
3. Set PDM mode.
4. Set DMAC mode.
5. Initiate transfer between the DMAC and PDM.
6. Stop transfer between the DMAC and PDM.
7. Restart transfer between the DMAC and PDM.

When restarting transfer at step 7, follow the procedure from step 5.

You do not need to follow the order of steps 3 and 4 to make initial settings for the PDM and DMAC

Regarding the mode setting registers of the individual IPs, refer to the functional specifications of these IPs.

8.1.3.11 Module Standby Function

For the register settings to make a transition to or from module standby mode, refer to section CPG.

8.1.3.11.1 Transition to Module Standby Mode

To make a transition to module standby mode by the module standby function, refer to the following transition procedure for each module.

(1) ADG

1. Set the CLK0_ON to CLK9_ON bits in CPG_CLKON_22 register to 0
2. Set the CLK9_ON bit in CPG_CLKON_15 register to 0

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

(2) ADMAC

1. Set the CLK8_ON bit in CPG_CLKON_15 register to 0

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

(3) SCU

1. Check the following registers' settings.
 - All bits in the SRCm control register (SRCm_CONTROL) (m = 0 to 9) are cleared to 0 (data transfer stopped).
 - All bits in the CMDn control register (CMDn_CONTROL) (n = 0, 1) are cleared to 0 (data transfer stopped).
 - All bits in the SRCm interrupt enable register 0 (SRCm_INT_ENABLE0) (m = 0 to 6, 9) are cleared to 0 (interrupt disabled).
 - All bits in the SRCn interrupt enable register 0 (SRCn_INT_ENABLE0) (n = 7, 8) are cleared to 0 (interrupt disabled).
2. Set the CLK0_ON to CLK14_ON bits in CPG_CLKON_23 register to 0.

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

(4) SSIU

1. Check the following registers' settings.

[SSIU]

- All bits in the SSIq control register (SSIQ_CONTROL) (q = 0-0, 1-0, 2-0, 3-0, 4-0, or 9-0) are cleared to 0 (data transfer stopped).
- All bits in the SSIr control register (SSIR_CONTROL) (r = 5 to 8) are cleared to 0 (data transfer stopped).
- All bits in the SSIq interrupt enable register (SSIQ_INT_ENABLE_MAIN) (q = 0-0, 1-0, 2-0, 3-0, 4-0, or 9-0) are cleared to 0 (interrupt disabled).
- All bits in the SSIr interrupt enable register (SSIR_INT_ENABLE_MAIN) (r = 5 to 8) are cleared to 0 (interrupt disabled).
- All bits in the SSI control register (SSI_CONTROL) are cleared to 0 (data transfer stopped)

[SSI]

- The DMEN and EN bits in the control register (SSICRn) (n = 0 to 9) are cleared to 0 (disabled).
- The IDST bit in the status register (SSISRn) (n = 0 to 9) is set to 1.
- The CONT bit in the WS mode register (SSIWSRn) (n = 0 to 9) is cleared to 0 (disabled).

2. Set the CLK0_ON to CLK10_ON bits in CPG_CLKON_24 register to 0

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

(5) SPDIF

1. If the module is SPDIF ch0, set the CLK14_ON bit in CPG_CLKON_15 register to 0.
If the module is SPDIF ch1, set the CLK15_ON bit in CPG_CLKON_15 register to 0.
If the module is SPDIF ch2, set the CLK0_ON bit in CPG_CLKON_16 register to 0.

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

(6) PDM

1. If the module is PDM ch0, set the CLK1_ON to CLK3_ON bits in CPG_CLKON_16 register to 0.
If the module is PDM ch1, set the CLK4_ON to CLK6_ON bits in CPG_CLKON_16 register to 0.

Remark The above bits should be set while the module operation has been completed and is placed in the idle state in which the module cannot be activated by external pins or other modules.

8.1.3.11.2 Releasing and Restarting from Module Standby Mode

After the transition to module standby mode, modules can be released from module standby mode by a power-on reset or the appropriate procedure from the list below.

(1) ADG

1. Supply the clock signal to ADG modules.
2. Set the CLK9_ON bit in CPG_CLKON_15 register to 1.

(2) ADMAC

1. Set the CLK8_ON bit in CPG_CLKON_15 register to 1.

(3) SCU

1. Set the CLK0_ON to CLK14_ON bits in CPG_CLKON_23 register to 1.

(4) SSIU

1. Supply the clock signal to SSIU modules.
2. Set the CLK0_ON to CLK10_ON bits in CPG_CLKON_24 register to 1

(5) SPDIF

1. If the module is SPDIF ch0, set the CLK14_ON bit in CPG_CLKON_15 register to 1.
If the module is SPDIF ch1, set the CLK15_ON bit in CPG_CLKON_15 register to 1.
If the module is SPDIF ch2, set the CLK0_ON bit in CPG_CLKON_16 register to 1.

(6) PDM

1. If the module is PDM ch0, set the CLK1_ON to CLK3_ON bits in CPG_CLKON_16 register to 1.
If the module is PDM ch1, set the CLK4_ON to CLK6_ON bits in CPG_CLKON_16 register to 1.

8.1.4 Usage Notes

8.1.4.1 Notes on Route Switching and Setting Changes

When switching the transfer route, stop the current transfer, specify the new transfer route, and then start transfer. Also, when changing the quantization bit count or the formats, once stop transfer, and then start transfer again after resetting. If this procedure is not followed, correct operation is not guaranteed.

8.1.4.2 Note when SSI0, SSI1, and SSI2, or SSI0, SSI1, SSI2, and SSI9 are Used for Single-Source Audio Data

When SSI0, SSI1, and SSI2, or SSI0, SSI1, SSI2, and SSI9 are used for single-source audio data, specifying eight-bit quantization and six- or eight-channel operation at the same time is not possible.

8.1.4.3 Note on Transfer

When an underflow or an overflow occurs in any audio module, stop the transfer.

8.1.4.4 Note on Usage of the SCU Module

Before starting data transfer, the SCU module needs certain “input data timing” and “output data timing”. When “input data timing” and “output data timing” are not inputted before the transmission starts, it does not operate correctly.

In addition, if the SSI module operates in master mode, and if the WS signal is used as “input/output timing signal for the SRC”, set SSIWSRn.CONT to 1.

(The WS signal inputted into the ADG module is the same signal as the SSI_WS terminal.)

8.1.4.5 Note on Software Reset During Data Transfer

If software reset*¹ is performed during data transfer, abnormal operation may occur. At this time, normal values cannot be guaranteed for the transferred data.

Note 1. RSTB1 to RSTB15 bits of CPG_RST_14 register in the CPG module
RSTB0 to RSTB5 bits of CPG_RST_15 register in the CPG module
SWRST bit of SRCm_SWRSR register (m = 0 to 9) in the SCU module
SWRST bit of CTUn_SWRSR register (n = 00, 01, 02, 03, 10, 11, 12, or 13) in the SCU module
SWRST bit of MIXp_SWRSR register (p = 0 or 1) in the SCU module
SWRST bit of DVCp_SWRSR register (p = 0 or 1) in the SCU module

8.1.4.6 Notes when Issuing SCU/SSIU Transfer Instructions

Follow the order below to issue an instruction to start or stop transfer by SCU and SSIU.

<To start transfer>

Issue the instruction to start DMAC/ADMAC transfer, then issue the transfer start instruction in order from the module on the output side of the path.

<To stop transfer>

Issue the transfer stop instruction in order from the module on the output side of the path, then issue the instruction to stop DMAC/ADMAC transfer.

The following describes two use cases as examples.

(1) In transmission (Memory → DMAC → SCU → ADMAC → SSIU)

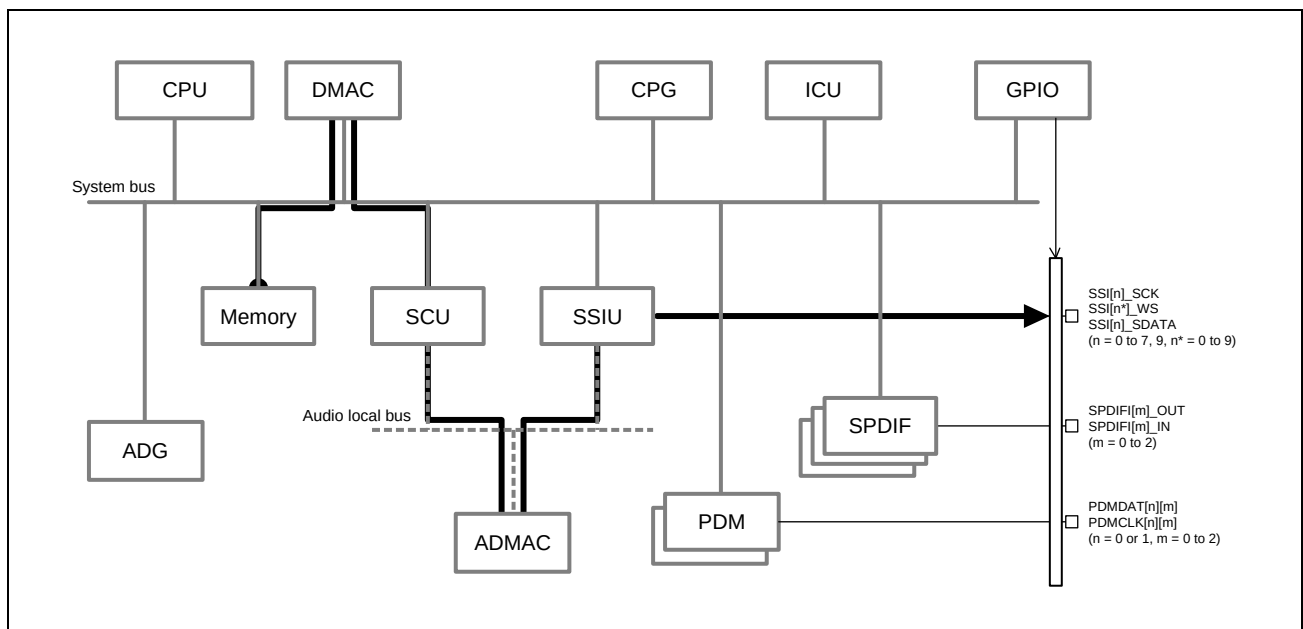


Figure 8.1-16 When Transmission is in the Order “Memory → DMAC → SCU → ADMAC → SSIU”

The order of issuing the transfer start instructions:

1. DMAC/ADMAC transfer start instruction
2. SSIU transfer start instruction
3. SCU transfer start instruction

The order of issuing the transfer stop instructions:

1. SSIU transfer stop instruction
2. SCU transfer stop instruction
3. DMAC/ADMAC transfer stop instruction

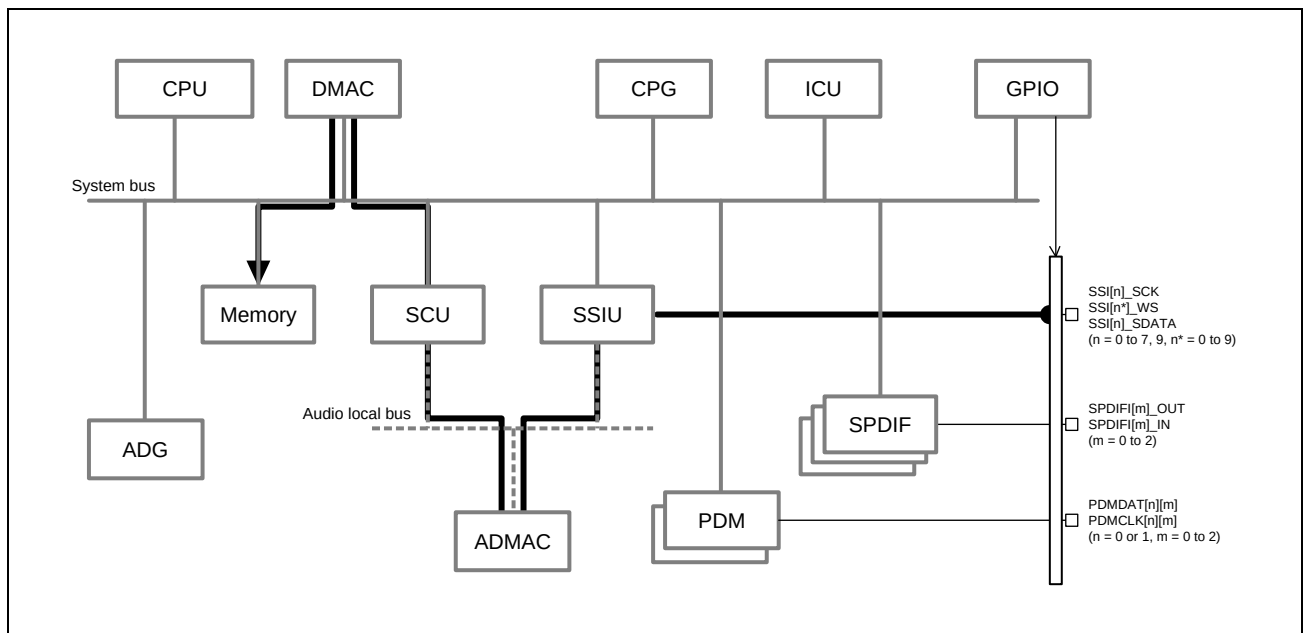
(2) In reception (SSIU → ADMAC → SCU → DMAC → Memory)

Figure 8.1-17 When Reception is in the Order “SSIU → ADMAC → SCU → DMAC → Memory”

Preparation to initiate transfer:

1. DMAC/ADMAC transfer start instruction
2. SCU transfer start instruction
3. SSIU transfer start instruction

Preparation to stop transfer:

1. SCU transfer stop instruction
2. SSIU transfer stop instruction
3. DMAC/ADMAC transfer stop instruction

SECTION 8 AUDIO

8.2 Sampling Rate Converter Unit (SCU)

This manual is a simplified version. For more information, refer to the User's Manual Additional Document.

8.2.1 Overview

The SCU has ten SRC modules (six for high-sound-quality type; four for general-sound-quality type) that are useful for synchronization of asynchronous data, which is necessary for data transfer with external memory or external devices. It also provides the functions to change the number of channels, perform mixing, and control the volume.

Remark When the SRCs are not in use, configure the system with the same clock sources for AUDIO CLOCK that is input to ADG or devices for I2S that are connected to external modules, and all thus operate at the same sampling frequency.

8.2.1.1 Features

Sampling Rate Converter (SRC)

- Asynchronous sampling rate conversion is available
- Supports resolutions up to 24 bits
- High-sound-quality type (THD + N*¹ is -132 dB) and general-sound-quality type (THD + N*¹ is -96 dB)
- Automatically generates antialiasing filter coefficients
- Four modules support one, two, four, six, or eight channels, and six modules support one or two channels.

Note 1. Total harmonic distortion plus noise

Channel Transfer Unit (CTU)

- Downmixing and splitter functions
 - Conversion of eight input channels into two output channels
 - Conversion of six input channels into two output channels
 - Conversion of two input channels into four sets of two output channels
 - Conversion of one input channel into eight sets of one output channel
 - No conversion

Mixer (MIX)

- Mixing (adds) two to four sources into one
- Ratio for adding sources is selectable
- Ratio is dynamically changeable
- Mixing with volume ramp is available (ramp period is selectable)

Digital Volume and Mute Function (DVC)

- Volume control function including digital volume, volume ramp, and zero-crossing mute
- The digital volume function is specified by a 24-bit fixed-point value within the range from 0 to 8 times (mute or -120 to 18 dB)
- The volume ramp function can be used for soft mute, fade-in, fade-out, or desired volume adjustment
- The volume ramp period can be changed within the sampling range from the 0th to 23rd power of 2
- The zero-crossing mute function silences the sound at the zero-crossing point of the audio data

The CTU, MIX, and DVC functional blocks are collectively called “CMD”.

SECTION 8 AUDIO

8.3 Audio Clock Generator (ADG)

This manual is a simplified version. For more information, refer to the User's Manual Additional Document.

8.3.1 Overview

The audio clock generator (ADG) selects and supplies the necessary clock for the SSIU, SCU, and SPDIF modules. It also divides the frequency of the selected clock and sends it outside the chip.

8.3.1.1 Features

- Selects the clock signal from the AUDIO_CLKA, AUDIO_CLKB, or AUDIO_CLKC pin or the internal clock and supplies it to the SSIU, SCU, and SPDIF modules.
- The frequency of the clock signals from the AUDIO_CLKA, AUDIO_CLKB, and AUDIO_CLKC pins and the internal clock can be divided before use.
- The divided clock can be output through the AUDIO_CLKOUT pin.

SECTION 8 AUDIO

8.4 Audio-DMAC-Peripheral-Peripheral (ADMAC)

This manual is a simplified version. For more information, refer to the User's Manual Additional Document.

8.4.1 Overview

The Audio-DMAC-Peripheral-Peripheral module controls data transfer between the audio modules (SSIU, SCU) connected to the audio local bus.

8.4.1.1 Features

- Number of channels: 29 channels
- Data transfer size: Longword (4 bytes)
- Addressing mode: Dual addressing; fixed access size
- Transfer count: Not programmable
- Interrupt processing: None

SECTION 8 AUDIO

8.5 Serial Sound Interface Unit (SSIU)

This manual is a simplified version. For more information, refer to the User's Manual Additional Document.

8.5.1 Overview

The SSIU, incorporating ten SSI modules, allows independent operation of SSI modules, operation of multiple SSI modules sharing the same serial clock, and connection or split of multi-channel data.

The serial sound interface (hereinafter referred to as the “SSI”) is a transceiver module designed to send or receive audio data interfacing with a variety of devices offering I2S format. It also supports multi-channel mode in addition to other common formats.

Remark Configure the system so that connected external devices can operate on the same clock source.

8.5.1.1 Features

- Incorporates ten SSI modules.
- Supports 6 channels/1 sound source with three SSI modules (SSI0, SSI1, and SSI2).
- Supports 8 channels/1 sound source with four SSI modules (SSI0, SSI1, SSI2, and SSI9).
- Operation of multiple modules on the same serial clock (SSI0/SSI1/SSI2/SSI3/SSI9, SSI3/SSI4/SSI9, SSI5/SSI6, or SSI7/SSI8).
- TDM format (Basic Configuration) corresponds to 4, 6, or 8-channel data.
- Handles 8-channel data on the serial bus and 6-channel data in the SoC (TDM extend mode).
- Handles 6-channel data on the serial bus and 8-channel data in the SoC (TDM extend mode).
- Connects monaural or stereo data to output TDM format data (TDM split mode).
- Splits TDM format input data into monaural or stereo data (TDM split mode).
- Connects stereo or multi data to output TDM format data (TDM ex-split mode).
- Splits TDM format input data into stereo or multi data (TDM ex-split mode).
- The frequency range of SCK signal is from 297.3 kHz to 12.5 MHz at master mode, and from 297.3 kHz to 12.5 MHz at slave mode.

The SSI has the following features:

- Number of channels: Maximum of four (when a multi-channel format is specified).
- The SSI module can serve as both a transmitter and a receiver.
- Asynchronous transfer takes place between the data buffer and the shift register.
- Only the MSB first data alignment is supported.
- A value as the dividing ratio for the clock used by the serial bus interface is selectable.

- Controlling of data transmission or reception with DMAC or interrupt requests is possible.
- TDM format is supported.
- The frequency range of SCK signal is from 297.3 kHz to 12.5 MHz at master mode, and from 297.3 kHz to 12.5 MHz at slave mode.
- The WS continue function by which operation can be performed without stopping the WS signal is supported.
- Monaural mode (8 bits or 16 bits) is supported.
- In monaural mode, the pulse width of the WS signal can be changed (short or long frame).
- If the sampling clock frequency is switched during transfer, the CPU is notified of an interrupt (a function to detect the frequency switching).
- Operating mode: Non-compressed mode (not support for compressed mode).
- Supports versatile serial audio formats (I2S/left justified/right justified).
- Supports master/slave functions.
- Programmable word clock, bit clock generation functions.

SECTION 8 AUDIO

8.6 SPDIF Interface (SPDIF)

This section describes the SPDIF interface (SPDIF).

8.6.1 Overview

The SPDIF complies with the IEC 60958 standard. For the features and configuration of the SPDIF, see the following subsections.

8.6.1.1 Features

The features of the SPDIF are described below.

- Compliant with the IEC 60958 standard (stereo and consumer use modes only)
- Sampling frequency: 32 kHz, 44.1 kHz, 48 kHz
- Audio word size: 16 to 24 bits/sample
- Biphase mark encoding
- Double buffer for data
- Serial data with parity
- Capable of simultaneous transmission and reception
- The receiver autodetects the IEC 61937 compressed mode data.
- DMAC support

The following restrictions apply to the SPDIF macro.

- Only an external-facing SPDIF device that can support 512 fs can be connected.
- Set the externally input oversampling clock (pa_audiox_a) frequency to 512 fs.

8.6.1.2 SPDIF Frame Format

A frame for the SPDIF consists of two sub-frames (channels 1 and 2), each containing a 4-bit preamble, up to 24 bits of audio data, V flag, user bit, channel status bit, and even parity bit. **Figure 8.6-1** shows the format of a sub-frame. The SPDIF performs modulation (channel coding) based on the biphase mark method, which minimizes the DC component on the transmission line for this format.

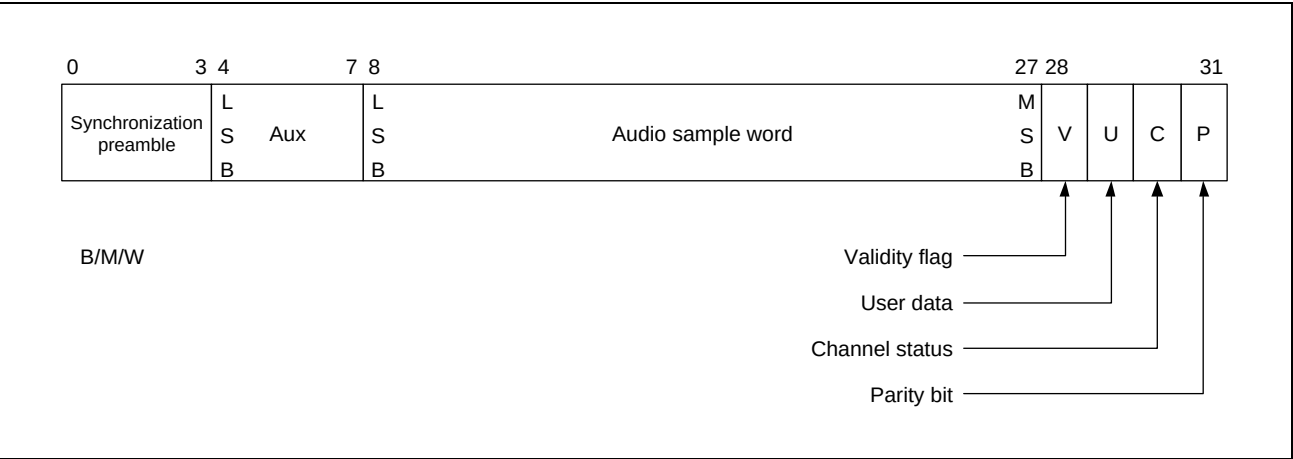


Figure 8.6-1 Sub-Frame Format

Figure 8.6-2 shows the format of a block, which consists of consecutive 192 frames. One block extends from the start frame (preamble B) to frame 191, which is the 192nd frame, and each sub-frame is identified by a preamble. 384 sub-frames are in one block and the frames consist of three types of sub-frames: sub-frame 0 indicating the start of a block, sub-frame 1 (generally, left channel), and sub-frame 2 (generally, right channel). Generally, the music data that are transmitted and received by the SPDIF are consecutive, so the data are transferred in consecutive blocks.

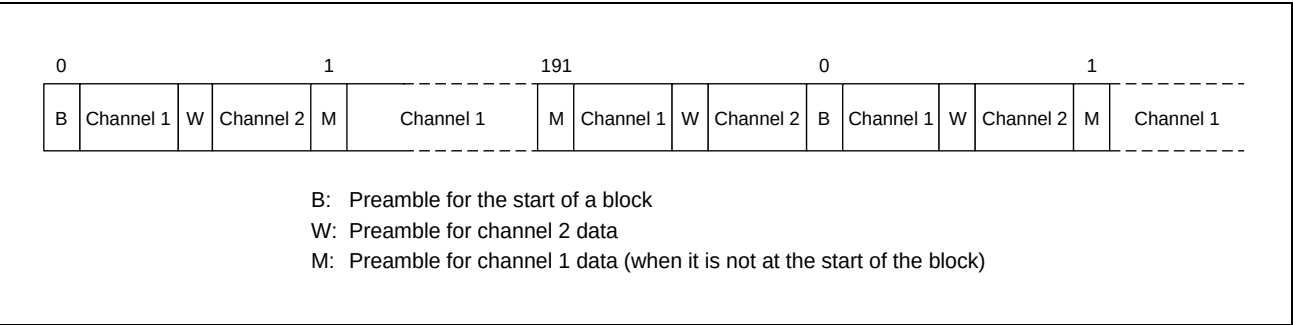


Figure 8.6-2 Block Format

Table 8.6-1 shows the values of the SPDIF preambles (in binary). The state of the preceding symbol (parity bit) determines the logical sense of each preamble.

Table 8.6-1 Preamble Values

Preamble	State of Preceding Symbol = 0	State of Preceding Symbol = 1
B	11101000	00010111
M	11100010	00011101
W	11100100	00011011

Remark: As shown in **Figure 8.6-1**, the even parity bit in time slot 31 of the sub-frame determines the preamble type in one transmission. Therefore, one of the set states to be normally sent via the SPDIF is selected. However, in IEC 60958 and so on, decoding of both types is defined in consideration of connection of the preamble with its polarity reversed. The SPDIF decodes preambles as listed in **Table 8.6-1**.

The channel status information is encoded in one bit of each sub-frame. Therefore, the amount of channel status information for one block is 192 bits for each of sub-frames 1 and 2. For the format of the channel status, refer to the IEC 60958 standard.

8.6.1.3 About Sampling Frequency Selection Method

The SPDIF macro supports three sampling frequencies: 1) 32 kHz, 2) 44.1 kHz, and 3) 48 kHz. To support each sampling frequency, the oversampling clock at a frequency of 512 fs must be entered through the input terminal for the oversampling clock (pa_audiox_a) of the SPDIF macro. The pa_audio_a terminal input clock frequencies for each of the sampling frequencies are listed below.

Table 8.6-2 Sampling Frequency versus pa_audio_a Terminal Input Clock Frequency

Sampling Frequency	pa_audio_a Terminal Input Clock Frequency (512 fs)
32 kHz	16.384 MHz
44.1 kHz	22.5792 MHz
48 kHz	24.576 MHz

Clock input to the SPDIF pa_audiox_a terminal is possible from any of the following three externally input pins.

- 1) AUDIO_EXTAL/AUDIO_XTAL
- 2) AUDIO_CLKB
- 3) AUDIO_CLKC

For selection of the external input, see the sections PFC and ADG.

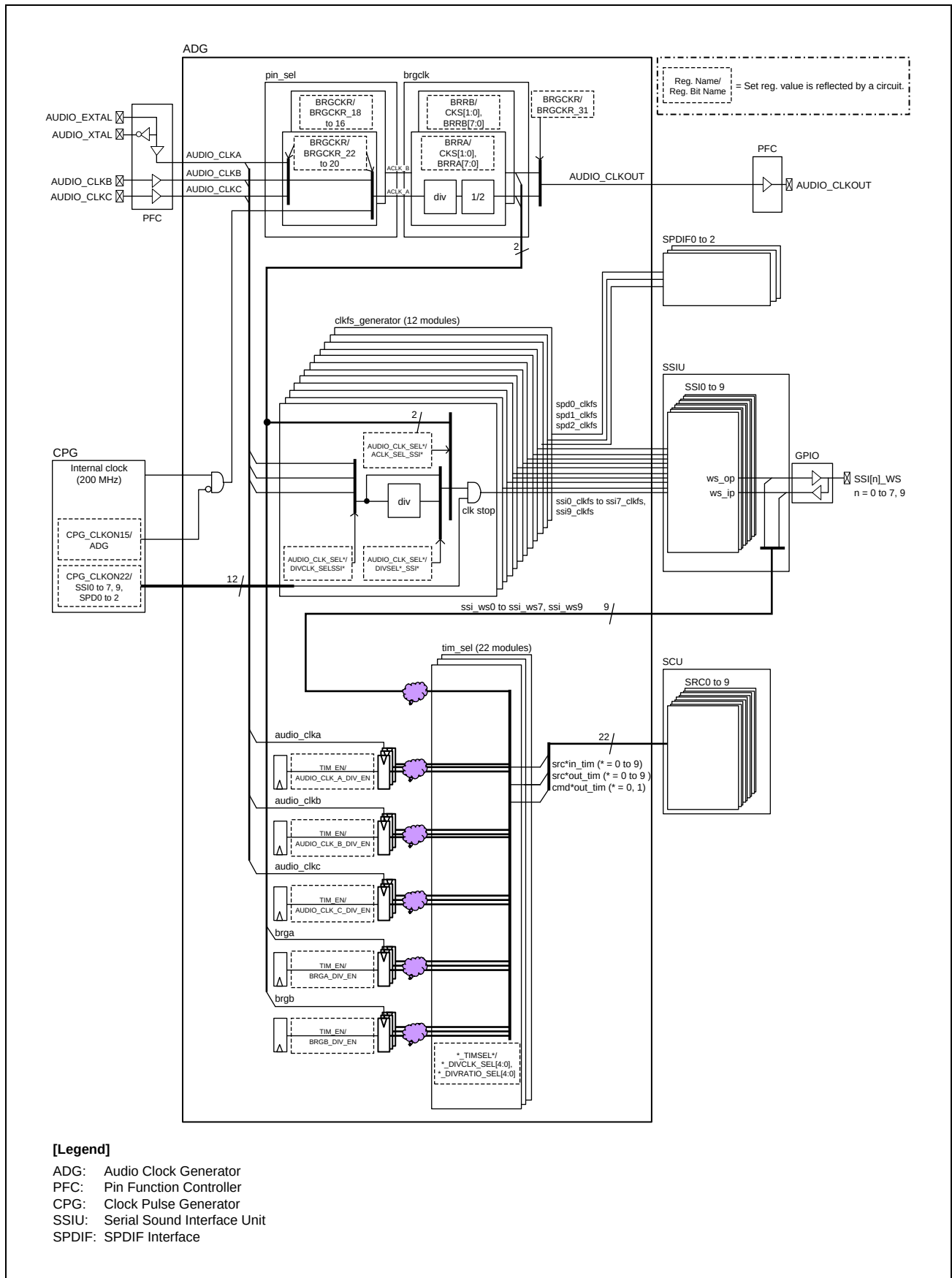


Figure 8.6-3 SPDIF External Clock Input Configuration

8.6.1.4 Block Diagram

The figure below is a block diagram of the SPDIF.

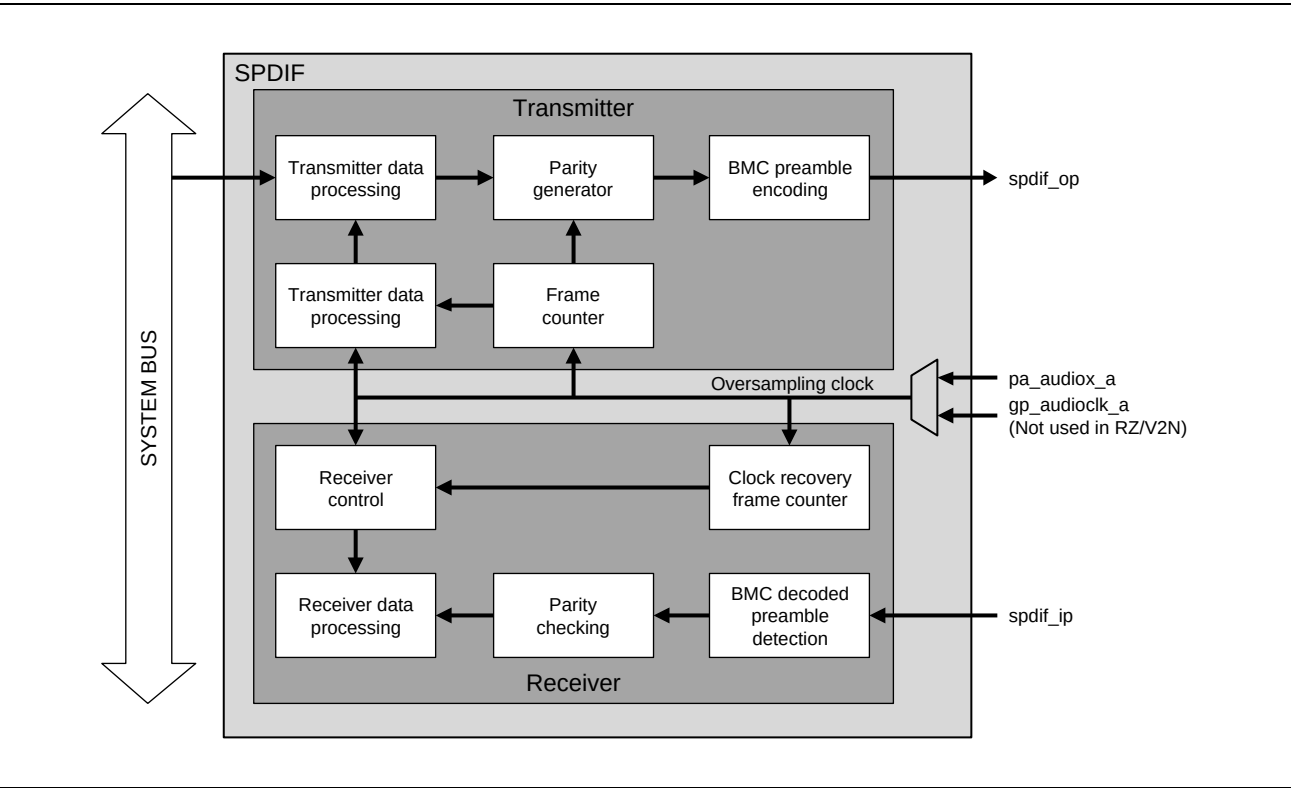


Figure 8.6-4 Block Diagram of SPDIF

8.6.1.5 External Pins

Table 8.6-3 lists the specifications of the SPDIF external pins.

Table 8.6-3 Pin Configuration

Pin Name	Input/ Output	Function
SPDIFn_OUT	Output	SPDIF signal transmission pin
SPDIFn_IN	Input	SPDIF signal reception pin

Note: n = 0 to 2

8.6.1.6 Connected Units

Table 8.6-4 lists the units that are connected to the SPDIF.

Table 8.6-4 Connected Units

Unit Name	Connected Unit Name	Function of Connected Unit
SPDIF	CPG	Clock pulse generator (CPG)
	SYSTEM BUS	Internal bus
	ICU	Interrupt control unit (ICU_TOP)
	PFC	Pin Function Controller (PFC)
	ADG	Audio clock generator (ADG)

8.6.2 Registers

The base addresses for the respective channels are as follows.

Table 8.6-5 Register Base Addresses

Base Register Name	Unit Name	Base Address
<SPDIF0_base>	SPDIF0	0_1440_2400h (5440_2400h ^{*1} , 4440_2400h ^{*2})
<SPDIF1_base>	SPDIF1	0_1440_2800h (5440_2800h ^{*1} , 4440_2800h ^{*2})
<SPDIF2_base>	SPDIF2	0_1440_2C00h (5440_2C00h ^{*1} , 4440_2C00h ^{*2})

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

8.6.2.1 List of Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Transmitter Channel 1 Audio Register	SPDIFm_TLCA	0000_0000h	0000h	32
Transmitter Channel 2 Audio Register	SPDIFm_TRCA	0000_0000h	0004h	32
Transmitter Channel 1 Status Register	SPDIFm_TLCS	0000_0000h	0008h	32
Transmitter Channel 2 Status Register	SPDIFm_TRCS	0000_0000h	000Ch	32
Transmitter User Data Register	SPDIFm_TUI	0000_0000h	0010h	32
Receiver Channel 1 Audio Register	SPDIFm_RLCA	0000_0000h	0014h	32
Receiver Channel 2 Audio Register	SPDIFm_RRCA	0000_0000h	0018h	32
Receiver Channel 1 Status Register	SPDIFm_RLCS	0000_0000h	001Ch	32
Receiver Channel 2 Status Register	SPDIFm_RRCS	0000_0000h	0020h	32
Receiver User Data Register	SPDIFm_RUI	0000_0000h	0024h	32
Control Register	SPDIFm_CTRL	0000_0000h	0028h	32
Status Register	SPDIFm_STAT	0000_C000h	002Ch	32
Transmitter DMA Audio Data Register	SPDIFm_TDAD	0000_0000h	0030h	32
Receiver DMA Audio Data Register	SPDIFm_RDAD	0000_0000h	0034h	32

Always access the above registers as 32-bit units because they are all longword registers.

Always write 0 to a bit for which the writing of 0 is stated in the register descriptions on the following pages (when the bit is writable). 0 is always read from the bit when such a bit is read (when the bit is readable).

8.6.3 Register Description

The prefix (SPDIFm_) of the register names is omitted in this and subsequent sections.

8.6.3.1 Transmitter Channel 1 Audio Register (SPDIFm_TLCA)

This register is for writing the channel 1 audio PCM data for the transmitter.

Access Size :		32 bits																
Address :		<SPDIFm_base> + 0000h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	PCM_DATA[15:0]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
Bit	Bit Name																	
Initial Value		All 0																
R/W		W																
Description		Reserved																
		A register that can be written. Reading is invalid. The write value should always be 0b.																
Bit		PCM_DATA																
		[23:0]																
Initial Value		0h																
R/W		W																
Description		The read value is undefined																
		Audio PCM data																
		PCM-encoded audio data (stored flush against the LSB side)																

8.6.3.2 Transmitter Channel 2 Audio Register (SPDIFm_TRCA)

This register is for writing the channel 2 audio PCM data for the transmitter.

Access Size :		32 bits															
Address :		<SPDIFm_base> + 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PCM_DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	
Bit	Bit Name		Initial Value	R/W	Description												
31 to 24	-		All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.												
23 to 0	PCM_DATA [23:0]		0h	W	The read value is undefined Audio PCM data PCM-encoded audio data (stored flush against the LSB side)												

8.6.3.3 Transmitter Channel 1 Status Register (SPDIFm_TLCS)

The channel status information to be transmitted is stored in a 30-bit register. The channel status information for each channel consists of 192 bits per one frame. However, the essential data are only the 30 bits to be set in the register below, so 0 is transmitted after transmission of the first 30 bits.

Refer to IEC 60958 regarding the values to be set in the fields.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0008h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	CLAC[1:0]		FS[3:0]				CHNO[3:0]				SRCNO[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CATCD[7:0]								-	-	CTL[4:0]				-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.
29, 28	CLAC[1:0]	0h	W	The read value is undefined Clock accuracy 00b: Level 2 01b: Level 1 10b: Level 3 11b: Reserved
27 to 24	FS[3:0]	0h	W	The read value is undefined Sampling frequency (fs) 0000b: 44.1 kHz 0100b: 48 kHz 1100b: 32 kHz
23 to 20	CHNO[3:0]	0h	W	The read value is undefined Channel number 0000b: Unspecified 0001b: A (left channel) 0010b: B (right channel) 0011b: C
19 to 16	SRCNO[3:0]	0h	W	The read value is undefined Transmission source number 0000b: Unspecified 0001b: 1 0010b: 2 0011b: 3
15 to 8	CATCD[7:0]	0h	W	The read value is undefined Category code (example) 0000_0000b: 2-channel general-purpose format 0000_0001b: 2-channel compact disk (IEC 908) 0000_0010b: 2-channel PCM encoder/decoder 0000_0011b: 2-channel digital audio tape recorder
7, 6	-	All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.
5 to 1	CTL[4:0]	0h	W	The read value is undefined Control The control bits are copied from the transmission source (refer to the IEC 60958 standard).
0	-	0h	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.

8.6.3.4 Transmitter Channel 2 Status Register (SPDIFm_TRCS)

The channel status information to be transmitted is stored in a 30-bit register. The channel status information for each channel consists of 192 bits per one frame. However, the essential data are only the 30 bits to be set in the register below, so 0 is transmitted after transmission of the first 30 bits.

Refer to IEC 60958 regarding the values to be set in the fields.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 000Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	CLAC[1:0]		FS[3:0]				CHNO[3:0]				SRCNO[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CATCD[7:0]								-	-	CTL[4:0]				-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.
29, 28	CLAC[1:0]	0h	W	The read value is undefined Clock accuracy 00b: Level 2 01b: Level 1 10b: Level 3 11b: Reserved
27 to 24	FS[3:0]	0h	W	The read value is undefined Sampling frequency (fs) 0000b: 44.1 kHz 0100b: 48 kHz 1100b: 32 kHz
23 to 20	CHNO[3:0]	0h	W	The read value is undefined Channel number 0000b: Unspecified 0001b: A (left channel) 0010b: B (right channel) 0011b: C
19 to 16	SRCNO[3:0]	0h	W	The read value is undefined Transmission source number 0000b: Unspecified 0001b: 1 0010b: 2 0011b: 3
15 to 8	CATCD[7:0]	0h	W	The read value is undefined Category code (example) 0000_0000b: 2-channel general-purpose format 0000_0001b: 2-channel compact disk (IEC 908) 0000_0010b: 2-channel PCM encoder/decoder 0000_0011b: 2-channel digital audio tape recorder
7, 6	-	All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.
5 to 1	CTL[4:0]	0h	W	The read value is undefined Control The control bits are copied from the transmission source (refer to the IEC 60958 standard).
0	-	0h	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.

8.6.3.5 Transmitter User Data Register (SPDIFm_TUI)

This register is for writing the U-bit data of the sub-frame. Update the U-bit data every 16 frames because the data are transmitted in order of sub-frame 1 and then sub-frame 2. For the contents of the user byte, refer to the standard for the user information that corresponds to the SPDIF device to be used. The user bits to be transmitted are set in order from the LSB side.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BYTE4[7:0]								BYTE3[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BYTE2[7:0]								BYTE1[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	BYTE4[7:0]	0h	W	The read value is undefined The U-bit information is stored.
23 to 16	BYTE3[7:0]	0h	W	The read value is undefined The U-bit information is stored.
15 to 8	BYTE2[7:0]	0h	W	The read value is undefined The U-bit information is stored.
7 to 0	BYTE1[7:0]	0h	W	The read value is undefined The U-bit information is stored.

8.6.3.6 Receiver Channel 1 Audio Register (SPDIFm_RLCA)

This register is for reading the channel 1 audio PCM data for the receiver.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCM_DATA[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W		Description									
31 to 24	-		All 0		R		Reserved Whenever it is read, 0 is read. The written value will be ignored.									
23 to 0	PCM_DATA [23:0]		0h		R		Audio PCM data PCM-encoded audio data (stored flush against the LSB side)									

8.6.3.7 Receiver Channel 2 Audio Register (SPDIFm_RRCA)

This register is for reading the channel 2 audio PCM data for the receiver.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0018h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCM_DATA[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
23 to 0	PCM_DATA [23:0]	0h	R	Audio PCM data PCM-encoded audio data (stored flush against the LSB side)

8.6.3.8 Receiver Channel 1 Status Register (SPDIFm_RLCS)

The channel status is stored from the start of the block to received sub-frame 1 in order from the LSB bit of this register. For the contents of the channel status, refer to IEC 60958.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 001Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	CLAC[1:0]		FS[3:0]				CHNO[3:0]				SRCNO[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CATCD[7:0]								-	-	CTL[4:0]				-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
29, 28	CLAC[1:0]	0h	R	Clock accuracy 00b: Level 2 01b: Level 1 10b: Level 3 11b: Reserved
27 to 24	FS[3:0]	0h	R	Sampling frequency (fs) 0000b: 44.1 kHz 0100b: 48 kHz 1100b: 32 kHz
23 to 20	CHNO[3:0]	0h	R	Channel number 0000b: Unspecified 0001b: A (left channel) 0010b: B (right channel) 0011b: C
19 to 16	SRCNO[3:0]	0h	R	Transmission source number 0000b: Unspecified 0001b: 1 0010b: 2 0011b: 3
15 to 8	CATCD[7:0]	0h	R	Category code (example) 0000_0000b: 2-channel general-purpose format 0000_0001b: 2-channel compact disk (IEC 908) 0000_0010b: 2-channel PCM encoder/decoder 0000_0011b: 2-channel digital audio tape recorder
7, 6	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
5 to 1	CTL[4:0]	0h	R	Control The control bits are copied from the transmission source (refer to the IEC 60958 standard).
0	-	0h	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.

8.6.3.9 Receiver Channel 2 Status Register (SPDIFm_RRCS)

The channel status is stored from the start of the block to received sub-frame 2 in order from the LSB bit of the register. For the contents of the channel status, refer to IEC 60958.

Access Size :		32 bits															
Address :		<SPDIFm_base> + 0020h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	CLAC[1:0]		FS[3:0]				CHNO[3:0]				SRCNO[3:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	CATCD[7:0]								-	-	CTL[4:0]				-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
29, 28	CLAC[1:0]	0h	R	Clock accuracy 00b: Level 2 01b: Level 1 10b: Level 3 11b: Reserved
27 to 24	FS[3:0]	0h	R	Sampling frequency (fs) 0000b: 44.1 kHz 0100b: 48 kHz 1100b: 32 kHz
23 to 20	CHNO[3:0]	0h	R	Channel number 0000b: Unspecified 0001b: A (left channel) 0010b: B (right channel) 0011b: C
19 to 16	SRCNO[3:0]	0h	R	Transmission source number 0000b: Unspecified 0001b: 1 0010b: 2 0011b: 3
15 to 8	CATCD[7:0]	0h	R	Category code (example) 0000_0000b: 2-channel general-purpose format 0000_0001b: 2-channel compact disk (IEC 908) 0000_0010b: 2-channel PCM encoder/decoder 0000_0011b: 2-channel digital audio tape recorder
7, 6	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
5 to 1	CTL[4:0]	0h	R	Control The control bits are copied from the transmission source (refer to the IEC 60958 standard).
0	-	0h	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.

8.6.3.10 Receiver User Data Register (SPDiFm_RUI)

This register stores the U-bit data of the sub-frame. Read the U-bit data every 16 frames because the data are stored in order of sub-frame 1 and then sub-frame 2 from the LSB side. For the contents of the user byte, refer to the standard that corresponds to the device to be used.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0024h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BYTE4[7:0]								BYTE3[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BYTE2[7:0]								BYTE1[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	BYTE4[7:0]	0h	R	The U-bit information is stored.
23 to 16	BYTE3[7:0]	0h	R	The U-bit information is stored.
15 to 8	BYTE2[7:0]	0h	R	The U-bit information is stored.
7 to 0	BYTE1[7:0]	0h	R	The U-bit information is stored.

8.6.3.11 Control Register (SPDIFm_CTRL)

This register controls the SPDIF unit. All control is applied in accord with the settings of this register.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0028h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	RASS[1:0]		TASS[1:0]		RDE	TDE	NCSI	AOS	RME	TME
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REIE	TEIE	UBOI	UBUI	CREI	PAEI	PREI	CSEI	ABOI	ABUI	RUII	TUII	RCSI	RCBI	TCSI	TCBI
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
28	-	0h	RW	Reserved. The write value should always be 0.
27	-	0h	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
26	-	0h	RW	Reserved. The write value should always be 0.
25, 24	RASS[1:0]	0h	RW	Receiver audio sample bit size (audio word size) Specify the number of the receiver audio sample bits (16, 20, or 24 bits) for data alignment. 00b: 16-bit sample 01b: 20-bit sample 10b: 24-bit sample 11b: Reserved
23, 22	TASS[1:0]	0h	RW	Transmitter audio sample bit size (audio word size) Specify the number of the transmitter audio sample bits (16, 20, or 24 bits) for data alignment. 00b: 16-bit sample 01b: 20-bit sample 10b: 24-bit sample 11b: Reserved
21	RDE	0h	RW	Receiver DMA enable Enables receiver DMA requests. 0b: Disables receiver DMA. 1b: Enables receiver DMA.
20	TDE	0h	RW	Transmitter DMA enable Enables transmitter DMA requests. 0b: Disables transmitter DMA. 1b: Enables transmitter DMA.
19	NCSI	0h	RW	New channel status information Set this bit to 1 when new channel status information that requires correction is present in the transmitter. 0b: No new channel status information is present. 1b: Channel status information is present.
18	AOS	0h	RW	Audio only sample Set this bit to 0 when user information is present in the user data register. When this bit is set to 1, the user bits are set to all 0s. 0b: User information is present. 1b: No user information is present.
17	RME	0h	RW	Receiver enable Enables the receiver. 0b: The receiver is disabled. 1b: The receiver is enabled.

Bit	Bit Name	Initial Value	R/W	Description
16	TME	0h	RW	Transmitter enable Enables the transmitter. 0b: The transmitter is disabled. 1b: The transmitter is enabled.
15	REIE	0h	RW	Receiver error interrupt enable Clearing this bit masks all error interrupts from the receiver. Setting this bit to 1 enables all receiver error interrupts. 0b: Disables receiver error interrupts. 1b: Enables receiver error interrupts.
14	TBE	0h	RW	Transmitter error interrupt enable Clearing this bit masks all error interrupts from the transmitter. Setting this bit to 1 enables all transmitter error interrupts. 0b: Disables transmitter error interrupts. 1b: Enables transmitter error interrupts.
13	UBOI	0h	RW	User buffer overrun interrupt enable Enables the user buffer overrun interrupt. 0b: Disables the user buffer overrun interrupt. 1b: Enables the user buffer overrun interrupt.
12	UBUI	0h	RW	User buffer underrun interrupt enable Enables the user buffer underrun interrupt. 0b: Disables the user buffer underrun interrupt. 1b: Enables the user buffer underrun interrupt.
11	CREI	0h	RW	Clock recovery error interrupt enable Enables the clock recovery error interrupt. 0b: Disables the clock recovery error interrupt. 1b: Enables the clock recovery error interrupt.
10	PAEI	0h	RW	Parity error interrupt enable Enables the parity checking error interrupt. 0b: Disables the parity checking error interrupt. 1b: Enables the parity checking error interrupt.
9	PREI	0h	RW	Preamble error interrupt enable Enables the preamble checking error interrupt. 0b: Disables the preamble error interrupt. 1b: Enables the preamble error interrupt.
8	CSEI	0h	RW	Channel status error interrupt enable Enables the channel status error interrupt. 0b: Disables the channel status error interrupt. 1b: Enables the channel status error interrupt.
7	ABOI	0h	RW	Audio buffer overrun interrupt enable Enables the receiver audio buffer overrun interrupt. 0b: Disables the audio buffer overrun interrupt. 1b: Enables the audio buffer overrun interrupt.
6	ABUI	0h	RW	Audio buffer underrun interrupt enable Enables the transmitter audio buffer underrun interrupt. 0b: Disables the audio buffer underrun interrupt. 1b: Enables the audio buffer underrun interrupt.
5	RUII	0h	RW	Receiver user information interrupt enable Enables the receiver user information register full interrupt. 0b: Disables the receiver user information interrupt. 1b: Enables the receiver user information interrupt.
4	TUII	0h	RW	Transmitter user information interrupt enable Enables the transmitter user information register empty interrupt. 0b: Disables the transmitter user information interrupt. 1b: Enables the transmitter user information interrupt.
3	RCSI	0h	RW	Receiver channel status interrupt enable Enables the receiver channel status register full interrupt. 0b: Disables the receiver channel status interrupt. 1b: Enables the receiver channel status interrupt.
2	RCBI	0h	RW	Receiver channel buffer interrupt enable Enables the receiver audio channel buffer full interrupt. 0b: Disables the receiver audio channel interrupt. 1b: Enables the receiver audio channel interrupt.

Bit	Bit Name	Initial Value	R/W	Description
1	TCSI	0h	RW	Transmitter channel status interrupt enable Enables the transmitter channel status register empty interrupt. 0b: Disables the transmitter channel status interrupt. 1b: Enables the transmitter channel status interrupt.
0	TCBI	0h	RW	Transmitter channel buffer interrupt enable Enables the transmitter audio channel buffer empty interrupt. 0b: Disables the transmitter audio channel interrupt. 1b: Enables the transmitter audio channel interrupt.

8.6.3.12 Status Register (SPDIFm_STAT)

This register indicates the status information of the SPDIF unit.

Access Size : 32 bits

Address : <SPDIFm_base> + 002Ch

Initial Value : 0000_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMD
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIS	TIS	UBO	UBU	CE	PARE	PREE	CSE	ABO	ABU	RUIR	TUIR	CSRX	CBRX	CSTX	CBTX
Initial Value	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
16	CMD	0h	R	Compressed mode data This bit is set when the received data are compressed mode data (the V flag and bit 1 of channel status = 1). 0b: The data are not compressed mode data. 1b: The data are compressed mode data.
15	RIS	1h	R	Receiver idle state This bit is set when the receiver is in the idle state. 0b: The receiver is not in the idle state. 1b: The receiver is in the idle state.
14	TIS	1h	R	Transmitter idle state This bit is set when the transmitter is in the idle state. 0b: The transmitter is not in the idle state. 1b: The transmitter is in the idle state.
13	UBO	0h	RW0	User buffer overrun* ¹ This bit is set in the case of an overrun of the receiver user buffer. This bit is cleared by writing 0 to it. When the REIE and UBOI bits of the control register are set, an interrupt occurs. 0b: The user buffer has not overrun. 1b: The user buffer has overrun.
12	UBU	0h	RW0	User buffer underrun* ¹ This bit is set in the case of an underrun of the transmitter user buffer. This bit is cleared by writing 0 to it. When the TEIE and UBUI bits of the control register are set, an interrupt occurs. 0b: The user buffer has not underrun. 1b: The user buffer has underrun.
11	CE	0h	RW0	Clock error* ¹ This bit is set when synchronization cannot be obtained in clock recovery. This bit is cleared by writing 0 to it. When the REIE and CREI bits of the control register are set, an interrupt occurs. 0b: Normal clock recovery 1b: Clock recovery error
10	PARE	0h	RW0	Parity error* ¹ This bit is set when the result of parity checking is the detection of an error. This bit is cleared by writing 0 to it. When the REIE and PAEI bits of the control register are set, an interrupt occurs. 0b: The result of parity checking is normal. 1b: Parity error

Bit	Bit Name	Initial Value	R/W	Description
9	PREE	0h	RW0	Preamble error^{*1} This bit is set when a preamble is undefined or a preamble does not appear in the right position. This bit is cleared by writing 0 to it. When the REIE and PREI bits of the control register are set, an interrupt occurs. Note: The bit is only set after a start of block preamble. 0b: A preamble is in the right position. 1b: Preamble error
8	CSE	0h	RW0	Channel status error^{*1} This bit is set when the channel status information is written before the 32nd frame that is in the transfer state. This bit is cleared by writing 0 to it. When the TEIE and CSEI bits of the control register are set, an interrupt occurs. 0b: Normal channel status 1b: Channel status error
7	ABO	0h	RW0	Audio buffer overrun^{*1} This bit indicates that both the first and second stages of the receiver audio buffer are full so that further received data will be written over data that have already been received. This bit is cleared by writing 0 to it. When the REIE and ABOI bits of the control register are set, an interrupt occurs. 0b: The receiver audio buffer has not overrun. 1b: The receiver audio buffer has overrun.
6	ABU	0h	RW0	Audio buffer underrun^{*1} This bit indicates that both the first and second stages of the transmitter audio buffer are empty so that further transmission will be repetition of the last data to have been transmitted. This bit is cleared by writing 0 to it. When the TEIE and ABUI bits of the control register are set, an interrupt occurs. 0b: The transmitter audio buffer has not underrun. 1b: The transmitter audio buffer has underrun.
5	RUIR	0h	R	Receiver user information register This bit indicates the state of the receiver user information register. Reading the receiver user register clears this bit. When the RUII bit of the control register is set, an interrupt occurs. 0b: The receiver user information register is empty. 1b: The receiver user information register is full.
4	TUIR	0h	R	Transmitter user information register This bit indicates the state of the transmitter user information register. Writing to the transmitter user register clears this bit. When the TUII bit of the control register is set, an interrupt occurs. 0b: The transmitter user information register is full. 1b: The transmitter user information register is empty.
3	CSRX	0h	R	Channel 1 and 2 status (for receiver) This bit indicates the state of the receiver channel status register. Reading the receiver channel status register clears this bit. When the RCSI bit of the control register is set, an interrupt occurs. 0b: The receiver channel status register is empty. 1b: The receiver channel status register is full.
2	CBRX	0h	R	Channel 1 and 2 buffers (for receiver) This bit indicates the state of the receiver audio channel register. Reading the receiver audio channel register clears this bit. When the RCBI bit of the control register is set, an interrupt occurs. 0b: The receiver audio channel register is empty. 1b: The receiver audio channel register is full.
1	CSTX	0h	R	Channel 1 and 2 status (for transmitter) This bit indicates the state of the transmitter channel status register. Writing the transmitter channel status register clears this bit. When the TCSI bit of the control register is set, an interrupt occurs. 0b: The transmitter channel status register is full. 1b: The transmitter channel status register is empty.
0	CBTX	0h	R	Channel 1 and 2 buffers (for transmitter) This bit indicates the state of the transmitter audio channel register. Writing to the transmitter audio channel register clears this bit. When the TCBI bit of the control register is set, an interrupt occurs. 0b: The transmitter audio channel register is full. 1b: The transmitter audio channel register is empty.

Note 1. When an error bit is detected during DMA transfer, re-making the DMA transfer settings is required. In this case, set the enable bit (the RME or TME bit) and the DMA enable bit (the RDE or TDE bit) to "disabled" to release the error state, and then re-set the direct memory access controller. After that, set the module enable bit to "enabled", and the DMA transfer can be resumed.

8.6.3.13 Transmitter DMA Audio Data Register (SPDIFm_TDAD)

This register is for DMA writing of the channel 1/2 audio PCM data for the transmitter.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0030h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCM_DATA[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	W	Reserved A register that can be written. Reading is invalid. The write value should always be 0b.
23 to 0	PCM_DATA [23:0]	0h	W	The read value is undefined Audio PCM data PCM-encoded audio data (stored flush against the LSB side)

8.6.3.14 Receiver DMA Audio Data Register (SPDIFm_RDAD)

This register is for DMA reading of the channel 1/2 audio PCM data for the receiver.

Access Size :		32 bits														
Address :		<SPDIFm_base> + 0034h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PCM_DATA[23:16]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCM_DATA[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value	R/W	Description											
31 to 24	-		All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.											
23 to 0	PCM_DATA [23:0]		0h	R	Audio PCM data PCM-encoded audio data (stored flush against the LSB side)											

8.6.4 Operation

8.6.4.1 Transmitter Control

8.6.4.1.1 Overview of Transmitter Control

The transmitter encodes the PCM data and auxiliary information to be sent with the biphase mark modulation in conformance with the IEC 60958 standard (SPDIF) and sends them.

■ Clock

The clock for the transmitter is the externally supplied oversampling clock. An oversampling value that is 8 times the clock frequency required for the biphase mark encoding is usually selected for this clock. In this case, the clock frequency that is required for transmitting the 32 time slots in a sub-frame will be 512 times the sampling frequency of the audio data.

■ Audio Data and Channel Status Information

The audio data and channel status information for channel 1 and then channel 2 of the module are written.

Writing of the channel status is only required when the information is changed. Writing of the channel status information will be required by the SPDIF module after 30 frames (when all transmission of the current channel status data has been completed), so write it from the 31st frame and before the next block (= 192 frames) starts.

Audio data are stored in a double-buffer configuration. There are two ways to check that data are not present in the first stage of the double buffer.

- Sending an interrupt request
- Reading the status register

■ DMA Transfer

Channel 1 audio data are transmitted with the use of DMA transfer in response to the first request and channel 2 audio data are transmitted in response to the second request.

■ Channel Status

The channel status information to be transmitted is set in the 30-bit registers for channels 1 and 2. Each block has 192 bits per channel of channel status information. However, only 30 of the 192 bits are essential data, so 0 is transmitted after the first 30 bits have been transmitted until completion of the block.

■ User Data

User data are in a 32-bit double-buffer arrangement. Issue an interrupt request or read the status register to check that the first stage of the double buffer is empty. Generally, the data length between blocks is insufficient for the user data information, and the user data information is transmitted in order of channel 1 and channel 2. After the 384 bits of user data in the interval of one block have been transmitted, the next block is transmitted in succession.

■ Audio Data

The format of audio data handled by the SPDIF is linear PCM, and the maximum settable size is 24 bits. For this reason, the V flag that indicates the audio data are linear PCM is always 0. No register setting is required in relation to the V flag. An even parity bit is generated for each 32 bits of the serial output data (that is, without including the preamble).

8.6.4.1.2 Transmitter Initialization

The transmitter is placed in the idle state by default after a reset. It can also be placed in the idle state by writing 0 to the TME bit of the CTRL register. In the idle state, the settings of the transmitter will be as follows.

- The idle status bit (TIS) of the transmitter is set to 1, and the other status bits are all cleared to 0.
- Preamble generation is invalid.
- Synchronization of channels 1 and 2 is set to 0 (channel 1: 0, channel 2: 1).
- The word count and frame count will become 0.
- Output from the encoder based on the biphase mark will be 0.

The channel status register, user data register, and audio data register retain their values before the state transition to the idle state. Write 1 to the TME bit of the CTRL register to cause return from the idle state.

8.6.4.1.3 Transmitter Initialization Settings

Setting the TME bit to 1 causes the TUIR and CSTX bits to be set to 1. At this time, writing data in the order 1) TUI, then 2) TLCS, TRCS causes a channel status error. To suppress the occurrence of an error, write data in the order 1) TLCS, TRCS, then 2) TUI.

In addition, after the TME bit is set to 1, writing of the first audio data (writing to TLCA and TRCA by CPU access or writing to TDAD by DMA transfer) should proceed after clearing of CSTX and TUIR has been checked following writing to TLCS, TRCS, and TUI.

8.6.4.1.4 Data Transfer to the Transmitter

When the transmitter has returned from the idle state, it is ready for data transfer. Data transfer can be started in any of three ways:

- Using interrupts
- Using DMA requests
- Reading the status register

Interrupt signals that are common to transmission and reception and individual DMA request signals for the transmitter are prepared.

For details of the transfer of data in response to a DMA request, see **8.6.4.1.5 DMA Data Transfer to the Transmitter**.

Figure 8.6-5 shows the flow of data transfer to the transmitter by using interrupts.

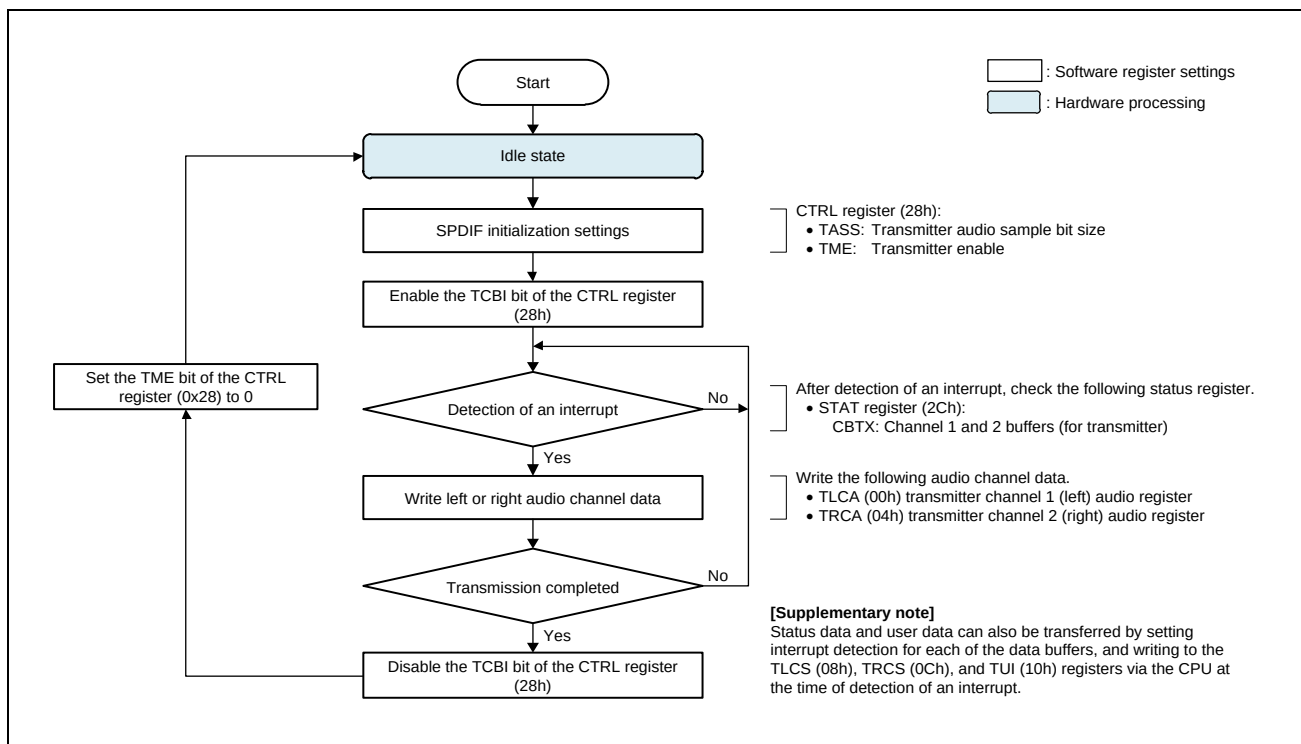


Figure 8.6-5 Flow of Data Transfer to the Transmitter (Using Interrupts)

The channel status information is updated in response to a change in the information.

Because the updating should proceed before transmission of the next block, write the channel status information for updating after the transmission of 30 frames has been completed. Completion of the transmission of 30 frames can be indicated by an interrupt or detected by reading the status bit. Writing the channel status information before the transmission of 30 frames has been completed (during transmission of the current information) leads to setting of the channel status error bit (CSE) of the status register and generates an interrupt.

Remark All valid information of one channel status block can be transmitted with 30 frames.

8.6.4.1.5 DMA Data Transfer to the Transmitter

This section describes the procedure for transmitting the SPDIF data with the use of the DMAC for DMA transfer.

Figure 8.6-6 shows the flow of data transfer to the transmitter by using DMA requests.

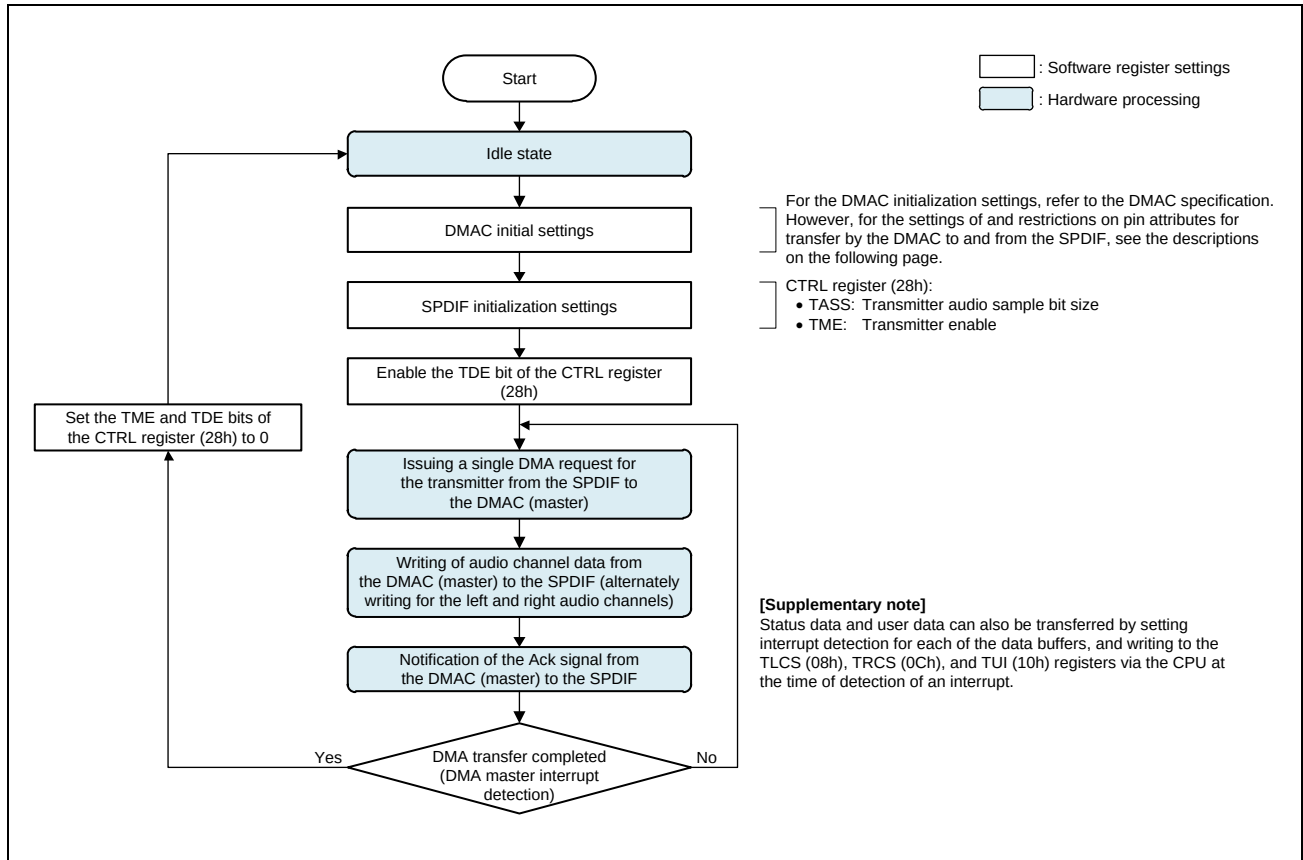


Figure 8.6-6 Flow of Data Transfer to the Transmitter (Using DMA Requests)

Set the DMAC master (in the CHCFG register) at the time of data transfer by DMA for the transmitter as follows.

1. Set the single transfer mode (one round of DMA transfer processing proceeds in response to one DMA request).
Register setting on the DMA master side: the TM bit: 0b
2. DMA transfer is in 32-bit units.
Register setting on the DMA master side: the DDS bit: 32-bit setting
3. Set DMAC REQ and DMAC ACK as level signals (DMA ACK is asserted until DMA REQ is deasserted). The active sense of the REQ signal is set to the high level.
Register settings on the DMA master side: AM[2:0] = 001b, LVL = 1b, HIEN = 1b, LOEN = 0b
Note: The active sense of the ACK signal is fixed to the high level.
4. Set the DMAC ACK signal to be activated at the time of write access to the SPDIF.
Register setting on the DMA master side: REQD = 1

5. Data for DMA transfer are written to the TDAD register (transmission) of the SPDIF unit.
6. When writing to the TDAD register at the time of DMA transfer, audio data are alternately written to the left and right channels.

For the SPDIF initialization at the time of DMA transfer, initialize the SPDIF unit according to **8.6.4.1.2 Transmitter Initialization**. At that time, enabling DMA transfer by setting the DMA enable bit (TDE = 1b) of the SPDIF unit is also required.

8.6.4.2 Receiver Control

8.6.4.2.1 Overview of Receiver Control

The receiver demodulates the data and clock signals from the input encoded in accord with the IEC 60958 standard.

The demodulated data are stored in the audio data register in a linear PCM format, and the channel status and user information that are received at the same time as auxiliary information are stored in the appropriate registers.

■ Clock

The main clock for the receiver is the externally supplied oversampling clock.

The receiver operates at 4 times the frequency of the oversampling clock.

Remark The oversampling clock is the same for the transmitter and receiver.

■ Clock Recovery

Clock recovery proceeds with the use of the pulse-width counter and averaging filter, and the sampling pulse is generated at the mid-point of each bit of the input data. The clock error status bit indicates that clock synchronization has failed. Synchronization starts when a first preamble appears in the input data. As long as changes in the jitter and clock frequency are in the range of clock recovery specification, effects of these changes on clock recovery can be constrained by continual adjustment.

When the clock recovery has been successful, the decoder starts preamble detection with the use of the biphase mark method. The decoder searches for a preamble to indicate the start of a block (see **Table 8.6-1**). The preamble error status bit indicates that the next preamble did not appear with the right timing. This can be considered as being due to an omission or the failure of transfer.

■ Parity Checking

An even-parity checking is run on the decoded data. If a mismatch is encountered, the parity error status bit is set.

■ Audio Data

Audio data are stored in a double-buffer configuration. Data can be detected to be readable by transmitting an interrupt request due to the buffer-full state or reading the corresponding status bit. Channel 1 audio data are transmitted with the use of DMA transfer in response to the first request and channel 2 audio data are transmitted in response to the second request.

■ User Data and Channel Status Information

The SPDIF module acquires the user data and channel status information as well as the audio data.

The channel status information is stored in a 30-bit register. Storage in the register is not completed until a total of 30 frames for each channel have been received because one bit is received per sub-frame. The new channel status information is compared with the current value, and the CPU only reads the new information if a change has occurred. The user data are also received at the same time. However, the reception is completed per 16 frames because the user data are stored in the register per sub-frame.

Remark 1. Requests for channel status information cannot be transferred by DMA transfer.

Remark 2. When an overrun of the receiver user buffer occurs, the current data in the SPDIF buffer are overwritten by the next data input from the SPDIF.

8.6.4.2.2 Receiver Initialization

The receiver is placed in the idle state by default after a reset. It can also be placed in the idle state by writing 0 to the RME bit of the CTRL register. In the idle state, the settings of the receiver will be as follows.

- The idle status bit of the receiver is set to 1, and the other status bits are all cleared to 0.
- Synchronization of channels 1 and 2 is set to 0 (channel 1: 0, channel 2: 1).
- The word count and frame count will become 0.

The channel status register, user data register, and audio data register retain their values before the state transition to the idle state. Write 1 to the RME bit of the CTRL register to cause return from the idle state.

8.6.4.2.3 Data Transfer from the Receiver

When the receiver has returned from the idle state, it is ready for data transfer. Data transfer can be started in any of three ways:

- Using interrupts
- Reading the status register
- Using DMA requests

For details of the transfer of data in response to a DMA request, see **8.6.4.2.4 DMA Data Transfer from the Receiver**.

Interrupt signals that are common to transmission and reception and individual DMA request signals for the receiver are prepared. An interrupt may be generated by an error signal due to any of the following factors during data transfer from the receiver.

1. Clock recovery error
2. Failure of or a fault in transfer is indicated by a preamble error
3. Parity checking error

A failure of or fault in transfer may lead to the preamble for starting a sub-frame or block having a shifted position or not appearing.

A parity checking error occurs when the parity bit is wrong. This error may occur due to any of the factors above.

■ Tolerance of Clock Recovery

The margin for clock recovery in reception is expressed by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

M: Reception margin

N: Oversampling rate

L: Frame length = 33

D: Duty cycle = 0.6

F: Oversampling clock deviation = Level II accuracy = $1000 \times 10e^{-6}$

Figure 8.6-7 shows the reception margin M.

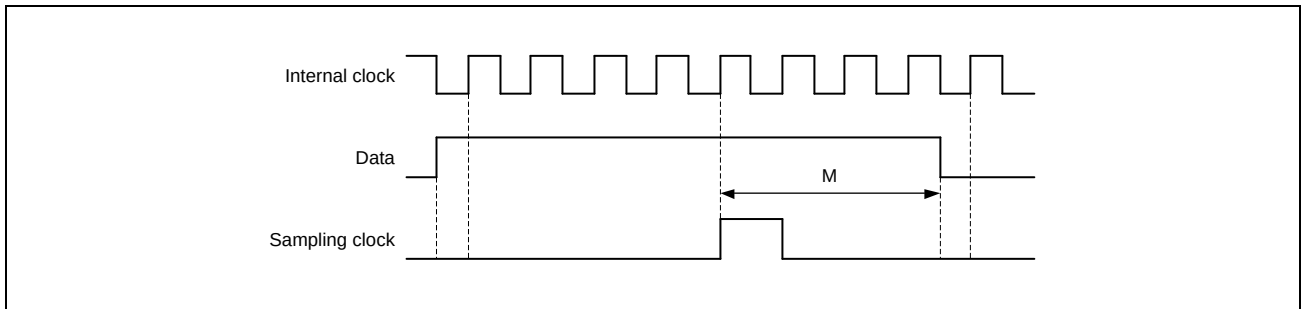


Figure 8.6-7 Reception Margin

Introducing jitter causes the previous formula to become the following inequality.

$$J \leq \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

J: Clock jitter

8 times the oversampling rate: reception margin = 39.25%

4 times the oversampling rate: reception margin = 31.75%

Twice the oversampling rate: reception margin = 16.75%

The fastest sampling frequency is 48 kHz. In this case, the required clock frequency will be $128 \times 48 \text{ kHz} = 6.144 \text{ MHz}$. In this case, the worst-case jitter in one cycle is defined as $40 \text{ ns} = 24.5\%$ of a cycle. Therefore, the above inequality will be satisfied when the oversampling rate is greater than or equal to 4.

Figure 8.6-8 shows the flow of data transfer from the receiver by using interrupts.

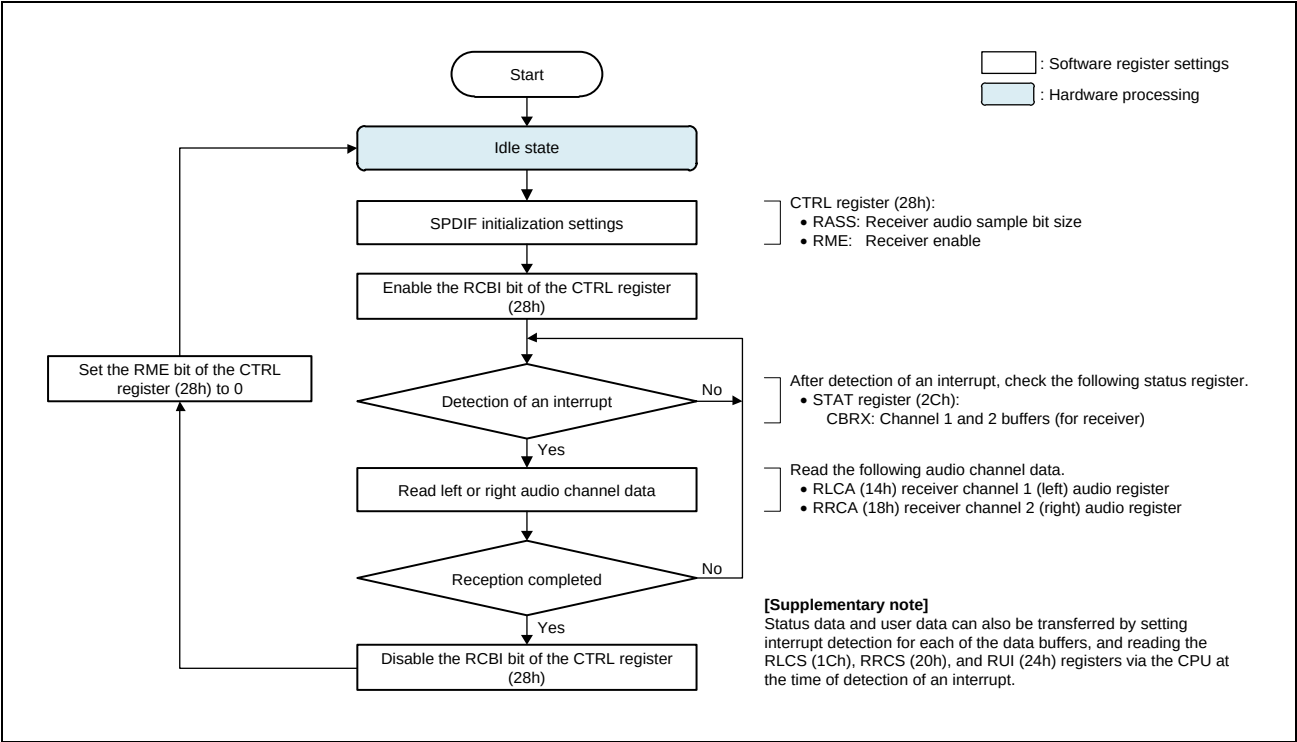


Figure 8.6-8 Flow of Data Transfer from the Receiver (Using Interrupts)

The interrupt that provides notification of information having been stored in the channel status information register is only generated when the information is found to have changed after reception of the 30th frame has been completed.

8.6.4.2.4 DMA Data Transfer from the Receiver

This section describes the procedure for receiving the SPDIF data with the use of the DMAC for DMA transfer.

Figure 8.6-9 shows the flow of data transfer from the receiver by using DMA requests.

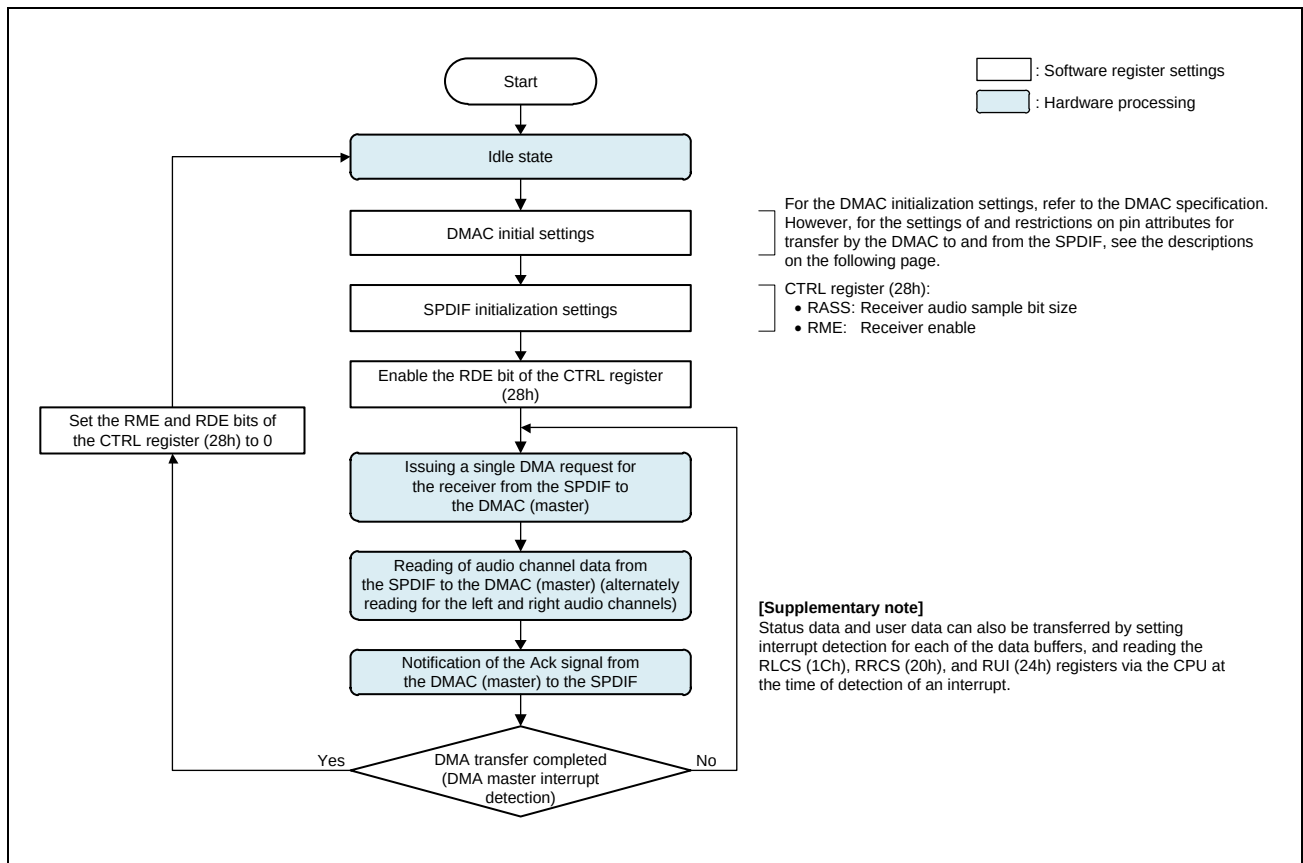


Figure 8.6-9 Flow of Data Transfer from the Receiver (Using DMA Requests)

Set the DMAC master (in the CHCFG register) at the time of data transfer by DMA from the receiver as follows.

1. Set the single transfer mode (one round of DMA transfer processing proceeds in response to one DMA request).
Register setting on the DMA master side: the TM bit: 0b
2. One DMA transfer is in 32-bit units.
Register setting on the DMA master side: the SDS bit: 32-bit setting
3. Set DMAC REQ and DMAC ACK as level signals (DMA ACK is asserted until DMA REQ is deasserted). The active sense of the REQ signal is set to the high level.
Register settings on the DMA master side: AM[2:0] = 001b, LVL = 1b, HIEN = 1b, LOEN = 0b
Note: The active sense of the ACK signal is fixed to the high level.
4. Set the DMAC ACK signal to be activated at the time of read access to the SPDIF.
Register setting on the DMA master side: REQD = 0b
5. Data for DMA transfer are read from the RDAD register (reception) of the SPDIF unit.

6. Reading of the RDAD register at the time of DMA transfer is alternately of data for the left and right channels.

For the SPDIF initialization at the time of DMA transfer, initialize the SPDIF unit according to **8.6.4.2.2 Receiver Initialization**. At that time, enabling DMA transfer by setting the DMA enable bit (RDE = 1b) of the SPDIF unit is also required.

8.6.4.3 Stopping Transmitter and Receiver

The operation of the transmitter and receiver can be invalidated by writing 0b to the enable bits in the control register (the TME bit for the transmitter, and the RME bit for the receiver). This idle state can be detected by reading the idle bits (the TIS bit for the transmitter, and the RIS bit for the receiver).

8.6.4.4 Compressed Mode Data

Compressed mode data is defined in the IEC 61937 specification. The SPDIF module only detects the compressed mode data. Check the validity flag (V flag) and bit 1 of the channel status data, and when both are 1, set the CMD bit of the status register to 1 to select compressed mode data.

Remark Only the receiver detects compressed mode data. This information has no effect on the transmitter.

8.6.4.5 Reset and Clock Control

In terms of controlling a reset and clock, the following reset and clock pins require control.

Table 8.6-6 Relationship between Reset and Clock

	Reset Signal	Clock Signal	Description
System 1	rst	clkp	For internal bus control circuit
System 2	rst* ¹	pa_audio_a	For external interface control circuit

Note 1. No measures for non-synchronization with the circuit.

The above reset and clock pins are controlled by the CPG unit, and each reset and clock signal can be switched off and on by the CPG register. The mechanism for stopping the clkp clock before and after release from the reset when the rst reset is released is implemented on the CPG side. However, a mechanism for stopping before and after release from the rst reset is not implemented for the pa_audio_a clock, so handling of reset and clock control in the way described below is required as a measure for non-synchronization. Perform the reset and clock control while the operation of the SPDIF unit is stopped, and while access by related units is also stopped.

- Release the SPDIF from the reset state and supply the clock:
 - Step 1: Start supply of the clkp clock through the register setting of the CPG unit.
 - Step 2: Release (deassert) the rst reset signal through the register setting of the CPG unit.
 - Step 3: Start supply of the pa_audiox_a clock through the register setting of the CPG unit.
- Enable the SPDIF reset and stop the clock:
 - Step 1: Stop supply of the pa_audiox_a clock through the register setting of the CPG unit.
 - Step 2: Enable (assert) the rst reset signal through the register setting of the CPG unit.
 - Step 3: Stop supply of the clkp clock through the register setting of the CPG unit.
- Apply a brief hardware reset of the whole SPDIF macro.
 - Step 1: Activate (assert) the rst reset signal through the register setting of the CPG unit.
 - Step 2: Stop supply of the pa_audiox_a clock through the register setting of the CPG unit.
 - Step 3: Release (deassert) the rst reset signal through the register setting of the CPG unit.
 - Step 4: Start supply of the pa_audiox_a clock through the register setting of the CPG unit.

Note: The state immediately after starting the device is as stated below.

clkp clock: Supply stopped state, pa_audiox_a clock: Supply stopped state, rst reset: Reset state

Note: The above steps are performed while the clkp clock is being supplied.

8.6.5 Interrupt

8.6.5.1 Interrupt List

Table 8.6-7 lists the interrupt signal of this unit.

Table 8.6-7 Interrupt List

Interrupt signal	Function	Pulse/Level	Min. Pulse	Active Level	Clock
intreq_spdif_n	SPDIF interrupt signal	Level	—	Low	clkp

The above interrupt signal operates as the logical OR of multiple interrupt sources.

The specifiable interrupt sources are as follows.

- Receiver error interrupt
- Transmitter error interrupt
- User buffer overrun interrupt
- User buffer underrun interrupt
- Clock recovery error interrupt
- Parity error interrupt
- Preamble error interrupt
- Channel status error interrupt
- Audio buffer overrun interrupt
- Audio buffer underrun interrupt
- Receiver user information interrupt
- Transmitter user information interrupt
- Receiver channel status interrupt
- Receiver channel buffer interrupt
- Transmitter channel status interrupt
- Transmitter channel buffer interrupt

Switching the interrupt sources listed above on and off can be specified in the control register (CTRL).

Checking the interrupt source when an interrupt occurs and clearing the source can be done through the status register (STAT).

8.6.6 Usage Notes

8.6.6.1 Clearing TUIR

After writing to TUI and until TUIR is cleared, a wait for the completion of the transmission of up to 1 frame is required. When writing data to TUI in response to a transmitter user information interrupt, check that TUIR has been cleared, and then stop the interrupt processing routine to prevent the interrupt being accepted again.

8.6.6.2 Frequency of Audio Input Clock

The frequency of the externally input oversampling clock (pa_audiox_a) requires setting to 512 fs.

In addition, the frequency of the externally input oversampling clock (pa_audiox_a) must be kept lower than the SPDIF internal HPB bus clock (clkp) frequency.

SECTION 8 AUDIO

8.7 PDM Interface (PDM)

This section describes the PDM interface.

8.7.1 Overview

This PDM interface is a unit for receiving voice data from an external PDM (pulse density modulation) device. For features and configuration, see the following sections.

8.7.1.1 Feature

Table 8.7-1 lists the features of this PDM interface.

Table 8.7-1 PDM-IF IP Specifications

Item	Description
Number of channels	Maximum 3 channels per 1 IP (× 2 units)
Functions	• Capable of filtering 1-bit digital input data PDM_DATn (n = 0, 1, 2) and converting them into 20-bit or 16-bit digital data. The bit order is little-endian. • The IP supports stereo microphone (L/R sampling by rising/falling clock edge). • The IP supports sound activity detector. • Each channel of IP includes programmable filters: 4th order sinc filter, high-pass filter (for suppression of DC bias), correction filter (for sinc passband distortion), half-band decimation filter (for aliasing distortion). – IP supports programmable and flexible decimation ratios. – The sinc filter is selectable as first-, second-, third-, and fourth-order. • The IP supports DMA operation through APB. • Each channel of IP has an internal buffer. – 64 stages when designed – Capable of storing voice data during low power mode • Error detection functions can be used for debugging.
Interrupt sources	Maximum 7 sources in 1 IP 1. Data reception interrupt per channel (Maximum 3 interrupts) 2. Sound detection interrupt (shared between channels) 3. Error detection interrupt per channel (Maximum 3 interrupts)
Low power consumption function	The IP supports microphone low power configuration (slower clock). The IP can switch to the higher clock speed after sound detection recognized.
Bus interface	The IP has an APB interface which is confirmed to APB4.

Since this LSI supports $24 \text{ MHz}/5 = 4.8 \text{ MHz CLK (CCLK)}$, the PDM_CLK frequency can be discretely selected in the range of 2.4 MHz to 0.15 MHz for the following target sampling frequencies. For details, see **8.7.3.1 Clock Specifications**.

Target sampling frequencies: 48, 40, 30, 25, 24, 20, 16, 15, 12, 10, 8 kHz

8.7.1.2 Block Diagram

The following is a block diagram of the PDM interface.

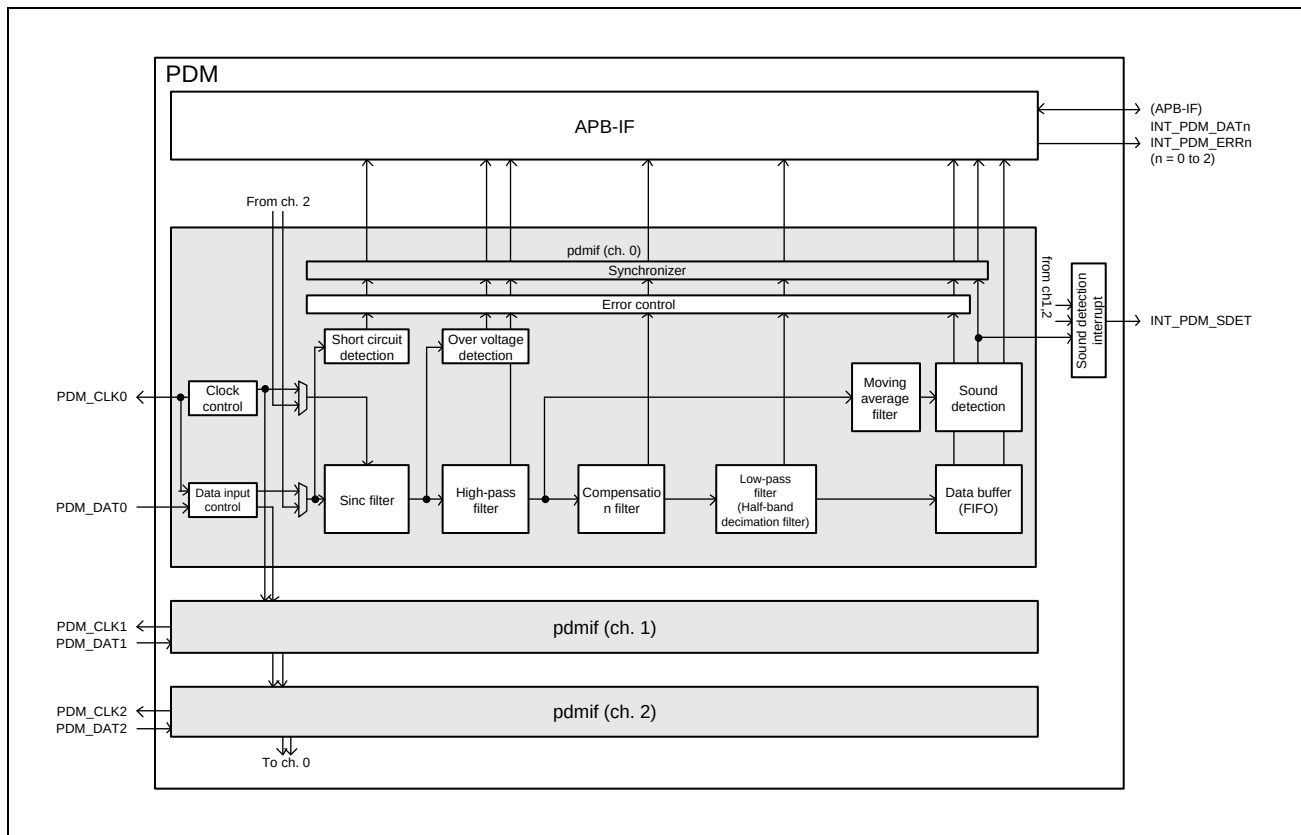


Figure 8.7-1 Three-Channel PDM-IF Block Diagram

Each channel of PDM-IF consists of following blocks:

- Clock control: Generates PDM_CLK_n ($n = 0, 1, 2$) to the microphone, and internal timing signals to the sinc filter.
- Data input control: Receives PDM_DAT_n ($n = 0, 1, 2$) from the microphone.
- Sinc filter: Converts the 1-bit digital data stream input from PDM_DAT_n ($n = 0, 1, 2$) to 34-bit signed data. Then, the 34-bit signed data is clipped to 20-bit signed data.
- High-pass filter: A filter for suppression of DC bias
- Compensation filter: A filter for sinc passband distortion
- Low-pass (half-band decimation) filter: A filter for aliasing distortion
- Moving average filter: A simple filter for noise reduction before sound detection
- Sound detection: Generates the wakeup interrupt to recover from low power mode.
- Data buffer: Stores 20-bit or 16-bit data filtered by some filters.
- Short circuit detection: Detects the short circuit error which indicates consecutive 0's or 1's input from PDM_DAT_n ($n = 0, 1, 2$).
- Overvoltage detection: Detects the overvoltage error.
- Error control: Generates error flags from other blocks.

The 1-bit digital data stream input from PDM_DATn pin (n = 0, 1, 2) is converted to 34-bit signed data by the sinc filter and then the data is clipped to 20-bit signed data. The 20-bit signed data is used for sound detection and data storing in buffer.

The sound detection block receives 20-bit signed data through high-pass filter block and moving average filter block. The block compares received data to the programmed thresholds and generates a wakeup interrupt in case that the data is above the upper threshold or below the lower threshold.

The data buffer block receives 20-bit signed data through high-pass filter, compensation filter, and low-pass filter. The block stores the 20-bit signed data or the clipped 16-bit signed data. And then the block sends the 20-bit signed data or 16-bit signed data to APB-IF once APB read access occurs.

8.7.1.3 External Pins

Table 8.7-2 lists the PDM external pin specifications.

Table 8.7-2 Pin Configuration

External Pin Name	Input/Output	Function	Pin Name in This Section
PDMDATn0	Input	External PDM CH0 data reception pin	PDM_DAT0
PDMDATn1	Input	External PDM CH1 data reception pin	PDM_DAT1
PDMDATn2	Input	External PDM CH2 data reception pin	PDM_DAT2
PDMCLKn0	Output	External PDM CH0 clock output pin	PDM_CLK0
PDMCLKn1	Output	External PDM CH1 clock output pin	PDM_CLK1
PDMCLKn2	Output	External PDM CH2 clock output pin	PDM_CLK2

Note: n = 0, 1

8.7.1.4 Connected Unit

Table 8.7-3 lists the connected units.

Table 8.7-3 Connected Unit

Unit Name	Connected Unit Name	Function of Connected Unit
PDMm	CPG	Clock pulse generator (CPG)
	SYSTEMBUS	System bus
	ICU	Interrupt control unit (ICU)
	PFC	Pin function controller (PFC)

8.7.2 Registers

The register addresses of PDM are given as offsets from the individual base addresses <PDMm_base>. The register base addresses of each PDM are listed in the following table.

Table 8.7-4 Register Base Addresses

Base Address Name	Unit	Base Address
<PDM0_base>	PDM0	0_1104_0000h (5104_0000h* ¹ , 4104_0000h* ²)
<PDM1_base>	PDM1	0_1105_0000h (5105_0000h* ¹ , 4105_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

8.7.2.1 List of Registers

The list of PDM registers and the memory addresses are as follows.

For the actual addresses, the offset values indicated in the following table are added to the base addresses.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]* ¹
Channel Software Start Trigger Register	PDMm_PDCSTRTR	0000_0000h	0000h	8, 16, 32
Channel Software Stop Trigger Register	PDMm_PDCSTPTR	0000_0000h	0004h	8, 16, 32
Channel Software Change Trigger Register	PDMm_PDCCHGTR	0000_0000h	0008h	8, 16, 32
Channel Interrupt Control Register	PDMm_PDCICR	0000_0000h	000Ch	8, 16, 32
Channel Status Register	PDMm_PDCSR	0000_0000h	0010h	8, 16, 32
Channel Status Clear Register	PDMm_PDCSCR	0000_0000h	0014h	8, 16, 32
Reserve	-	-	0018h to 001Fh	-
Channel Sound Detection Control Register	PDMm_PDCSDCR	0000_0000h	0020h	8, 16, 32
Channel Data Read Control Register	PDMm_PDCDRCR	0000_0000h	0024h	8, 16, 32
Channel Data Clear Register	PDMm_PDCDCR	0000_0000h	0028h	8, 16, 32
Reserve	-	-	002Ch to 007Fh	-
Version Register	PDMm_PDVR	0000_0010h	0080h	8, 16, 32
Reserve	-	-	0084h to 00FFh	-
Software Start Trigger Register Channel 0	PDMm_PDSTRTRCH0	0000_0000h	0100h	8, 16, 32
Software Stop Trigger Register Channel 0	PDMm_PDSTPTRCH0	0000_0000h	0104h	8, 16, 32
Software Change Trigger Register Channel 0	PDMm_PDCHGTRCH0	0000_0000h	0108h	8, 16, 32
Interrupt Control Register Channel 0	PDMm_PDICRCH0	0000_0000h	010Ch	8, 16, 32
Status Detection Control Register Channel 0	PDMm_PDSDCRCH0	0000_0000h	0110h	8, 16, 32
Status Register Channel 0	PDMm_PDSRCH0	0000_0000h	0114h	8, 16, 32
Status Clear Register Channel 0	PDMm_PDSCRCH0	0000_0000h	0118h	8, 16, 32
Reserve	-	-	011Ch to 011Fh	-
Mode Setting Register Channel 0	PDMm_PDMSRCH0	0000_0000h	0120h	8, 16, 32
Sinc filter Control Register Channel 0	PDMm_PDSFCRCH0	057C_0000h	0124h	8, 16, 32
High-pass filter Coefficient s(0) Register Channel 0	PDMm_PDHFCs0RCH0	0000_3F61h	0128h	8, 16, 32
High-pass filter Coefficient k(1) Register Channel 0	PDMm_PDHFK1RCH0	0000_3EC1h	012Ch	8, 16, 32
High-pass filter Coefficient h(0) Register Channel 0	PDMm_PDHfCH0RCH0	0000_4000h	0130h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
High-pass filter Coefficient h(1) Register Channel 0	PDMm_PDHFCH1RCH0	0000_C000h	0134h	8, 16, 32
Compensation filter Coefficient h(0) Register Channel 0	PDMm_PDCFCH00RCH0	0000_1FE8h	0138h	8, 16, 32
Compensation filter Coefficient h(1) Register Channel 0	PDMm_PDCFCH01RCH0	0000_0039h	013Ch	8, 16, 32
Compensation filter Coefficient h(2) Register Channel 0	PDMm_PDCFCH02RCH0	0000_003Ch	0140h	8, 16, 32
Compensation filter Coefficient h(3) Register Channel 0	PDMm_PDCFCH03RCH0	0000_1E56h	0144h	8, 16, 32
Compensation filter Coefficient h(4) Register Channel 0	PDMm_PDCFCH04RCH0	0000_01DCh	0148h	8, 16, 32
Compensation filter Coefficient h(5) Register Channel 0	PDMm_PDCFCH05RCH0	0000_06E1h	014Ch	8, 16, 32
Compensation filter Coefficient h(6) Register Channel 0	PDMm_PDCFCH06RCH0	0000_01DCh	0150h	8, 16, 32
Compensation filter Coefficient h(7) Register Channel 0	PDMm_PDCFCH07RCH0	0000_1E56h	0154h	8, 16, 32
Compensation filter Coefficient h(8) Register Channel 0	PDMm_PDCFCH08RCH0	0000_003Ch	0158h	8, 16, 32
Compensation filter Coefficient h(9) Register Channel 0	PDMm_PDCFCH09RCH0	0000_0039h	015Ch	8, 16, 32
Compensation filter Coefficient h(10) Register Channel 0	PDMm_PDCFCH10RCH0	0000_1FE8h	0160h	8, 16, 32
Low-pass filter Coefficient h0(10) Register Channel 0	PDMm_PDLFCH010RCH0	0000_0400h	0164h	8, 16, 32
Low-pass filter Coefficient h1(0) Register Channel 0	PDMm_PDLFCH10RCH0	0000_1FF8h	0168h	8, 16, 32
Low-pass filter Coefficient h1(1) Register Channel 0	PDMm_PDLFCH101RCH0	0000_000Ah	016Ch	8, 16, 32
Low-pass filter Coefficient h1(2) Register Channel 0	PDMm_PDLFCH102RCH0	0000_1FF0h	0170h	8, 16, 32
Low-pass filter Coefficient h1(3) Register Channel 0	PDMm_PDLFCH103RCH0	0000_0018h	0174h	8, 16, 32
Low-pass filter Coefficient h1(4) Register Channel 0	PDMm_PDLFCH104RCH0	0000_1FDCh	0178h	8, 16, 32
Low-pass filter Coefficient h1(5) Register Channel 0	PDMm_PDLFCH105RCH0	0000_0034h	017Ch	8, 16, 32
Low-pass filter Coefficient h1(6) Register Channel 0	PDMm_PDLFCH106RCH0	0000_1FB3h	0180h	8, 16, 32
Low-pass filter Coefficient h1(7) Register Channel 0	PDMm_PDLFCH107RCH0	0000_0076h	0184h	8, 16, 32
Low-pass filter Coefficient h1(8) Register Channel 0	PDMm_PDLFCH108RCH0	0000_1F2Eh	0188h	8, 16, 32
Low-pass filter Coefficient h1(9) Register Channel 0	PDMm_PDLFCH109RCH0	0000_0289h	018Ch	8, 16, 32
Low-pass filter Coefficient h1(10) Register Channel 0	PDMm_PDLFCH110RCH0	0000_0289h	0190h	8, 16, 32
Low-pass filter Coefficient h1(11) Register Channel 0	PDMm_PDLFCH111RCH0	0000_1F2Eh	0194h	8, 16, 32
Low-pass filter Coefficient h1(12) Register Channel 0	PDMm_PDLFCH112RCH0	0000_0076h	0198h	8, 16, 32
Low-pass filter Coefficient h1(13) Register Channel 0	PDMm_PDLFCH113RCH0	0000_1FB3h	019Ch	8, 16, 32
Low-pass filter Coefficient h1(14) Register Channel 0	PDMm_PDLFCH114RCH0	0000_0034h	01A0h	8, 16, 32
Low-pass filter Coefficient h1(15) Register Channel 0	PDMm_PDLFCH115RCH0	0000_1FDCh	01A4h	8, 16, 32
Low-pass filter Coefficient h1(16) Register Channel 0	PDMm_PDLFCH116RCH0	0000_0018h	01A8h	8, 16, 32
Low-pass filter Coefficient h1(17) Register Channel 0	PDMm_PDLFCH117RCH0	0000_1FF0h	01ACh	8, 16, 32
Low-pass filter Coefficient h1(18) Register Channel 0	PDMm_PDLFCH118RCH0	0000_000Ah	01B0h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Low-pass filter Coefficient h1(19) Register Channel 0	PDMm_PDLFCH119RCH0	0000_1FF8h	01B4h	8, 16, 32
Sound Detection Lower Threshold Register Channel 0	PDMm_PSDLTRCH0	0000_0000h	01B8h	8, 16, 32
Sound Detection Upper Threshold Register Channel 0	PDMm_PSDUTRCH0	0000_0000h	01BCh	8, 16, 32
Data Buffer Control Register Channel 0	PDMm_PDDBCRCH0	0000_0000h	01C0h	8, 16, 32
Short Circuit Threshold Setting Register Channel 0	PDMm_PDSCSRCH0	0000_0000h	01C4h	8, 16, 32
Overvoltage Lower Threshold Register Channel 0	PDMm_PDOVLTRCH0	0000_0000h	01C8h	8, 16, 32
Overvoltage Upper Threshold Register Channel 0	PDMm_PDOVUTRCH0	0000_0000h	01CCh	8, 16, 32
Reserve	-	-	01D0h to 01DFh	-
Data Read Control Register Channel 0	PDMm_PDDRCRCH0	0000_0000h	01E0h	8, 16, 32
Data Clear Register Channel 0	PDMm_PDDCRCH0	0000_0000h	01E4h	8, 16, 32
Data Read Register Channel 0	PDMm_PDDRRCH0	0000_0000h	01E8h	8, 16, 32
Data Status Register Channel 0	PDMm_PDDSRCH0	0000_0000h	01ECh	8, 16, 32
Reserve	-	-	01F0h to 01FFh	-
Software Start Trigger Register Channel 1	PDMm_PDSTRTRCH1	0000_0000h	0200h	8, 16, 32
Software Stop Trigger Register Channel 1	PDMm_PDSTPTRCH1	0000_0000h	0204h	8, 16, 32
Software Change Trigger Register Channel 1	PDMm_PDCHGTRCH1	0000_0000h	0208h	8, 16, 32
Interrupt Control Register Channel 1	PDMm_PDICRCH1	0000_0000h	020Ch	8, 16, 32
Status Detection Control Register Channel 1	PDMm_PSDCRCH1	0000_0000h	0210h	8, 16, 32
Status Register Channel 1	PDMm_PDSRCH1	0000_0000h	0214h	8, 16, 32
Status Clear Register Channel 1	PDMm_PDSCRCH1	0000_0000h	0218h	8, 16, 32
Reserve	-	-	021Ch to 021Fh	-
Mode Setting Register Channel 1	PDMm_PMDSRCH1	0000_0000h	0220h	8, 16, 32
Sinc filter Control Register Channel 1	PDMm_PDSFCRCH1	057C_0000h	0224h	8, 16, 32
High-pass filter Coefficient s(0) Register Channel 1	PDMm_PDHFCS0RCH1	0000_3F61h	0228h	8, 16, 32
High-pass filter Coefficient k(1) Register Channel 1	PDMm_PDHFCK1RCH1	0000_3EC1h	022Ch	8, 16, 32
High-pass filter Coefficient h(0) Register Channel 1	PDMm_PDHFCH0RCH1	0000_4000h	0230h	8, 16, 32
High-pass filter Coefficient h(1) Register Channel 1	PDMm_PDHFCH1RCH1	0000_C000h	0234h	8, 16, 32
Compensation filter Coefficient h(0) Register Channel 1	PDMm_PDCFCH00RCH1	0000_1FE8h	0238h	8, 16, 32
Compensation filter Coefficient h(1) Register Channel 1	PDMm_PDCFCH01RCH1	0000_0039h	023Ch	8, 16, 32
Compensation filter Coefficient h(2) Register Channel 1	PDMm_PDCFCH02RCH1	0000_003Ch	0240h	8, 16, 32
Compensation filter Coefficient h(3) Register Channel 1	PDMm_PDCFCH03RCH1	0000_1E56h	0244h	8, 16, 32
Compensation filter Coefficient h(4) Register Channel 1	PDMm_PDCFCH04RCH1	0000_01DCh	0248h	8, 16, 32
Compensation filter Coefficient h(5) Register Channel 1	PDMm_PDCFCH05RCH1	0000_06E1h	024Ch	8, 16, 32
Compensation filter Coefficient h(6) Register Channel 1	PDMm_PDCFCH06RCH1	0000_01DCh	0250h	8, 16, 32
Compensation filter Coefficient h(7) Register Channel 1	PDMm_PDCFCH07RCH1	0000_1E56h	0254h	8, 16, 32
Compensation filter Coefficient h(8) Register Channel 1	PDMm_PDCFCH08RCH1	0000_003Ch	0258h	8, 16, 32
Compensation filter Coefficient h(9) Register Channel 1	PDMm_PDCFCH09RCH1	0000_0039h	025Ch	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Compensation filter Coefficient h(10) Register Channel 1	PDMm_PDCFCH10RCH1	0000_1FE8h	0260h	8, 16, 32
Low-pass filter Coefficient h0(10) Register Channel 1	PDMm_PDLFCH010RCH1	0000_0400h	0264h	8, 16, 32
Low-pass filter Coefficient h1(0) Register Channel 1	PDMm_PDLFCH10RCH1	0000_1FF8h	0268h	8, 16, 32
Low-pass filter Coefficient h1(1) Register Channel 1	PDMm_PDLFCH101RCH1	0000_000Ah	026Ch	8, 16, 32
Low-pass filter Coefficient h1(2) Register Channel 1	PDMm_PDLFCH102RCH1	0000_1FF0h	0270h	8, 16, 32
Low-pass filter Coefficient h1(3) Register Channel 1	PDMm_PDLFCH103RCH1	0000_0018h	0274h	8, 16, 32
Low-pass filter Coefficient h1(4) Register Channel 1	PDMm_PDLFCH104RCH1	0000_1FDCh	0278h	8, 16, 32
Low-pass filter Coefficient h1(5) Register Channel 1	PDMm_PDLFCH105RCH1	0000_0034h	027Ch	8, 16, 32
Low-pass filter Coefficient h1(6) Register Channel 1	PDMm_PDLFCH106RCH1	0000_1FB3h	0280h	8, 16, 32
Low-pass filter Coefficient h1(7) Register Channel 1	PDMm_PDLFCH107RCH1	0000_0076h	0284h	8, 16, 32
Low-pass filter Coefficient h1(8) Register Channel 1	PDMm_PDLFCH108RCH1	0000_1F2Eh	0288h	8, 16, 32
Low-pass filter Coefficient h1(9) Register Channel 1	PDMm_PDLFCH109RCH1	0000_0289h	028Ch	8, 16, 32
Low-pass filter Coefficient h1(10) Register Channel 1	PDMm_PDLFCH110RCH1	0000_0289h	0290h	8, 16, 32
Low-pass filter Coefficient h1(11) Register Channel 1	PDMm_PDLFCH111RCH1	0000_1F2Eh	0294h	8, 16, 32
Low-pass filter Coefficient h1(12) Register Channel 1	PDMm_PDLFCH112RCH1	0000_0076h	0298h	8, 16, 32
Low-pass filter Coefficient h1(13) Register Channel 1	PDMm_PDLFCH113RCH1	0000_1FB3h	029Ch	8, 16, 32
Low-pass filter Coefficient h1(14) Register Channel 1	PDMm_PDLFCH114RCH1	0000_0034h	02A0h	8, 16, 32
Low-pass filter Coefficient h1(15) Register Channel 1	PDMm_PDLFCH115RCH1	0000_1FDCh	02A4h	8, 16, 32
Low-pass filter Coefficient h1(16) Register Channel 1	PDMm_PDLFCH116RCH1	0000_0018h	02A8h	8, 16, 32
Low-pass filter Coefficient h1(17) Register Channel 1	PDMm_PDLFCH117RCH1	0000_1FF0h	02ACh	8, 16, 32
Low-pass filter Coefficient h1(18) Register Channel 1	PDMm_PDLFCH118RCH1	0000_000Ah	02B0h	8, 16, 32
Low-pass filter Coefficient h1(19) Register Channel 1	PDMm_PDLFCH119RCH1	0000_1FF8h	02B4h	8, 16, 32
Sound Detection Lower Threshold Register Channel 1	PDMm_PDSDLTRCH1	0000_0000h	02B8h	8, 16, 32
Sound Detection Upper Threshold Register Channel 1	PDMm_PDSDUTRCH1	0000_0000h	02BCh	8, 16, 32
Data Buffer Control Register Channel 1	PDMm_PDDBCRCH1	0000_0000h	02C0h	8, 16, 32
Short Circuit Threshold Setting Register Channel 1	PDMm_PDSCTSRCH1	0000_0000h	02C4h	8, 16, 32
Overvoltage Lower Threshold Register Channel 1	PDMm_PDOVLTRCH1	0000_0000h	02C8h	8, 16, 32
Overvoltage Upper Threshold Register Channel 1	PDMm_PDOVUTRCH1	0000_0000h	02CCh	8, 16, 32
Reserve	-	-	02D0h to 02DFh	-
Data Read Control Register Channel 1	PDMm_PDDRCRCH1	0000_0000h	02E0h	8, 16, 32
Data Clear Register Channel 1	PDMm_PDDCRCH1	0000_0000h	02E4h	8, 16, 32
Data Read Register Channel 1	PDMm_PDDRRCH1	0000_0000h	02E8h	8, 16, 32
Data Status Register Channel 1	PDMm_PDDSRCH1	0000_0000h	02ECh	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]*1
Reserve	-	-	02F0h to 02FFh	-
Software Start Trigger Register Channel 2	PDMm_PDSTRTRCH2	0000_0000h	0300h	8, 16, 32
Software Stop Trigger Register Channel 2	PDMm_PDSTPTRCH2	0000_0000h	0304h	8, 16, 32
Software Change Trigger Register Channel 2	PDMm_PDCHGTRCH2	0000_0000h	0308h	8, 16, 32
Interrupt Control Register Channel 2	PDMm_PDICRCH2	0000_0000h	030Ch	8, 16, 32
Status Detection Control Register Channel 2	PDMm_PSDSCRCH2	0000_0000h	0310h	8, 16, 32
Status Register Channel 2	PDMm_PDSRCH2	0000_0000h	0314h	8, 16, 32
Status Clear Register Channel 2	PDMm_PDSCRCH2	0000_0000h	0318h	8, 16, 32
Reserve	-	-	031Ch to 031Fh	-
Mode Setting Register Channel 2	PDMm_PMDSRCH2	0000_0000h	0320h	8, 16, 32
Sinc filter Control Register Channel 2	PDMm_PDSFCRCH2	057C_0000h	0324h	8, 16, 32
High-pass filter Coefficient s(0) Register Channel 2	PDMm_PDHFCS0RCH2	0000_3F61h	0328h	8, 16, 32
High-pass filter Coefficient k(1) Register Channel 2	PDMm_PDHFCK1RCH2	0000_3EC1h	032Ch	8, 16, 32
High-pass filter Coefficient h(0) Register Channel 2	PDMm_PDHFCH0RCH2	0000_4000h	0330h	8, 16, 32
High-pass filter Coefficient h(1) Register Channel 2	PDMm_PDHFCH1RCH2	0000_C000h	0334h	8, 16, 32
Compensation filter Coefficient h(0) Register Channel 2	PDMm_PDCFCH00RCH2	0000_1FE8h	0338h	8, 16, 32
Compensation filter Coefficient h(1) Register Channel 2	PDMm_PDCFCH01RCH2	0000_0039h	033Ch	8, 16, 32
Compensation filter Coefficient h(2) Register Channel 2	PDMm_PDCFCH02RCH2	0000_003Ch	0340h	8, 16, 32
Compensation filter Coefficient h(3) Register Channel 2	PDMm_PDCFCH03RCH2	0000_1E56h	0344h	8, 16, 32
Compensation filter Coefficient h(4) Register Channel 2	PDMm_PDCFCH04RCH2	0000_01DCh	0348h	8, 16, 32
Compensation filter Coefficient h(5) Register Channel 2	PDMm_PDCFCH05RCH2	0000_06E1h	034Ch	8, 16, 32
Compensation filter Coefficient h(6) Register Channel 2	PDMm_PDCFCH06RCH2	0000_01DCh	0350h	8, 16, 32
Compensation filter Coefficient h(7) Register Channel 2	PDMm_PDCFCH07RCH2	0000_1E56h	0354h	8, 16, 32
Compensation filter Coefficient h(8) Register Channel 2	PDMm_PDCFCH08RCH2	0000_003Ch	0358h	8, 16, 32
Compensation filter Coefficient h(9) Register Channel 2	PDMm_PDCFCH09RCH2	0000_0039h	035Ch	8, 16, 32
Compensation filter Coefficient h(10) Register Channel 2	PDMm_PDCFCH10RCH2	0000_1FE8h	0360h	8, 16, 32
Low-pass filter Coefficient h0(10) Register Channel 2	PDMm_PDLFCH010RCH2	0000_0400h	0364h	8, 16, 32
Low-pass filter Coefficient h1(0) Register Channel 2	PDMm_PDLFCH10RCH2	0000_1FF8h	0368h	8, 16, 32
Low-pass filter Coefficient h1(1) Register Channel 2	PDMm_PDLFCH101RCH2	0000_000Ah	036Ch	8, 16, 32
Low-pass filter Coefficient h1(2) Register Channel 2	PDMm_PDLFCH102RCH2	0000_1FF0h	0370h	8, 16, 32
Low-pass filter Coefficient h1(3) Register Channel 2	PDMm_PDLFCH103RCH2	0000_0018h	0374h	8, 16, 32
Low-pass filter Coefficient h1(4) Register Channel 2	PDMm_PDLFCH104RCH2	0000_1FDCh	0378h	8, 16, 32
Low-pass filter Coefficient h1(5) Register Channel 2	PDMm_PDLFCH105RCH2	0000_0034h	037Ch	8, 16, 32
Low-pass filter Coefficient h1(6) Register Channel 2	PDMm_PDLFCH106RCH2	0000_1FB3h	0380h	8, 16, 32
Low-pass filter Coefficient h1(7) Register Channel 2	PDMm_PDLFCH107RCH2	0000_0076h	0384h	8, 16, 32
Low-pass filter Coefficient h1(8) Register Channel 2	PDMm_PDLFCH108RCH2	0000_1F2Eh	0388h	8, 16, 32
Low-pass filter Coefficient h1(9) Register Channel 2	PDMm_PDLFCH109RCH2	0000_0289h	038Ch	8, 16, 32
Low-pass filter Coefficient h1(10) Register Channel 2	PDMm_PDLFCH110RCH2	0000_0289h	0390h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]* ¹
Low-pass filter Coefficient h1(11) Register Channel 2	PDMm_PDLFCH111RCH2	0000_1F2Eh	0394h	8, 16, 32
Low-pass filter Coefficient h1(12) Register Channel 2	PDMm_PDLFCH112RCH2	0000_0076h	0398h	8, 16, 32
Low-pass filter Coefficient h1(13) Register Channel 2	PDMm_PDLFCH113RCH2	0000_1FB3h	039Ch	8, 16, 32
Low-pass filter Coefficient h1(14) Register Channel 2	PDMm_PDLFCH114RCH2	0000_0034h	03A0h	8, 16, 32
Low-pass filter Coefficient h1(15) Register Channel 2	PDMm_PDLFCH115RCH2	0000_1FDCh	03A4h	8, 16, 32
Low-pass filter Coefficient h1(16) Register Channel 2	PDMm_PDLFCH116RCH2	0000_0018h	03A8h	8, 16, 32
Low-pass filter Coefficient h1(17) Register Channel 2	PDMm_PDLFCH117RCH2	0000_1FF0h	03ACh	8, 16, 32
Low-pass filter Coefficient h1(18) Register Channel 2	PDMm_PDLFCH118RCH2	0000_000Ah	03B0h	8, 16, 32
Low-pass filter Coefficient h1(19) Register Channel 2	PDMm_PDLFCH119RCH2	0000_1FF8h	03B4h	8, 16, 32
Sound Detection Lower Threshold Register Channel 2	PDMm_PDSLTRCH2	0000_0000h	03B8h	8, 16, 32
Sound Detection Upper Threshold Register Channel 2	PDMm_PDSUTRCH2	0000_0000h	03BCh	8, 16, 32
Data Buffer Control Register Channel 2	PDMm_PDDBCRCH2	0000_0000h	03C0h	8, 16, 32
Short Circuit Threshold Setting Register Channel 2	PDMm_PDSCTSRCH2	0000_0000h	03C4h	8, 16, 32
Overvoltage Lower Threshold Register Channel 2	PDMm_PDOVLTRCH2	0000_0000h	03C8h	8, 16, 32
Overvoltage Upper Threshold Register Channel 2	PDMm_PDOVUTRCH2	0000_0000h	03CCh	8, 16, 32
Reserve	-	-	03D0h to 03DFh	-
Data Read Control Register Channel 2	PDMm_PDDRCRCH2	0000_0000h	03E0h	8, 16, 32
Data Clear Register Channel 2	PDMm_PDDCRCH2	0000_0000h	03E4h	8, 16, 32
Data Read Register Channel 2	PDMm_PDDRRCH2	0000_0000h	03E8h	8, 16, 32
Data Status Register Channel 2	PDMm_PDDSRCH2	0000_0000h	03ECh	8, 16, 32

Note 1. The read access size is fixed to 32 bits.

8.7.2.2 Register Description

The prefix (PDMm_) of the register names is omitted in this and subsequent sections.

8.7.2.2.1 Channel Software Start Trigger Register (PDMm_PDCSTRTR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0000h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	STRTRG[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W	R0W	R0W
Bit	Bit Name															
31 to 3	-		All0		R0		Reserved									
	Whenever it is read, 0b is read. The written value will be ignored.															
2 to 0	STRTRG[2:0]		0h		R0W		The read value is always 0b.									
	Channel n start trigger															
	0b: Do nothing															
	1b: Start channel n															

Note: The read access size is fixed to 32 bits.

8.7.2.2.2 Channel Software Stop Trigger Register (PDMm_PDCSTPTR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	STPTRG[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W	R0W	R0W
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	STPTRG[2:0]		0h		R0W		The read value is always 0b. Channel n stop trigger 0b: Do nothing 1b: Stop channel n									

Note: The read access size is fixed to 32 bits.

8.7.2.2.3 Channel Software Change Trigger Register (PDMm_PDCCHGTR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0008h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	CHGTRG[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0W	R0W	R0W	R0	R0	R0	R0	R0	R0W	R0W	R0W
Bit	Bit Name		Initial Value		R/W		Description									
31 to 11	-		All0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
10 to 8	-		All0		R0W		Reserved. This register is readable and the initial value can be written to it.									
7 to 3	-		All0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	CHGTRG[2:0]		0h		R0W		The read value is always 0b. Channel n change trigger 0b: Do nothing 1b: Change clock (PDM_CLKn) setting									

Note: The read access size is fixed to 32 bits.

8.7.2.2.4 Channel Interrupt Control Register (PDMm_PDCICR) (m = 0, 1)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 000Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	IEDE[2:0]			-	-	-	-	-	IDRE[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	RW	RW	RW	R0	R0	R0	R0	R0	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	ISDE[2:0]			-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	RW	RW	RW	R0	R0	R0	R0	R0	R0	R0	R0

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26 to 24	IEDE[2:0]	0h	RW	Channel n error detection interrupt enable bit 0b: Don't allow to issue INT_PDM_ERRn interrupt 1b: Allow to issue INT_PDM_ERRn interrupt
23 to 19	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18 to 16	IDRE[2:0]	0h	RW	Channel n data reception interrupt enable bit 0b: Don't allow to issue INT_PDM_DATn interrupt 1b: Allow to issue INT_PDM_DATn interrupt
15 to 11	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10 to 8	ISDE[2:0]	0h	RW	Channel n sound detection interrupt enable bit 0b: Don not allow to issue INT_PDM_SDET interrupt when the sound for channel n is detected. 1b: Allow to issue INT_PDM_SDET interrupt when the sound for channel n is detected.
7 to 0	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Note: The read access size is fixed to 32 bits.

8.7.2.2.5 Channel Status Register (PDMm_PDCSR) (m = 0, 1)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	EDF[2:0]			-	-	-	-	-	DRF[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R	R	R	R0	R0	R0	R0	R0	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SDF[2:0]			-	-	-	-	-	STATE[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R	R	R	R0	R0	R0	R0	R0	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26 to 24	EDF[2:0]	0h	R	Channel n error detection flag 0b: Indicates that no data is detected. 1b: Indicates that an error is detected. Refer to the PDSRCHn register for details of errors.
23 to 19	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18 to 16	DRF[2:0]	0h	R	Channel n data reception flag 0b: Indicates that the number of data stored in buffer does not exceed the threshold. 1b: Indicates that the number of data stored in buffer exceeded the threshold.
15 to 11	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10 to 8	SDF[2:0]	0h	R	Channel n sound detection flag 0b: Indicates that the sound which exceeded the threshold is not detected. 1b: Indicates that the sound which exceeded the threshold is detected.
7 to 3	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	STATE[2:0]	0h	R	Channel n state 0b: Channel n stop 1b: Channel n in operation

Note: The read access size is fixed to 32 bits.

EDFn

[Condition to become “1b”]

- When any error occurs. Refer to the PDSRCHn register for the kind of error.

[Condition to become “0b”]

- When all error flags are cleared by writing the PDSCRCHn, this flag is also cleared.

DRFn

Refer to **8.7.3.7 Data Buffer** for details.

[Condition to become “1b”]

- When PDCDRCR.DATREn = 1b and PDDSRCHn.DATNUM[7:0] is greater than or equal to a threshold which is configured by PDDBCRCn.DATRITHR[2:0].

[Condition to become “0b”]

- When PDCDRCR.DATREn = 0b or PDDSRCHn.DATNUM[7:0] is less than a threshold.

SDFn

Refer to **8.7.3.6 Sound Detection** for details.

[Conditions to become “1b”]

- When PDCSDCR.SDEn = 1b and a moving average filter result is greater than or equal to PDSUTRCHn.SDETU[19:0].
- When PDCSDCR.SDEn = 1b and a moving average filter result is less than or equal to PSDLTRCHn.SDETL[19:0].

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDCSCR.SDFCn).

STATEn

Refer to **8.7.3.2 Channel Start and Channel Stop** for details.

[Condition to become “1b”]

- When “1b” is written to PDCSTRTR.STRTRGn.

[Condition to become “0b”]

- When “1b” is written to PDCSTPTR.STPTRGn.

8.7.2.2.6 Channel Status Clear Register (PDMm_PDCSCR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SDFC[2:0]		-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0W	R0W	R0W	R0	R0	R0	R0	R0	R0	R0	R0

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10 to 8	SDFC[2:0]	0h	R0W	The read value is always 0b. Channel n sound detection flag clear 0b: Do nothing 1b: Clear PDCSR.SDFn
7 to 0	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Note: The read access size is fixed to 32 bits.

8.7.2.2.7 Channel Sound Detection Control Register (PDMm_PDCSDCR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDE[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 3	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
2 to 0	SDE[2:0]		0h	RW	Channel n sound detection enable bit 0b: Do not allow to detect sound 1b: Allow to detect sound											

Note: The read access size is fixed to 32 bits.

8.7.2.2.8 Channel Data Read Control Register (PDMm_PDCDRCR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0024h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DATRE[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	DATRE[2:0]		0h		RW		Channel n data read enable bit 0b: Do not allow to read data from data buffer 1b: Allow to read data from data buffer									

Note: The read access size is fixed to 32 bits.

8.7.2.2.9 Channel Data Clear Register (PDMm_PDCDCR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0028h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DATC[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W	R0W	R0W
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	DATC[2:0]		0h		R0W		The read value is always 0b. Channel n data clear 0b: Do nothing 1b: Clear data									

Note: The read access size is fixed to 32 bits.

8.7.2.2.10 Version Register (PDMm_PDVR) (m = 0, 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0080h														
Initial Value :		0000_0010h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	VER[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W		Description									
31 to 12	-		All 0		R0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
11 to 0	VER[11:0]		10h		R		IP version is shown.									

Note: The read access size is fixed to 32 bits.

8.7.2.2.11 Software Start Trigger Register Channel n (PDMm_PDSTRTRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0100h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STRTR G
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W
Bit	Bit Name		Initial Value	R/W	Description											
31 to 1	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
0	STRTRG		0h	R0W	The read value is always 0b. Start trigger 0b: Do nothing 1b: Start the channel											

Note: The read access size is fixed to 32 bits.

8.7.2.2.12 Software Stop Trigger Register Channel n (PDMm_PDSTPTRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0104h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STPTR G
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W
Bit	Bit Name		Initial Value	R/W	Description											
31 to 1	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
0	STPTRG		0h	R0W	The read value is always 0b. Stop trigger 0b: Do nothing 1b: Stop the channel											

Note: The read access size is fixed to 32 bits.

8.7.2.2.13 Software Change Trigger Register Channel n (PDMm_PDCHGTRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0108h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CHGTR G
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0W	R0	R0	R0	R0	R0	R0	R0	R0W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	-	0h	R0W	Reserved This register is readable and the initial value can be written to it.
7 to 1	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	CHGTRG	0h	R0W	The read value is always 0b. Change trigger 0b: Do nothing 1b: Change settings of PDM_CLKn and sinc filter

Note: The read access size is fixed to 32 bits.

8.7.2.2.14 Interrupt Control Register Channel n (PDMm_PDICRCHn) (m = 0 to 1, n = 0 to 2)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 010Ch + n × 0100h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IEDE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IDRE	ISDE	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	R0

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	IEDE	0h	RW	Error detection interrupt enable bit 0b: Do not allow to issue INT_PDM_ERRn interrupt 1b: Allow to issue INT_PDM_ERRn interrupt
15 to 3	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	IDRE	0h	RW	Data reception interrupt enable bit 0b: Do not allow to issue INT_PDM_DATn interrupt 1b: Allow to issue INT_PDM_DATn interrupt
1	ISDE	0h	RW	Sound detection interrupt enable bit 0b: Do not allow to issue INT_PDM_SDET interrupt 1b: Allow to issue INT_PDM_SDET interrupt
0	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Note: The read access size is fixed to 32 bits.

8.7.2.2.15 Status Detection Control Register Channel n (PDMm_PDSDCRCHn) (m = 0 to 1, n = 0 to 2)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0110h + n × 0100h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	BFLOW DE	-	-	-	-	-	-	-	-	OVUDE	OVLDE	SCDE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SDE	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	R0

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	BFLOWDE	0h	RW	Buffer overwriting detection enable bit 0b: Do not allow to detect buffer overwriting. 1b: Allow to detect buffer overwriting.
26 to 19	-	All 0	RW	Reserved This register is readable and the initial value can be written to it.
18	OVUDE	0h	RW	Overvoltage upper limit exceeded detection enable bit 0b: Do not allow to detect data exceeding the upper limit. 1b: Allow to detect data exceeding the upper limit
17	OVLDE	0h	RW	Overvoltage lower limit exceeded detection enable bit 0b: Do not allow to detect data falling below the lower limit. 1b: Allow to detect data falling below the lower limit
16	SCDE	0h	RW	Short circuit detection enable bit 0b: Do not allow to detect the PDM_DATn pin short circuit 1b: Allow to detect the PDM_DATn pin short circuit
15 to 2	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	SDE	0h	RW	Sound detection enable bit 0b: Do not allow to detect sound 1b: Allow to detect sound
0	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Note: The read access size is fixed to 32 bits.

8.7.2.2.16 Status Register Channel n (PDMm_PDSRCHn) (m = 0 to 1, n = 0 to 2)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0114h + n × 0100h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	BFWDF	-	-	-	-	-	-	-	-	OVUDF	OVLDF	SCDF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DRF	SDF	STATE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	BFWDF	0h	R	Buffer overwriting detection flag 0b: Indicates that buffer overwriting does not occur. 1b: Indicates that buffer overwriting occurred.
26 to 19	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18	OVUDF	0h	R	Overvoltage upper limit exceeded detection flag 0b: Indicates that the data is not above the upper limit 1b: Indicates that the data has exceeded the upper limit
17	OVLDF	0h	R	Overvoltage lower limit exceeded detection flag 0b: Indicates that the data is not below the lower limit 1b: Indicates that the data has falling below the lower limit
16	SCDF	0h	R	Short circuit detection flag 0b: Indicates that the PDM_DATn pin short circuit is not detected 1b: Indicates that a short circuit on the PDM_DATn pin has been detected
15 to 3	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	DRF	0h	R	Data reception flag 0b: Indicates that the number of data stored in buffer is less than a threshold 1b: Indicates that the number of data stored in buffer is greater than or equal to a threshold
1	SDF	0h	R	Sound detection flag 0b: Indicates that the sound is not detected 1b: Indicates that the sound is detected
0	STATE	0h	R	State 0b: Channel stop 1b: Channel in operation

Note: The read access size is fixed to 32 bits.

BFWDF

Refer to **8.7.3.7 Data Buffer** for details.

[Condition to become “1b”]

- When PDSRCHn.BFWDF = 1b and data buffer is full and data is written to the buffer without reading.

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDSRCHn.BFWDFC).

OVUDF

Refer to **8.7.3.9 Overvoltage Detection** for details.

[Condition to become “1b”]

- When PDSDCRCHn.OVUDE = 1b and a clipped sinc filter result is larger than an upper limit (PDOVUTRCHn.OVDU[19:0]).

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDSCRCHn.OVUDFC).

OVLDF

Refer to **8.7.3.9 Overvoltage Detection** for details.

[Condition to become “1b”]

- When PDSDCRCHn.OVLDE = 1b and a clipped sinc filter result is smaller than a lower limit (PDOVLTRCHn.OVDL[19:0]).

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDSCRCHn.OVLDFC).

SCDF

Refer to **8.7.3.8 Short-Circuit Detection** for details.

[Conditions to become “1b”]

- When PDSDCRCHn.SCDE = 1b and 1b is continuously input to the PDM_DATn pin, exceeding the number of times set in PDSCTSRCHn.SCDH[12:0].
- When PDSDCRCHn.SCDE = 1b and 0b is continuously input to the PDM_DATn pin, exceeding the number of times set in PDSCTSRCHn.SCDL[12:0].

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDSCRCHn.SCDFC).

DRF

Refer to **8.7.3.7 Data Buffer** for details.

[Condition to become “1b”]

- When PDDRCRCHn.DATRE = 1b and PDDSRCHn.DATNUM[7:0] is greater than or equal to a threshold which is configured by PDDBCRCn.DATRITHR[2:0].

[Condition to become “0b”]

- When PDDRCRCHn.DATRE = 0b or PDDSRCHn.DATNUM[7:0] is less than a threshold.

SDF

Refer to **8.7.3.6 Sound Detection** for details.

[Conditions to become “1b”]

- When PDSDCRCHn.SDE = 1b and a moving average filter result is PDSDUTRCHn.SDETU[19:0] or greater.
- When PDSDCRCHn.SDE = 1b and a moving average filter result is PSDLTRCHn.SDETL[19:0] or less.

[Condition to become “0b”]

- When writing 1b to a flag clear register bit (PDSCRCHn.SDFC).

STATE

Refer to **8.7.3.2 Channel Start and Channel Stop** for details.

[Condition to become “1b”]

- When “1b” is written to PDSTRTRCHn.STRTRG.

[Condition to become “0b”]

- When “1b” is written to PDSTPTRCHn.STPTRG.

8.7.2.2.17 Status Clear Register Channel n (PDMm_PDSCRCHn) (m = 0 to 1, n = 0 to 2)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0118h + n × 0100h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	BFOW DFC	-	-	-	-	-	-	-	-	OVUDF C	OVLDF C	SCDFC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SDFC	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W	R0

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	BFOWDFC	0h	R0W	The read value is always 0b. Buffer overwriting detection flag clear 0b: Do nothing 1b: Clear PDSRCHn.BFOWDF
26 to 19	-	All 0	R0W	Reserved This register is readable and the initial value can be written to it.
18	OVUDFC	0h	R0W	The read value is always 0b. Overvoltage upper limit exceeded detection flag clear 0b: Do nothing 1b: Clear PDSRCHn.OVUDF
17	OVLDFC	0h	R0W	The read value is always 0b. Overvoltage lower limit exceeded detection flag clear 0b: Do nothing 1b: Clear PDSRCHn.OVLDF
16	SCDFC	0h	R0W	The read value is always 0b. Short circuit detection flag clear 0b: Do nothing 1b: Clear PDSRCHn.SCDF
15 to 2	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	SDFC	0h	R0W	The read value is always 0b. Sound detection flag clear 0b: Do nothing 1b: Clear PDSRCHn.SDF
0	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Note: The read access size is fixed to 32 bits.

8.7.2.2.18 Mode Setting Register Channel n (PDMm_PDMSRCHn) (m = 0 to 1, n = 0 to 2)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0120h + n × 0100h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DBIS[3:0]				-	-	SDMAMD[1:0]		-	-	-	-	-	-	LFIS[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	R0	R0	RW	RW	R0	R0	R0	RW	R0	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CFIS[1:0]		-	-	HFIS[1:0]		-	SFMD[2:0]			-	-	-	INPSEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	RW	RW	RW	R0	RW	RW	RW	R0	RW	RW	RW	R0	R0	R0	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	DBIS[3:0]	0h	RW	Data buffer input shift setting 0000b: 20-bit mode, {1{S},[18:0]} 0001b: 20-bit mode, {2{S},[18:1]} 0010b: 20-bit mode, {3{S},[18:2]} 0011b: 20-bit mode, {4{S},[18:3]} (0100b-0111b: Setting prohibited) 1000b: 16-bit mode, {S,D[18:4]} 1001b: 16-bit mode, {S,D[17:3]} 1010b: 16-bit mode, {S,D[16:2]} 1011b: 16-bit mode, {S,D[15:1]} 1100b: 16-bit mode, {S,D[14:0]} (1101b-1111b: Setting prohibited) *S: Sign bit
27, 26	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25, 24	SDMAMD[1:0]	0h	RW	Moving average mode for sound detection data 00b: 1-order (filter is skipped) (default) 01b: 2-order 10b: 4-order 11b: Setting prohibited
23 to 21	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	-	0h	RW	Reserved This register is readable and the initial value can be written to it.
19	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18	-	0h	RW	Reserved This register is readable and the initial value can be written to it.
17, 16	LFIS[1:0]	0h	RW	Low-pass (half-band decimation) filter input shift setting 00b: No shift 01b: 1-bit right shift 10b: 2-bit right shift 11b: 3-bit right shift
15	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14	-	0h	RW	Reserved This register is readable and the initial value can be written to it.
13, 12	CFIS[1:0]	0h	RW	Compensation filter input shift setting 00b: No shift 01b: 1-bit right shift 10b: 2-bit right shift 11b: 3-bit right shift

Bit	Bit Name	Initial Value	R/W	Description
11	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10	-	0h	RW	Reserved This register is readable and the initial value can be written to it.
9, 8	HFIS[1:0]	0h	RW	High-pass filter input shift setting 00b: No shift 01b: 1-bit right shift 10b: 2-bit right shift 11b: 3-bit right shift
7	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 4	SFMD[2:0]	0h	RW	Sinc filter mode setting 001b: 1-order 010b: 2-order 011b: 3-order Other: 4-order (default)
3 to 1	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	INPSEL	0h	RW	Input data select 0b: Rise-edge data of channel n 1b: Fall-edge data of channel n-1 (In case of n = 0, fall-edge data of channel 2 is selected.)

Note: The read access size is fixed to 32 bits.

8.7.2.2.19 Sinc Filter Control Register Channel n (PDMm_PDSFCRCHn) (m = 0 to 1, n = 0 to 2)

See **Table 8.7-8** for the setting values of SINCDEC[7:0] and SINCRNG[4:0].

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0124h + n × 0100h														
Initial Value :		057C_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	SINCRNG[4:0]					SINCDEC[7:0]							
Initial Value	0	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0
R/W	R0	R0	R0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	CKDIV[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 24	SINCRNG[4:0]		5h	RW	Sinc filter output valid range 00000b: {S, [32:14]} 10000b: {S, [16:0],00b} 00001b: {S, [31:13]} 10001b: {S, [15:0],000b} 00010b: {S, [30:12]} 10010b: {S, [14:0],0000b} 00011b: {S, [29:11]} 10011b: {S, [13:0],00000b} 00100b: {S, [28:10]} 10100b: {S, [12:0],000000b} 00101b: {S, [27:9]} (default) 10101b: {S, [11:0],0000000b} 00110b: {S, [26:8]} 10110b: {S, [10:0],00000000b} 00111b: {S, [25:7]} 10111b: {S, [9:0],000000000b} 01000b: {S, [24:6]} 11000b: {S, [8:0],0000000000b} 01001b: {S, [23:5]} 11001b: {S, [7:0],00000000000b} 01010b: {S, [22:4]} 11010b: {S, [6:0],000000000000b} 01011b: {S, [21:3]} 11011b: {S, [5:0],0000000000000b} 01100b: {S, [20:2]} 11100b: {S, [4:0],00000000000000b} 01101b: {S, [19:1]} 11101b: {S, [3:0],000000000000000b} 01110b: {S, [18:0]} 11110b: {S, [2:0],0000000000000000b} 01111b: {S, [17:0], 0b} 11111b: {S, [1:0],0000000000000000b} (prohibited) S: Signed bit. When the overflow occurs, 7FFFFh is output. When the underflow occurs, 80000h is output.											
23 to 16	SINCDEC[7:0]		7Ch	RW	Sinc filter decimation ratio M indicates the decimation ratio. Decimation ratio M is set as follows. M = SINCDEC + 1 00h: M = 1 (prohibited) 01h: M = 2 (prohibited) 02h: M = 3 (prohibited) 03h: M = 4 ... 7Ch: M = 125 (default) ... FFh: M = 256 Setting a value less than M = 4 is prohibited. If such a value is set, operation of the sinc filter is not guaranteed. The setting should satisfy D*M > 12. Otherwise, operation of other filters is not guaranteed. (D: PDM_CLKn's division ratio)											
15 to 4	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CKDIV[3:0]	0h	RW	PDM_CLKn's division ratio to core clock 0000b: 1/2 (default) 0001b: 1/4 0010b: 1/6 ... 1101b: 1/28 1110b: 1/30 1111b: 1/32

Note: The read access size is fixed to 32 bits.

8.7.2.2.20 High-Pass Filter Coefficient s(0) Register Channel n (PDMm_PDHFCS0RCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0128h + n × 0100h														
Initial Value :		0000_3F61h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HFSm[15:0]															
Initial Value	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	HFSm[15:0]	3F61h	RW	High-pass filter coefficient s0 16-bit signed data (Fraction bit [13:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
HFS0	s(0)	3F61h	0.99029541015625	Cut-off 100 Hz

8.7.2.2.21 High-Pass Filter Coefficient k(1) Register Channel n (PDMm_PDHFCK1RCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 012Ch + n × 0100h														
Initial Value :		0000_3EC1h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HFKm[15:0]															
Initial Value	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	HFKm[15:0]	3EC1h	RW	High-pass filter coefficient k1 16-bit signed data (Fraction bit [13:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
HFK1	k(1)	3EC1h	0.98052978515625	Cut-off 100 Hz

8.7.2.2.22 High-Pass Filter Coefficient h(p) Register Channel n (PDMm_PDHFCHpRCHn)
(m = 0 to 1, n = 0 to 2, p = 0 to 1)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0130h + n × 0100h + p × 0004h														
Initial Value :		0000_xxxxh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HFH0[15:0]															
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	HFH0[15:0]	All x	RW	High-pass filter coefficient h(p) 16-bit signed data (Fraction bit [13:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
HFH0	h(0)	4000h	1	
HFH1	h(1)	C000h	-1	

8.7.2.2.23 Compensation Filter Coefficient h(p) Register Channel n (PDMm_PDCFCHpRCHn) (m = 0 to 1, n = 0 to 2, p = 00 to 10)

Access Size : 8, 16, 32 bits

Address : <PDMm_base> + 0138h + n × 0100h + p × 0004h

Initial Value : 0000_xxxxh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	CFH00[12:0]												
Initial Value	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R0	R0	R0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	CFH00[12:0]	All x	RW	Compensation filter coefficients h(p) 13-bit signed data (Fraction bit [10:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
CFH00	h(0)	1FE8h	-0.01171875	
CFH01	h(1)	0039h	0.02783203125	
CFH02	h(2)	003Ch	0.029296875	
CFH03	h(3)	1E56h	-0.2080078125	
CFH04	h(4)	01DCh	0.232421875	
CFH05	h(5)	06E1h	0.85986328125	
CFH06	h(6)	01DCh	0.232421875	
CFH07	h(7)	1E56h	-0.2080078125	
CFH08	h(8)	003Ch	0.029296875	
CFH09	h(9)	0039h	0.02783203125	
CFH10	h(10)	1FE8h	-0.01171875	

8.7.2.2.24 Low-Pass Filter Coefficient h0(10) Register Channel n
(PDMm_PDLFCH010RCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0164h + n × 0100h														
Initial Value :		0000_0400h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	LFH010[12:0]												
Initial Value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	LFH010[12:0]	0400h	RW	Low-pass (half-band decimation) filter coefficient h0(m) 13-bit signed data (Fraction bit [10:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
LFH010	h0(10)	0400h	0.5	Cut-off 7 kHz

8.7.2.2.25 Low-Pass Filter Coefficient h1(p) Register Channel n (PDMm_PDLFCH1pRCHn) (m = 0 to 1, n = 0 to 2, p = 0 to 19)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 0168h + n × 0100h + p × 0004h														
Initial Value :		0000_xxxxh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	LFH100[12:0]												
Initial Value	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R0	R0	R0	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

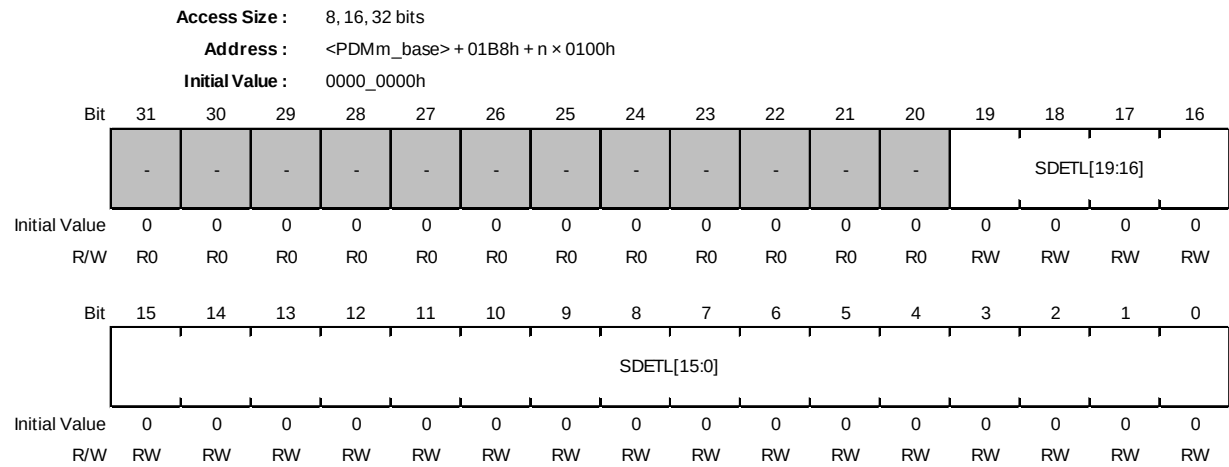
Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	LFH100[12:0]	All x	RW	Low-pass (half-band decimation) filter coefficient h1(p) 13-bit signed data (Fraction bit [10:0]) For the initial value, see Value after Reset of Each Coefficient

Note: The read access size is fixed to 32 bits.

● Value after Reset of Each Coefficient

Symbol Name	Coefficient Name	Value after Reset		Comment
		Hex	Decimal	
LFH100	h1(0)	1FF8h	-0.00390625	Cut-off 7 kHz
LFH101	h1(1)	000Ah	0.0048828125	
LFH102	h1(2)	1FF0h	-0.0078125	
LFH103	h1(3)	0018h	0.01171875	
LFH104	h1(4)	1FDCh	-0.017578125	
LFH105	h1(5)	0034h	0.025390625	
LFH106	h1(6)	1FB3h	-0.03759765625	
LFH107	h1(7)	0076h	0.0576171875	
LFH108	h1(8)	1F2Eh	-0.1025390625	
LFH109	h1(9)	0289h	0.31689453125	
LFH110	h1(10)	0289h	0.31689453125	
LFH111	h1(11)	1F2Eh	-0.1025390625	
LFH112	h1(12)	0076h	0.0576171875	
LFH113	h1(13)	1FB3h	-0.03759765625	
LFH114	h1(14)	0034h	0.025390625	
LFH115	h1(15)	1FDCh	-0.017578125	
LFH116	h1(16)	0018h	0.01171875	
LFH117	h1(17)	1FF0h	-0.0078125	
LFH118	h1(18)	000Ah	0.0048828125	
LFH119	h1(19)	1FF8h	-0.00390625	

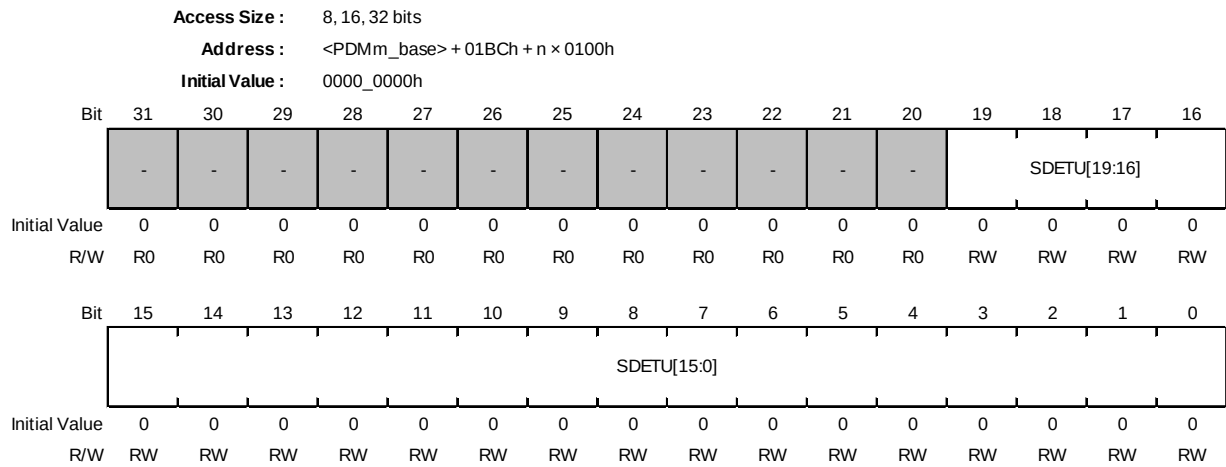
8.7.2.2.26 Sound Detection Lower Threshold Register Channel n (PDMm_PDSDLTRCHn)
(m = 0 to 1, n = 0 to 2)



Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 0	SDETL[19:0]	0h	RW	Sound detection lower limit This bit sets the lower limit value (20-bit signed data) for sound detection.

Note: The read access size is fixed to 32 bits.

8.7.2.2.27 Sound Detection Upper Threshold Register Channel n (PDMm_PDSDUTRCHn)
(m = 0 to 1, n = 0 to 2)



Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 0	SDETU[19:0]	0h	RW	Sound detection upper limit This bit sets the upper limit value (20-bit signed data) for sound detection.

Note: The read access size is fixed to 32 bits.

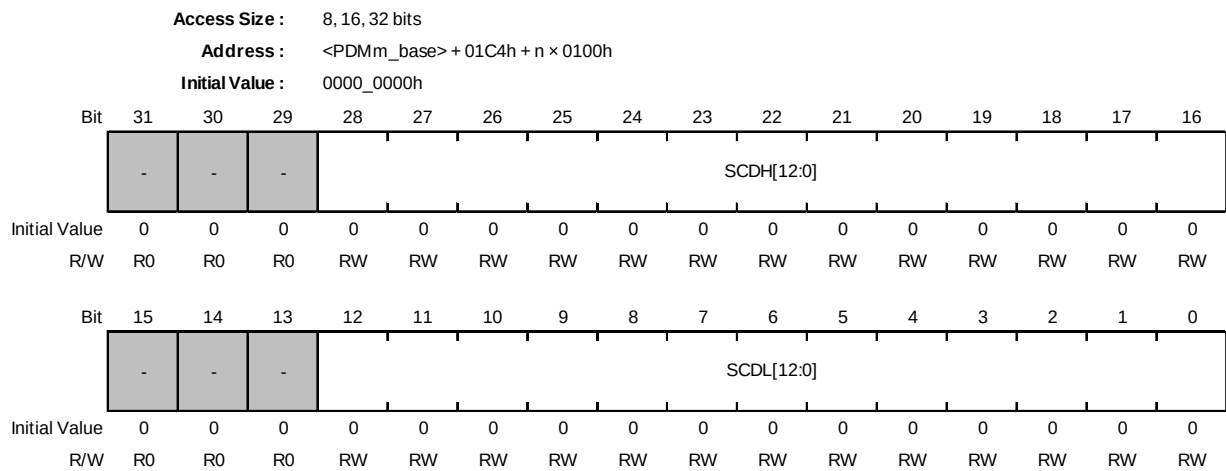
8.7.2.2.28 Data Buffer Control Register Channel n (PDMm_PDDBCRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01C0h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DATRITHR[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	R0	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	-	0h	RW	Reserved This register is readable and the initial value can be written to it.
3	-	0h	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	DATRITHR [2:0]	0h	RW	Data reception interrupt threshold 000b: Output interrupt when receiving 1 or more data 001b: Output interrupt when receiving 2 or more data 010b: Output interrupt when receiving 4 or more data 011b: Output interrupt when receiving 8 or more data others: Output interrupt when receiving 16 or more data

Note: The read access size is fixed to 32 bits.

8.7.2.2.29 Short Circuit Threshold Setting Register Channel n (PDMm_PDSCTSRCHn) (m = 0 to 1, n = 0 to 2)



Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	SCDH[12:0]	0h	RW	Short circuit detection High Continuous detection count Set the upper limit for the number of times "1" are input continuously to the PDM_DATn pin.
15 to 13	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	SCDL[12:0]	0h	RW	Continuous detection count Set the upper limit for the number of times "0" are input continuously to the PDM_DATn pin.

Note: The read access size is fixed to 32 bits.

8.7.2.2.30 Overvoltage Lower Threshold Register Channel n (PDMm_PDOVLTRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01C8h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	OVDL[19:16]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVDL[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 0	OVDL[19:0]	0h	RW	Overvoltage detection lower limit These bits set the lower limit of the active voltage value (20-bit signed data) for overvoltage detection.

Note: The read access size is fixed to 32 bits.

8.7.2.2.31 Overvoltage Upper Threshold Register Channel n (PDMm_PDOVUTRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01CCh + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	OVDU[19:16]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVDU[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 0	OVDU[19:0]	0h	RW	Overvoltage detection upper limit These bits set the upper limit of the active voltage value (20-bit signed data) for overvoltage detection.

Note: The read access size is fixed to 32 bits.

8.7.2.2.32 Data Read Control Register Channel n (PDMm_PDDRCRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01E0h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DATRE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All0		R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	DATRE		0h		RW	Data read enable bit 0b: Do not allow to read data from buffer 1b: Allow to read data from buffer										

Note: The read access size is fixed to 32 bits.

8.7.2.2.33 Data Clear Register Channel n (PDMm_PDDCRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01E4h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DATC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0W
Bit	Bit Name		Initial Value	R/W	Description											
31 to 1	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
0	DATC		0h	R0W	The read value is always 0b. Data clear 0b: Do nothing 1b: Clear data											

Note: The read access size is fixed to 32 bits.

8.7.2.2.34 Data Read Register Channel n (PDMm_PDDRRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01E8h + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	DAT[19:16]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R0	R0	R0	R0	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAT[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 20	-	All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 0	DAT[19:0]	0h	R	Data 20-bit signed sound data or 16-bit signed sound data is read out. In case of 20-bit mode (PDMSRCHn.DBIS[3:0] = 0xxb), the signed bit is [19]. In case of 16-bit mode (PDMSRCHn.DBIS[3:0] = 1xxb), the signed bits are [19:15]. Writing is invalid.

Note: The read access size is fixed to 32 bits.

8.7.2.235 Data Status Register Channel n (PDMm_PDDSRCHn) (m = 0 to 1, n = 0 to 2)

Access Size :		8, 16, 32 bits														
Address :		<PDMm_base> + 01ECh + n × 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	DATNUM[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value	R/W	Description											
31 to 8	-		All0	R0	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
7 to 0	DATNUM[7:0]		0h	R	Number of data stored in buffer When PDDRCRCHn.DATRE = 0b, 0b is read out. When PDDRCRCHn.DATRE = 1b, the number of data stored in buffer is read out. In case that buffer size is 64 stages, the maximum value is 63.											

Note: The read access size is fixed to 32 bits.

8.7.2.3 Mirroring Register

The common registers except for the version register have mirroring registers in each channel. PDCSR.EDFn is a register which ORs all the error flags of the mirroring registers in each channel.

Use the common registers for synchronous control between channels.

Table 8.7-5 Mirroring Register

Common Register		Mirroring Register in Each Channel	
Register Name	Bit Name	Register Name	Bit Name
PDCSTRTR	STRTRGn	PDSTRTRCHn	STRTRG
PDCSTPTR	STPTRGn	PDSTPTRCHn	STPTRG
PDCCHGTR	CHGTRGn	PDCHGTRCHn	CHGTRG
PDCICR	ISDEn	PDICRCHn	ISDE
	IDREn		IDRE
	IEDEn		IEDE
PDCSR	STATEn	PDSRCHn	STATE
	SDFn		SDF
	DRFn		DRF
	EDFn		SCDF or OVLDF or OVUDF or BFOWDF
PDCSCR	SDFCn	PDSCRCHn	SDFC
PDCSDCR	SDEn	PDSDCRCHn	SDE
PDCDRCR	DATREn	PDDRCRCHn	DATRE
PDCDCR	DATCn	PDDCRCHn	DATC

8.7.3 Operation

This section describes the PDM basic control method. See **8.7.4.6 Setting PDM_CLKn (n = 0, 1, 2) Frequency and Sampling Frequency Generated by the Unit** for information on setting the PDM_CLKn (n = 0, 1, 2) frequency and the sampling frequency generated by the PDM unit, and see **8.7.4.7 DMA Transfer Method** for information on transferring data with the DMA master.

8.7.3.1 Clock Specifications

The frequency-divided core clock is generated and output to PDM_CLKn.

Table 8.7-6 lists the guaranteed clock frequency and clock mode in PDM-IF. Core clock (CCLK) and PCLK are asynchronous.

Table 8.7-6 Guaranteed Clock Frequency and Clock Mode

CCLK	PCLK PCLK_SFR	PDM_CLKn		
Frequency (MHz)	Frequency (MHz)	Frequency (MHz)	Supported Core Clock Division Ratio	Setting Value (PDSFCRCHn.CKDIV[3:0])
4.8	1 to 100	2.4	2	0000b
		1.2	4	0001b
		0.8	6	0010b
		0.6	8	0011b
		0.48	10	0100b
		0.4	12	0101b
		0.34	14	0110b
		0.3	16	0111b
		0.27	18	1000b
		0.24	20	1001b
		0.22	22	1010b
		0.2	24	1011b
		0.18	26	1100b
		0.17	28	1101b
		0.16	30	1110b
		0.15	32	1111b

8.7.3.2 Channel Start and Channel Stop

The channel start and the channel stop is controlled by the registers listed in **Table 8.7-7**.

Set the channels to be synchronized to 1 at the same time by using the common registers.

Set the channel start according to **8.7.4.1 Start Flow**.

Set the channel stop according to **8.7.4.2 Stop Flow**.

Table 8.7-7 Channel Start Trigger and Channel Stop Trigger

Category	Register Bit Name	
	Common Register	Channel Register
Channel start	PDCSTRTR.STRTRG0	PDSTRTRCH0.STRTRG
	PDCSTRTR.STRTRG1	PDSTRTRCH1.STRTRG
	PDCSTRTR.STRTRG2	PDSTRTRCH2.STRTRG
Channel stop	PDCSTPTR.STPTRG0	PDSTPTRCH0.STPTRG
	PDCSTPTR.STPTRG1	PDSTPTRCH1.STPTRG
	PDCSTPTR.STPTRG2	PDSTPTRCH2.STPTRG

8.7.3.3 PDM_CLKn's Frequency Change

The external microphone transitions the power mode according to the input clock frequency. PDM-IF enables the transition of microphone's power mode by changing PDM_CLKn's operating frequency which is output to the microphone when operating. The PDM_CLKn's operating frequency can be changed by using the PDSFCRCHn.CKDIV[3:0] bits. When the frequency is changed, the decimation ratio of the sinc filter also needs to be changed. Set PDSFCRCHn.SINCDEC[7:0] in accordance with the sampling frequency after the decimation. Furthermore, when the decimation ratio of the sinc filter is changed, its output range is changed. Set PDSFCRCHn.SINCRNG[4:0] together in accordance with the output range. Refer to **Table 8.7-8**, which lists example settings when the sampling frequency after the decimation is 30 kHz*¹.

For the method of calculating the set values of PDSFCRCHn.CKDIV[3:0], PDSFCRCHn.SINCDEC[7:0], and PDSFCRCHn.SINCRNG[4:0] for the expected sampling frequency of PDM_CLKn, refer to **8.7.4.6 Setting PDM_CLKn (n = 0, 1, 2) Frequency and Sampling Frequency Generated by the Unit**.

After changing the configuration of these registers, software needs to wait for the settling time of microphone and PDM-IF channel. Refer to **8.7.4.4 Low Power Mode Transition Flow** and **8.7.4.5 Normal Mode Transition Flow** for details.

Note 1. If the same filter coefficient has been set in filtering after the sinc filter, the characteristics of the filter changes with the change of the sampling frequency.

8.7.3.4 Filter

8.7.3.4.1 Data Clipping and Shifting between Filters

Voice sample data is clipped or shifted between filters. **Figure 8.7-2** shows this operation.

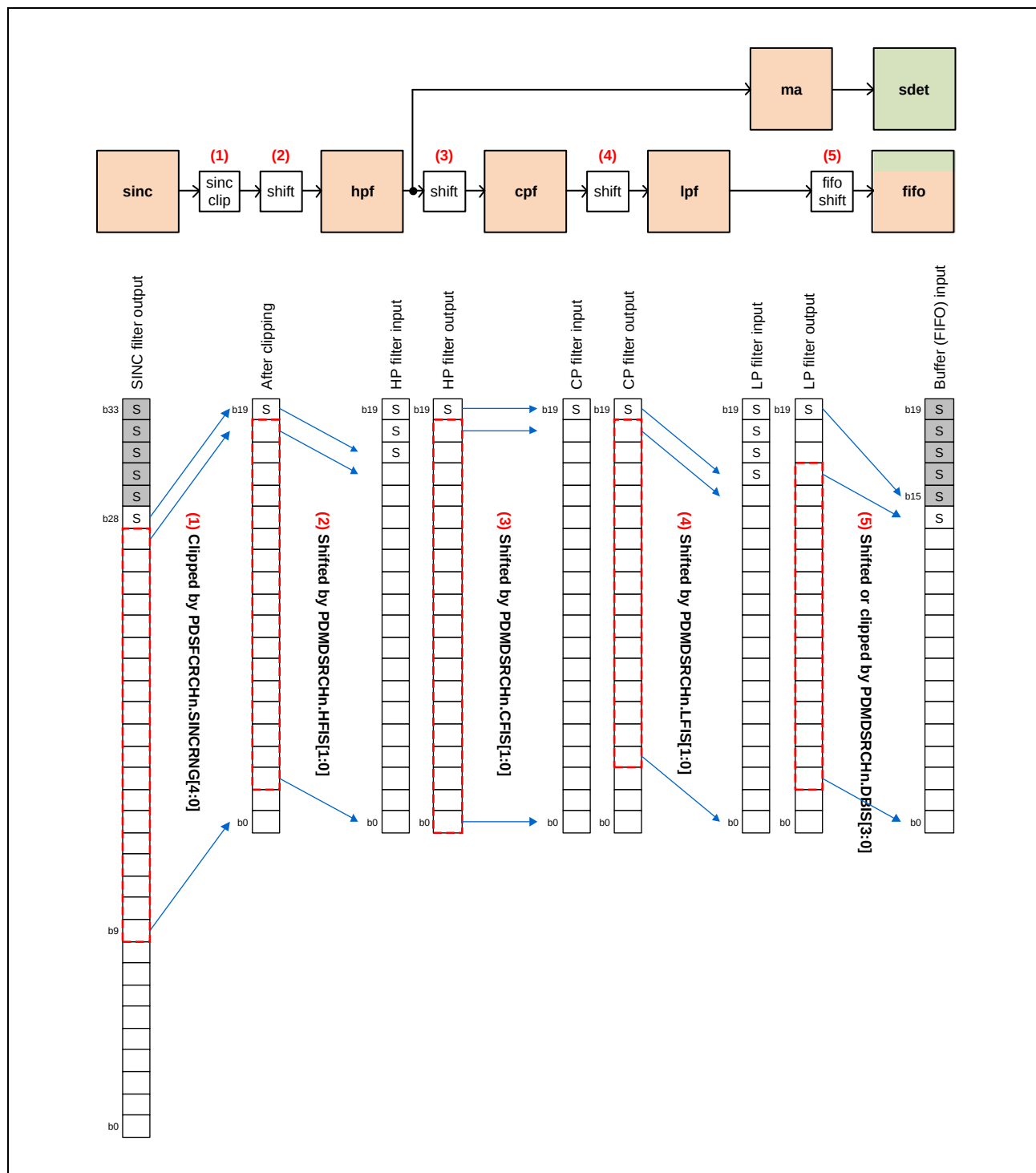


Figure 8.7-2 Data Clipping and Shifting between Filters

a) Sinc filter output clip

It is controlled by PDSFCRCHn.SINCRNG[4:0].

Table 8.7-8 lists the possible combinations of settings when the sampling frequency is 30 kHz.

Table 8.7-8 Sinc Filter Output Clip (1/2)

CCLK (MHz)	Sinc Filter Order	PDMDSRCHn. SFMD[2:0]	PDM_CLK's Division Ratio	PDSFCRCHn. CKDIV[3:0]	Sinc filter Decimation Ratio	PDSFCRCHn. SINCDEC[7:0]	Sinc Filter Output Effective Bits	Sinc Filter Output Clipping (20 bits)	PDSFCRCHn. SINCRNG[4:0]	Resolution	PDM_CLKn Frequency (MHz)	PDMIF Output Sampling Frequency (kHz)
4.8	4	0h	2	0h	40	27h	b22 to b0	b22 to b3	01011b	20	2.4	30
			4	1h	20	13h	b18 to b0	b18 to b0, 0b	01111b	19	1.2	30
			6	2h	13	Ch	b15 to b0	b15 to b0, 0000b	10010b	16	0.8	30.77
			8	3h	10	9h	b14 to b0	b14 to b0, 00000b	10011b	15	0.6	30
			10	4h	8	7h	b13 to b0	b13 to b0, 000000b	10010b	14	0.48	30
			12	5h	6	5h	b11 to b0	b11 to b0, 00000000b	10110b	12	0.4	33.33
			14	6h	6	5h	b11 to b0	b11 to b0, 00000000b	10110b	12	0.34	28.57
			16	7h	5	4h	b10 to b0	b10 to b0, 000000000b	10111b	11	0.3	30
			18	8h	4	3h	b9 to b0	b9 to b0, 0000000000b	11000b	10	0.27	33.33
			20	9h	4	3h	b9 to b0	b9 to b0, 0000000000b	11000b	10	0.24	30
			22	Ah	4	3h	b9 to b0	b9 to b0, 0000000000b	11000b	10	0.22	27.27
4.8	3	3h	2	0h	40	27h	b16 to b0	b16 to b0, 000b	10001b	20	2.4	30
			4	1h	20	13h	b13 to b0	b13 to b0, 000000b	10100b	19	1.2	30
			6	2h	13	Ch	b12 to b0	b12 to b0, 0000000b	10101b	16	0.8	30.77
			8	3h	10	9h	b10 to b0	b10 to b0, 000000000b	10111b	15	0.6	30
			10	4h	8	7h	b10 to b0	b10 to b0, 000000000b	10111b	14	0.48	30
			12	5h	6	5h	b8 to b0	b8 to b0, 0000000000b	11001b	12	0.4	33.33
			14	6h	6	5h	b8 to b0	b8 to b0, 0000000000b	11001b	12	0.34	28.57
			16	7h	5	4h	b7 to b0	b7 to b0, 00000000000b	11010b	11	0.3	30
			18	8h	4	3h	b7 to b0	b7 to b0, 000000000000b	11010b	10	0.27	33.33
			20	9h	4	3h	b7 to b0	b7 to b0, 000000000000b	11010b	10	0.24	30
			22	Ah	4	3h	b7 to b0	b7 to b0, 000000000000b	11010b	10	0.22	27.27
4.8	2	2h	2	0h	40	27h	b11 to b0	b11 to b0, 00000000b	10110b	20	2.4	30
			4	1h	20	13h	b9 to b0	b9 to b0, 0000000000b	11000b	19	1.2	30
			6	2h	13	Ch	b8 to b0	b8 to b0, 00000000000b	11001b	16	0.8	30.77
			8	3h	10	9h	b7 to b0	b7 to b0, 000000000000b	11010b	15	0.6	30
			10	4h	8	7h	b7 to b0	b7 to b0, 000000000000b	11010b	14	0.48	30
			12	5h	6	5h	b6 to b0	b6 to b0, 000000000000b	11011b	12	0.4	33.33
			14	6h	6	5h	b6 to b0	b6 to b0, 000000000000b	11011b	12	0.34	28.57
			16	7h	5	4h	b5 to b0	b5 to b0, 0000000000000b	11100b	11	0.3	30
			18	8h	4	3h	b5 to b0	b5 to b0, 0000000000000b	11100b	10	0.27	33.33
			20	9h	4	3h	b5 to b0	b5 to b0, 0000000000000b	11100b	10	0.24	30
			22	Ah	4	3h	b5 to b0	b5 to b0, 0000000000000b	11100b	10	0.22	27.27

Table 8.7-8 Sinc Filter Output Clip (2/2)

CCLK (MHz)	Sinc Filter Order	PDMSRCHn.SFMD[2:0]	PDM_CLK's Division Ratio	PDSFCRCHn.CKDIV[3:0]	Sinc filter Decimation Ratio	PDSFCRCHn.SINCDEC[7:0]	Sinc Filter Output Effective Bits	Sinc Filter Output Clipping (20 bits)	PDSFCRCHn.SINCRNG[4:0]	Resolution	PDM_CLKn Frequency (MHz)	SINC Filter Output Sampling Frequency (kHz)
4.8	1	1h	2	0h	40	27h	b6 to b0	b6 to b0, 000000000000000b	11011b	20	2.4	30
			4	1h	20	13h	b5 to b0	b5 to b0, 000000000000000b	11100b	19	1.2	30
			6	2h	13	Ch	b4 to b0	b4 to b0, 000000000000000b	11101b	16	0.8	30.77
			8	3h	10	9h	b4 to b0	b4 to b0, 000000000000000b	11101b	15	0.6	30
			10	4h	8	7h	b4 to b0	b4 to b0, 000000000000000b	11101b	14	0.48	30
			12	5h	6	5h	b3 to b0	b3 to b0, 000000000000000b	11110b	12	0.4	33.33
			14	6h	6	5h	b3 to b0	b3 to b0, 000000000000000b	11110b	12	0.34	28.57
			16	7h	5	4h	b3 to b0	b3 to b0, 000000000000000b	11110b	11	0.3	30
			18	8h	4	3h	b3 to b0	b3 to b0, 000000000000000b	11110b	10	0.27	33.33
			20	9h	4	3h	b3 to b0	b3 to b0, 000000000000000b	11110b	10	0.24	30
			22	Ah	4	3h	b3 to b0	b3 to b0, 000000000000000b	11110b	10	0.22	27.27

An overflow or an underflow never occurs as long as the combinations which are shown in **Table 8.7-8** are set to PDMSRCHn.SFMD[2:0], PDSFCRCHn.CKDIV[3:0], PDSFCRCHn.SINCDEC[7:0] and PDSFCRCHn.SINCRNG[4:0]. However, when other combinations are set to them, the overflow or the underflow might occur. When the overflow occurs, 7FFFh is output. When the underflow occurs, 8000h is output. Use the overvoltage detection function if you want to know if the overflow or the underflow occurs.

b) High-pass filter input shift

The clipped sinc filter output is right-shifted and input to the high-pass filter.

It is controlled by PDMSRCHn.HFIS[1:0].

Table 8.7-9 High-Pass Filter Input Shift

PDMSRCHn.HFIS[1:0]	Function	Shifted Data*1
0h	No shift	S, b18-b0
1h	1-bit right-shift	S, S, b18-b1
2h	2-bit right-shift	S, S, S, b18-b2
3h	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

c) Compensation filter input shift

The high-pass filter output is right-shifted and input to the compensation filter.

It is controlled by PDMSRCHn.CFIS[1:0].

Table 8.7-10 Compensation Filter Input Shift

PDMSRCHn.CFIS[1:0]	Function	Shifted Data* ¹
0h	No shift	S, b18-b0
1h	1-bit right-shift	S, S, b18-b1
2h	2-bit right-shift	S, S, S, b18-b2
3h	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

d) Low-pass filter input shift

The compensation filter output is right-shifted and input to the low-pass (half-band decimation) filter.

It is controlled by PDMSRCHn.LFIS[1:0].

Table 8.7-11 Low-Pass Filter Input Shift

PDMSRCHn.LFIS[1:0]	Function	Shifted Data* ¹
0h	No shift	S, b18-b0
1h	1-bit right-shift	S, S, b18-b1
2h	2-bit right-shift	S, S, S, b18-b2
3h	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

e) Buffer input shift or clip

The compensation filter output is right-shifted, or 16-bit data is clipped from the compensation filter output. Then, the shifted or clipped data is input to buffer.

It is controlled by PDMSRCHn.DBIS[3:0].

Table 8.7-12 Buffer Input Shift or Clip

PDMSRCHn.DBIS[3:0]	Function	Valid Data Width	Shifted or Clipped Data* ¹
0000b	No shift	20-bit	S, b18-b0
0001b	1-bit right-shift	20-bit	S, S, b18-b1
0010b	2-bit right-shift	20-bit	S, S, S, b18-b2
0011b	3-bit right-shift	20-bit	S, S, S, S, b18-b3
1000b	Signed bit and b18-b4 is clipped	16-bit	S, b18-b4
1001b	Signed bit and b17-b3 is clipped* ²	16-bit	S, b17-b3
1010b	Signed bit and b16-b2 is clipped* ²	16-bit	S, b16-b2
1011b	Signed bit and b15-b1 is clipped* ²	16-bit	S, b15-b1
1100b	Signed bit and b14-b0 is clipped* ²	16-bit	S, b14-b0

Note 1. S: Signed bit (= bit 19)

Note 2. When an overflow occurs, 7FFFh is input to the buffer.
When an underflow occurs, 8000h is input to the buffer.

8.7.3.4.2 Sinc Filter

The operating clock of the sinc filter is CCLK. Sinc filter is operated by using the 1-bit stream data latched with PDM_CLKn's rise-edge or fall-edge. Refer to **8.7.3.5 Data Input Control for Stereo Sound** for details.

Figure 8.7-3 shows a sinc filter block diagram.

The differentiation stage is operated by decimation clock (the frequency is $1/M$ of PDM_CLKn). M is the decimation ratio and is set in the PDSFCRCHn.SINCDEC[7:0] bits. The filter result is output in register every decimation clock.

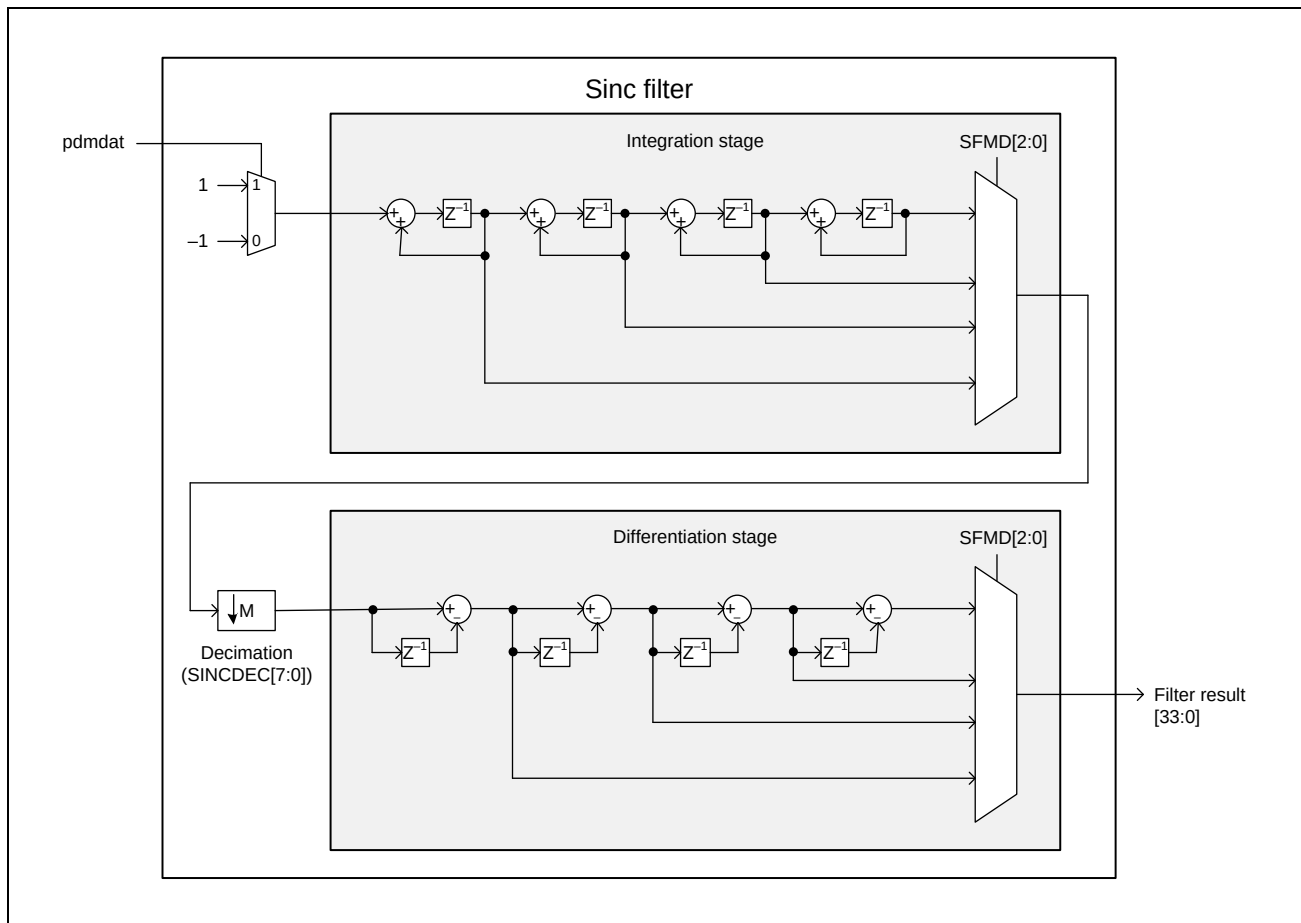


Figure 8.7-3 Sinc Filter Block Diagram

8.7.3.4.3 High-Pass Filter

The operating clock of the high-pass filter is CCLK. The high-pass filter is operated by using the result of the sinc filter.

Figure 8.7-4 shows a high-pass filter block diagram.

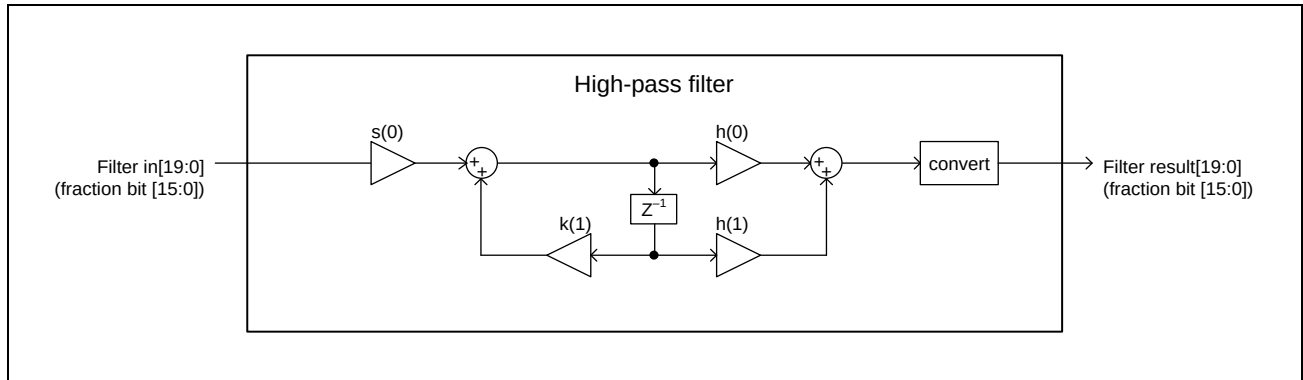


Figure 8.7-4 High-Pass Filter Block Diagram

Table 8.7-13 lists the high-pass filter specification.

Table 8.7-13 High-Pass Filter Specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 16-bit (fraction 14-bit) Refer to 8.7.2.2.20 High-Pass Filter Coefficient s(0) Register Channel n (PDMm_PDHFCS0RCHn) (m = 0 to 1, n = 0 to 2), 8.7.2.2.21 High-Pass Filter Coefficient k(1) Register Channel n (PDMm_PDHFCK1RCHn) (m = 0 to 1, n = 0 to 2), and 8.7.2.2.22 High-Pass Filter Coefficient h(p) Register Channel n (PDMm_PDHFCHpRCHn) (m = 0 to 1, n = 0 to 2, p = 0 to 1) for details.
Delay Z	Bit width	Signed 32-bit (fraction 18-bit)
Multiplier	Input A bit width	Signed 32-bit (fraction 18-bit)
	Input B bit width	Signed 16-bit (fraction 14-bit)
	Output bit width	Signed 47-bit (fraction 32-bit)
Adder	Input A bit width	Signed 32-bit (fraction 18-bit)
	Input B bit width	Signed 32-bit (fraction 18-bit)
	Output bit width	Signed 33-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 32-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output.

8.7.3.4.4 Compensation Filter

The operating clock of the compensation filter is CCLK. The compensation filter is operated by using the result of the high-pass filter.

Figure 8.7-5 shows a compensation filter block diagram.

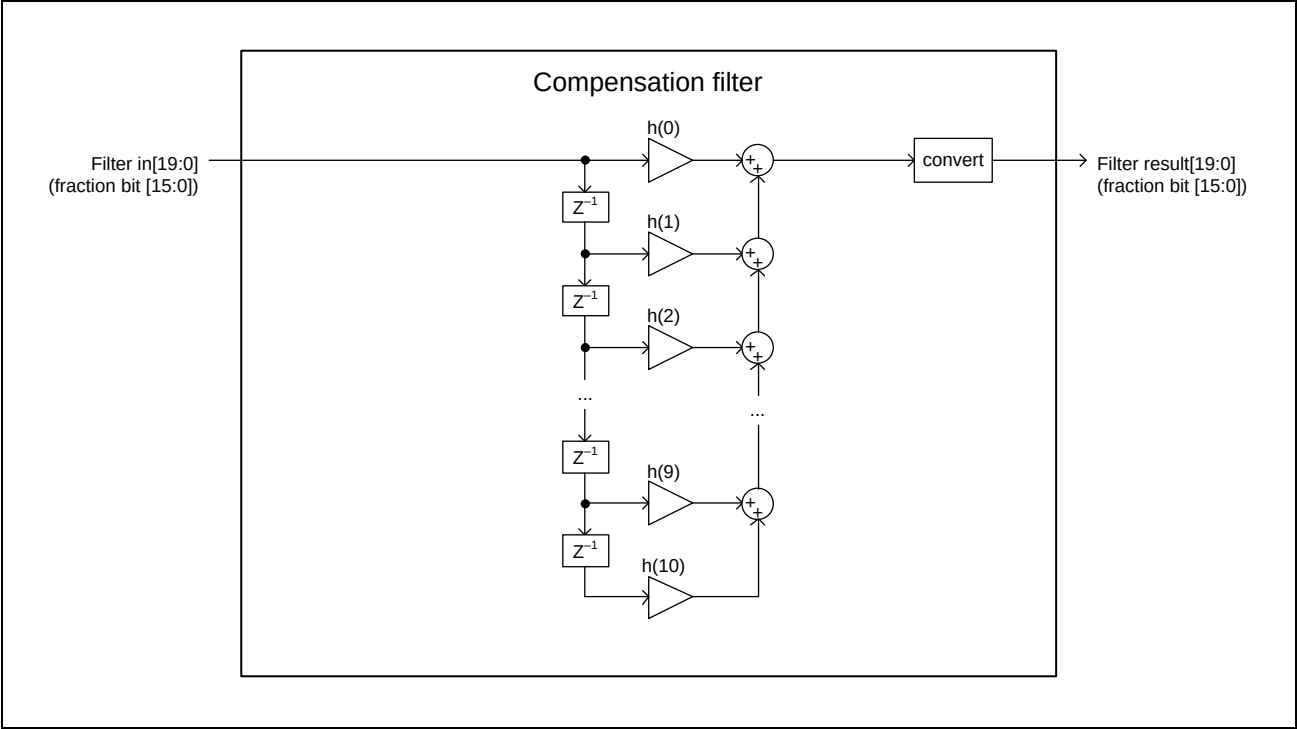


Figure 8.7-5 Compensation Filter Block Diagram

Table 8.7-14 shows a compensation filter specification.

Table 8.7-14 Compensation Filter Specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 13-bit (fraction 11-bit) Refer to 8.7.2.2.23 Compensation Filter Coefficient h(p) Register Channel n (PDMm_PDCFCHpRCHn) (m = 0 to 1, n = 0 to 2, p = 00 to 10) for details.
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Multiplier	Input A bit width	Signed 20-bit (fraction 16-bit)
	Input B bit width	Signed 13-bit (fraction 11-bit)
	Output bit width	Signed 32-bit (fraction 27-bit)
Adder	Input A bit width	Signed 22-bit (fraction 18-bit)
	Input B bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 23-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output.

8.7.3.4.5 Low-Pass (Half-Band Decimation) Filter

The operating clock of the low-pass filter is CCLK. The low-pass filter is operated by using the result of the compensation filter. Output data is decimated to a half of input data.

Figure 8.7-6 shows a low-pass filter block diagram.

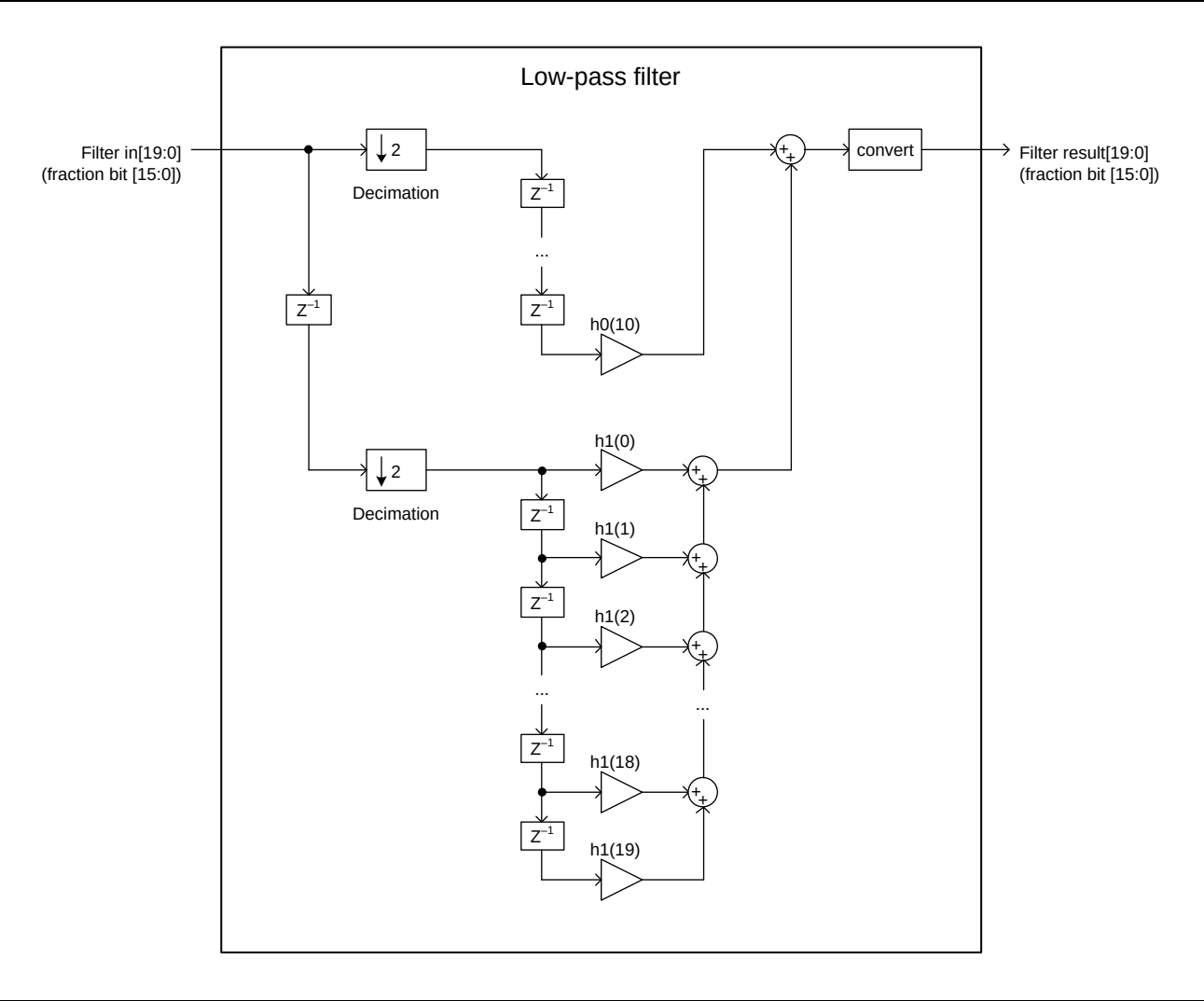


Figure 8.7-6 Low-Pass Filter Block Diagram

Table 8.7-15 lists a low-pass filter specification.

Table 8.7-15 Low-Pass Filter Specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 13-bit (fraction 11-bit) Refer to 8.7.2.2.24 Low-Pass Filter Coefficient h0(10) Register Channel n (PDMm_PDLFCH010RCHn) (m = 0 to 1, n = 0 to 2) and 8.7.2.2.25 Low-Pass Filter Coefficient h1(p) Register Channel n (PDMm_PDLFCH1pRCHn) (m = 0 to 1, n = 0 to 2, p = 0 to 19) for details.
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Multiplier	Input A bit width	Signed 20-bit (fraction 16-bit)
	Input B bit width	Signed 13-bit (fraction 11-bit)
	Output bit width	Signed 32-bit (fraction 27-bit)
Adder	Input A bit width	Signed 22-bit (fraction 18-bit)
	Input B bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 23-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output.

8.7.3.4.6 Moving Average Filter

The operating clock of the moving average filter is CCLK. The moving average filter is operated by using the result of the high-pass filter. The order of the moving average filter can be changed by the setting of the PDMSRCHn.SDMAMD[1:0] bits.

Figure 8.7-7 shows a moving average filter block diagram.

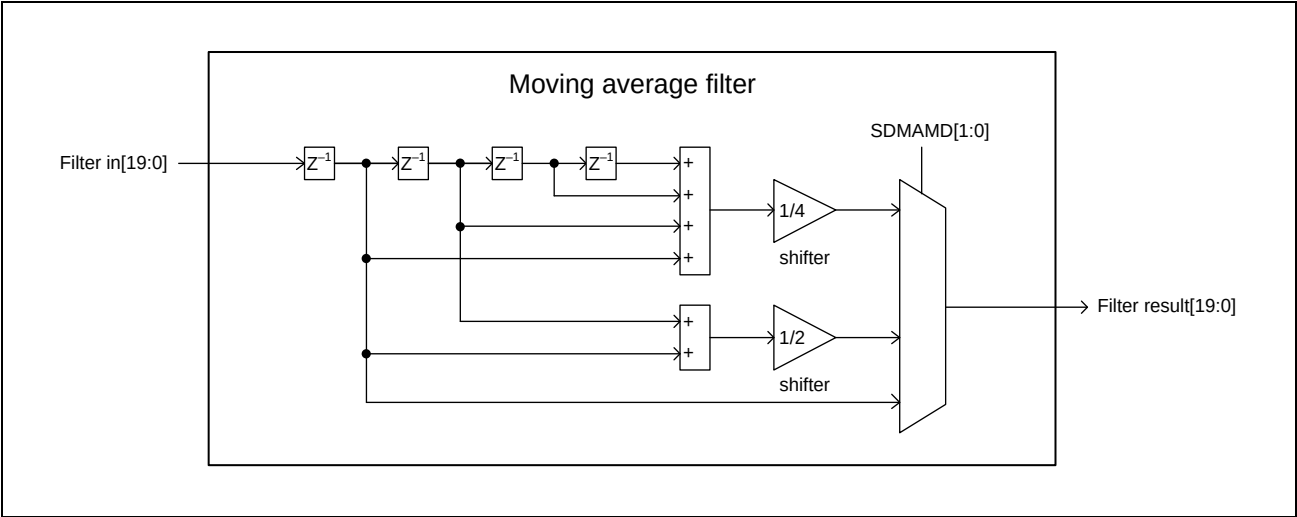


Figure 8.7-7 Moving Average Filter Block Diagram

Table 8.7-16 lists a moving average filter specification.

Table 8.7-16 Moving Average Filter Specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at shifter.

8.7.3.4.7 Settling Time of Channel Activation / Setting Change

The settling time of channel activation is the time after a start trigger PDCSTRTR.STRTRGn (or PDSTRTRCHn.STRTRG) is set before the filter result is stable and output. The settling time of channel setting change is the time after a change trigger PDCCHGTR.CHGTRGn (or PDCHGTRCHn.CHGTRG) is set before the filter result is stable and output. **Figure 8.7-8** shows the settling time measurement point.

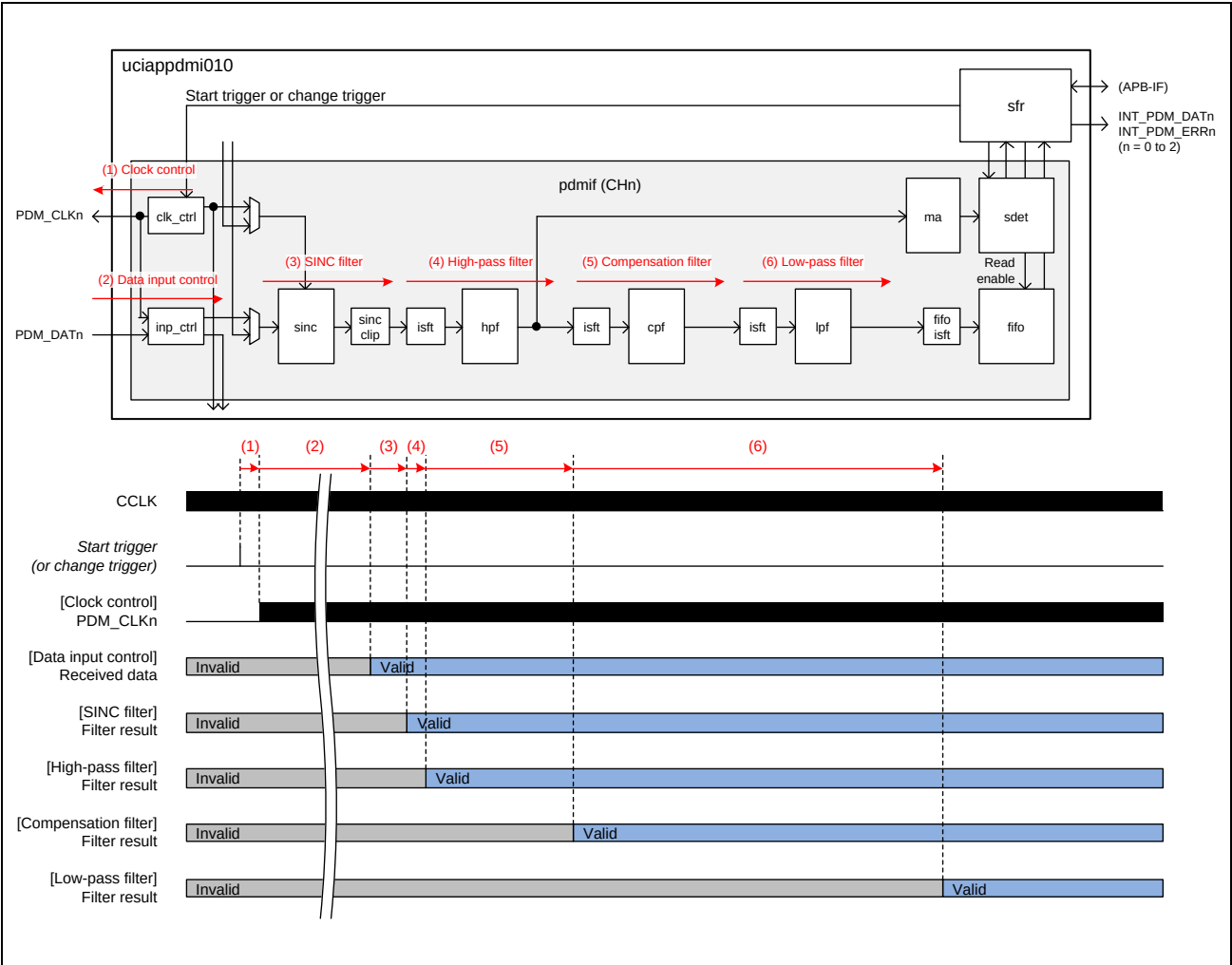


Figure 8.7-8 Settling Time of Channel Activation

The settling time is calculated as shown in **Table 8.7-17**.

Table 8.7-17 Settling Time of Channel Activation / Setting Change

No.	Function	Settling Time	
		Channel Activation	Channel Setting Change
(1)	Clock control	$(4 + (D/2)) \times C_p$	$(3 + D_o + (D/2)) \times C_p$
(2)	Data input control	$(D/2) \times C_p + (\text{Stable time for microphone}^{*1})$	
(3)	Sinc filter	$(K \times D \times M) \times C_p$	
(4)	High-pass filter	$(D \times M + 5) \times C_p$	
(5)	Compensation filter	$(10 \times D \times M + 12) \times C_p$	
(6)	Low-pass filter	$(38 \times D \times M + 22) \times C_p$	

Note: C_p : CCLK period

D : PDM_CLKn's division ratio

D_o : PDM_CLKn's division ratio (old setting)

K : Sinc filter order

M : Sinc filter decimation ratio

Note 1. Wakeup time from low power mode, Switching time, etc. Refer to the datasheet of microphone for details.

8.7.3.5 Data Input Control for Stereo Sound

The microphone which supports the stereo sound outputs the right sound synchronized with PDM_CLKn’s rise-edge and the left sound synchronized with PDM_CLKn’s fall-edge.

PDM-IF can process the right sound and the left sound using different channels.

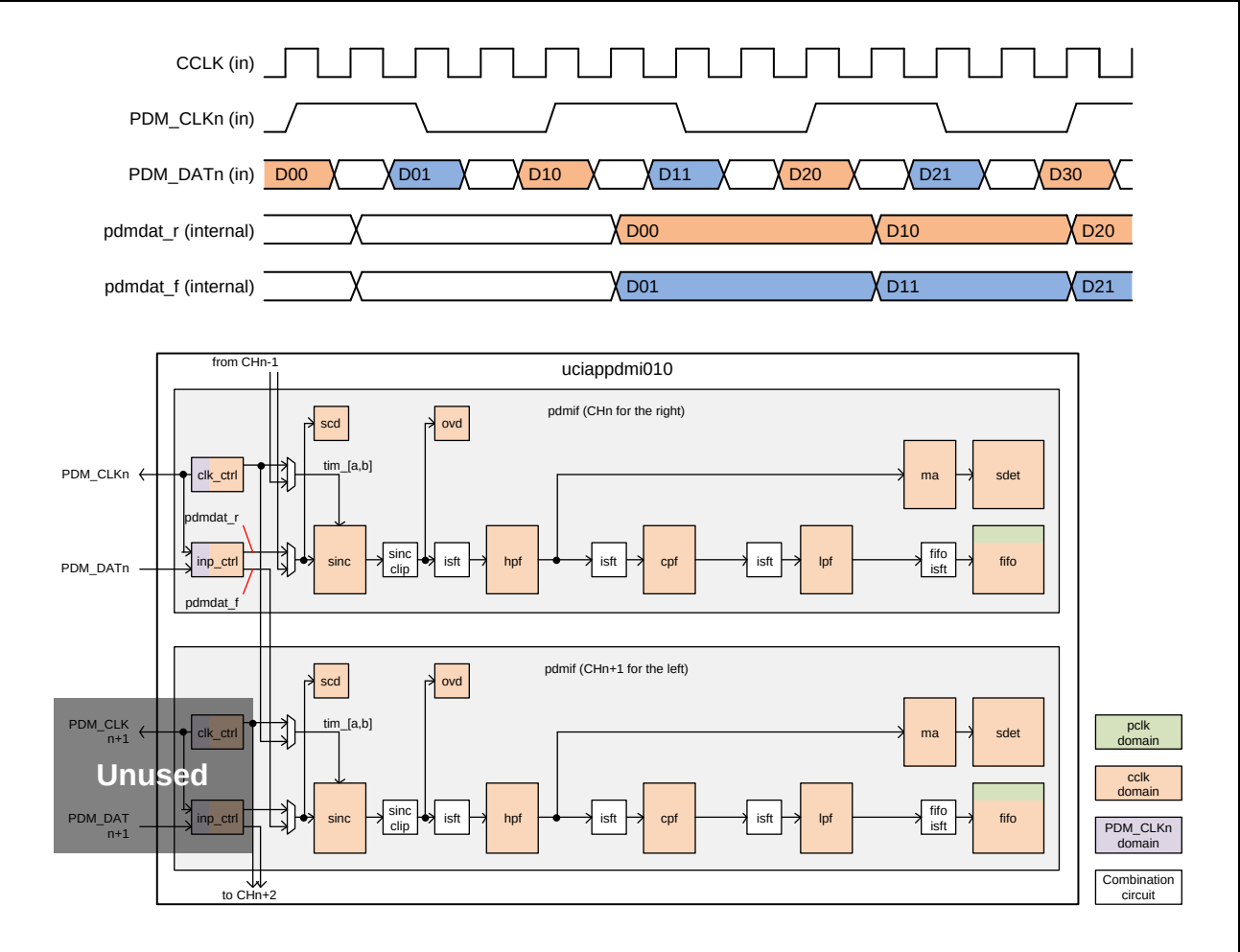


Figure 8.7-9 Data Input Control for Stereo Sound

Table 8.7-18 lists the channel combinations of stereo sound processing.

Table 8.7-18 Channel Combinations of Stereo Sound Processing

Input	Three Channels PDM-IF		Two Channels PDM-IF	
	Data Processing Latched at Rise-Edge (Right Sound)	Data Processing Latched at Fall-Edge (Left Sound)	Data Processing Latched at Rise-Edge (Right Sound)	Data Processing Latched at Fall-Edge (Left Sound)
PDM_CLK0 PDM_DAT0	Channel 0	Channel 1	Channel 0	Channel 1
PDM_CLK1 PDM_DAT1	Channel 1	Channel 2	Channel 1	Channel 0
PDM_CLK2 PDM_DAT2	Channel 2	Channel 0	Not implemented	

Table 8.7-19 lists an example of register settings for stereo sound processing using channel 0 and channel 1.

Table 8.7-19 Register Settings for Stereo Sound Processing

Register	Bit	Channel 0 Setting	Channel 1 Setting
PDICRCHn	All bits	Channel unique settings	Channel unique settings
PDSDCRCHn	All bits	Channel unique settings	Channel unique settings
PDMDSRCHn	INPSEL	0 (Use channel 0 data synchronized with rise-edge)	1 (Use channel 0 data synchronized with fall-edge)
	SFMD[2:0]	Same settings for channel 0 and channel 1 (Mandatory)	
	HFIS[1:0]	Channel unique settings	Channel unique settings
	HFMD	Same settings for channel 0 and channel 1 (Mandatory)	
	CFIS[1:0]	Channel unique settings	Channel unique settings
	CFMD	Same settings for channel 0 and channel 1 (Mandatory)	
	LFIS[1:0]	Channel unique settings	Channel unique settings
	LFMD	Same settings for channel 0 and channel 1 (Mandatory)	
	SDHFMD	Same settings for channel 0 and channel 1 (Mandatory)	
	SDMAMD[1:0]	Same settings for channel 0 and channel 1 (Mandatory)	
	DBIS[3:0]	Channel unique settings	Channel unique settings
PDSFCRCHn	CKDIV[3:0]	Setting of the clock which is supplied to the microphone connected to channel 0	Unuse
	SINCDEC[7:0]	Same settings for channel 0 and channel 1 (Mandatory)	
	SINCRNG[4:0]	Same settings for channel 0 and channel 1 (Mandatory)	
PDHFCS0RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCK1RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCH0RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCH1RCHn	All bits	Channel unique settings	Channel unique settings
PDCFCHmRCHn	All bits	Channel unique settings	Channel unique settings
PDLFCH0mRCHn	All bits	Channel unique settings	Channel unique settings
PDLFCH1mRCHn	All bits	Channel unique settings	Channel unique settings
PDSLTRCHn	All bits	Channel unique settings	Channel unique settings
PDSUTRCHn	All bits	Channel unique settings	Channel unique settings
PDDBCRCHn	All bits	Same settings for channel 0 and channel 1 (Mandatory)	
PDSCTSRCHn	All bits	Channel unique settings	Channel unique settings
PDOVLTRCHn	All bits	Channel unique settings	Channel unique settings
PDOVUTRCHn	All bits	Channel unique settings	Channel unique settings

8.7.3.6 Sound Detection

The operating clock of the sound detection function is CCLK. When PDSDCRCHn.SDE = 1b, the function judges the sound detection using the moving average filter results.

A sound detection flag (PDSRCHn.SDF) is set to 1b when the result is above the upper limit PDSDUTRCHn.SDETU[19:0]. The sound detection flag is also set to 1b when the result is below the lower limit PDSDLTRCHn.SDETL[19:0]. The sound detection flag is cleared to 0b when 1b is written to PDSCRCHn.SDFC.

Figure 8.7-10 shows the sound detection function.

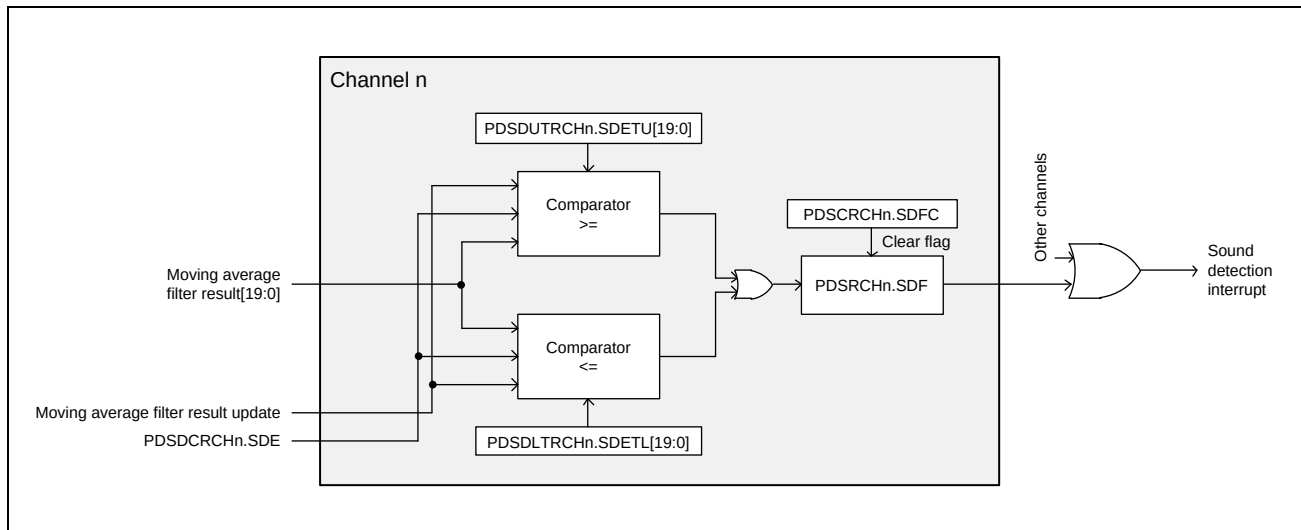


Figure 8.7-10 Sound Detection Function

8.7.3.7 Data Buffer

Data buffer writes sound data to FIFO synchronizing with CCLK and reads sound data from FIFO synchronizing with PCLK. Data buffer stores sound data in FIFO until it is read out.

While $PDDRCRCHn.DATRE = 0b$ (hereinafter, $DATRE$), $PDDSRCHn.DATNUM[7:0]$ (hereinafter, $DATNUM[7:0]$) shows 0 and PDM-IF does not allow to read sound data. After $DATRE$ is changed from 0b to 1b, $DATNUM[7:0]$ shows the buffer size minus 2 and PDM-IF allows to read sound data. While $DATRE = 1b$, data buffer increments $DATNUM[7:0]$ when data is written and decrements $DATNUM[7:0]$ when data is read.

When $DATNUM[7:0]$ is greater than or equal to a threshold which is configured by $PDDBCRCHn.DATRITHR[2:0]$, a data reception flag $PDSRCHn.DRF$ (hereinafter, DRF) is set to 1b. When $DATNUM[7:0]$ is less than the threshold by reading data or etc., a data reception flag DRF is cleared to 0b.

When $DATNUM[7:0]$ is the buffer size minus 2 and data is written without reading, data buffer overwrites the oldest data with a new one and a buffer overwriting detection flag $PDSRCHn.BFOWDF$ (hereinafter, $BFOWDF$) is set to 1b. And then $DATNUM[7:0]$ is set to the buffer size minus 2 again after overwriting. However, data buffer overwrites but $BFOWDF$ is not set after $DATRE$ changes from 0b to 1b to before the first data is read.

Figure 8.7-11 shows the data buffer function.

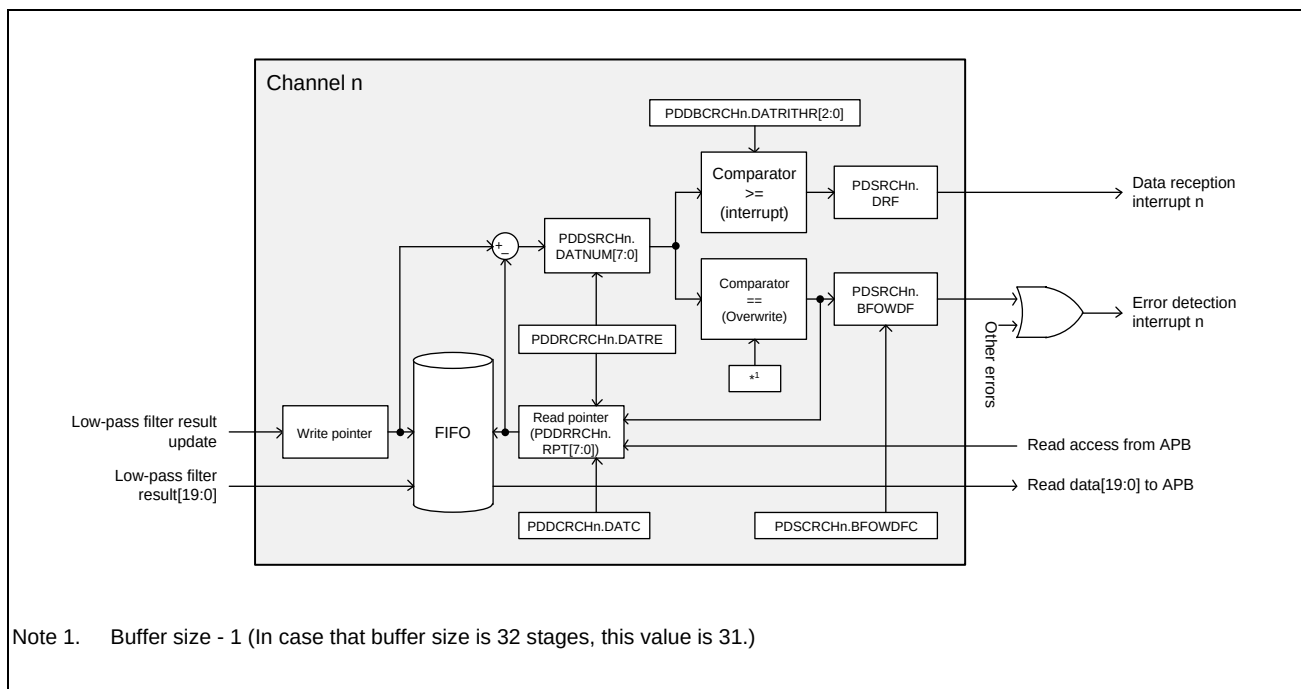


Figure 8.7-11 Data Buffer Function

8.7.3.8 Short-Circuit Detection

When Short-circuit detection enable (PDSDCRCHn.SCDE) is valid, the short-circuit detection can operate.

A dedicated 13-bit counter counts consecutive 0's or 1's input from pdm_dat which is outputs by data input control block of channel n or channel n+1.

When the number of consecutive 0's exceeds the value set in the PDCTSRCHn.SCDL[12:0] register or the number of consecutive 1's exceeds the value set in the PDCTSRCHn.SCDH[12:0] register, a short-circuit detection interrupt request is generated.

Figure 8.7-12 shows the configuration for the short-circuit detection function.

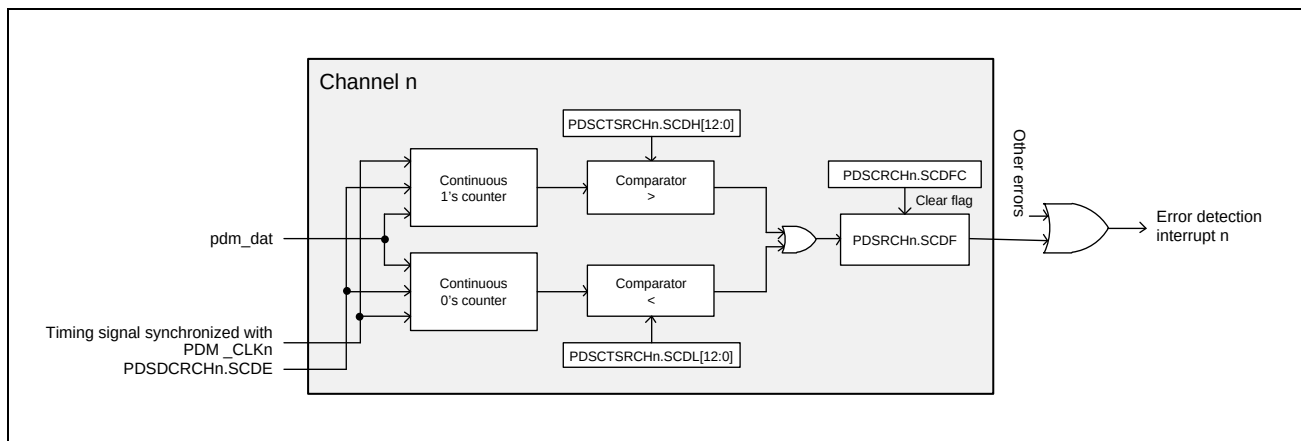


Figure 8.7-12 Short-Circuit Detection Function

8.7.3.9 Overvoltage Detection

When the overvoltage upper limit detection enable (PDSDCRCHn.OVUDE) is valid, the overvoltage upper limit detection can operate. When the overvoltage lower limit detection enable (PDSDCRCHn.OVLDE) is valid, the overvoltage lower limit detection can operate.

The sound data is clipped as a 20-bit data from 34-bit data output from sinc filter by the setting of the PDSFCRCHn.SINCRNG[4:0] bits. Refer to **Table 8.7-8** for details.

An overvoltage upper limit detection flag is set to 1b when the clipped sound data is greater than PDOVUTRCHn.OVDU[19:0].

An overvoltage lower limit detection flag is set to 1b when the clipped sound data is less than PDOVLTRCHn.OVDL[19:0].

Figure 8.7-13 shows the overvoltage detection function.

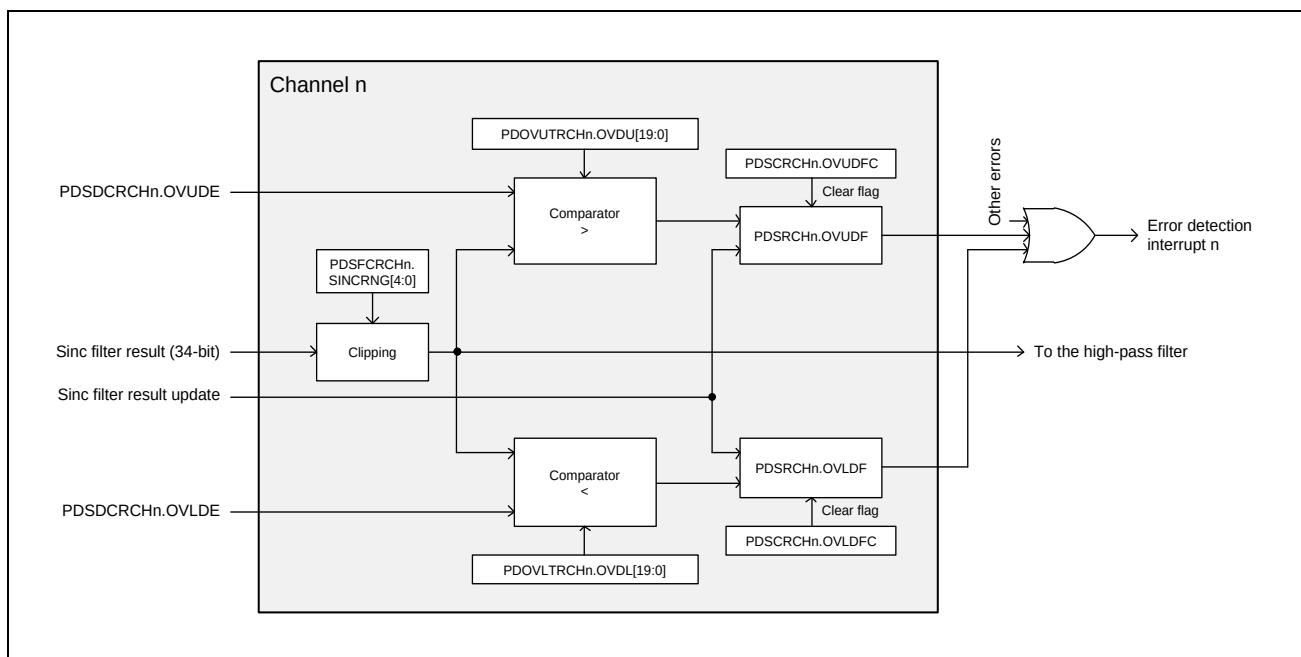


Figure 8.7-13 Overvoltage Detection Function

8.7.3.10 Error

Table 8.7-20 shows a PDM-IF error list.

Table 8.7-20 PDM-IF Error List

No.	Register Name	Description	Comment	Function (Related Section)
1	PDSRCHn.SCDF	Short circuit detection flag		8.7.3.8 Short-Circuit Detection
2	PDSRCHn.OVLDF	Overvoltage lower limit detection flag		8.7.3.9 Overvoltage Detection
3	PDSRCHn.OVUDF	Overvoltage upper limit detection flag		8.7.3.9 Overvoltage Detection
4	PDSRCHn.BFOWDF	Buffer overwriting detection flag	For debugging	8.7.3.7 Data Buffer

The errors in No.4 are for debugging. Software needs to set registers and operate PDM-IF so that these errors do not occur. If any of these error occur, stop PDM-IF following **8.7.4.2 Stop Flow** and change the settings.

8.7.3.11 Interrupt Function

PDM-IF has the following interrupt function.

Table 8.7-21 PDM-IF Interrupt List

No.	Signal Name	Interrupt Name	Number of Signals	Pulse or Level	Related Register Bit Name		Function (Related Section)
					Common Register	Channel Register	
1	INT_PDM_DATn (n = 0, 1, 2)	Data reception interrupt CHn	3	Level	PDCSR.DRFn	PDSRCHn.DRF	8.7.3.7 Data Buffer
2	INT_PDM_SDET	Sound detection interrupt (3-channel status flags are ORed.)	1	Level	PDCSR.SDFn	PDSRCHn.SDF	8.7.3.6 Sound Detection
3	INT_PDM_ERRn (n = 0, 1, 2)	Error detection interrupt CHn	3	Level	PDCSR.EDFn	PDSRCHn.SCDF	8.7.3.8 Short-Circuit Detection
						PDSRCHn.OVLDF	8.7.3.9 Overvoltage Detection
						PDSRCHn.OVUDF	8.7.3.9 Overvoltage Detection
						PDSRCHn.BFOWDF	8.7.3.7 Data Buffer

8.7.4 Setting Flow

Figure 8.7-14 and Figure 8.7-15 show the example of the main procedure of PDM-IF.

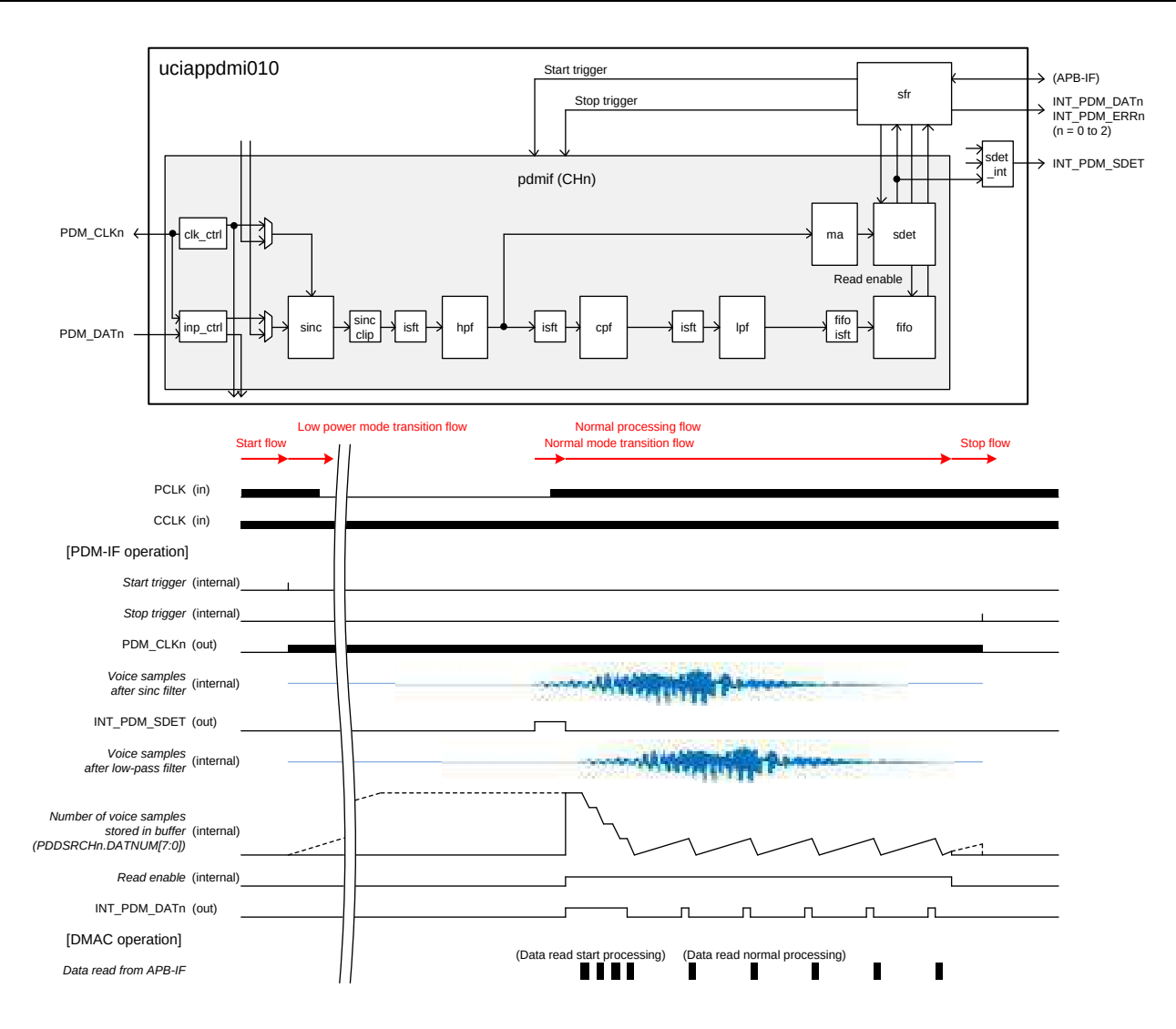


Figure 8.7-14 PDM-IF Operation Example

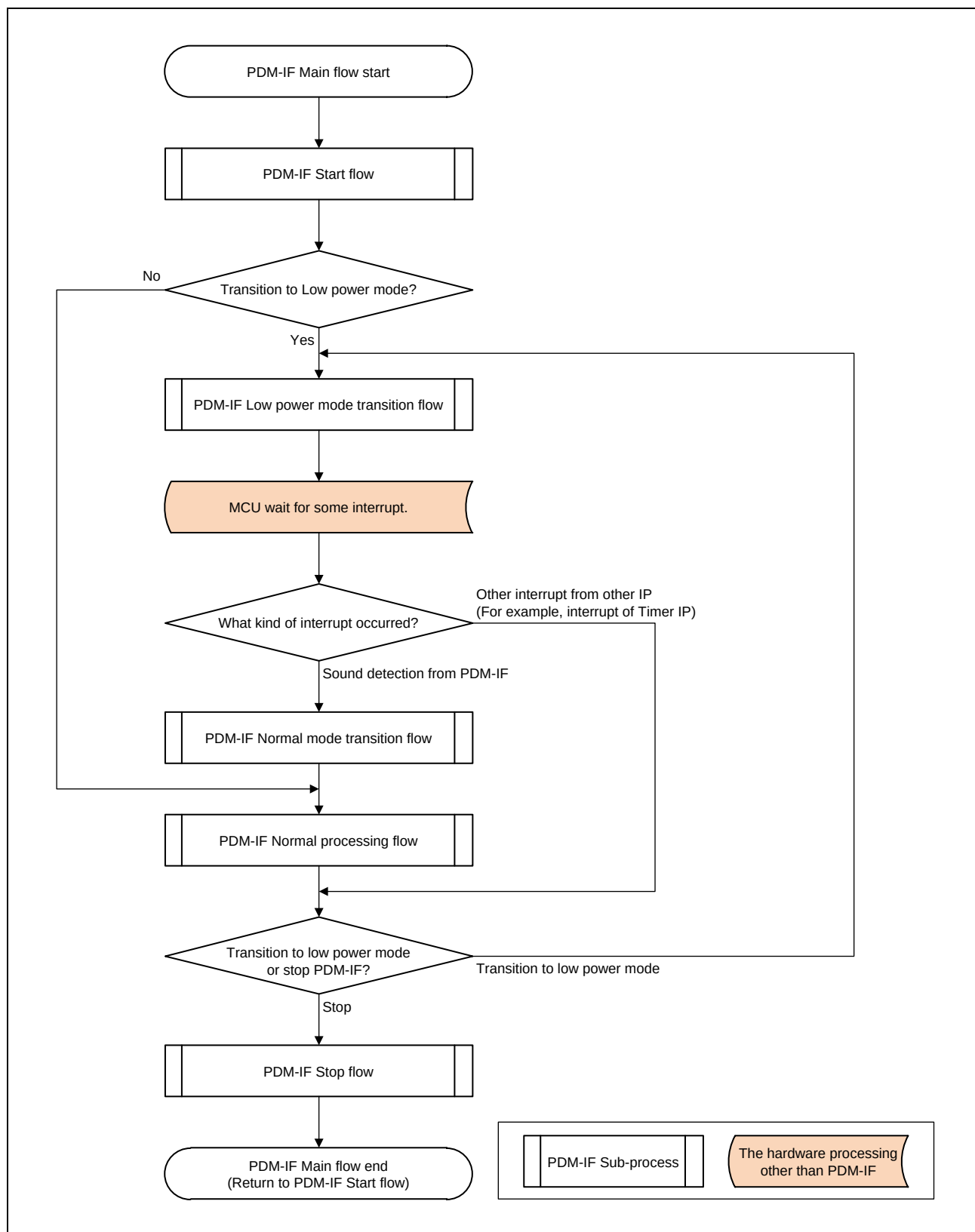


Figure 8.7-15 PDM-IF Setting Main Flow with Low Power Mode Transition

Figure 8.7-16 shows the example of the main procedure without low-power mode transition.

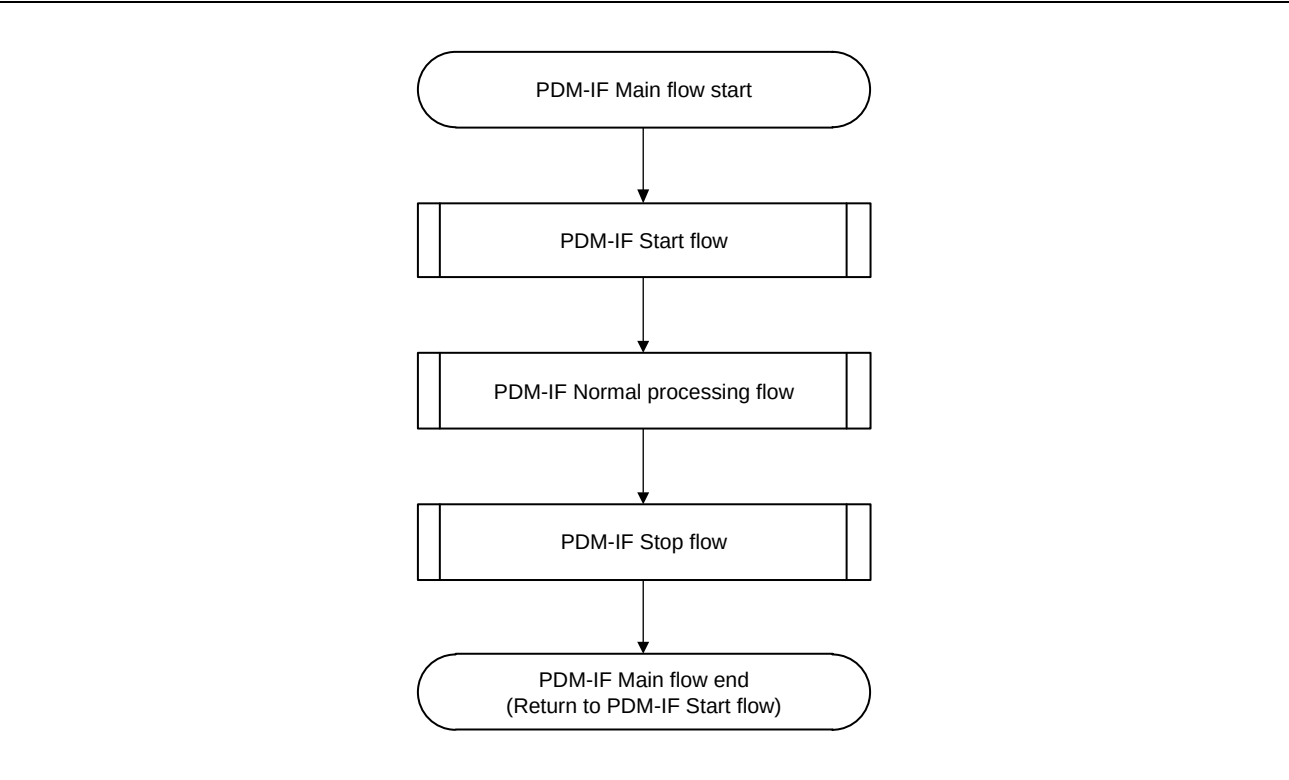


Figure 8.7-16 PDM-IF Setting Main Flow without Low Power Mode Transition

8.7.4.1 Start Flow

Figure 8.7-17 shows the example of the start procedure of PDM-IF.

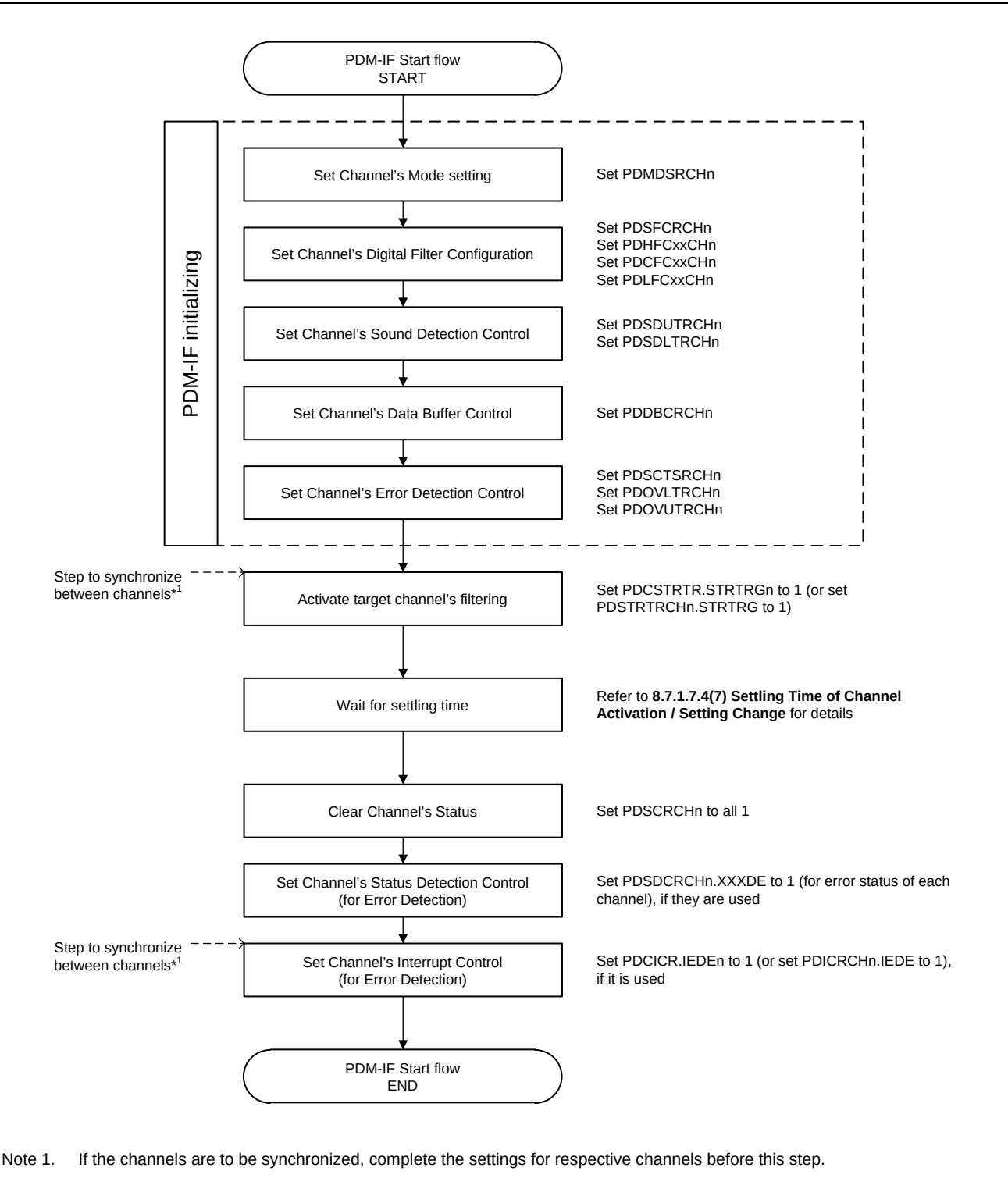


Figure 8.7-17 PDM-IF Start Flow

8.7.4.2 Stop Flow

Figure 8.7-18 shows the example of the stop procedure of PDM-IF.

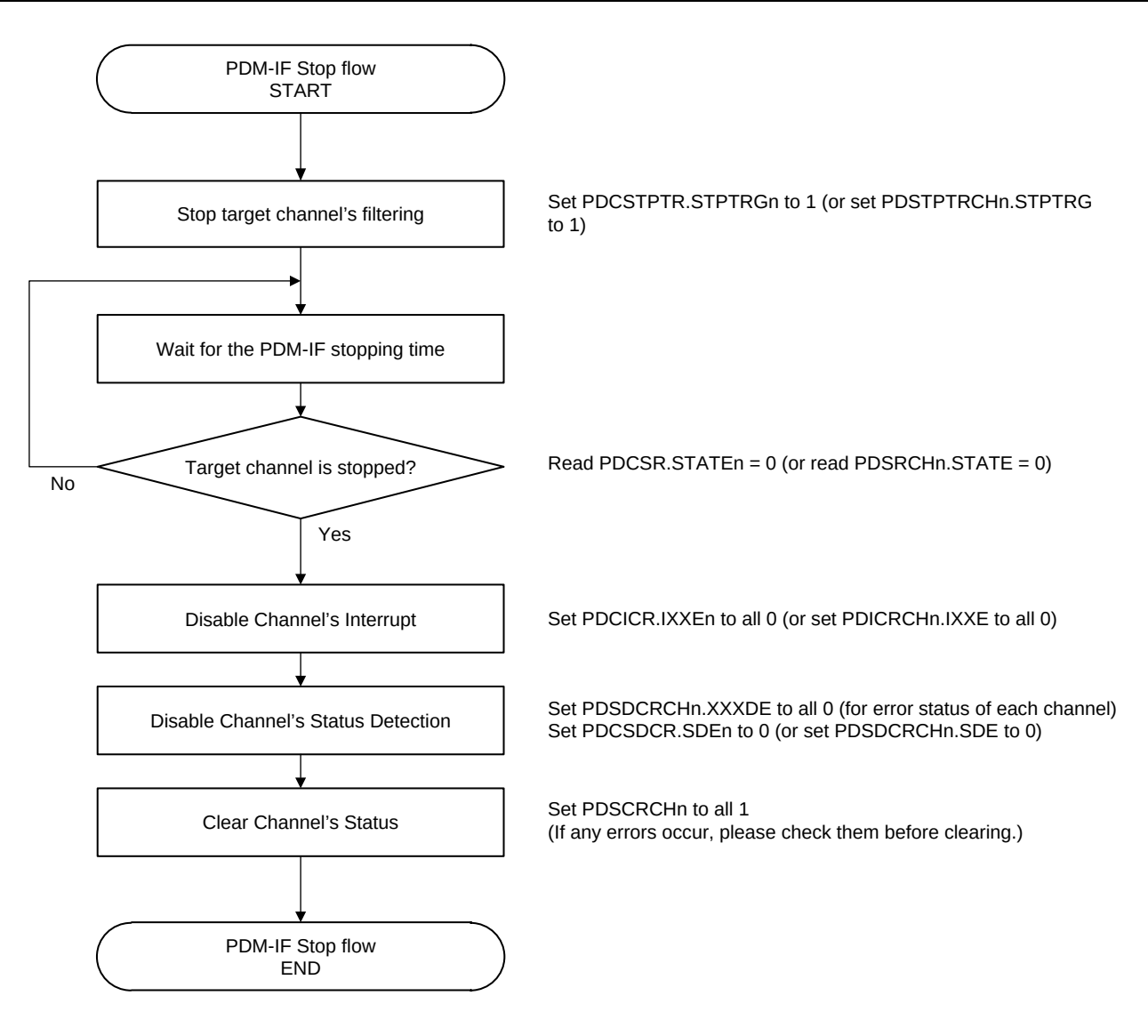


Figure 8.7-18 PDM-IF Stop Flow

8.7.4.3 Normal Processing Flow

Figure 8.7-19 shows the example of the normal processing procedure of PDM-IF.

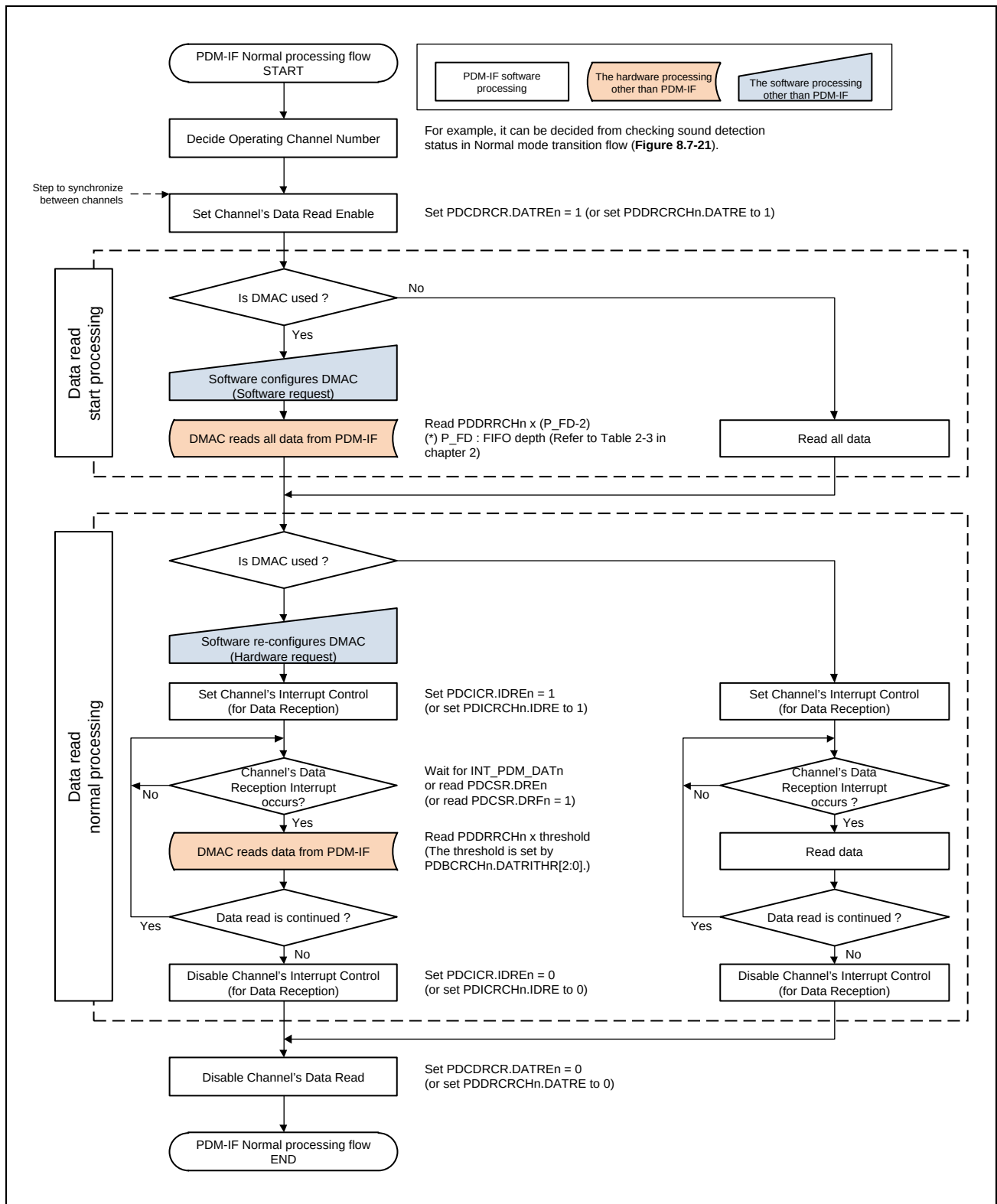


Figure 8.7-19 PDM-IF Normal Processing Flow

8.7.4.4 Low Power Mode Transition Flow

Figure 8.7-20 shows the example of the low power mode transition procedure of PDM-IF.

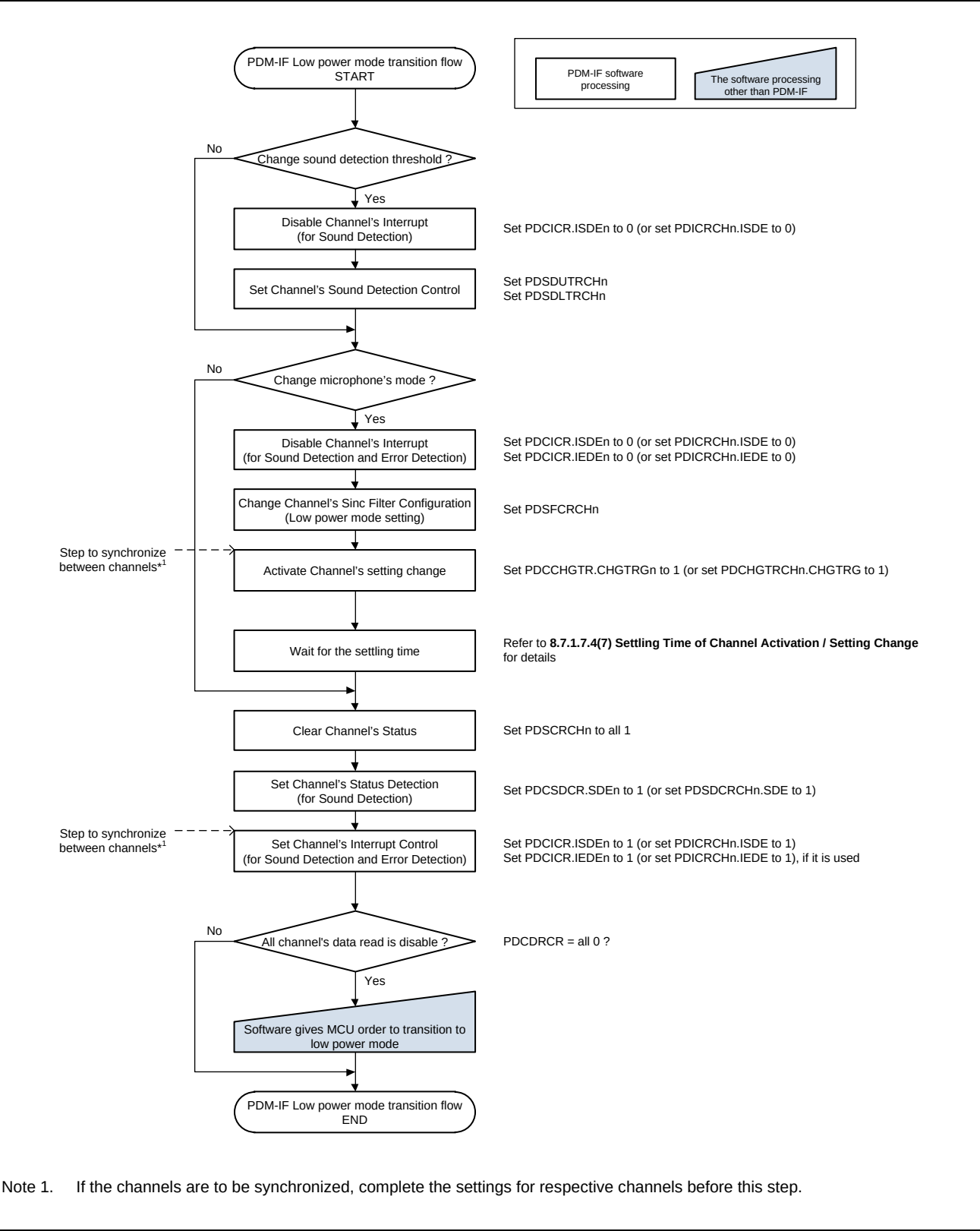


Figure 8.7-20 PDM-IF Low Power Mode Transition Flow

8.7.4.5 Normal Mode Transition Flow

Figure 8.7-21 shows the example of normal mode transition procedure of PDM-IF.

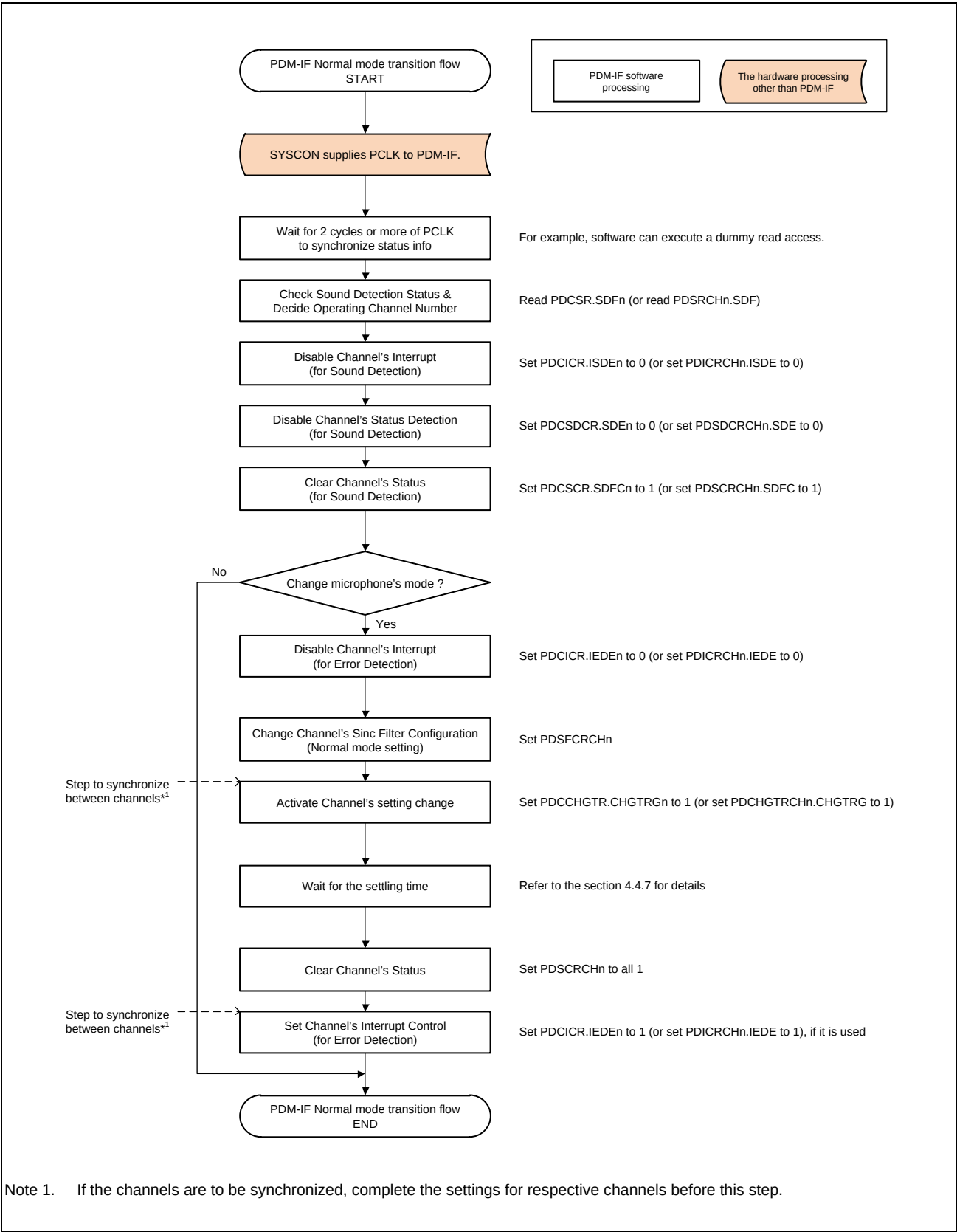


Figure 8.7-21 PDM-IF Normal Mode Transition Flow

8.7.4.6 Setting PDM_CLKn (n = 0, 1, 2) Frequency and Sampling Frequency Generated by the Unit

The frequency of PDM_CLKn (n = 0, 1, 2), which is the reference for receiving data from the external device, and the sampling frequency generated by the PDM unit for external input PDM data can be changed by setting the register. The PDM_CLKn (n = 0, 1, 2) frequency must match the clock input specifications of the external device. The sampling frequency generated by the unit should be set to the expected value.

In this LSI, the PDM reference clock CCLK is fixed at 4.8 MHz, so the PDM_CLK output frequency and the unit-generation sampling frequency follow the equation below when implementing the circuit.

- PDM_CLKn (n = 0, 1, 2) frequency

$$= \frac{\text{CCLK frequency}}{(\text{CKDIV} + 1) \times 2};$$

- Unit-generation sampling frequency

$$= \frac{\text{PDM\_CLK frequency}}{(\text{SINCDEC} + 1) \times 2^{*1}} = \frac{\text{CCLK frequency}}{(\text{CKDIV} + 1) \times 2} \div (\text{SINCDEC} + 1) \times 2^{*1};$$

Remark: CKDIV: PDSFCRCHn.CKDIV[3:0] register set value (Set the PDM_CLKn division ratio to CCLK)
SINCDEC: PDSFCRCHn.SINCDEC[7:0] register set value (Set the decimation ratio of the Sinc filter)

Note 1. Reason for ×2: Low-pass filter decimation ratio

- PDMIF sampling rate:

$$\frac{\text{CCLK frequency}}{((\text{CKDIV} + 1) \times 2) \times (\text{SINCDEC} + 1) \times 2}$$

The output range of the sinc filter is changed by changing the decimation ratio of the sinc filter (SINCDEC) with the change of the sampling frequency above. Since the maximum number of effective bits for processing of audio data within the PDM following sinc filtering is 20 bits, PDSFCRCHn.SINCRNG[4:0] must also be changed according to the output range of the sinc filter. This allows avoiding the occurrence of overflow and underflow and maximizing the effective audio amplitude.

Recommended setting: Set PDSFCRCHn.SINCRNG[4:0] such that the higher-order 20 bits of valid data of the sinc filter output are effective in calculation processing following sinc filtering.

For details on clipping or sifting of the sinc filter output, see **8.7.3.4.1 Data Clipping and Shifting between Filters**.

- Output range of the sinc filter
= ±decimation ratio (PDSFCRCHn.SINCDEC[7:0]) ^ order (PDMDSRCHn.SFMD[2:0]);

[Example]

When SINCDEC[7:0] = 27h (decimation ratio: 40) and SFMD[2:0] = 0 (4-order), the output range of the sinc filter is ±2560000 (23 bits consisting of 22 bits of significant digits and a sign bit). The recommended setting of PDSFCRCHn.SINCRNG[4:0] in this case is 0_1011b (b22-b3).

The following describes the impact of the circuit structure on the frequency calculations.

The table below shows the relationship between the selectable N/M values (register settings) with respect to the PDM_CLKn (n = 0, 1, 2) frequency and the unit-generation sampling frequency, according to the above calculation formula, and the target-generated sampling frequency.

In the table below, “Not selectable” indicates that the N value or M value cannot be selected for the target-generated sampling frequency. Since the N value can be selected from 2 to 32 in multiples of 2 and the M value can only be selected as an integer of 1, 2, 3, ..., 256, the corresponding target-generated sampling frequency may not be generated. However, in some cases, a frequency near the target-generation sampling frequency may be selectable. If that frequency is acceptable, refer to the PDM_CLKn (n = 0, 1, 2) frequency and unit-generation sampling frequency calculation formula above to derive the desired N value/M value and the corresponding register setting value.

Table 8.7-22 Selectable PDM_CLKn (n = 0, 1, 2) Frequency and Unit-Generation Sampling Frequency

	Sampling frequency (kHz)	48	40	30	25	24	20	16	15	12	10	8
N value	PDM_CLK frequency (MHz)	M value	M value	M value	M value	M value	M value	M value	M value	M value	M value	M value
2	2.4	25	30	40	48	50	60	75	80	100	120	150
4	1.2	Not selectable	15	20	24	25	30	Not selectable	40	50	60	75
6	0.8	Not selectable	10	Not selectable	16	Not selectable	20	25	Not selectable	Not selectable	40	50
8	0.6	Not selectable	Not selectable	10	12	Not selectable	15	Not selectable	20	25	30	Not selectable
10	0.48	5	6	8	Not selectable	10	12	15	16	20	24	30
12	0.4	Not selectable	5	Not selectable	8	Not selectable	10	Not selectable	Not selectable	Not selectable	20	25
14	0.342857143	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable
16	0.3	Not selectable	Not selectable	5	6	Not selectable	Not selectable	Not selectable	10	Not selectable	15	Not selectable
18	0.266666667	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable
20	0.24	Not selectable	3	4	Not selectable	5	6	Not selectable	8	10	12	15
22	0.218181818	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable
24	0.2	Not selectable	Not selectable	Not selectable	4	Not selectable	5	Not selectable	Not selectable	Not selectable	10	Not selectable
26	0.184615385	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable
28	0.171428571	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable	Not selectable
30	0.16	Not selectable	2	Not selectable	Not selectable	Not selectable	4	5	Not selectable	Not selectable	8	10
32	0.15	Not selectable	Not selectable	Not selectable	3	Not selectable	Not selectable	Not selectable	5	Not selectable	Not selectable	Not selectable

The N value/M value in the above table is based on the following formula.

Definition of N value: $(CKDIV + 1) \times 2$

Definition of M value: $(SINCDEC + 1)$

Remark: CKDIV: PDSFCRCHn.CKDIV[3:0] register set value
SINCDEC: PDSFCRCHn.SINCDEC[7:0] register set value

8.7.4.7 DMA Transfer Method

This PDM unit can only receive voice data from external sources, and does not have a transmit function. The PDM unit does not have a dedicated REQ/ACK signal to the DMAC master, but only reads data from the PDM buffer read register via the APB I/F. Reading data via the CPU and reading data via the DMAC master are not different in terms of PDM functionality.

Figure 8.7-22 shows an overview of the PDM data read process.

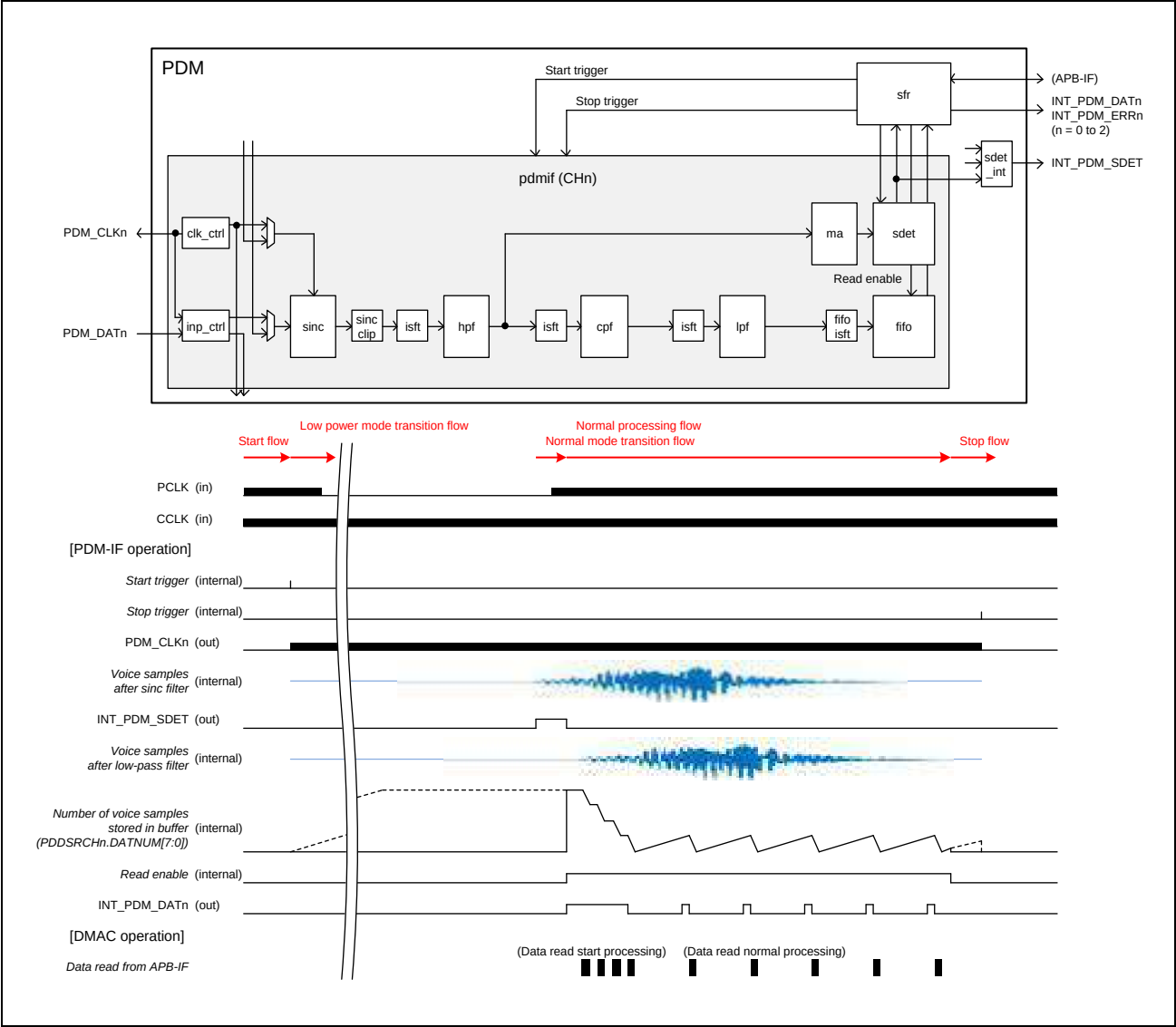


Figure 8.7-22 PDM-IF Operation Example

The DMA data transfer using the DMA master must follow the description in **8.7.4.3 Normal Processing Flow**. The following explains the connection configuration with the DMA master, initialization settings on the DMA master, and recommended settings for the INT_PDM_DAT0/1/2 signal at PDM initialization.

The connection configuration with the DMA master in this LSI is as follows, issuing a DMA request to the DMA master based on the INT_PDM_DAT0/1/2 signal of the PDM. The INT_PDM_DAT0/1/2 interrupt signal is active (high) depending on the amount of data stored in the receive buffer in the PDM (the amount of interrupt generation buffer can be changed by setting the register / PDDBCRCHn (n = 0, 1, 2).DATRITHR[2:0]).

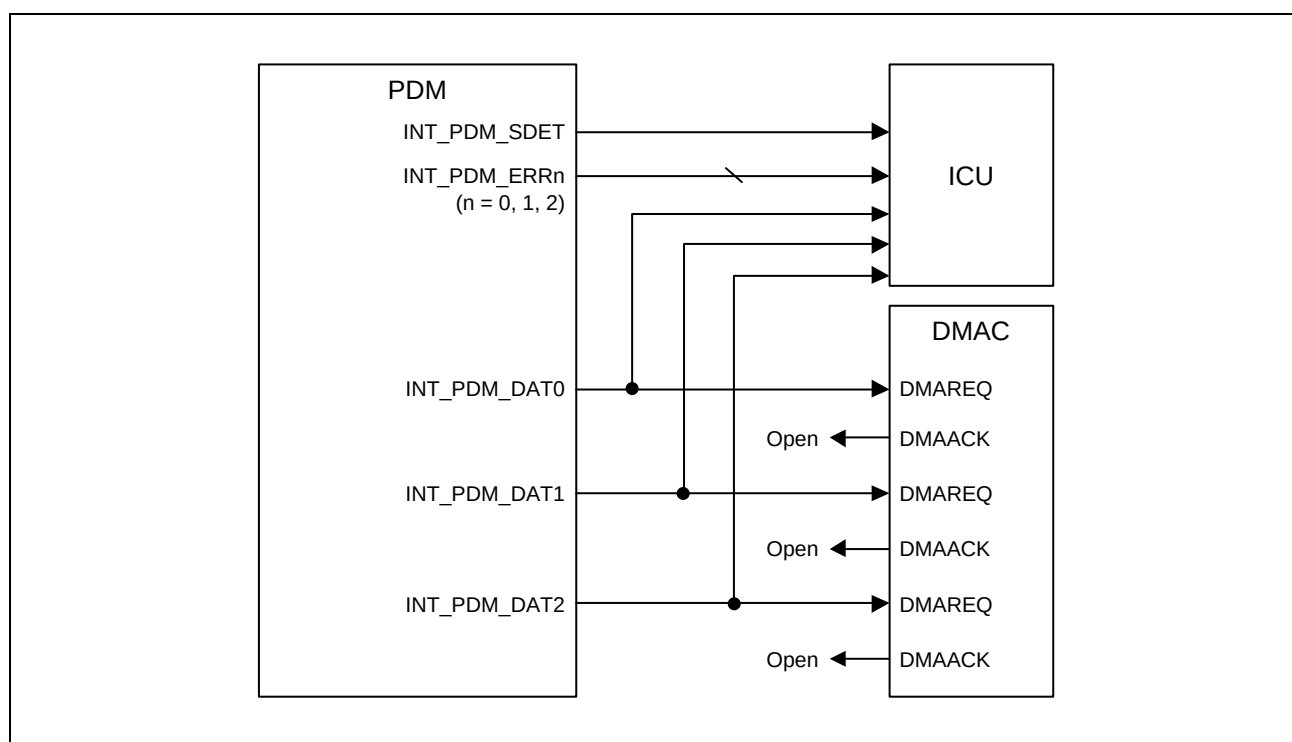


Figure 8.7-23 PDM Connection Configuration

This section describes the restrictions at DMA transfer and the active condition of the INT_PDM_DAT0/1/2 signal at PDM initialization.

1) Restrictions at DMA transfer:

The initialization setting on the DMA master is required in “Software configures DMAC/Software re-configures DMAC” in **Figure 8.7-19** of **8.7.4.3 Normal Processing Flow**. The restrictions on PDM DMA transfers are described below.

- DMA transfer mode (1): Single
(One DMA transfer process is performed for one DMA request.)
Reference value for the DMA master setting: TM register = 0b
- DMA transfer mode (2): Bus cycle mode
(DMAREQ is masked in the bus cycle period (while DMAACK is active.))
Reference value for the DMA master setting: AM[2:0] = 010b
- One DMA transfer: 32 bits
Reference value for the DMA master: SDS register = 32 bits
- DMA REQ signal attribute: Level/High
(The active polarity of the INT_PDM_DAT0/1/2 signal is high, so the DMA request signal must also be high.)
Reference value for the DMA master: LVL = 1b, HIEN = 1b, LOEN = 0b
- DMA REQ signal active release timing: After the completion of data read from the PDM unit
(DMA REQ is masked while DMA ACK is enabled.)
Reference value for the DMA master: REQD = 0b
- DMA transfer data read register: PDDRRCHn (n = 0, 1, 2).DAT[19:0] register

2) Interrupt setting of the INT_PDM_DAT0/1/2 signal in the PDM receive initialization setting:

The interrupt setting for the INT_PDM_DAT0/1/2 signal is required in “Set Channel’s Data Buffer Control” in **Figure 8.7-17** of **8.7.4.1 Start Flow**.

As the receive buffer amount of PDM in this LSI is fixed at 64 stages, it is recommended to set PDDBCRCHn (n = 0, 1, 2).DATRITHR[2:0] = 4 (16 stages). Since a time difference may occur between the active timing of the INT_PDM_DAT0/1/2 signal and the request decision on the DMA master, the active timing of the INT_PDM_DAT0/1/2 signal must always be selected to be at least 2 stages.

8.7.4.8 Reset Clock Control

For the reset clock control, the following reset clock pins must be controlled.

Table 8.7-23 List of Reset Clocks

	Reset Signal	Clock Signal	Clock Source	Description
System 1	PRESETn	PCLK, PCLK_SFR	CPG unit (Divider clock for CPG PLL output)	For internal bus control circuit
System 2	CRESETn	CCLK	CPG unit (External system clock: 4.8 MHz fixed at 24 MHz divided by 5)	For external IF control circuit

The above reset clock pins are controlled by the CPG unit, and each reset clock can be controlled ON/OFF with the CPG register. When each reset is released, a feature is implemented on the CPG to stop the target clock before and after the reset is released.

For each system, reset and clock control should be performed according to the following procedure.

However, the reset and clock control between the two systems below are independent, and there are no restrictions on the order of settings between the systems.

Furthermore, when controlling the reset clock, the PDM unit should be stopped and accesses from related units should also be stopped.

- Reset (PRESETn) and clock (PCLK, PCLK_SFR) for the internal bus control circuit:
 - PDM operation start condition (reset/clock supply): clock supply → reset release
 - PDM stop condition (reset/clock stop): reset enabled → clock stop
 - PDM reset (reset ON/OFF, clock supply continued): Reset can be enabled or disabled at any timing
- Reset (CRESETn) and clock (CCLK) for the external IF control circuit:
 - PDM operation start condition (reset/clock supply): clock supply → reset release
 - PDM stop condition (reset/clock stop): reset enabled → clock stop
 - PDM reset (reset ON/OFF, clock supply continued): Reset can be enabled or disabled at any timing

As for PCLK/PCLK_SFR, the CPG unit can change the clock frequency variable (100 MHz, 50 MHz, 25 MHz, 12.5 MHz, 3.125 MHz). When lowering the PCLK/PCLK_SFR frequency, the performance of reading voice data from the PDM unit will be degraded. Use caution when lowering the frequency.

8.7.5 Interrupt

8.7.5.1 Interrupt List

Table 8.7-24 is a list of interrupt signals for this unit. For details on interrupts, refer to **4.6 Interrupt Controller**.

Table 8.7-24 PDM-IF Interrupt List

No.	Signal Name	Interrupt Name	Number of Signals	Pulse or Level	Related Register Bit Name		Function (Related Section)
					Common Register	Channel Register	
1	INT_PDM_DATn (n = 0, 1, 2)	Data reception interrupt CHn (n = 0, 1, 2)	3	Level	PDCSR.DRFn	PDSRCHn (n = 0, 1, 2).DRF	8.7.3.7 Data Buffer
2	INT_PDM_SDET	Sound detection interrupt (3-channel status flags are ORed.)	1	Level	PDCSR.SDFn	PDSRCHn (n = 0, 1, 2).SDF	8.7.3.6 Sound Detection
3	INT_PDM_ERRn (n = 0, 1, 2)	Error detection interrupt CHn (n = 0, 1, 2)	3	Level	PDCSR.EDFn	PDSRCHn (n = 0, 1, 2).SCDF	8.7.3.8 Short-Circuit Detection
						PDSRCHn (n = 0, 1, 2).OVLDF	8.7.3.9 Overvoltage Detection
						PDSRCHn (n = 0, 1, 2).OVUDF	8.7.3.9 Overvoltage Detection
						PDSRCHn (n = 0, 1, 2) BFLOWDF	8.7.3.7 Data Buffer

8.7.6 Usage Notes

8.7.6.1 Data Reception Interrupt Threshold Register Setting

A data reception interrupt INT_PDM_DATn is issued as a level output. Therefore, if the number of samples in buffer still exceeds the data reception interrupt threshold set in PDDBCRCHn.DATRITHR[2:0] when data reading of DMA finishes, the interrupt controller might not detect the edge of INT_PDM_DATn. The situation happens if data reading time exceeds sampling rate*¹ x threshold. Set the appropriate threshold so that the interrupt controller can detect the interrupt.

In case the threshold cannot avoid the situation, read PDDBCRCHn.DATRITHR[2:0] and re-start DMA by software after reading the data.

Note 1. Sampling rate: Time to spend to store a sample in buffer.

8.7.6.2 Frequency Characteristics of Filters with Default Settings

The following figure and table show the frequency characteristics of the filters with the default settings*¹ for reference information.

Note 1. SINC filter output sampling frequency = 32 kHz
PDM output sampling frequency = 16 kHz

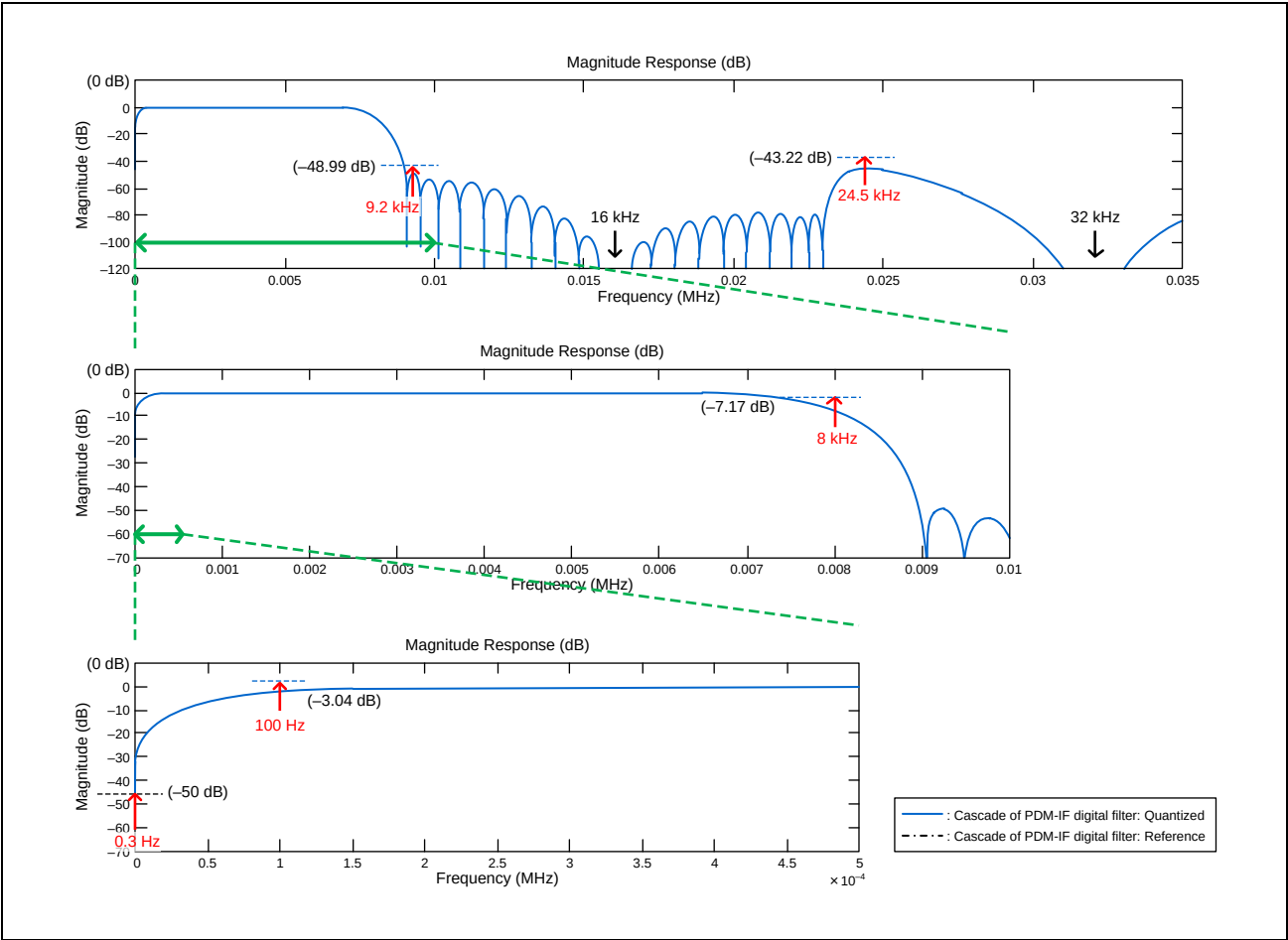


Figure 8.7-24 Frequency Characteristics of Filters with Default Settings

Table 8.7-25 Frequency Characteristics of Filters with Default Settings

Frequency (Hz)		Magnitude (dB)
Cut-off frequency of LPF	24.5 kHz	-43.22
	...	...
	9.2 kHz	-48.99
	8 kHz	-7.17
Passband	7 kHz	-0.39
	6 kHz	-0.07
	...	...
	150 Hz	-1.63
	100 Hz	-3.04
Cut-off frequency of HPF	50 Hz	-7.02
	0.3 Hz	-50.00

SECTION 9 IMAGE

9.1 Image Overview

This sections describes the Image unit of this LSI. For details, refer to the sections of each unit.

■ Camera Data Receive Unit (CRU) (See 9.2)

This LSI equips 2ch of CRU.

■ Image Scaling Unit (ISU) (See 9.3)

■ LCD Controller (LCDC) (See 9.4)

■ MIPI DSI (See 9.5)

Conforms to the following standards.

- MIPI® Alliance Specification for Display Serial Interface Version 1.3.1
- MIPI® Alliance Specification for D-PHY Version 1.2

■ H.265/H.264 Multi Codec (VCD) (See 9.6)

■ 3D Graphics Engine (GE3D) (See 9.7)

This LSI includes the Arm® Mali™-G31 GPU as a 3D Graphics Engine (GE3D). This unit is an optional function.

■ Image Signal Processor (ISP) (See 9.8)

This LSI includes the Arm® Mali™-C55 ISP core.

SECTION 9 IMAGE

9.2 Camera Data Receiver Unit (CRU)

9.2.1 Overview

Each CRU has an MIPI CSI-2 input and then decodes the data type specified by the packet, handles conversion of the color space of pixels, LUT conversion, pixel format conversion, and image conversion such as demosaicing and obtaining statistics. After converting processed data to various pixel formats, it transfers the data to external memory.

Each CRU consists of the MIPI CSI2 block and an image converter. The main functions of each of these blocks are as follows.

MIPI CSI2

The MIPI CSI2 block receives signals compliant with the MIPI CSI-2 V2.1(DPHY V1.2) standard, extracts video signals from various packets, and sends data to the image converter, which is the next stage. Output from CRU0 to CRU1 is possible.

The details of its functions are described in **9.2.3 MIPI CSI2**.

Image Converter

The image converter receives video data from the MIPI CSI2 block, that is, from the previous stage, and handles image processing corresponding to the given data. MIPI CSI2 data input to CRU0 can be selected for CRU1. Data to which image processing has been applied are initially stored in a FIFO buffer and then transferred to external memory.

The details of its functions are described in **9.2.4 Image Converter**.

Figure 9.2-1 is a schematic diagram of the CRUs. A total of two CRU channels are present.

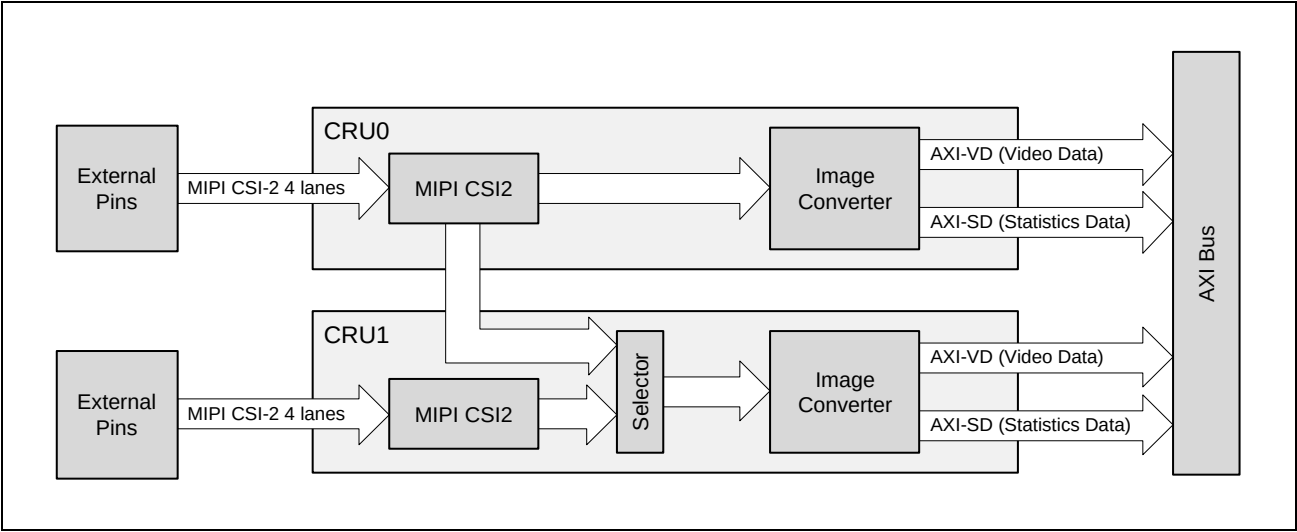


Figure 9.2-1 Schematic Diagram of CRU

9.2.2 Connected Units

Table 9.2-1 lists differences between the functions of CRU0 to 1.

Figure 9.2-2 and Figure 9.2-3 show the units that are connected to CRU0 and 1, respectively.

Table 9.2-1 Differences between Functions of CRU0 to 1

Ch.	Image Converter Input Data Ch. Selection
CRU0	CRU0
CRU1	CRU0/1

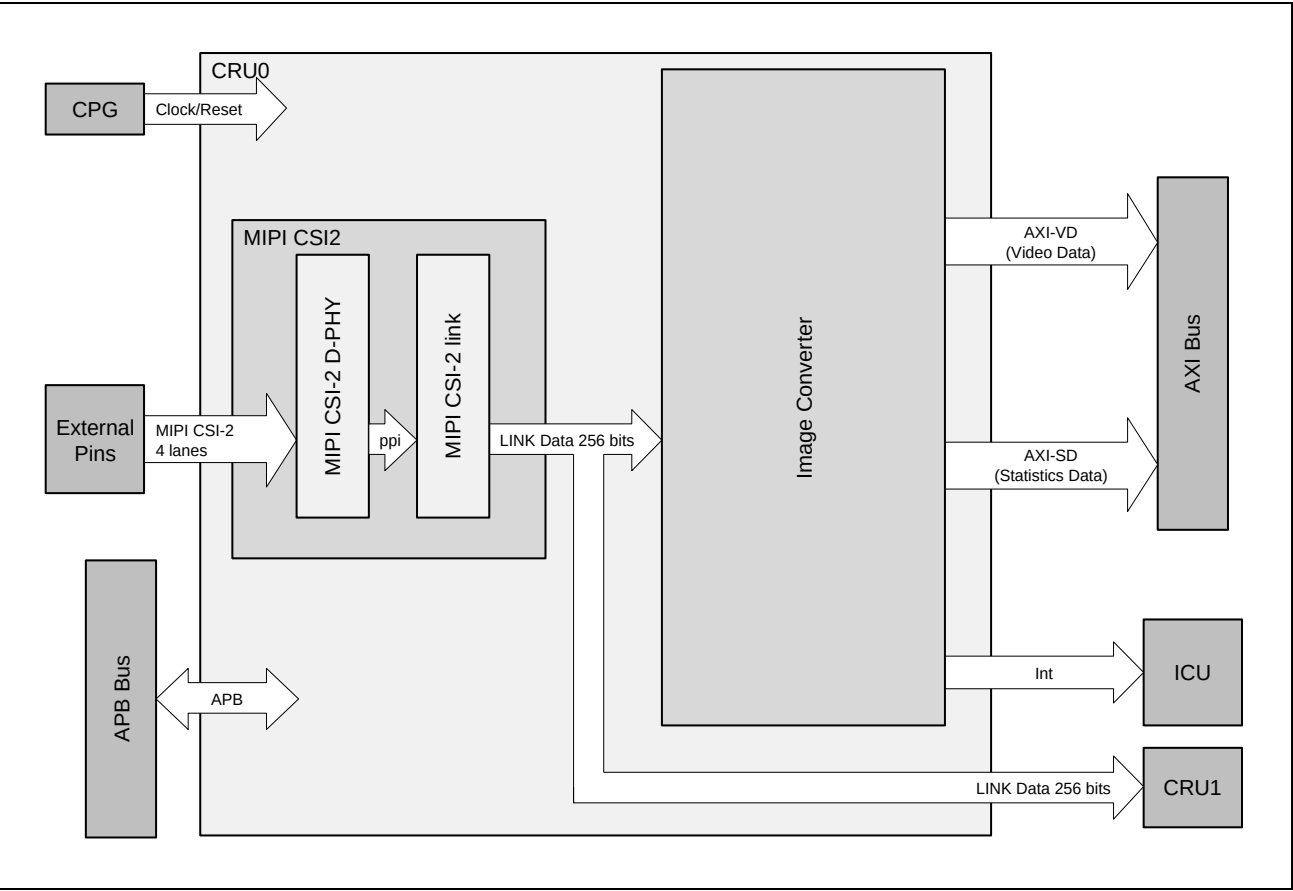


Figure 9.2-2 Block Diagram of CRU0

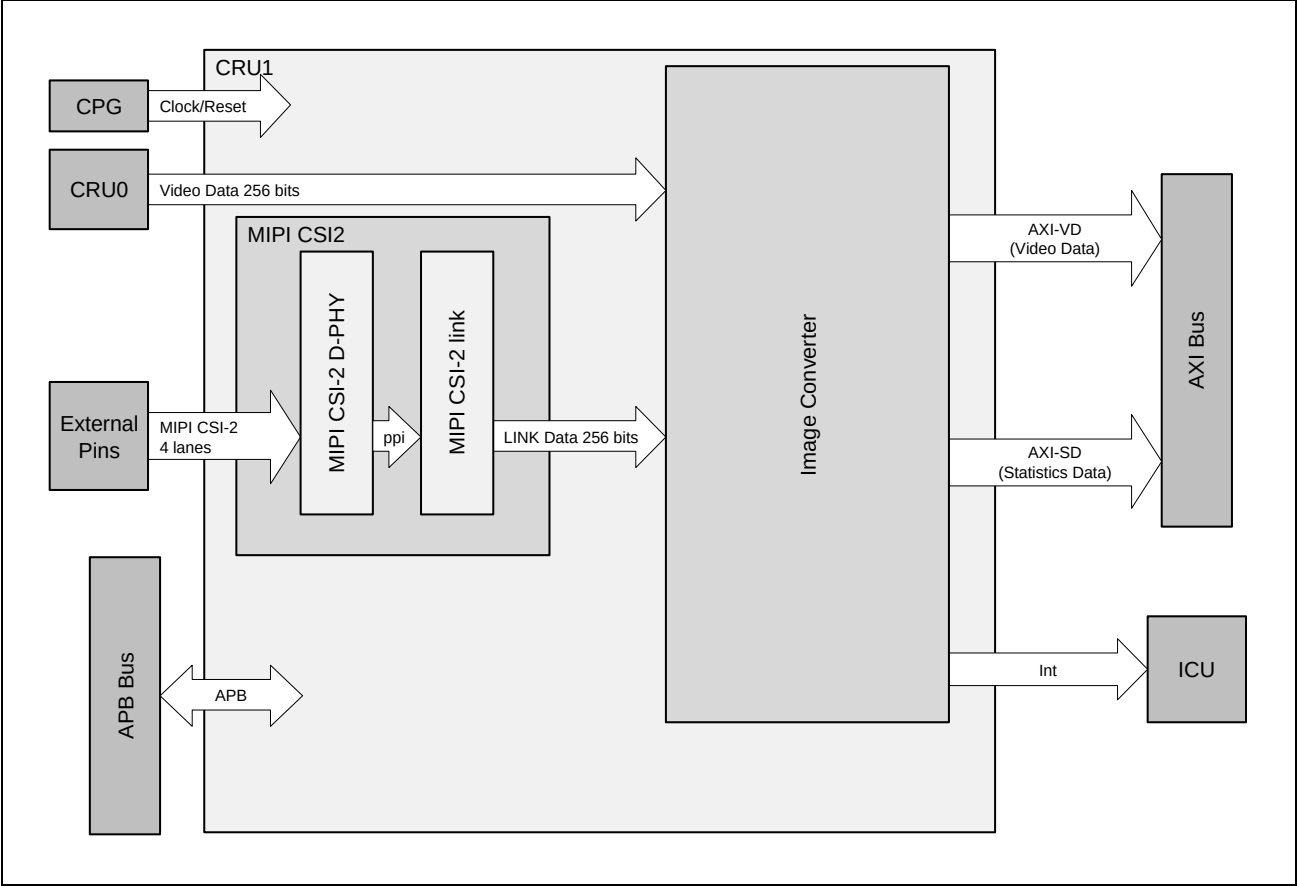


Figure 9.2-3 Block Diagram of CRU1

9.2.3 MIPI CSI2

9.2.3.1 Overview

MIPI CSI2 (camera serial interface 2) receives signals compliant with the DPHY standard from a camera module connected to external pins and decodes them as image data.

Data decoded by this module are sent to the image converter block as the next stage.

9.2.3.1.1 Features

This module has the following functions.

COMMON

1) Data input range

[Pixel size]

- Maximum number of pixels: 8.8 Mpixels; minimum number of pixels: QVGA (320 × 240) = 76.8 Kpixels
- Maximum number of effective pixels in the horizontal direction: 4096 pixels / 8192 pixels*¹
- Maximum number of effective pixels in the vertical direction: 4096 lines / 8192 pixels*¹

[Other than images]

- Data size range of input long packets: 1 to 24576 bytes

Note 1. 4096 lines when image processing and non-image are selected
8192 lines when full bypass is selected

2) Restrictions on the input bandwidth

Use the input bandwidth within the range which satisfies all conditions in **Table 9.2-2** and **Table 9.2-3**.

Table 9.2-2 Maximum Rate per Lane

Lane to be Used	Transfer Rate per Lane (Mbps)		Transfer Rate for All Lanes (Mbps)	
	MIN	MAX	MIN	MAX
1	80	2100	80	2100
2	80	2100	160	4200
4	80	2100	320	8400

Table 9.2-3 Maximum Transfer Rate per Format for Total of Four Lanes (1/2)

CRUnCTRL.ATH	Input Format	MIPI CSI-2 Transfer Rate (Mbps (max.))
0	YUV422 8-bit	8400
0	YUV422 10-bit	8400
0	RGB444	8400
0	RGB555	8400
0	RGB565	8400
0	RGB666	8400
0	RGB888	8400

Table 9.2-3 Maximum Transfer Rate per Format for Total of Four Lanes (2/2)

CRUnCTRL.ATH	Input Format	MIPI CSI-2 Transfer Rate (Mbps (max.))
0	RAW6	3751
0	RAW7	4359
0	RAW8	5040
0	RAW10	6203
0	RAW12	7331
0	RAW14	8400
0	RAW16	8400
0	RAW20	8400
1 (full bypass)	MIPI recommended memory storage format	8400

3) Restrictions on blanking

At least the time given below is required during the period from the last valid data on a line to the first valid data on the next line.

- When demosaicing is in use: $(300 + H\text{-size}) \times 1/vclk$ (MHz) or more
e.g. when $vclk = 630$ MHz, for VGA (640x480): $(300 + 640) \times 1/630$ (MHz) = approx. 1.50 μ s or more
- When demosaicing is not in use: $300 \times 1/vclk$ (MHz) or more
e.g. when $vclk = 630$ MHz: $300 \times 1/630$ (MHz) = approx. 477 ns or more

4) Compliant with the specification for camera serial interface 2 (CSI-2)SM

5) Receiver error detection and reporting

- D-PHY level errors
- Packet level errors
- Protocol decoding level errors

D-PHY

- 1) Support MIPI D-PHY V1.2 (80 Mbps to 2100 Mbps/lane)
- 2) Support 1, 2, and 4 lanes
- 3) Support for receiver deskewing by D-PHY
 - In operation at over 1.5 Gbps, periodic deskewing is essential.
 - The timing for periodic deskewing calibration: 2^{13} to 10 μ s
 - The timing for initial deskewing calibration: 2^{15} UI to 100 μ s
- 4) D-PHY Data lane type: CIL-SFEN (forward ULPS only).
- 5) Low Power Mode Max Speed 10 Mbps

LINK

- 1) Support MIPI CSI-2 V2.1
- 2) Support data de-scrambling
- 3) Support latency reduction and transport efficiency (LRTE)
- 4) ECC 1-bit error correction and 2-bit error detection in packet headers

5) CRC checking for payload data

6) Support 16 virtual channels

Note: Although the MIPI CSI2 block can receive 16 virtual channels (VC0 to VC15), 4 virtual channels can be selected from VC0 to VC15 in the image converter. Refer to **9.2.4.1.1 Features** for more detail.

7) Number of stages of the FIFO for generic short packets: 16

8) Support data interleaving

- Data type interleaving
 - Packet-level interleaved data transmission
 - Frame-level interleaved data transmission
- Virtual channel identifier interleaving

9.2.3.1.2 Block Diagram

Figure 9.2-4 is a block diagram of MIPI CSI2.

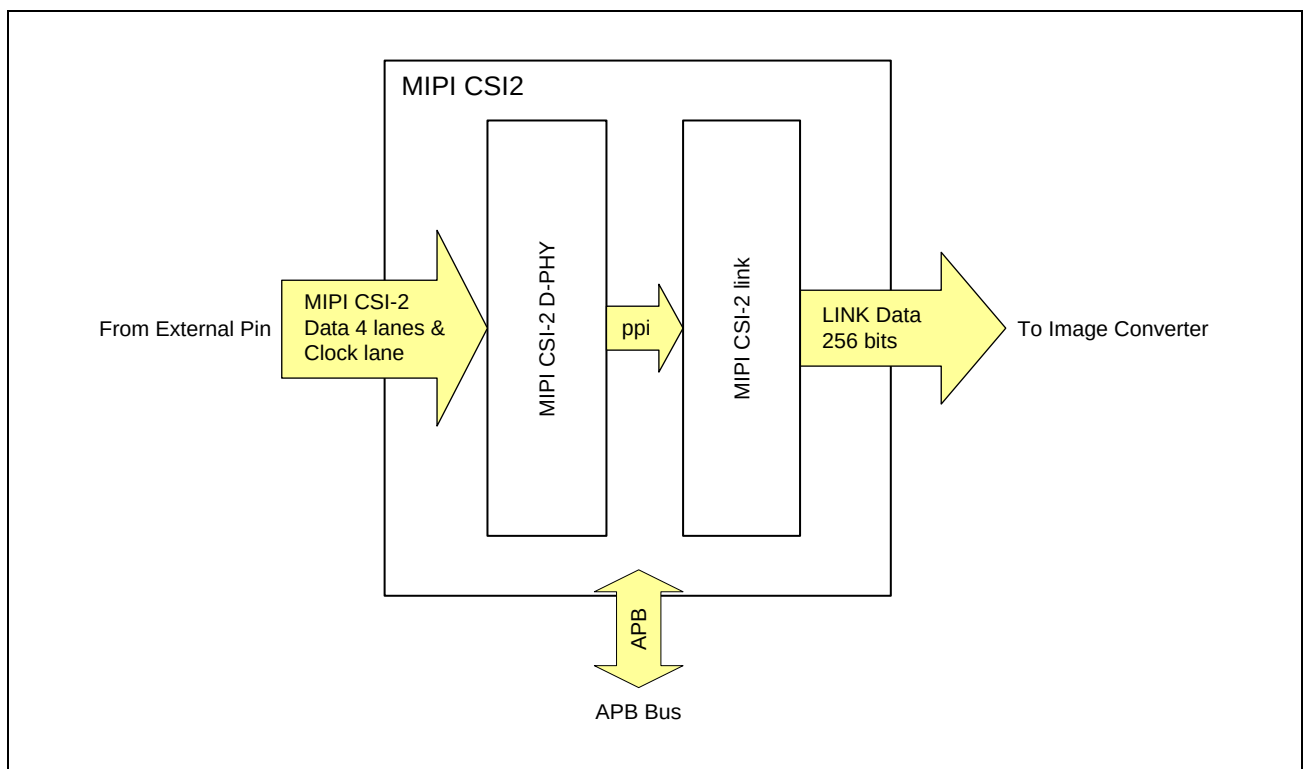


Figure 9.2-4 Block Diagram of MIPI-CSI2

9.2.3.1.3 External Pins

Table 9.2-4 lists the input/output pins of MIPI CSI2.

Table 9.2-4 MIPI CSI2 Input/Output Pins

Category	Pin Name	Input/Output	Description
DPHY Interface for MIPI CSI-2	CSI0_CLKP	Input	MIPI CSI-2 Rx clock input for CRU0 (positive)
	CSI0_CLKN	Input	MIPI CSI-2 Rx clock input for CRU0 (negative)
	CSI0_DATA0P	Input	MIPI CSI-2 Rx serial data input lane 0 for CRU0 (positive)
	CSI0_DATA0N	Input	MIPI CSI-2 Rx serial data input lane 0 for CRU0 (negative)
	CSI0_DATA1P	Input	MIPI CSI-2 Rx serial data input lane 1 for CRU0 (positive)
	CSI0_DATA1N	Input	MIPI CSI-2 Rx serial data input lane 1 for CRU0 (negative)
	CSI0_DATA2P	Input	MIPI CSI-2 Rx serial data input lane 2 for CRU0 (positive)
	CSI0_DATA2N	Input	MIPI CSI-2 Rx serial data input lane 2 for CRU0 (negative)
	CSI0_DATA3P	Input	MIPI CSI-2 Rx serial data input lane 3 for CRU0 (positive)
	CSI0_DATA3N	Input	MIPI CSI-2 Rx serial data input lane 3 for CRU0 (negative)
DPHY Interface for MIPI CSI-2	CSI1_CLKP	Input	MIPI CSI-2 Rx clock input for CRU1 (positive)
	CSI1_CLKN	Input	MIPI CSI-2 Rx clock input for CRU1 (negative)
	CSI1_DATA0P	Input	MIPI CSI-2 Rx serial data input lane 0 for CRU1 (positive)
	CSI1_DATA0N	Input	MIPI CSI-2 Rx serial data input lane 0 for CRU1 (negative)
	CSI1_DATA1P	Input	MIPI CSI-2 Rx serial data input lane 1 for CRU1 (positive)
	CSI1_DATA1N	Input	MIPI CSI-2 Rx serial data input lane 1 for CRU1 (negative)
	CSI1_DATA2P	Input	MIPI CSI-2 Rx serial data input lane 2 for CRU1 (positive)
	CSI1_DATA2N	Input	MIPI CSI-2 Rx serial data input lane 2 for CRU1 (negative)
	CSI1_DATA3P	Input	MIPI CSI-2 Rx serial data input lane 3 for CRU1 (positive)
	CSI1_DATA3N	Input	MIPI CSI-2 Rx serial data input lane 3 for CRU1 (negative)

9.2.3.2 MIPI CSI2 Registers

The base addresses of the CRUm registers are shown below.

Base Address Name	Base Address
<CRU0_base>	0_1600_0000h (5600_0000h* ¹ , 4600_0000h* ²)
<CRU1_base>	0_1601_0000h (5601_0000h* ¹ , 4601_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.2.3.2.1 List of Registers

(1) CRU LINK Registers (m = 0, 1)

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Module Configuration Register	CRUm_CSI2nMCG	0010_0400h	0400h	32
Reserve	-	-	0404h to 040Fh	-
Module Control Register 0	CRUm_CSI2nMCT0	0200_0004h	0410h	32
Reserve	-	-	0414h to 0417h	-
Module Control Register 2	CRUm_CSI2nMCT2	0000_0000h	0418h	32
Module Control Register 3	CRUm_CSI2nMCT3	0000_0000h	041Ch	32
Reserve	-	-	0420h to 0427h	-
Reset Control Register	CRUm_CSI2nRTCT	0000_0000h	0428h	32
Reset Status Register	CRUm_CSI2nRTST	0000_0000h	042Ch	32
Reserve	-	-	0430h to 043Fh	-
EPD Option Control Register	CRUm_CSI2nEPCT	0000_0000h	0440h	32
Reserve	-	-	0444h to 044Fh	-
Module Interrupt Status Register	CRUm_CSI2nMIST	0000_0000h	0450h	32
Reserve	-	-	0454h to 045Fh	-
Receive Data Type Enable Low Register	CRUm_CSI2nDTEL	0000_000Fh	0460h	32
Receive Data Type Enable High Register	CRUm_CSI2nDTEH	0000_0000h	0464h	32
Reserve	-	-	0468h to 046Fh	-
Receive Status Register	CRUm_CSI2nRXST	0000_0000h	0470h	32
Receive Status Clear Register	CRUm_CSI2nRXSC	0000_0000h	0474h	32
Receive Interrupt Enable Register	CRUm_CSI2nRXIE	0000_0000h	0478h	32
Reserve	-	-	047Ch to 047Fh	-
Data Lane 0 Status Register	CRUm_CSI2nDLST0	0000_0000h	0480h	32
Data Lane 0 Status Clear Register	CRUm_CSI2nDLSC0	0000_0000h	0484h	32
Data Lane 0 Interrupt Enable Register	CRUm_CSI2nDLIE0	0000_0000h	0488h	32
Reserve	-	-	048Ch to 048Fh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Data Lane 1 Status Register	CRUm_CSI2nDLST1	0000_0000h	0490h	32
Data Lane 1 Status Clear Register	CRUm_CSI2nDLSC1	0000_0000h	0494h	32
Data Lane 1 Interrupt Enable Register	CRUm_CSI2nDLIE1	0000_0000h	0498h	32
Reserve	-	-	049Ch to 049Fh	-
Data Lane 2 Status Register	CRUm_CSI2nDLST2	0000_0000h	04A0h	32
Data Lane 2 Status Clear Register	CRUm_CSI2nDLSC2	0000_0000h	04A4h	32
Data Lane 2 Interrupt Enable Register	CRUm_CSI2nDLIE2	0000_0000h	04A8h	32
Reserve	-	-	04ACh to 04AFh	-
Data Lane 3 Status Register	CRUm_CSI2nDLST3	0000_0000h	04B0h	32
Data Lane 3 Status Clear Register	CRUm_CSI2nDLSC3	0000_0000h	04B4h	32
Data Lane 3 Interrupt Enable Register	CRUm_CSI2nDLIE3	0000_0000h	04B8h	32
Reserve	-	-	04BCh to 04FFh	-
Virtual Channel 0 Status Register	CRUm_CSI2nVCST0	0000_0000h	0500h	32
Virtual Channel 0 Status Clear Register	CRUm_CSI2nVCSC0	0000_0000h	0504h	32
Virtual Channel 0 Interrupt Enable Register	CRUm_CSI2nVCIE0	0000_0000h	0508h	32
Reserve	-	-	050Ch to 05FFh	-
Virtual Channel 1 Status Register	CRUm_CSI2nVCST1	0000_0000h	0510h	32
Virtual Channel 1 Status Clear Register	CRUm_CSI2nVCSC1	0000_0000h	0514h	32
Virtual Channel 1 Interrupt Enable Register	CRUm_CSI2nVCIE1	0000_0000h	0518h	32
Reserve	-	-	051Ch to 051Fh	-
Virtual Channel 2 Status Register	CRUm_CSI2nVCST2	0000_0000h	0520h	32
Virtual Channel 2 Status Clear Register	CRUm_CSI2nVCSC2	0000_0000h	0524h	32
Virtual Channel 2 Interrupt Enable Register	CRUm_CSI2nVCIE2	0000_0000h	0528h	32
Reserve	-	-	052Ch to 052Fh	-
Virtual Channel 3 Status Register	CRUm_CSI2nVCST3	0000_0000h	0530h	32
Virtual Channel 3 Status Clear Register	CRUm_CSI2nVCSC3	0000_0000h	0534h	32
Virtual Channel 3 Interrupt Enable Register	CRUm_CSI2nVCIE3	0000_0000h	0538h	32
Reserve	-	-	053Ch to 053Fh	-
Virtual Channel 4 Status Register	CRUm_CSI2nVCST4	0000_0000h	0540h	32
Virtual Channel 4 Status Clear Register	CRUm_CSI2nVCSC4	0000_0000h	0544h	32
Virtual Channel 4 Interrupt Enable Register	CRUm_CSI2nVCIE4	0000_0000h	0548h	32
Reserve	-	-	054Ch to 054Fh	-
Virtual Channel 5 Status Register	CRUm_CSI2nVCST5	0000_0000h	0550h	32
Virtual Channel 5 Status Clear Register	CRUm_CSI2nVCSC5	0000_0000h	0554h	32
Virtual Channel 5 Interrupt Enable Register	CRUm_CSI2nVCIE5	0000_0000h	0558h	32
Reserve	-	-	055Ch to 055Fh	-
Virtual Channel 6 Status Register	CRUm_CSI2nVCST6	0000_0000h	0560h	32
Virtual Channel 6 Status Clear Register	CRUm_CSI2nVCSC6	0000_0000h	0564h	32
Virtual Channel 6 Interrupt Enable Register	CRUm_CSI2nVCIE6	0000_0000h	0568h	32
Reserve	-	-	056Ch to 056Fh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Virtual Channel 7 Status Register	CRUm_CSI2nVCST7	0000_0000h	0570h	32
Virtual Channel 7 Status Clear Register	CRUm_CSI2nVCSC7	0000_0000h	0574h	32
Virtual Channel 7 Interrupt Enable Register	CRUm_CSI2nVCIE7	0000_0000h	0578h	32
Reserve	-	-	057Ch to 057Fh	-
Virtual Channel 8 Status Register	CRUm_CSI2nVCST8	0000_0000h	0580h	32
Virtual Channel 8 Status Clear Register	CRUm_CSI2nVCSC8	0000_0000h	0584h	32
Virtual Channel 8 Interrupt Enable Register	CRUm_CSI2nVCIE8	0000_0000h	0588h	32
Reserve	-	-	058Ch to 058Fh	-
Virtual Channel 9 Status Register	CRUm_CSI2nVCST9	0000_0000h	0590h	32
Virtual Channel 9 Status Clear Register	CRUm_CSI2nVCSC9	0000_0000h	0594h	32
Virtual Channel 9 Interrupt Enable Register	CRUm_CSI2nVCIE9	0000_0000h	0598h	32
Reserve	-	-	059Ch to 059Fh	-
Virtual Channel 10 Status Register	CRUm_CSI2nVCST10	0000_0000h	05A0h	32
Virtual Channel 10 Status Clear Register	CRUm_CSI2nVCSC10	0000_0000h	05A4h	32
Virtual Channel 10 Interrupt Enable Register	CRUm_CSI2nVCIE10	0000_0000h	05A8h	32
Reserve	-	-	05ACh to 05AFh	-
Virtual Channel 11 Status Register	CRUm_CSI2nVCST11	0000_0000h	05B0h	32
Virtual Channel 11 Status Clear Register	CRUm_CSI2nVCSC11	0000_0000h	05B4h	32
Virtual Channel 11 Interrupt Enable Register	CRUm_CSI2nVCIE11	0000_0000h	05B8h	32
Reserve	-	-	05BCh to 05BFh	-
Virtual Channel 12 Status Register	CRUm_CSI2nVCST12	0000_0000h	05C0h	32
Virtual Channel 12 Status Clear Register	CRUm_CSI2nVCSC12	0000_0000h	05C4h	32
Virtual Channel 12 Interrupt Enable Register	CRUm_CSI2nVCIE12	0000_0000h	05C8h	32
Reserve	-	-	05CCh to 05CFh	-
Virtual Channel 13 Status Register	CRUm_CSI2nVCST13	0000_0000h	05D0h	32
Virtual Channel 13 Status Clear Register	CRUm_CSI2nVCSC13	0000_0000h	05D4h	32
Virtual Channel 13 Interrupt Enable Register	CRUm_CSI2nVCIE13	0000_0000h	05D8h	32
Reserve	-	-	05DCh to 05DFh	-
Virtual Channel 14 Status Register	CRUm_CSI2nVCST14	0000_0000h	05E0h	32
Virtual Channel 14 Status Clear Register	CRUm_CSI2nVCSC14	0000_0000h	05E4h	32
Virtual Channel 14 Interrupt Enable Register	CRUm_CSI2nVCIE14	0000_0000h	05E8h	32
Reserve	-	-	05ECh to 05EFh	-
Virtual Channel 15 Status Register	CRUm_CSI2nVCST15	0000_0000h	05F0h	32
Virtual Channel 15 Status Clear Register	CRUm_CSI2nVCSC15	0000_0000h	05F4h	32
Virtual Channel 15 Interrupt Enable Register	CRUm_CSI2nVCIE15	0000_0000h	05F8h	32
Reserve	-	-	05FCh to 05FFh	-
Power Management Status Register	CRUm_CSI2nPMST	000F_4000h	0600h	32
Power Management Status Clear Register	CRUm_CSI2nPMSC	0000_0000h	0604h	32
Power Management Interrupt Enable Register	CRUm_CSI2nPMIE	0000_0000h	0608h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Reserve	-	-	060Ch to 067Fh	-
Generic Short Packet Control Register	CRUm_CSI2nGSCT	0001_0000h	0680h	32
Generic Short Packet Status Register	CRUm_CSI2nGSST	0000_0000h	0684h	32
Generic Short Packet Status Clear Register	CRUm_CSI2nGSSC	0000_0000h	0688h	32
Generic Short Packet Interrupt Enable Register	CRUm_CSI2nGSIE	0000_0000h	068Ch	32
Generic Short Packet Register	CRUm_CSI2nGSHT	0008_0000h	0690h	32
Generic Short Packet Information Update Register	CRUm_CSI2nGSIU	0000_0000h	0694h	32

(2) CRU PHY Registers (m = 0, 1)

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Slave Timing Control Register	CRUm_S_TIMCTL	0000_0000h	081Ch	32
Reserve	-	-	0820h to 0833h	-
Slave D-PHY Control Register (MSB)	CRUm_S_DPHYCTL_MSB	0000_0090h	0834h	32
DP/DN Swap Control Register	CRUm_SWAPCTL	0000_0000h	0838h	32
Reserve	-	-	083Ch to 086Bh	-
RX Skew Calibration Results 0	CRUm_S_ATB_OUT_0	0000_0000h	086Ch	32
RX Skew Calibration Results 1	CRUm_S_ATB_OUT_1	0000_0000h	0870h	32
RX Skew Calibration Results 2	CRUm_S_ATB_OUT_2	0000_0000h	0874h	32
RX Skew Calibration Results 3	CRUm_S_ATB_OUT_3	0000_0000h	0878h	32

9.2.3.2.2 Register Descriptions

(1) Module Configuration Register (CRUm_CSI2nMCG) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0400h															
Initial Value :		0010_0400h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	GSNM[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	SDLN[3:0]				-	-	-	-	VER[3:0]				
Initial Value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	GSNM[7:0]	10h	R	Indicate the number of stages of the generic short packet FIFO (16 stages).
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 8	SDLN[3:0]	4h	R	Indicate the supported maximum number of data lanes. 4h: Operable with four lanes, two lanes or one lane.
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	VER[3:0]	0h	R	Indicate the version of this core.

(2) Module Control Register 0 (CRUm_CSI2nMCT0) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0410h

Initial Value : 0200_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	LFSRE N	ECCV1 3	-	-	-	-	RVMD	-	EDMD	ZLMD
Initial Value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	R	R	R	R0W	RW	R	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	VDLN[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25	LFSREN	1h	RW	Enables or disables de-scrambling. 0b: Disables de-scrambling. 1b: Enables de-scrambling.
24	ECCV13	0h	RW	CSI-2 specification-compliant mode for ECC checking 0b: 26 bits to be checked by ECC 1b: 24 bits to be checked by ECC It is prohibited to set this bit to 1b when receiving a packet whose virtual channel ID is from 4 to 15.
23 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	-	0h	R0W	The read value is always 0b. Do not change the value of this bit from 0h.
19	RVMD	0h	RW	For a packet whose data type is 38h to 3Fh ("Reserved" in "Table 10 Data Type Classes" of the CSI-2 specification), set the receive operation mode to 0. 0b: Discards the data (notifies an ErrID). 1b: Receives data as a long packet (outputs data to the image processing module when the corresponding bit in CSI2nDTEH is 1b, or discards the data and notifies an ErrID when the corresponding bit in CSI2nDTEH is 0b).
18	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	EDMD	0h	RW	Notifies ErrFrameData when an ECC 2-bit error or a packet of less than four bytes is received between FS and FE. 0b: Does not notify ErrFrameData. 1b: Notifies ErrFrameData.
16	ZLMD	0h	RW	Sets whether to output a long packet whose word count is 0 to the image processing module. Set this bit to 0b. 0b: Output 1b: No output
15 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	VDLN[3:0]	4h	RW	Set the number of used data lanes. Change this field when CSI2nMCT3.RXEN = 0b. A value equal to or less than the CSI2nMCG.SDLN value can be specified. 1h: Operation with one lane 2h: Operation with two lanes 4h: Operation with four lanes Others: Setting prohibited

(3) Module Control Register 2 (CRUm_CSI2nMCT2) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0418h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	FRRSKW[8:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	FRRCLK[8:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24 to 16	FRRSKW[8:0]	0h	RW	Internal setting parameter Set the following value by using the ROUNDDOWN function that rounds down the decimal places. (ROUNDDOWN((3 * vclk frequency/hsc1k1 frequency), 0) + 1) where vclk = 630 MHz, hsc1k1 = transfer rate/8 Example: Set 10h when the vclk frequency is 630 MHz and the transfer rate is 1 Gbps. ROUNDDOWN((3 × 630/(1000/8)), 0) + 1 = 16d
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8 to 0	FRRCLK[8:0]	0h	RW	Parameter for the IP to determine whether packets have been received if RxWordClkHS stops at the timing when the IP cannot recognize that RxValidHS is low. Set the following value by using the ROUNDDOWN function that rounds down the decimal places. (ROUNDDOWN((1.5 * vclk frequency/hsc1k1 frequency), 0) + 1) where vclk = 630 MHz, hsc1k1 = transfer rate/8 Example: Set 8h when the vclk frequency is 266 MHz and the transfer rate is 1 Gbps. ROUNDDOWN((1.5 × 630/(1000/8)), 0) + 1 = 8d

(4) Module Control Register 3 (CRUm_CSI2nMCT3) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 041Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RXEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	RXEN		0h		RW		Enables LINK data reception. 0b: Disables reception. 1b: Enables reception.									

(5) Reset Control Register (CRUm_CSI2nRTCT) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0428h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	VSRST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W1
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	VSRST		0h		R0W1		The read value is always 0b. Writing 1 to this bit resets the internal LINK signal by a software reset.									

(6) Reset Status Register (CRUm_CSI2nRTST) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 042Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	VSRSTS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	VSRSTS		0h		R	Indicates the execution status of the software reset generated by CSI2nRTCT.VSRST. 0b: Not during a reset 1b: During a reset										

(7) EPD Option Control Register (CRUm_CSI2nEPCT) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0440h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDEN	SSP[14:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDOP	SLP[14:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	EPDEN	0h	RW	Enables the EPD. 0b: Disables the EPD. 1b: Enables the EPD.
30 to 16	SSP[14:0]	0h	RW	Specify the number of spacers inserted after a short packet. This field is valid when EPDEN = 1b. The following shows the specifiable minimum number of spacers when EPDEN = 1b. Setting a value less than the minimum value is prohibited. When EPDOP = 1b, the minimum spacer value is (1) The larger value of the following two values: • Minimum number of spacers due to the D-PHY setting • Minimum number of spacers due to the number of used lanes (2) Minimum number of spacers due to the D-PHY setting: The minimum value can be found by using the ROUNDUP function that rounds up the decimal places. $\text{ROUNDUP}(((105/U[\text{ns}] + 20) / 8), 0)$ (3) Minimum number of spacers due to the number of used lanes: 1 lane: minimum spacer = 64 2 lanes: minimum spacer = 32 4 lanes: minimum spacer = 32
15	EPDOP	0h	RW	Selects an EPD option. Set this bit to 1b. 0b: D-PHY EPD Option 1 (0 cannot be set when EPDEN = 1b) 1b: D-PHY EPD Option 2
14 to 0	SLP[14:0]	0h	RW	Specify the number of spacers inserted after a long packet. This field is valid when EPDEN = 1b. When EPDEN = 1b, the minimum number of spacers is the value described in the SSP field. Setting a value less than the minimum value is prohibited.

(8) Module Interrupt Status Register (CRUm_CSI2nMIST) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0450h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC15S	VC14S	VC13S	VC12S	VC11S	VC10S	VC9S	VC8S	VC7S	VC6S	VC5S	VC4S	VC3S	VC2S	VC1S	VC0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RXS	GSTS	PMS	-	-	-	-	DL3S	DL2S	DL1S	DL0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	VC15S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 15. Check the CSI2nVCST15 register for the details of the interrupt source.
30	VC14S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 14. Check the CSI2nVCST14 register for the details of the interrupt source.
29	VC13S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 13. Check the CSI2nVCST13 register for the details of the interrupt source.
28	VC12S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 12. Check the CSI2nVCST12 register for the details of the interrupt source.
27	VC11S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 11. Check the CSI2nVCST11 register for the details of the interrupt source.
26	VC10S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 10. Check the CSI2nVCST10 register for the details of the interrupt source.
25	VC9S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 9. Check the CSI2nVCST9 register for the details of the interrupt source.
24	VC8S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 8. Check the CSI2nVCST8 register for the details of the interrupt source.
23	VC7S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 7. Check the CSI2nVCST7 register for the details of the interrupt source.
22	VC6S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 6. Check the CSI2nVCST6 register for the details of the interrupt source.
21	VC5S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 5. Check the CSI2nVCST5 register for the details of the interrupt source.
20	VC4S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 4. Check the CSI2nVCST4 register for the details of the interrupt source.
19	VC3S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 3. Check the CSI2nVCST3 register for the details of the interrupt source.
18	VC2S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 2. Check the CSI2nVCST2 register for the details of the interrupt source.
17	VC1S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 1. Check the CSI2nVCST1 register for the details of the interrupt source.
16	VC0S	0h	R	When this bit is set to 1b, there is an interrupt source for virtual channel 0. Check the CSI2nVCST0 register for the details of the interrupt source.
15 to 11	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10	RXS	0h	R	When this bit is set to 1b, there is an interrupt source for the reception in general. Check the CSI2nRXST register for the details of the interrupt source.
9	GSTS	0h	R	When this bit is set to 1b, there is an interrupt source for reception of generic short packets. Check the CSI2nGSST register for the details of the interrupt source.
8	PMS	0h	R	When this bit is set to 1b, there is an interrupt source for the power management. Check the CSI2nPMST register for the details of the interrupt source.
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
3	DL3S	0h	R	When this bit is set to 1b, there is an interrupt source for data lane 3. Check the CSI2nDLST3 register for the details of the interrupt source.
2	DL2S	0h	R	When this bit is set to 1b, there is an interrupt source for data lane 2. Check the CSI2nDLST2 register for the details of the interrupt source.
1	DL1S	0h	R	When this bit is set to 1b, there is an interrupt source for data lane 1. Check the CSI2nDLST1 register for the details of the interrupt source.
0	DL0S	0h	R	When this bit is set to 1b, there is an interrupt source for data lane 0. Check the CSI2nDLST0 register for the details of the interrupt source.

(9) Receive Data Type Enable Low Register (CRUm_CSI2nDTEL) (m = 0, 1)

Access Size :		32 bits																	
Address :		<CRUm_base> + 0460h																	
Initial Value :		0000_000Fh																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
DTEN[31:16]																			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
DTEN[15:0]																			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1			
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R			
Bit	Bit Name																Initial Value	R/W	Description
31 to 0	DTEN[31:0]																0000_000Fh	RW (R)	Receive data of a packet in which data type is one of the following. 0b: Does not receive data. 1b: Receives data.
																	Note: After multiple data types are selected here, only one data type is selected by the INF[5:0] bits in the ICnIPMC_C0/1/2/3 register in the image processing module. See Table 9.2-5 for details on setting this bit.		

Table 9.2-5 Receive Data Type Enable Low Register (CSI2nDTEL)

Bit	Data Type	R/W	Description
31	YUV422 10-bit	RW	When this bit is set to 1b, data is output to the image processing module.
30	YUV422 8-bit	RW	
29	YUV420 10-bit	RW	
28	YUV420 8-bit	RW	
27	Reserved	RW	
26	Legacy YUV420 8-bit	RW	
25	YUV420 10-bit	RW	
24	YUV420 8-bit	RW	
23	Reserved	RW	
22	Generic long packet data type 4	RW	
21	Generic long packet data type 3	RW	
20	Generic long packet data type 2	RW	
19	Generic long packet data type 1	RW	
18	Embedded 8-bit non-image data	RW	
17	Blanking data	RW	
16	Null	RW	
15	Generic Short Packet Code 8	RW	Fixed to 1b for each bit.
14	Generic Short Packet Code 7	RW	
13	Generic Short Packet Code 6	RW	When each bit is set to 1b and CSI2nGSCT.GFIF = 1, data is output to the generic short packet FIFO.
12	Generic Short Packet Code 5	RW	
11	Generic Short Packet Code 4	RW	
10	Generic Short Packet Code 3	RW	
9	Generic Short Packet Code 2	RW	
8	Generic Short Packet Code 1	RW	
7 to 4	Reserved	R	Fixed to 0.
3 to 0	Reserved	R	Fixed to 1.

(10) Receive Data Type Enable High Register (CRUm_CSI2nDTEH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0464h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTEN[63:48]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEN[47:32]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DTEN[63:32]	0000_0000h	RW	Receive data of a packet in which data type is one of the following. 0b: Does not receive data. 1b: Receives data.

Note: See **Table 9.2-6** for details on setting this bit.

Table 9.2-6 Receive Data Type Enable High Register (CSI2nDTEH)

Bit	Data Type	R/W	Description
31	Reserved	RW	Changing the reserved bit from the initial value is prohibited.
30 to 24	Reserved	RW	When this bit is set to 1b and CSI2nMCT0.RVMD = 1, data is output to the image processing module.
23	User Defined 8-bit Data Type 8	RW	When this bit is set to 1b, data is output to the image processing module.
22	User Defined 8-bit Data Type 7	RW	
21	User Defined 8-bit Data Type 6	RW	
20	User Defined 8-bit Data Type 5	RW	
19	User Defined 8-bit Data Type 4	RW	
18	User Defined 8-bit Data Type 3	RW	
17	User Defined 8-bit Data Type 2	RW	
16	User Defined 8-bit Data Type 1	RW	
15	RAW20	RW	
14	RAW16	RW	
13	RAW14	RW	
12	RAW12	RW	
11	RAW10	RW	
10	RAW8	RW	
9	RAW7	RW	
8	RAW6	RW	
7 to 5	Reserved	RW	
4	RGB888	RW	
3	RGB666	RW	
2	RGB565	RW	
1	RGB555	RW	
0	RGB444	RW	

(11) Receive Status Register (CRUm_CSI2nRXST) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0470h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RACTDET	RACT
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRM15	FRM14	FRM13	FRM12	FRM11	FRM10	FRM9	FRM8	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RACTDET	0h	R	When this bit is set to 1b, CSI2nRXST.RACT = 1b has been detected. Write 1 to CSI2nRXSC.RACTDETC to clear this bit.
16	RACT	0h	R	Indicates that this IP is receiving packets.
15	FRM15	0h	R	Indicates that frames on virtual channel = 15 are being processed.
14	FRM14	0h	R	Indicates that frames on virtual channel = 14 are being processed.
13	FRM13	0h	R	Indicates that frames on virtual channel = 13 are being processed.
12	FRM12	0h	R	Indicates that frames on virtual channel = 12 are being processed.
11	FRM11	0h	R	Indicates that frames on virtual channel = 11 are being processed.
10	FRM10	0h	R	Indicates that frames on virtual channel = 10 are being processed.
9	FRM9	0h	R	Indicates that frames on virtual channel = 9 are being processed.
8	FRM8	0h	R	Indicates that frames on virtual channel = 8 are being processed.
7	FRM7	0h	R	Indicates that frames on virtual channel = 7 are being processed.
6	FRM6	0h	R	Indicates that frames on virtual channel = 6 are being processed.
5	FRM5	0h	R	Indicates that frames on virtual channel = 5 are being processed.
4	FRM4	0h	R	Indicates that frames on virtual channel = 4 are being processed.
3	FRM3	0h	R	Indicates that frames on virtual channel = 3 are being processed.
2	FRM2	0h	R	Indicates that frames on virtual channel = 2 are being processed.
1	FRM1	0h	R	Indicates that frames on virtual channel = 1 are being processed.
0	FRM0	0h	R	Indicates that frames on virtual channel = 0 are being processed.

(12) Receive Status Clear Register (CRUm_CSI2nRXSC) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0474h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RACTD ETC	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W1	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RACTDETC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nRXST.RACTDET. Writing 0b to this bit has no effect.
16 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(13) Receive Interrupt Enable Register (CRUm_CSI2nRXIE) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0478h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RACTD ETE	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RACTDETE	0h	RW	Enables interrupts generated by CSI2nRXST.RACTDET. 0b: Does not assert csi2_int_rx when CSI2nRXST.RACTDET = 1b. 1b: Asserts csi2_int_rx when CSI2nRXST.RACTDET = 1b.
16 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(14) Data Lane (N) Status Register (CRUm_CSI2nDLST(N)) (m = 0, 1)

(N) indicates a logical data lane number.

(N) = 0, 1, 2, or 3

Access Size : 32 bits**Address :** <CRUm_base> + 0480h + (N) x 010h**Initial Value :** 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	ULP (N)	-	-	-	-	-	-	RUL (N)	EUL (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	EES (N)	ECT (N)	-	ESH (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	ULP(N)	0h	R	Indicates the RxUlpEsc status of logical data lane (N).
23 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RUL(N)	0h	R	When this bit is set to 1b, logical data lane (N) has transitioned to the ULPS receive mode (RxUlpEsc). Write 1b to CSI2nDLSC(N).RULC(N) to clear this bit.
16	EUL(N)	0h	R	When this bit is set to 1b, logical data lane (N) has exited from the ULPS receive mode (RxUlpEsc). Write 1b to CSI2nDLSC(N).EULC(N) to clear this bit.
15 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	EES(N)	0h	R	When this bit is set to 1b, an escape mode entry error (ErrESC) has occurred on logical data lane (N). Write 1b to CSI2nDLSC(N).EESC(N) to clear this bit.
2	ECT(N)	0h	R	When this bit is set to 1b, a control error (ErrControl) has occurred on logical data lane (N). Write 1b to CSI2nDLSC(N).ECTC(N) to clear this bit.
1	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	ESH(N)	0h	R	When this bit is set to 1b, an SoT error (ErrSoT) has occurred on logical data lane (N). Write 1b to CSI2nDLSC(N).ESH(N) to clear this bit.

(15) Data Lane (N) Status Clear Register (CRUm_CSI2nDLSC(N)) (m = 0, 1)

(N) indicates a logical data lane number.

(N) = 0, 1, 2, or 3

Access Size : 32 bits

Address : <CRUm_base> + 0484h + (N) x 010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RULC (N)	EULC (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W1	R0W1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	EESC (N)	ECTC (N)	-	ESHC (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R0W1	R0W1	R	R0W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RULC(N)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nDLST(N).RUL(N). Writing 0b to this bit has no effect.
16	EULC(N)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nDLST(N).EUL(N). Writing 0b to this bit has no effect.
15 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	EESC(N)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nDLST(N).EES(N). Writing 0b to this bit has no effect.
2	ECTC(N)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nDLST(N).ECT(N). Writing 0b to this bit has no effect.
1	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	ESHC(N)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nDLST(N).ESH(N). Writing 0b to this bit has no effect.

(16) Data Lane (N) Interrupt Enable Register (CRUm_CSI2nDLIE(N)) (m = 0, 1)

(N) indicates a logical data lane number.

(N) = 0, 1, 2, or 3

Access Size : 32 bits

Address : <CRUm_base> + 0488h + (N) x 010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RULE (N)	EULE (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	EESE (N)	ECTE (N)	-	ESHE (N)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	RULE(N)	0h	RW	Enables interrupts generated by CSI2nDLST(N).RUL(N). 0b: Does not assert csi2_int_dl when CSI2nDLST(N).RUL(N) = 1b. 1b: Asserts csi2_int_dl when CSI2nDLST(N).RUL(N) = 1b.
16	EULE(N)	0h	RW	Enables interrupts generated by CSI2nDLST(N).EUL(N). 0b: Does not assert csi2_int_dl when CSI2nDLST(N).EUL(N) = 1b. 1b: Asserts csi2_int_dl when CSI2nDLST(N).EUL(N) = 1b.
15 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	EESE(N)	0h	RW	Enables interrupts generated by CSI2nDLST(N).EES(N). 0b: Does not assert csi2_int_dl when CSI2nDLST(N).EES(N) = 1b. 1b: Asserts csi2_int_dl when CSI2nDLST(N).EES(N) = 1b.
2	ECTE(N)	0h	RW	Enables interrupts generated by CSI2nDLST(N).ECT(N). 0b: Does not assert csi2_int_dl when CSI2nDLST(N).ECT(N) = 1b. 1b: Asserts csi2_int_dl when CSI2nDLST(N).ECT(N) = 1b.
1	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	ESHE(N)	0h	RW	Enables interrupts generated by CSI2nDLST(N).ESH(N). 0b: Does not assert csi2_int_dl when CSI2nDLST(N).ESH(N) = 1b. 1b: Asserts csi2_int_dl when CSI2nDLST(N).ESH(N) = 1b.

(17) Virtual Channel (M) Status Register (CRUm_CSI2nVCST(M)) (m = 0, 1)

(M) indicates a virtual channel number.

(M) = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15

Access Size : 32 bits

Address : <CRUm_base> + 0500h + (M) x 010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	LER (M)	LSR (M)	FER (M)	FSR (M)	-	-	-	-	-	-	-	OVF (M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	FRD (M)	FRS (M)	-	ECN (M)	ECC (M)	WCE (M)	IDE (M)	CRC (M)	ECD (M)	MLF (M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	LER(M)	0h	R	When this bit is set to 1b, a packet whose data type is "Line End Code" has been received. Write 1b to CSI2nVCSC(M).LERC(M) to clear this bit.
26	LSR(M)	0h	R	When this bit is set to 1b, a packet whose data type is "Line Start Code" has been received. Write 1b to CSI2nVCSC(M).LSRC(M) to clear this bit.
25	FER(M)	0h	R	When this bit is set to 1b, a packet whose data type is "Frame End Code" has been received. Write 1b to CSI2nVCSC(M).FERC(M) to clear this bit.
24	FSR(M)	0h	R	When this bit is set to 1b, a packet whose data type is "Frame Start Code" has been received. Write 1b to CSI2nVCSC(M).FSRC(M) to clear this bit.
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	OVF(M)	0h	R	When this bit is set to 1b, the generic short packets on virtual channel M have been discarded due to an overflow of the generic short packet FIFO. Write 1b to CSI2nVCSC(M).OVFC(M) to clear this bit.
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9	FRD(M)	0h	R	When this bit is set to 1b, ErrFrameData has been detected. This bit is set to 1 if any of the following errors occurs during the period between FS and FE: • ErrCRC • ErrWC • ECC 2-bit error (when CSI2nMCT0.EDMD = 1b) • Reception of a packet of less than four bytes (when CSI2nMCT0.EDMD = 1b) Note that this bit is set to 1 if virtual channel = 0 between FS and FE because the virtual channel for the packet with an ECC 2-bit error or the virtual channel which has received a packet of less than four bytes cannot be determined. Write 1b to CSI2nVCSC(M).FRDC(M) to clear this bit.
8	FRS(M)	0h	R	When this bit is set to 1b, ErrFrameSync has been detected. This bit is set to 1b if any of the following conditions occurs: • After an FS was received, another FS has been received on the same virtual channel without receiving an FE. • Although an FE was received, an FS has not been received on the same virtual channel. Write 1b to CSI2nVCSC(M).FRSC(M) to clear this bit.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6	ECN(M)	0h	R	When this bit is set to 1b, no error was found in ECC checking on the received packets. Write 1b to CSI2nVCSC(M).ECNC(M) to clear this bit.
5	ECC(M)	0h	R	When this bit is set to 1b, a 1-bit error was found in ECC checking on the received packets and the error was corrected. Write 1b to CSI2nVCSC(M).ECCC(M) to clear this bit.

Bit	Bit Name	Initial Value	R/W	Description
4	WCE(M)	0h	R	When this bit is set to 1b, the data payload length of the packet being shorter than the value indicated by WC has been detected. Write 1b to CSI2nVCSC(M).WCEC(M) to clear this bit.
3	IDE(M)	0h	R	When this bit is set to 1b, an ErrID was detected and the packet was discarded. This bit is set to 1 if any of the following conditions occurs: • A packet which is defined as "Reserved" in "Chapter 9 Low Level Protocol" of the CSI-2 specification has been received. - A packet whose data type is 04h to 07h has been received. - A packet whose data type is 38h to 3Fh has been received when CSI2nMCT0.RVMD = 0b. • A generic short packet (data type = 08h to 0Fh) for which the corresponding bit in CSI2nDTEL.DTEN is 0 has been received. • A long packet (data type = 10h to 3Fh) for which the corresponding bit in CSI2nDTEL(H).DTEN is 0 has been received. Write 1 to CSI2nVCSC(M).IDEC(M) to clear this bit. Note that this bit may also be set to 1b if more spacers than the count specified in CSI2nEPCT.SSP/SLP are received when EPD option 2 is in use.
2	CRC(M)	0h	R	When this bit is set to 1b, a CRC error was detected in the received packet. Write 1b to CSI2nVCSC(M).CRCC(M) to clear this bit.
1	ECD(M)	0h	R	When this bit is set to 1b, a 2-bit error was found in ECC checking on the received packets. Because the virtual channel for the packet with an error cannot be determined, the ECD0 field is set to 1b for all virtual channels. To clear the ECD0 field for all virtual channels, write 1b to CSI2nVCSC(M).AECDC(M) for any virtual channel. To clear the ECD0 field of virtual channel M, write 1b to CSI2nVCSC(M).ECDC(M).
0	MLF(M)	0h	R	When this bit is set to 1b, a packet of less than 4 bytes has been received. Because the virtual channel for the packet with an error cannot be determined, the MLF0 field is set to 1b for all virtual channels. To clear the MLF0 field for all virtual channels, write 1b to CSI2nVCSC(M).AMLFC(M) for any virtual channel. To clear the MLF0 field of virtual channel 0, write 1b to CSI2nVCSC(M).MLFC(M).

(18) Virtual Channel (M) Status Clear Register (CRUm_CSI2nVCSC(M)) (m = 0, 1)

(M) indicates a virtual channel number.

(M) = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15

Access Size : 32 bits

Address : <CRUm_base> + 0504h + (M) x 010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	LERC (M)	LSRC (M)	FERC (M)	FSRC (M)	-	-	-	-	-	-	-	OVFC (M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R0W1	R0W1	R0W1	R0W1	R	R	R	R	R	R	R	R0W1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AECDC (M)	AMLFC (M)	-	-	-	-	FRDC (M)	FRSC (M)	-	ECNC (M)	ECCC (M)	WCEC (M)	IDEC (M)	CRCC (M)	ECDC (M)	MLFC (M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0W1	R0W1	R	R	R	R	R0W1	R0W1	R	R0W1	R0W1	R0W1	R0W1	R0W1	R0W1	R0W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	LERC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).LER(M). Writing 0b to this bit has no effect.
26	LSRC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).LSR(M). Writing 0b to this bit has no effect.
25	FERC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).FER(M). Writing 0b to this bit has no effect.
24	FSRC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).FSR(M). Writing 0b to this bit has no effect.
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	OVFC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).OVF(M). Writing 0b to this bit has no effect.
15	AECDC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).ECD(M) for all virtual channels. Writing 0b to this bit has no effect.
14	AMLFC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).MLF(M) for all virtual channels. Writing 0b to this bit has no effect.
13 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9	FRDC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).FRD(M). Writing 0b to this bit has no effect.
8	FRSC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).FRS(M). Writing 0b to this bit has no effect.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6	ECNC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).ECN(M). Writing 0b to this bit has no effect.
5	ECCC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).ECC(M). Writing 0b to this bit has no effect.

Bit	Bit Name	Initial Value	R/W	Description
4	WCEC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear the CSI2nVCST(M).WCE(M). Writing 0b to this bit has no effect.
3	IDEC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).IDEC(M). Writing 0b to this bit has no effect.
2	CRCC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).CRC(M). Writing 0b to this bit has no effect.
1	ECDC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).ECD(M). Writing 0b to this bit has no effect.
0	MLFC(M)	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nVCST(M).MLF(M). Writing 0b to this bit has no effect.

(19) Virtual Channel (M) Interrupt Enable Register (CRUm_CSI2nVCIE(M)) (m = 0, 1)

(M) indicates a virtual channel number.

(M) = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15

Access Size : 32 bits

Address : <CRUm_base> + 0508h + (M) x 010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	LERE(M)	LSRE(M)	FERE(M)	FSRE(M)	-	-	-	-	-	-	-	OVFE(M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	FRDE(M)	FRSE(M)	-	ECNE(M)	ECCE(M)	WCEE(M)	IDEE(M)	CRCE(M)	ECDE(M)	MLFE(M)
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27	LERE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).LER(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).LER(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).LER(M) = 1b.
26	LSRE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).LSR(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).LSR(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).LSR(M) = 1b.
25	FERE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).FER(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).FER(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).FER(M) = 1b.
24	FSRE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).FSR(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).FSR(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).FSR(M) = 1b.
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	OVFE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).OVF(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).OVF(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).OVF(M) = 1b.
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9	FRDE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).FRD(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).FRD(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).FRD(M) = 1b.
8	FRSE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).FRS(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).FRS(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).FRS(M) = 1b.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6	ECNE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).ECN(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).ECN(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).ECN(M) = 1b.
5	ECCE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).ECC(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).ECC(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).ECC(M) = 1b.
4	WCEE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).WCE(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).WCE(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).WCE(M) = 1b.
3	IDEE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).IDE(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).IDE(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).IDE(M) = 1b.

Bit	Bit Name	Initial Value	R/W	Description
2	CRCE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).CRC(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).CRC(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).CRC(M) = 1b.
1	ECDE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).ECD(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).ECD(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).ECD(M) = 1b.
0	MLFE(M)	0h	RW	Enables interrupts generated by CSI2nVCST(M).MLF(M). 0b: Does not assert csi2_int_vc when CSI2nVCST(M).MLF(M) = 1b. 1b: Asserts csi2_int_vc when CSI2nVCST(M).MLF(M) = 1b.

(20) Power Management Status Register (CRUm_CSI2nPMST) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0600h

Initial Value : 000F_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	DLUL[3:0]				-	-	-	-	DLSS[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLUL	CLSS	-	-	-	-	-	-	CUN	CUX	DUN	DUX	CSN	CSX	DSN	DSX
Initial Value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 24	DLUL[3:0]	0h	R	Indicate the RxUlpsEsc status of a logical data lane. Bit 0: RxUlpsEsc of logical lane 0 Bit 1: RxUlpsEsc of logical lane 1 Bit 2: RxUlpsEsc of logical lane 2 Bit 3: RxUlpsEsc of logical lane 3
23 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 16	DLSS[3:0]	Fh	R	Indicate the stop state of a logical data lane. Bit 0: Stop state of logical lane 0 Bit 1: Stop state of logical lane 1 Bit 2: Stop state of logical lane 2 Bit 3: Stop state of logical lane 3
15	CLUL	0h	R	Indicates the inverted state of RxUlpsClkNot of the PPI clock lane. 0b: Not in ULPS 1b: In ULPS
14	CLSS	1h	R	Indicates the stop state of the PPI clock lane. 0b: Not in the stop state 1b: In the stop state
13 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7	CUN	0h	R	When this bit is set to 1b, the clock lane has transitioned to ULPS mode (inverse of RxUlpsClkNot). Write 1b to CSI2nPMSC.CUNC to clear this bit.
6	CUX	0h	R	When this bit is set to 1b, the clock lane has transitioned from ULPS mode (inverse of RxUlpsClkNot) to non-ULPS mode. Write 1b to CSI2nPMSC.CUXC to clear this bit.
5	DUN	0h	R	When this bit is set to 1b, all valid data lanes have transitioned to ULPS mode. Write 1b to CSI2nPMSC.DUNC to clear this bit.
4	DUX	0h	R	When this bit is set to 1b, all valid data lanes have transitioned from ULPS mode to any other state. Write 1b to CSI2nPMSC.DUXC to clear this bit.
3	CSN	0h	R	When this bit is set to 1b, the clock lane has transitioned to the stop state. Write 1b to CSI2nPMSC.CSNC to clear this bit.
2	CSX	0h	R	When this bit is set to 1b, the clock lane has transitioned from the stop state to a non-stop state. Write 1b to CSI2nPMSC.CSXC to clear this bit.
1	DSN	0h	R	When this bit is set to 1b, all valid data lanes have transitioned to the stop state. Write 1b to CSI2nPMSC.DSNC to clear this bit.
0	DSX	0h	R	When this bit is set to 1b, all valid data lanes have transitioned from the stop state to any other state. Write 1b to CSI2nPMSC.DSXC to clear this bit.

(21) Power Management Status Clear Register (CRUm_CSI2nPMSC) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0604h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CUNC	CUXC	DUNC	DUXC	CSNC	CSXC	DSNC	DSXC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R0W1	R0W1	R0W1	R0W1	R0W1	R0W1	R0W1	R0W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7	CUNC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.CUN. Writing 0b to this bit has no effect.
6	CUXC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.CUX. Writing 0b to this bit has no effect.
5	DUNC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.DUN. Writing 0b to this bit has no effect.
4	DUXC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.DUX. Writing 0b to this bit has no effect.
3	CSNC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.CSN. Writing 0b to this bit has no effect.
2	CSXC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.CSX. Writing 0b to this bit has no effect.
1	DSNC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.DSN. Writing 0b to this bit has no effect.
0	DSXC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nPMST.DSX. Writing 0b to this bit has no effect.

(22) Power Management Interrupt Enable Register (CRUm_CSI2nPMIE) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0608h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CUNE	CUXE	DUNE	DUXE	CSNE	CSXE	DSNE	DSXE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7	CUNE	0h	RW	Enables interrupts generated by CSI2nPMST.CUN. 0b: Does not assert csi2_int_pm when CSI2nPMST.CUN = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.CUN = 1b.
6	CUXE	0h	RW	Enables interrupts generated by CSI2nPMST.CUX. 0b: Does not assert csi2_int_pm when CSI2nPMST.CUX = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.CUX = 1b.
5	DUNE	0h	RW	Enables interrupts generated by CSI2nPMST.DUN. 0b: Does not assert csi2_int_pm when CSI2nPMST.DUN = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.DUN = 1b.
4	DUXE	0h	RW	Enables interrupts generated by CSI2nPMST.DUX. 0b: Does not assert csi2_int_pm when CSI2nPMST.DUX = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.DUX = 1b.
3	CSNE	0h	RW	Enables interrupts generated by CSI2nPMST.CSN. 0b: Does not assert csi2_int_pm when CSI2nPMST.CSN = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.CSN = 1b.
2	CSXE	0h	RW	Enables interrupts generated by CSI2nPMST.CSX. 0b: Does not assert csi2_int_pm when CSI2nPMST.CSX = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.CSX = 1b.
1	DSNE	0h	RW	Enables interrupts generated by CSI2nPMST.DSN. 0b: Does not assert csi2_int_pm when CSI2nPMST.DSN = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.DSN = 1b.
0	DSXE	0h	RW	Enables interrupts generated by CSI2nPMST.DSX. 0b: Does not assert csi2_int_pm when CSI2nPMST.DSX = 1b. 1b: Asserts csi2_int_pm when CSI2nPMST.DSX = 1b.

(23) Generic Short Packet Control Register (CRUm_CSI2nGSCT) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0680h

Initial Value : 0001_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	GFIF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	SHTH[6:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	-	0h	R0W	The read value is always 0b. Do not change the value of this bit from 0h.
16	GFIF	1h	RW	When this bit is set to 1b, the received generic short packet is stored in the generic short packet FIFO (GS FIFO).
15 to 7	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 0	SHTH[6:0]	0h	RW	Asserts CSI2nGSST.GTH while the number of packets stored in the generic short packet FIFO equals or exceeds the value of this field + 1. Set this field to 15 or less when using CSI2nGSST.GTH. Set the default value when not using CSI2nGSST.GTH.

(24) Generic Short Packet Status Register (CRUm_CSI2nGSST) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0684h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STRDS	GCD
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNUM[7:0]								-	-	-	GOV	-	-	GTH	GNE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	STRDS	0h	R	Indicates whether the received generic short packets can be stored in the generic short packet FIFO. 0b: Can be stored in the generic short packet FIFO. 1b: Cannot be stored in the generic short packet FIFO. This field indicates an inverted value of the store_en signal described in 9.2.3.3.1 Data Reception with a delay for an internal latency. The previous value may be read depending on the signal change timing.
16	GCD	0h	R	When this bit is set to 1b, the GS FIFO is being cleared by CSI2nGSCT.GFCLR.
15 to 8	PNUM[7:0]	0h	R	Indicate the number of packets stored in the generic short packet FIFO.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	GOV	0h	R	When this bit is set to 1b, the generic short packet FIFO has overflowed. Write 1b to CSI2nGSCT.GOV to clear this bit.
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	GTH	0h	R	When this bit is set to 1b, the number of packets stored in the generic short packet FIFO equals or exceeds "CSI2nGSCT.SHTH + 1".
0	GNE	0h	R	When this bit is set to 1b, the generic short packet FIFO is not empty.

(25) Generic Short Packet Status Clear Register (CRUm_CSI2nGSSC) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0688h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	GOVC	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R0W1	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	GOVC	0h	R0W1	The read value is always 0b. Write 1b to this bit to clear CSI2nGSST.GOV. Writing 0b to this bit has no effect.
3 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(26) Generic Short Packet Interrupt Enable Register (CRUm_CSI2nGSIE) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 068Ch

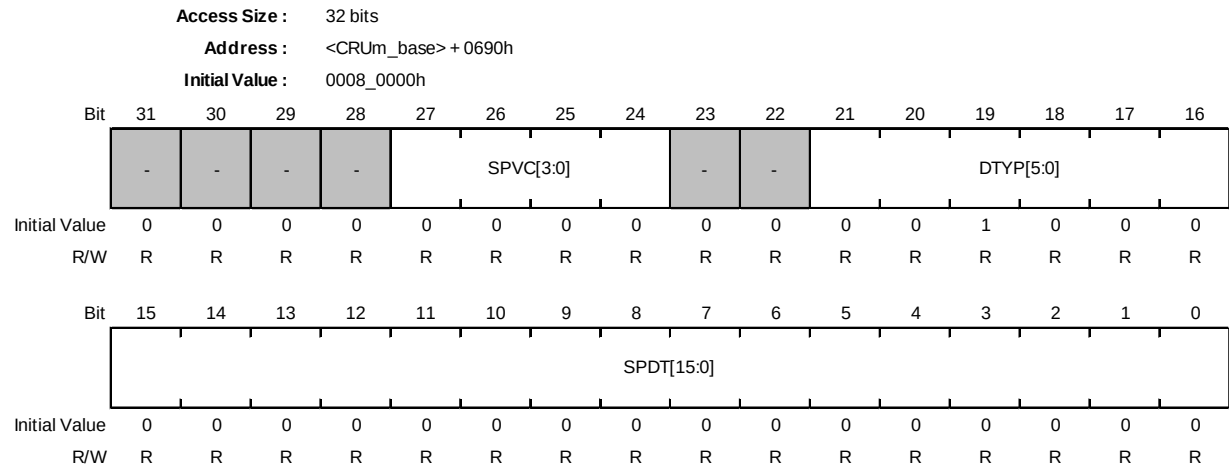
Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	GOVE	-	-	GTHE	GNEE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	GOVE	0h	RW	Enables interrupts generated by CSI2nGSST.GOV. 0b: Does not assert csi2_int_gst when CSI2nGSST.GOV = 1b. 1b: Asserts csi2_int_gst when CSI2nGSST.GOV = 1b.
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	GTHE	0h	RW	Enables interrupts generated by CSI2nGSST.GTH. 0b: Does not assert csi2_int_gst when CSI2nGSST.GTH = 1b. 1b: Asserts csi2_int_gst when CSI2nGSST.GTH = 1b.
0	GNEE	0h	RW	Enables interrupts generated by CSI2nGSST.GNE. 0b: Does not assert csi2_int_gst when CSI2nGSST.GNE = 1b. 1b: Asserts csi2_int_gst when CSI2nGSST.GNE = 1b.

(27) Generic Short Packet Register (CRUm_CSI2nGSHT) (m = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 24	SPVC[3:0]	0h	R	Indicate the virtual channel.
23, 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21 to 16	DTYP[5:0]	8h	R	Indicate the data type. 08h-0Fh
15 to 0	SPDT[15:0]	0h	R	Indicate 16-bit user-defined data.

(28) Generic Short Packet Information Update Register (CRUm_CSI2nGSIU) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0694h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	GFEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	GFCLR	-	-	-	-	-	-	-	FINC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R	R0W1

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	GFEN	0h	R0W1	The read value is always 0b. When the GS FIFO has overflowed, write 1b to this bit to enable storing of generic short packets in the GS FIFO which has been disabled.
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	GFCLR	0h	RW	Clears the GS FIFO. Writing 1b to this bit requests clearing of the FIFO. Writing 0b to this bit requests clearing of the FIFO to be released.
7 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	FINC	0h	R0W1	The read value is always 0b. Write 1b to this bit to update data of the packet indicated in the CSI2nGSHT register for the next packet. Writing 0b to this bit has no effect. Writing 1b to this bit while CSI2nGSST.PNUM = 0b is prohibited.

(29) Slave Timing Control Register (CRUm_S_TIMCTL) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 081Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S_HSSETTLECTL[7:0]								-	-	-	-	-	-	S_CLKSETTLECTL[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 8	S_HSSETTLECTL[7:0]	0h	RW	Slave Data Lane Control Register for THS-SETTLE. Set the values according to Table 9.2-7 .
7 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	S_CLKSETTLECTL[1:0]	0h	RW	Do not change the value of this bit from 0h.

Table 9.2-7 S_TIMCTL.S_HSETTLECTL[7:0] Register Settings

Transmission Rate TR (Mbps)	S_HSETTLECTL[7:0]	Transmission Rate TR (Mbps)	S_HSETTLECTL[7:0]
2070 < TR ≤ 2100	2Eh	1030 < TR ≤ 1080	17h
2030 < TR ≤ 2070	2Dh	990 < TR ≤ 1030	16h
1980 < TR ≤ 2030	2Ch	940 < TR ≤ 990	15h
1940 < TR ≤ 1980	2Bh	900 < TR ≤ 940	14h
1890 < TR ≤ 1940	2Ah	850 < TR ≤ 900	13h
1850 < TR ≤ 1890	29h	810 < TR ≤ 850	12h
1800 < TR ≤ 1850	28h	760 < TR ≤ 810	11h
1760 < TR ≤ 1800	27h	720 < TR ≤ 760	10h
1710 < TR ≤ 1760	26h	670 < TR ≤ 720	0Fh
1670 < TR ≤ 1710	25h	630 < TR ≤ 670	0Eh
1620 < TR ≤ 1670	24h	580 < TR ≤ 630	0Dh
1580 < TR ≤ 1620	23h	540 < TR ≤ 580	0Ch
1530 < TR ≤ 1580	22h	490 < TR ≤ 540	0Bh
1490 < TR ≤ 1530	21h	450 < TR ≤ 490	0Ah
1440 < TR ≤ 1490	20h	400 < TR ≤ 450	09h
1400 < TR ≤ 1440	1Fh	360 < TR ≤ 400	08h
1350 < TR ≤ 1400	1Eh	310 < TR ≤ 360	07h
1310 < TR ≤ 1350	1Dh	270 < TR ≤ 310	06h
1260 < TR ≤ 1310	1Ch	220 < TR ≤ 270	05h
1220 < TR ≤ 1260	1Bh	180 < TR ≤ 220	04h
1170 < TR ≤ 1220	1Ah	130 < TR ≤ 180	03h
1120 < TR ≤ 1170	19h	90 < TR ≤ 130	02h
1080 < TR ≤ 1120	18h	80 ≤ TR ≤ 90	01h

(30) Slave D-PHY Control Register (MSB) (CRUm_S_DPHYCTL_MSB) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0834h														
Initial Value :		0000_0090h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S_DPHYCTL[63:48]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S_DPHYCTL[47:32]																
Initial Value	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name	Initial Value	R/W	Description												
31 to 2	S_DPHYCTL [63:34]	24h	RW	Changing this bit from 24h to a different value is prohibited.												
1	S_DPHYCTL [33]	All 0	RW	DeSkew Calibration Function Enable/Disable Control • 0b: DeSkew Calibration Disable • 1b: DeSkew Calibration Enable												
0	S_DPHYCTL [32]	All 0	RW	Changing this bit from 0h to a different value is prohibited.												

(31) DP/DN Swap Control Register (CRUm_SWAPCTL) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0838h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	S_DPD N_SW AP_DA T	S_DPD N_SW AP_CL K	-	-	M_DP DN_S WAP_ DAT	M_DP DN_S WAP_ CLK
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5	S_DPDN_SWA P_DAT	0h	RW	Slave PPI signal This signal is for crossing PCB trace lines of DP and DN when it connects the Slave lane to a Master lane on the PCB board. This signal swaps all the DP and DN signals of Slave when set to HIGH. Tie this signal to the ground when the connections are correct.
4	S_DPDN_SWA P_CLK	0h	RW	Slave PPI signal This signal is for crossing PCB trace lines of DP and DN when it connects the Slave lane to a Master lane on the PCB board. This signal swaps all the DP and DN signals of Slave when set to HIGH. Tie this signal to the ground when the connections are correct.
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	M_DPDN_SWA P_DAT	0h	RW	Do not change the value of this bit from 0h.
0	M_DPDN_SWA P_CLK	0h	RW	Do not change the value of this bit from 0h.

(32) RX Skew Calibration Results n (CRUm_S_ATB_OUT_n) (m = 0, 1, n = 0 to 3)

Access Size : 32 bits																
Address : <CRUm_base> + 086Ch + n x 0004h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S_ATB_OUT_n	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	S_ATB_OUT_n	0h	R	The initial/periodic deskew calibration status for lane n (for debugging) 0b: Calibration in progress or calibration failed 1b: Calibration completed
0	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.2.3.3 Operation (Short Packet Reception)

The prefix “CRUm” of the register names is omitted in the following subsections.

9.2.3.3.1 Data Reception

(1) Common between short packets and long packets

Figure 9.2-5 shows the flow when receiving a packet whose data type is A and virtual channel is M (0, 1, 2, to 15).

This flow is common between short packets and long packets.

Upon receiving a packet, the LINK checks the length of the received packet.

If the length of the received packet is less than the packet header length (4 bytes), a virtual channel cannot be determined, resulting in setting CSI2nVCST(M).MLF(M) to 1b for all virtual channels. In addition, this packet is discarded because it is invalid.

When the length of the received packet is 4 bytes or more, ECC checking is performed. The check results are indicated in the CSI2nVCST(M) register.

- 2-bit error: Because the virtual channel of the packet with an error cannot be determined, CSI2nVCST(M).ECD(M) is set to 1b for all virtual channels.
In addition, the packet is discarded because error correction cannot be performed.
- 1-bit error: CSI2nVCST(M).ECC(M) is set to 1b for the relevant virtual channel.
Error correction proceeds because it is possible.
- No error: CSI2nVCST(M).ECN(M) is set to 1b for the relevant virtual channel.

If the ECC check results are “1-bit error” or “no error”, the data type field of the packet header is analyzed.

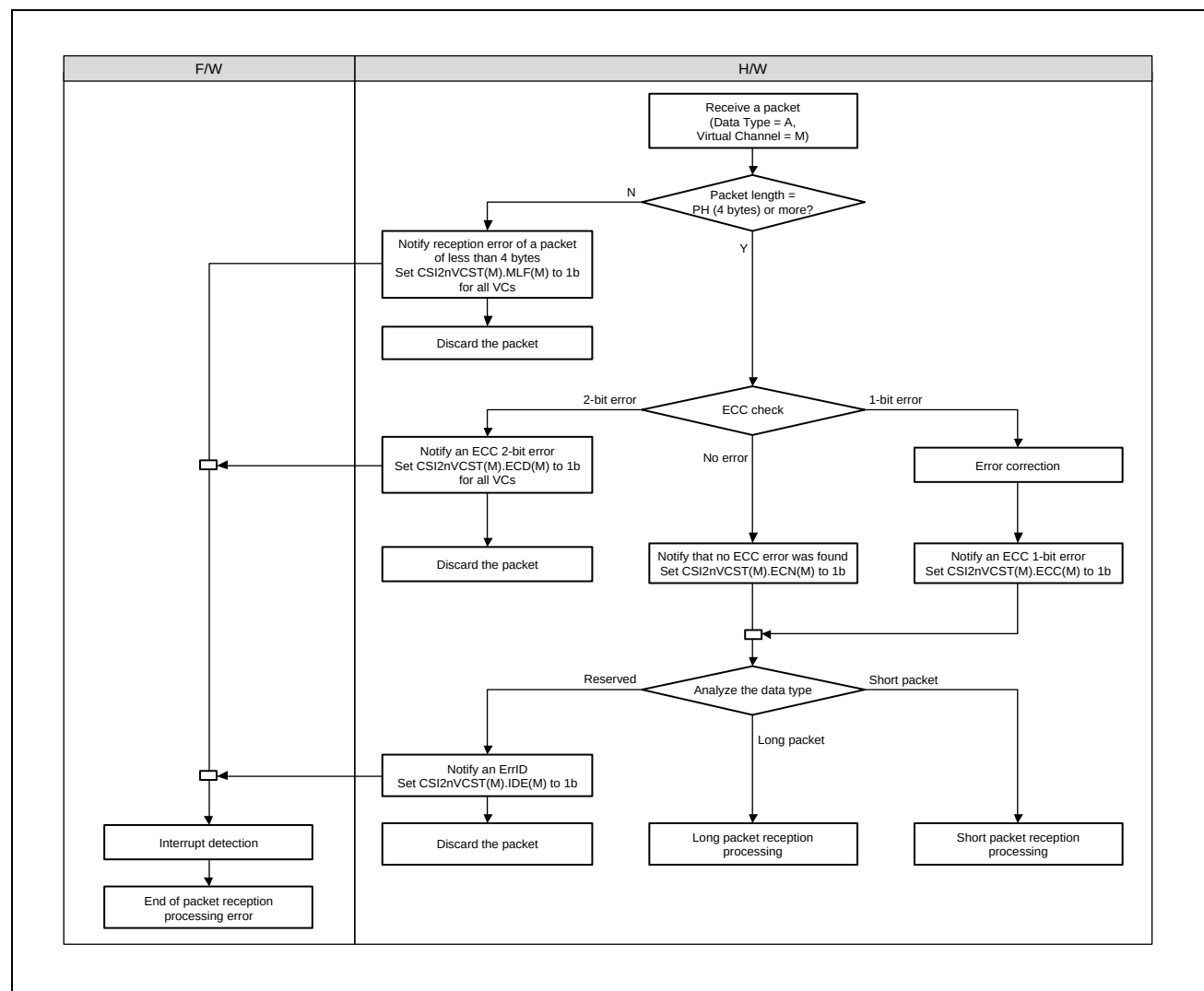


Figure 9.2-5 Packet Reception Processing

(2) Generic short packet data type code

Figure 9.2-6 shows the flow when receiving a packet whose data type is generic short packet code 1 to 8 and virtual channel is M.

When the corresponding bit in CSI2nDTEL.DTEN is set to 0b, CSI2nVCST(M).IDE(M)) is asserted and the packet is discarded.

When the corresponding bit in CSI2nDTEL.DTEN is set to 1b and CSI2nGSCT.GFIF is set to 1b, the generic short packet is output to the generic short packet FIFO. For details on the operation when using the generic short packet FIFO, see the flow in **Figure 9.2-6**.

The initial value of internal signal store_en which is described in the flow is 1. When store_en is set to 1, packets can be stored in the generic short packet FIFO. When it is set to 0, packets cannot be stored in the generic short packet FIFO.

- store_en is set to 0 when the generic short packet FIFO has overflowed.
- store_en is set to 1 when 1b is written to CSI2nGSIU.GFEN.

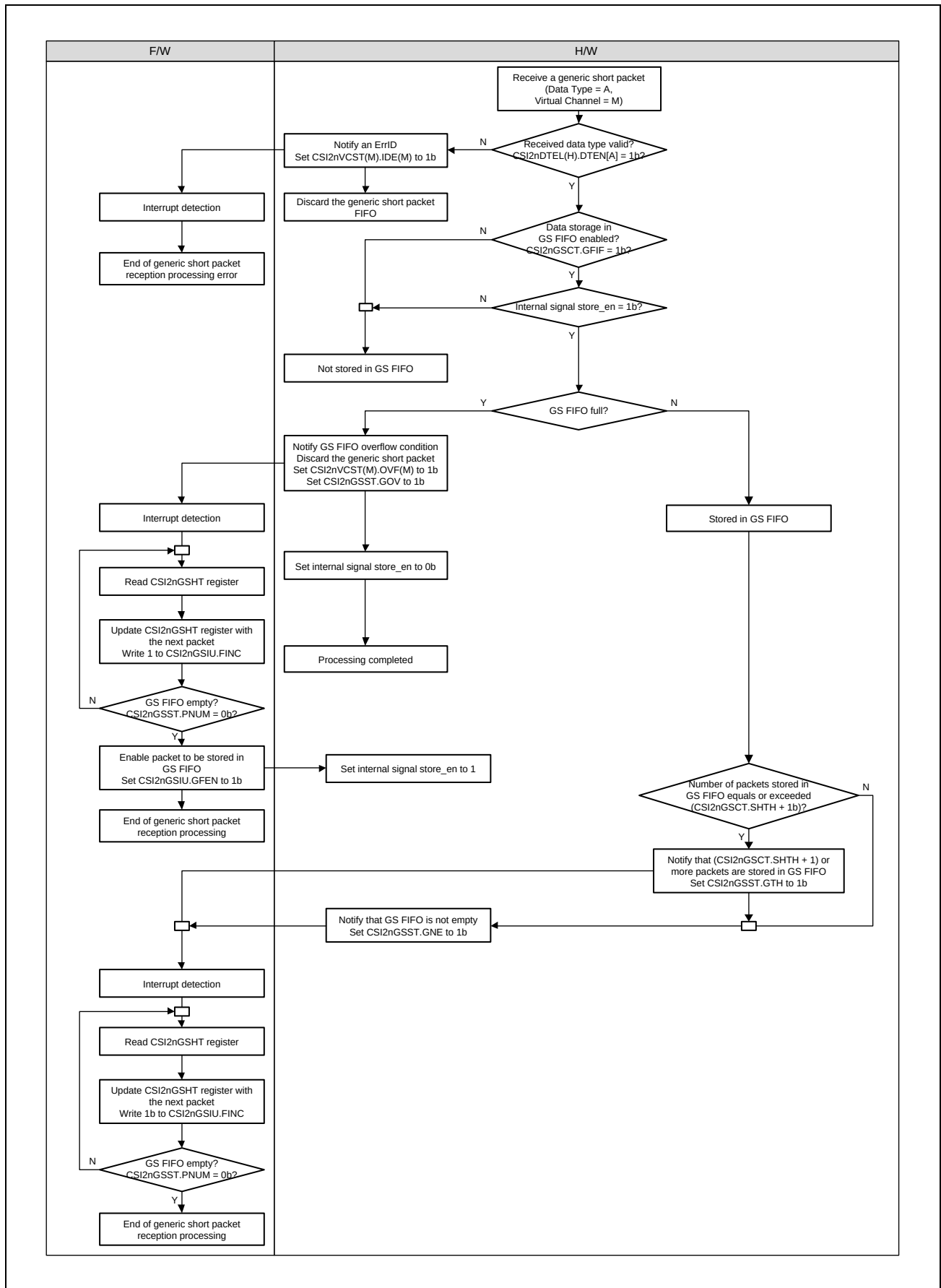


Figure 9.2-6 Generic Short Packet Reception Processing

(3) Clearing the generic short packet FIFO

Figure 9.2-7 shows how the generic short packet FIFO is cleared.

Writing 1b to CSI2nGSIU.GFCLR starts clearing the generic short packet FIFO.

After requesting to clear the generic short packet FIFO, check if CSI2nGSST.GCD is set to 1b to make sure that the FIFO is cleared. Then, writing 0b to CSI2nGSIU.GFCLR cancels the request for clearing the generic short packet FIFO. Check if CSI2nGSST.GCD is set to 0b to make sure that the request for clearing the generic short packet FIFO is canceled.

When CSI2nMCG.GSNM is set to 0b, it is prohibited to clear the generic short packet FIFO.

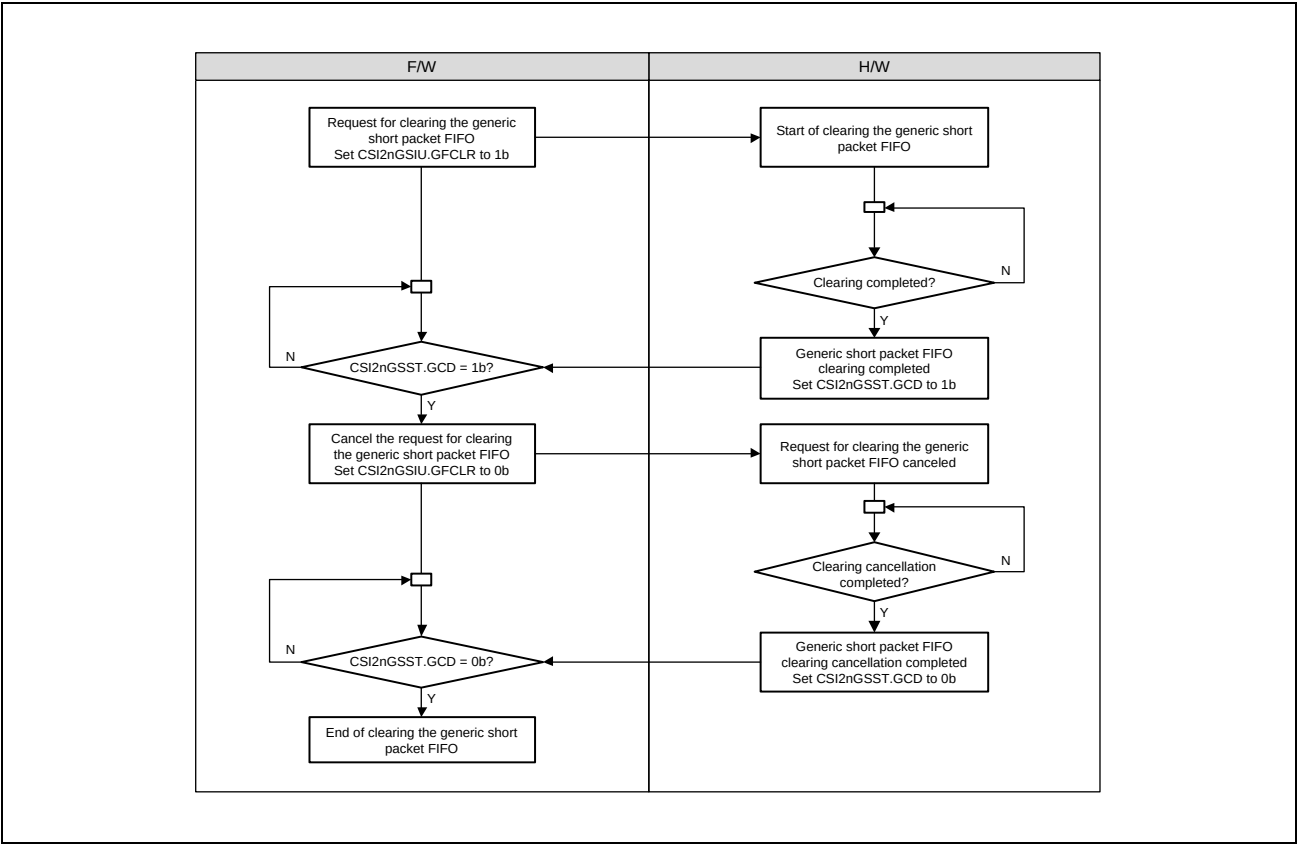


Figure 9.2-7 Clearing the Generic Short Packet FIFO

9.2.3.4 Interrupts

Figure 9.2-8 shows the structure of interrupt signals of MIPI CSI2. CRU0 and 1 each have an interrupt. The conditions for interrupt generation, interrupt source registers, and interrupt enable registers are listed in **Table 9.2-8**, **Table 9.2-9**, and **Table 9.2-10**. The CSI2nMIST register indicates the interrupt output state. When an interrupt is generated, read the CSI2nMIST register to identify the interrupt source.

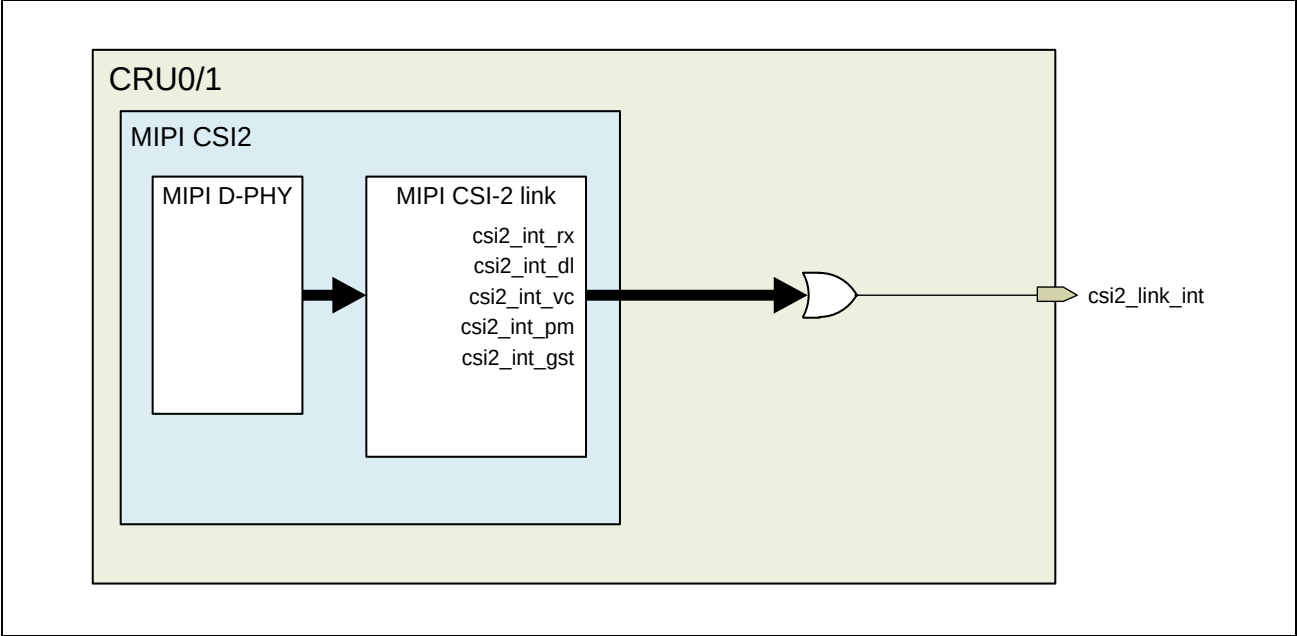


Figure 9.2-8 Interrupt Structure

Table 9.2-8 Interrupt States and Generation Conditions (1)

Interrupt State	Interrupt Source Register Field	Interrupt Enable Register Field	Generation Condition
CSI2nMIST.RXS	CSI2nRXST.RACTDET	CSI2nRXIE.RACTDETE	This IP received a packet.
CSI2nMIST.DLS0	CSI2nDLST0.ESH0	CSI2nDLIE0.ESHE0	ErrSotHS occurred on DL0.
	CSI2nDLST0.ECT0	CSI2nDLIE0.ECTE0	ErrControl occurred on DL0.
	CSI2nDLST0.EES0	CSI2nDLIE0.EESE0	ErrESC occurred on DL0.
	CSI2nDLST0.EUL0	CSI2nDLIE0.EULE0	DL0 exited from the ULPS.
	CSI2nDLST0.RUL0	CSI2nDLIE0.RULE0	DL0 transitioned to the ULPS.
CSI2nMIST.DLS1	CSI2nDLST1.ESH1	CSI2nDLIE1.ESHE1	ErrSotHS occurred on DL1.
	CSI2nDLST1.ECT1	CSI2nDLIE1.ECTE1	ErrControl occurred on DL1.
	CSI2nDLST1.EES1	CSI2nDLIE1.EESE1	ErrESC occurred on DL1.
	CSI2nDLST1.EUL1	CSI2nDLIE1.EULE1	DL1 exited from the ULPS.
	CSI2nDLST1.RUL1	CSI2nDLIE1.RULE1	DL1 transitioned to the ULPS.
CSI2nMIST.DLS2	CSI2nDLST2.ESH2	CSI2nDLIE2.ESHE2	ErrSotHS occurred on DL2.
	CSI2nDLST2.ECT2	CSI2nDLIE2.ECTE2	ErrControl occurred on DL2.
	CSI2nDLST2.EES2	CSI2nDLIE2.EESE2	ErrESC occurred on DL2.
	CSI2nDLST2.EUL2	CSI2nDLIE2.EULE2	DL2 exited from the ULPS.
	CSI2nDLST2.RUL2	CSI2nDLIE2.RULE2	DL2 transitioned to the ULPS.
CSI2nMIST.DLS3	CSI2nDLST3.ESH3	CSI2nDLIE3.ESHE3	ErrSotHS occurred on DL3.
	CSI2nDLST3.ECT3	CSI2nDLIE3.ECTE3	ErrControl occurred on DL3.
	CSI2nDLST3.EES3	CSI2nDLIE3.EESE3	ErrESC occurred on DL3.
	CSI2nDLST3.EUL3	CSI2nDLIE3.EULE3	DL3 exited from the ULPS.
	CSI2nDLST3.RUL3	CSI2nDLIE3.RULE3	DL3 transitioned to the ULPS.

Table 9.2-9 Interrupt States and Generation Conditions (2)

Interrupt State	Interrupt Source Register Field	Interrupt Enable Register Field	Generation Condition
CSI2nMIST.VCS0	CSI2nVCST0.MLF0	CSI2nVCIE0.MLFE0	Malformed packet was received on VC0.
	CSI2nVCST0.ECD0	CSI2nVCIE0.ECDE0	ErrEccDouble occurred on VC0.
	CSI2nVCST0.CRC0	CSI2nVCIE0.CRCE0	ErrCrc occurred on VC0.
	CSI2nVCST0.IDE0	CSI2nVCIE0.IDEE0	ErrID occurred on VC0.
	CSI2nVCST0.WCE0	CSI2nVCIE0.WCEE0	ErrWC occurred on VC0.
	CSI2nVCST0.ECC0	CSI2nVCIE0.ECCE0	ErrEccCorrected occurred on VC0.
	CSI2nVCST0.ECN0	CSI2nVCIE0.ECNE0	ErrEccNoError occurred on VC0.
	CSI2nVCST0.FRS0	CSI2nVCIE0.FRSE0	ErrFrameSync occurred on VC0.
	CSI2nVCST0.FRD0	CSI2nVCIE0.FRDE0	ErrFrameData occurred on VC0.
	CSI2nVCST0.OVF0	CSI2nVCIE0.OVFE0	Generic short packets on VC0 were discarded because GS FIFO overflowed.
	CSI2nVCST0.FSR0	CSI2nVCIE0.FSRE0	Frame start was received on VC0.
	CSI2nVCST0.FER0	CSI2nVCIE0.FERE0	Frame end was received on VC0.
	CSI2nVCST0.LSR0	CSI2nVCIE0.LSRE0	Line start was received on VC0.
	CSI2nVCST0.LER0	CSI2nVCIE0.LERE0	Line end was received on VC0.
CSI2nMIST.VCS1	CSI2nVCST1.MLF1	CSI2nVCIE1.MLFE1	Malformed packet was received on VC1.
	CSI2nVCST1.ECD1	CSI2nVCIE1.ECDE1	ErrEccDouble occurred on VC1.
	CSI2nVCST1.CRC1	CSI2nVCIE1.CRCE1	ErrCrc occurred on VC1.
	CSI2nVCST1.IDE1	CSI2nVCIE1.IDEE1	ErrID occurred on VC1.
	CSI2nVCST1.WCE1	CSI2nVCIE1.WCEE1	ErrWC occurred on VC1.
	CSI2nVCST1.ECC1	CSI2nVCIE1.ECCE1	ErrEccCorrected occurred on VC1.
	CSI2nVCST1.ECN1	CSI2nVCIE1.ECNE1	ErrEccNoError occurred on VC1.
	CSI2nVCST1.FRS1	CSI2nVCIE1.FRSE1	ErrFrameSync occurred on VC1.
	CSI2nVCST1.FRD1	CSI2nVCIE1.FRDE1	ErrFrameData occurred on VC1.
	CSI2nVCST1.OVF1	CSI2nVCIE1.OVFE1	Generic short packets on VC1 were discarded because GS FIFO overflowed.
	CSI2nVCST1.FSR1	CSI2nVCIE1.FSRE1	Frame start was received on VC1.
	CSI2nVCST1.FER1	CSI2nVCIE1.FERE1	Frame end was received on VC1.
	CSI2nVCST1.LSR1	CSI2nVCIE1.LSRE1	Line start was received on VC1.
	CSI2nVCST1.LER1	CSI2nVCIE1.LERE1	Line end was received on VC1.
...	...	...	...

Table 9.2-10 Interrupt States and Generation Conditions (3)

Interrupt State	Interrupt Source Register Field	Interrupt Enable Register Field	Generation Condition
CSI2nMIST.VCS15	CSI2nVCST15.MLF15	CSI2nVCIE15.MLFE15	Malformed packet was received on VC15.
	CSI2nVCST15.ECD15	CSI2nVCIE15.ECDE15	ErrEccDouble occurred on VC15.
	CSI2nVCST15.CRC15	CSI2nVCIE15.CRCE15	ErrCrc occurred on VC15.
	CSI2nVCST15.IDE15	CSI2nVCIE15.IDEE15	ErrID occurred on VC15.
	CSI2nVCST15.WCE15	CSI2nVCIE15.WCEE15	ErrWC occurred on VC15.
	CSI2nVCST15.ECC15	CSI2nVCIE15.ECCE15	ErrEccCorrected occurred on VC15.
	CSI2nVCST15.ECN15	CSI2nVCIE15.ECNE15	ErrEccNoError occurred on VC15.
	CSI2nVCST15.FRS15	CSI2nVCIE15.FRSE15	ErrFrameSync occurred on VC15.
	CSI2nVCST15.FRD15	CSI2nVCIE15.FRDE15	ErrFrameData occurred on VC15.
	CSI2nVCST15.OVF15	CSI2nVCIE15.OVFE15	Generic short packets on VC15 were discarded because GS FIFO overflowed.
	CSI2nVCST15.FSR15	CSI2nVCIE15.FSRE15	Frame start was received on VC15.
	CSI2nVCST15.FER15	CSI2nVCIE15.FERE15	Frame end was received on VC15.
	CSI2nVCST15.LSR15	CSI2nVCIE15.LSRE15	Line start was received on VC15.
	CSI2nVCST15.LER15	CSI2nVCIE15.LERE15	Line end was received on VC15.
CSI2nMIST.PMS	CSI2nPMST.DSX	CSI2nPMIE.DSXE	All data lanes exited from the stop state.
	CSI2nPMST.DSN	CSI2nPMIE.DSNE	All data lanes transitioned to the stop state.
	CSI2nPMST.CSX	CSI2nPMIE.CSXE	The clock lane exited from the stop state.
	CSI2nPMST.CSN	CSI2nPMIE.CSNE	The clock lane transitioned to the stop state.
	CSI2nPMST.DUX	CSI2nPMIE.DUXE	All data lanes exited from the ULPS.
	CSI2nPMST.DUN	CSI2nPMIE.DUNE	All data lanes transitioned to the ULPS.
	CSI2nPMST.CUX	CSI2nPMIE.CUXE	The clock lane exited from the ULPS.
	CSI2nPMST.CUN	CSI2nPMIE.CUNE	The clock lane transitioned to the ULPS.
CSI2nMIST.GSTS	CSI2nGSST.GNE	CSI2nGSIE.GNEE	GS FIFO is not empty.
	CSI2nGSST.GTH	CSI2nGSIE.GTHE	Data stored in GS FIFO equals or exceeds the threshold.
	CSI2nGSST.GOV	CSI2nGSIE.GOVE	GS FIFO overflowed.

9.2.4 Image Converter

9.2.4.1 Overview

The image converter module is a data conversion module which handles image clipping, frame subsampling, demosaicing (only selectable for SVC0, and only for RAW input), linear matrix, statistics (only selectable for SVC0, and only for RAW input), color space conversion, LUT conversion, pixel format conversion, and so on. It has an MIPI CSI2 (9.2.3 MIPI CSI2) input as an image sensor interface and its processing is roughly divided into two.

- Video system (Image Process, Non Image Process, All Bypass): Data are initially stored in FIFO-VD and transferred to external memory through the internal AXI-VD bus.
- Statistics system: Data are initially stored in FIFO-SD and transferred to external memory through the internal AXI-SD bus.

The video system further has image processing, non-image processing and full bypass systems. Full bypass is used when images are not to be processed by the image converter. Image processing and non-image processing can be used simultaneously (however, the data type cannot be set simultaneously within a single VC). The image processing system handles image processing and the non-image processing system does not handle image processing. Therefore, the data type for image processing, such as RGB, YUV, or RAW, can be selected for image processing, while several data types which are not for image processing, such as embedded 8-bit non-image data and a user-defined 8-bit data type, can be selected for non-image processing. In full bypass (CRUnCTRL.ATH = 1b), image processing within the CRU is not used and data from the MIPI CSI-2 link (in the recommended memory storage format for MIPI CSI-2) are directly transferred to external memory (image processing and non-image processing data cannot be used with full bypass).

The statistics system only handles data processing to produce statistics.

For each MIPI CSI-2 input, up to four channels (SVC0 to 3) can be selected from among 16 virtual channels (VC0 to 15) and only one data type for image processing is processed per channel and several data types can be selected for non-image processing.

Additionally, when only SVC0 is used, some parameters are adjustable from frame to frame, and when demosaicing processing for multiple virtual channels (the image sizes, etc. are the same) is to proceed, control is applied with switching on frame-by-frame demarking by changing a register setting. However, this function cannot be used for sensors that have different VC for each line.

9.2.4.1.1 Features

The operating range of the image converter is as follows.

Image processing

- Range of effective pixels: QVGA (320 × 240) to 8.8 Mpixels
- Range of effective pixels in the horizontal direction: 320 to 4096 pixels
- Range of effective pixels in the vertical direction: 240 to 4096 lines
- Non-image processing and full bypass:
The size range of application-specific payloads within input packets is from 1 to 24576 bytes.

Common

The maximum transfer rate at the time of MIPI CSI-2 input differs with the input format. For details, see **9.2.3.1.1 Features**.

- Input data format (type): See **9.2.4.3.1 Input Formats and Image Processing**.
 - Image processing:
Only one data type from among YUV, RGB, RAW, etc. is selectable for each SVC.
 - Non-image processing:
Several data types such as embedded 8-bit non-image data and user-defined 8-bit data are selectable for each SVC.
- Video output data format (type): Little endian is only supported.
 - 8-bit multiplexing of YCbCr422/YCbCr420 (with support for switching the order of Y and U/V)
 - YC separation of YCbCr422 and YCbCr420 (separation into Y and CbCr components)
 - Extraction of the Y component of YCbCr422
 - U and V in YUV422/YCbCr420 outputs are expressed in offset binary or as two's complements.
 - RGB-888 (24 bits/pixel)
 - RGB-888 (32 bits/pixel)
 - ARGB-8888 (32 bits/pixel)
 - RAW6, 7, 8, 10, 12, 14, 16, 20
 - MIPI CSI-2 recommended memory storage format
- Demosaicing
 - RAW6, 7, 8, 10, 12, 14, 16, 20 to RGB888
- Color correction by linear matrix operations
 - Offset calculation included
- Color space conversion
 - YCbCr422 to RGB888
 - RGB888 to YCbCr422

- Image format conversion
 - YCbCr422 → YC separation (separation into Y and CbCr components)
 - YCbCr422 → Y component extraction
 - YCbCr422 → YCbCr420
 - RGB-888 → RGB-888 (32 bits/pixel)
 - RGB-888 → ARGB-8888 (32 bits/pixel) B first and A last format
 - RGB-888 → ARGB-8888 (32 bits/pixel) A first and B last format
 - RGB-565 → RGB-888 conversion (intermediate processing)
 - RGB-666 → RGB-888 conversion (intermediate processing)
 - RGB-555 → RGB-888 conversion (intermediate processing)
 - RGB-444 → RGB-888 conversion (intermediate processing)
 - RAW6, 7 → RAW8 conversion (intermediate processing)
 - RAW20: 16 higher-order bits are used for processing (demosaic, statis_cal) (intermediate processing)
- 4 channels (SVC0 to 3) from among virtual channels 0 to 15 can be selected for input (however, only SVC0 is usable for demosaicing and statistics)
- Conversion by a lookup table (LUT): 10-bit to 8-bit conversion
- Data clipping: Suppressing the level of YCbCr data. For details see **9.2.4.3.11 Data Clipping**.
- Image clipping
- Frame subsampling
- Bypassing of each function (not performing the function): Several data types selectable
- Pattern generator (for debugging)
 - Y, U, and V register values are output.
- Statistics (statistical processing operations: horizontal detection area: 320 to 4096 pixels; vertical detection area: 240 to 4096)
 - Accumulation in the (16×16, 32×32, 64×64, and 128×128) areas for each of R, G, and B from RAW6, 7, 8, 10, 12, 14, 16, and 20 data
 - Summation of absolute values for differences from adjacent pixel values of G from RAW6, 7, 8, 10, 12, 14, 16, and 20 data
- FIFO
 - Video system: 16 Kbytes
 - Statistics system: 2 Kbytes
- AXI master: 2 channels (video system, statistics system)
- Memory bank address settings

- Support for 8 bank memory bank addresses (video system)
 - ◆ SVC offset address
 - ◆ Non-image processing offset address
 - ◆ Offset address when separation into Y and UV is selected
- Support for 8 memory bank addresses (statistics system: SVC0)
- Memory addressing
 - Video: Stored with image striding
 - Statistics: Stored with consecutive writing line
- Appending of a header (data type, VC, word count) to the start of a line: turning this off or on is possible.
- Appending of a footer (256'd1/256'd0)

A footer value (256'd1/256'd0) is inverted when writing to the same bank so that the CPU can judge how far data have been written.

 - Appending of a footer to the end of each line (video system)
 - Appending of a footer to the end of each frame (statistics system)
- APB slave: Only 32-bit access supported
- Interrupt sources
 - Scan line interrupt: 4 sources
 - Start-of-frame/end-of-frame interrupt: 8 sources
 - FIFO overflow interrupt (video system, statistics system): 2 sources
 - AXI SLVERR and DECERR error interrupts (video system, statistics system): 4 sources
 - AXI specified address transfer completed interrupt (video system: 5 addresses; statistics system: 2 addresses): 7 sources

9.2.4.1.2 Block Diagram

Figure 9.2-9 is a block diagram of the image converter.

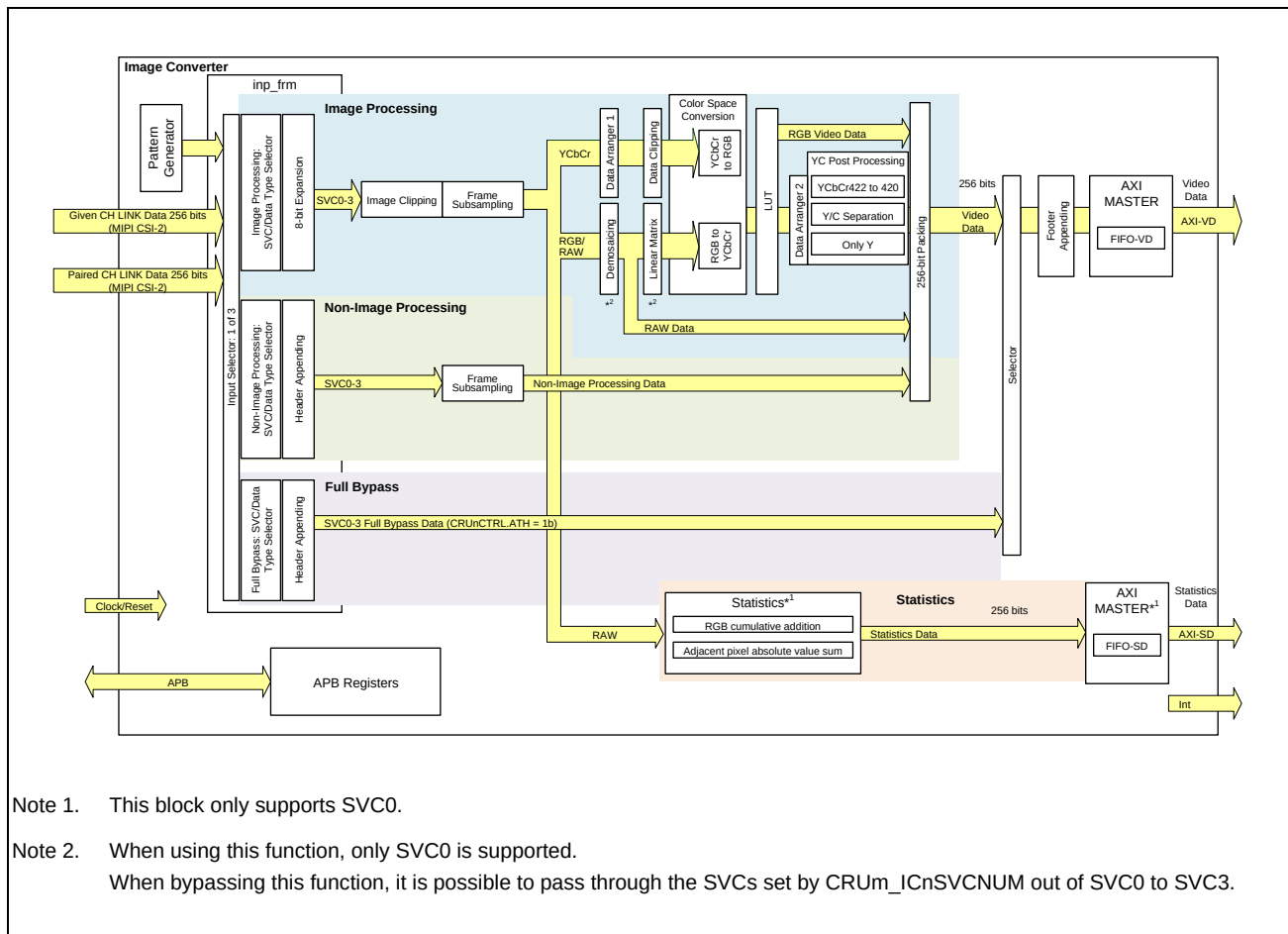


Figure 9.2-9 Block Diagram of Image Converter

Pattern Generator

This function is used for debugging and generates image data for testing. The data types are limited to YCbCr422.

Input Selector (1 of 3)

A register setting selects data for input to the image converter from among the following three sources.

- Image data for testing
- Link data from the paired CRU
- Link data from this CRU

Video

- Image Processing
 - SVC/Data Type Selector
A register setting selects one channel from among 16 VCs (MIPI CSI-2 virtual channels) for use by the image converter. Four channels are selectable and these are treated as SVC0 to 3.
A register setting selects one data type for use in image processing. Four data types are selectable.

- 8-bit Expansion
Input data less than 8 bits ((RGB444/RGB555/RGB565/RGB666, RAW6/7) are expanded to 8 bits (RGB888, RAW8).
- Image Clipping
Input images are cropped to the size selected by register settings for the output of small images.
- Frame Subsampling
Subsamples input image frames to lower the frame rate. This is used to reduce the bandwidth on the bus.
- Data Arrangers 1 and 2
Converts the negative expression of the UV component of YUV (YCbCr) data.
- Data Clipping
Limits data values of YUV (YCbCr) to upper and lower limits. This is used when standards having upper and lower limits such as ITU-R BT.601 and 702 are applicable.
- Demosaicing
Interpolates converted RGB data from RAW (Bayer-array) data.
- Linear Matrix
Applies matrix operations to correct the colors of RGB data converted by demosaicing.
- Color Space Conversion
Converts the color space from YCbCr to RGB and vice versa.
- LUT (Lookup Table)
Converts 10-bit data after color space conversion (YCbCr to RGB or RGB to YCbCr) to a table of 8-bit entries. A basic bit rounding function is set, but the table can also be used for non-linear image correction such as gamma correction.
- YC Post Processing
 - ◆ YCbCr422 to 420
Subsamples YCbCr422 data to convert it into YCbCr420.
 - ◆ Y/C Separation
Sorts and separates YCbCr data into Y and C components.
 - ◆ Only Y
Deletes the C components of YCbCr data so that only Y data remain.
- 256-bit Packing
Packs data into 256-bit units to suit the bus width for transmission through the AXI.
- Non-Image Processing
 - SVC/Data Type Selector
The same channels as those set as SVC0 to 3 described under “Image Processing” are used.
A register setting selects several data types for use in non-image processing.
 - Header Appending
Since non-image processing and full bypass involve the selection of several data types and storage of the data in a mixed state in the same SVC memory, a header is appended to the start of a line of non-image processing or full bypass data so that the data types and data lengths can be identified when read.
- Full Bypass
 - VC/Data Type Selector
The same channels as those set as SVC0 to 3 described under “Image Processing” are used.
A register setting selects several data types for use in full bypass.

- **Footer Appending**
Appends 256'd0/d1 as a footer, which is inverted every time the memory bank is overwritten, so that the CPU can recognize the bank number to which the memory data have been written.
- **AXI MASTER (AXI-VD)**
Video data are initially stored in a FIFO buffer and then written to the DRAM by the AXI master.

Statistics (only for CRU0 and 1)

- **Statistics**
Acquires statistics for each block which is divided from a display for 3A (AE/AWB/AF).
- **Footer Appending**
Appends 512 bits as a footer, which is inverted every time the memory bank is overwritten, so that the CPU can recognize the bank number to which the memory data have been written.
The footer value is written alternately: (1) higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1 and (2) higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0.
- **AXI MASTER (AXI-SD)**
Statistics data are initially stored in a FIFO buffer and then written to the DRAM by the AXI master.

APB Registers

Only 32-bit access is supported.

9.2.4.2 IMC Registers

See **9.2.3.2 MIPI CSI2 Registers** for the CRUm base addresses.

9.2.4.2.1 List of Registers (m = 0, 1)

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
CRU Control Register	CRUm_CRUnCTRL	0000_0000h	0000h	32
CRU Interrupt Enable Register1	CRUm_CRUnIE1	0000_0000h	0004h	32
CRU Interrupt Enable Register2	CRUm_CRUnIE2	0000_0000h	0008h	32
CRU Interrupt Status Register1	CRUm_CRUnINTS1	0000_0000h	000Ch	32
CRU Interrupt Status Register2	CRUm_CRUnINTS2	0000_0000h	0010h	32
Reserve	-	-	0014h to 0017h	-
CRU Reset Register	CRUm_CRUnRST	0000_0000h	0018h	32
Reserve	-	-	001Ch to 002Bh	-
CRU General Read/Write Register	CRUm_CRUnCOM	0000_0000h	002Ch	32
Reserve	-	-	0030h to 003Fh	-
Memory Bank 1 Base Address Lower Register for Video Data	CRUm_AMnMB1ADDRL	0000_0000h	0040h	32
Memory Bank 1 Base Address Higher Register for Video Data	CRUm_AMnMB1ADDRH	0000_0000h	0044h	32
Memory Bank 2 Base Address Lower Register for Video Data	CRUm_AMnMB2ADDRL	0000_0000h	0048h	32
Memory Bank 2 Base Address Higher Register for Video Data	CRUm_AMnMB2ADDRH	0000_0000h	004Ch	32
Memory Bank 3 Base Address Lower Register for Video Data	CRUm_AMnMB3ADDRL	0000_0000h	0050h	32
Memory Bank 3 Base Address Higher Register for Video Data	CRUm_AMnMB3ADDRH	0000_0000h	0054h	32
Memory Bank 4 Base Address Lower Register for Video Data	CRUm_AMnMB4ADDRL	0000_0000h	0058h	32
Memory Bank 4 Base Address Higher Register for Video Data	CRUm_AMnMB4ADDRH	0000_0000h	005Ch	32
Memory Bank 5 Base Address Lower Register for Video Data	CRUm_AMnMB5ADDRL	0000_0000h	0060h	32
Memory Bank 5 Base Address Higher Register for Video Data	CRUm_AMnMB5ADDRH	0000_0000h	0064h	32
Memory Bank 6 Base Address Lower Register for Video Data	CRUm_AMnMB6ADDRL	0000_0000h	0068h	32
Memory Bank 6 Base Address Higher Register for Video Data	CRUm_AMnMB6ADDRH	0000_0000h	006Ch	32
Memory Bank 7 Base Address Lower Register for Video Data	CRUm_AMnMB7ADDRL	0000_0000h	0070h	32
Memory Bank 7 Base Address Higher Register for Video Data	CRUm_AMnMB7ADDRH	0000_0000h	0074h	32
Memory Bank 8 Base Address Lower Register for Video Data	CRUm_AMnMB8ADDRL	0000_0000h	0078h	32
Memory Bank 8 Base Address Higher Register for Video Data	CRUm_AMnMB8ADDRH	0000_0000h	007Ch	32
UV Data Address Offset Lower Register for Video Data	CRUm_AMnUVAOFL	0000_0000h	0080h	32
UV Data Address Offset Higher Register for Video Data	CRUm_AMnUVAOFH	0000_0000h	0084h	32
Memory Bank Enable Register for Video Data	CRUm_AMnMBVALID	0000_0001h	0088h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Memory Address Lower Status Register for Video Data	CRUm_AMnMADRSL	0000_0000h	008Ch	32
Memory Address Higher Status Register for Video Data	CRUm_AMnMADRSH	0000_0000h	0090h	32
Non Image Process Address Offset Lower Register	CRUm_AMnNIAOFL	0000_0000h	0094h	32
Non Image Process Address Offset Higher Register	CRUm_AMnNIAOFH	0000_0000h	0098h	32
SVC Data Address Offset Lower Register for Video Data	CRUm_AMnSVCAOFL	0000_0000h	009Ch	32
SVC Data Address Offset Higher Register for Video Data	CRUm_AMnSVCAOFH	0000_0000h	00A0h	32
Reserve	-	-	00A4h to 00B3h	-
AXI-VD Bus Transfer Completion Event Address 0 Lower Register	CRUm_AMnIVT0ADDRL	0000_0000h	00B4h	32
AXI-VD Bus Transfer Completion Event Address 0 Higher Register	CRUm_AMnIVT0ADDRH	0000_0000h	00B8h	32
AXI-VD Bus Transfer Completion Event Address 1 Lower Register	CRUm_AMnIVT1ADDRL	0000_0000h	00BCh	32
AXI-VD Bus Transfer Completion Event Address 1 Higher Register	CRUm_AMnIVT1ADDRH	0000_0000h	00C0h	32
AXI-VD Bus Transfer Completion Event Address 2 Lower Register	CRUm_AMnIVT2ADDRL	0000_0000h	00C4h	32
AXI-VD Bus Transfer Completion Event Address 2 Higher Register	CRUm_AMnIVT2ADDRH	0000_0000h	00C8h	32
AXI-VD Bus Transfer Completion Event Address 3 Lower Register	CRUm_AMnIVT3ADDRL	0000_0000h	00CCh	32
AXI-VD Bus Transfer Completion Event Address 3 Higher Register	CRUm_AMnIVT3ADDRH	0000_0000h	00D0h	32
AXI-VD Bus Transfer Completion Event Address 4 Lower Register	CRUm_AMnIVT4ADDRL	0000_0000h	00D4h	32
AXI-VD Bus Transfer Completion Event Address 4 Higher Register	CRUm_AMnIVT4ADDRH	0000_0000h	00D8h	32
Reserve	-	-	00DCh to 00EBh	-
AXI-VD Bus Master Transfer Setting Register	CRUm_AMnAXIATTR	0020_0150h	00ECh	32
AXI-VD Bus Master FIFO Setting Register	CRUm_AMnFIFO	0000_0001h	00F0h	32
AXI-VD Bus Master Transfer Resume Register	CRUm_AMnFIFOTRST	0000_0000h	00F4h	32
AXI-VD Bus Master FIFO Pointer Register	CRUm_AMnFIFOPNTR	0000_0000h	00F8h	32
Reserve	-	-	00FCh to 010Fh	-
AXI-VD Bus Master Transfer Stop Register	CRUm_AMnAXISTP	0000_0000h	0110h	32
AXI-VD Bus Master Transfer Stop Status Register	CRUm_AMnAXISTPACK	0000_0000h	0114h	32
Reserve	-	-	0118h to 0127h	-
Image Stride Setting Register	CRUm_AMnIS	0000_0000h	0128h	32
Reserve	-	-	012Ch to 013Bh	-
Memory Bank 1 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB1ADDRL	0000_0000h	013Ch	32
Memory Bank 1 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB1ADDRH	0000_0000h	0140h	32
Memory Bank 2 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB2ADDRL	0000_0000h	0144h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Memory Bank 2 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB2ADDRH	0000_0000h	0148h	32
Memory Bank 3 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB3ADDRL	0000_0000h	014Ch	32
Memory Bank 3 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB3ADDRH	0000_0000h	0150h	32
Memory Bank 4 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB4ADDRL	0000_0000h	0154h	32
Memory Bank 4 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB4ADDRH	0000_0000h	0158h	32
Memory Bank 5 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB5ADDRL	0000_0000h	015Ch	32
Memory Bank 5 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB5ADDRH	0000_0000h	0160h	32
Memory Bank 6 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB6ADDRL	0000_0000h	0164h	32
Memory Bank 6 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB6ADDRH	0000_0000h	0168h	32
Memory Bank 7 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB7ADDRL	0000_0000h	016Ch	32
Memory Bank 7 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB7ADDRH	0000_0000h	0170h	32
Memory Bank 8 Base Address Lower Register for CRU Statistics Data	CRUm_AMnSDB8ADDRL	0000_0000h	0174h	32
Memory Bank 8 Base Address Higher Register for CRU Statistics Data	CRUm_AMnSDB8ADDRH	0000_0000h	0178h	32
Reserve	-	-	017Ch to 018Bh	-
Memory Bank Enable Register for CRU Statistics Data	CRUm_AMnSDBVALID	0000_0001h	018Ch	32
Memory Bank Status Register for CRU Statistics Data	CRUm_AMnSDBMS	0000_0000h	0190h	32
AXI-SD Bus Transfer Completion Event Address 0 Lower Register	CRUm_AMnSDIVT0ADDRL	0000_0000h	0194h	32
AXI-SD Bus Transfer Completion Event Address 0 Higher Register	CRUm_AMnSDIVT0ADDRH	0000_0000h	0198h	32
AXI-SD Bus Transfer Completion Event Address 1 Lower Register	CRUm_AMnSDIVT1ADDRL	0000_0000h	019Ch	32
AXI-SD Bus Transfer Completion Event Address 1 Higher Register	CRUm_AMnSDIVT1ADDRH	0000_0000h	01A0h	32
Reserve	-	-	01A4h to 01BFh	-
AXI-SD Bus Master FIFO Pointer Register	CRUm_AMnSDFIFOPNTR	0000_0000h	01C0h	32
Reserve	-	-	01C4h to 01D7h	-
AXI-SD Bus Master Transfer Stop Register	CRUm_AMnSDAXISTP	0000_0000h	01D8h	32
AXI-SD Bus Master Transfer Stop Status Register	CRUm_AMnSDAXISTPACK	0000_0000h	01DCh	32
Reserve	-	-	01E0h to 01EFh	-
CRU Image Converter Enable Register	CRUm_ICnEN	0000_0000h	01F0h	32
CRU Data Value Processing	CRUm_ICnDTVP	0000_0003h	01F4h	32
CRU SVC Number Register	CRUm_ICnSVCNUM	0000_0000h	01F8h	32
CRU VC Select Register	CRUm_ICnSVC	0000_3210h	01FCh	32
CRU Image Converter Main Control Register (SVC0)	CRUm_ICnIPMC_C0	001E_00FEh	0200h	32
CRU Non Image Process Data Type Code Select Lower Register (SVC0)	CRUm_ICnNIPDT_C0L	0000_0000h	0204h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
CRU Non Image Process Data Type Code Select Higher Register (SVC0)	CRUm_ICnNIPDT_C0H	0000_0000h	0208h	32
CRU Image Clipping Start Line Register (SVC0)	CRUm_ICnSLPrC_C0	0000_0000h	020Ch	32
CRU Image Clipping End Line Register (SVC0)	CRUm_ICnELPrC_C0	0000_0000h	0210h	32
CRU Image Clipping Start Pixel Register (SVC0)	CRUm_ICnSPPrC_C0	0000_0000h	0214h	32
CRU Image Clipping End Pixel Register (SVC0)	CRUm_ICnEPPrC_C0	0000_0000h	0218h	32
CRU Scan Line Interrupt Register (SVC0)	CRUm_ICnSI_C0	0000_0000h	021Ch	32
CRU Linear Matrix offset Register (SVC0 only)	CRUm_ICnLMXOF	0000_0000h	0220h	32
CRU Linear Matrix R coefficient 1 Register (SVC0 only)	CRUm_ICnLMXRC1	0000_0400h	0224h	32
CRU Linear Matrix R coefficient 2 Register (SVC0 only)	CRUm_ICnLMXRC2	0000_0000h	0228h	32
CRU Linear Matrix G coefficient 1 Register (SVC0 only)	CRUm_ICnLMXGC1	0000_0000h	022Ch	32
CRU Linear Matrix G coefficient 2 Register (SVC0 only)	CRUm_ICnLMXGC2	0000_0400h	0230h	32
CRU Linear Matrix B coefficient 1 Register (SVC0 only)	CRUm_ICnLMXBC1	0000_0000h	0234h	32
CRU Linear Matrix B coefficient 2 Register (SVC0 only)	CRUm_ICnLMXBC2	0400_0000h	0238h	32
CRU Statistics Control 1 Register (SVC0 only)	CRUm_ICnSTIC1	0000_0000h	023Ch	32
Reserve	-	-	0240h to 0243h	-
CRU Image Converter Register Setting Change Control Register	CRUm_ICnREGC	0000_0000h	0244h	32
Reserve	-	-	0248h to 0257h	-
CRU Image Converter Main Control Register (SVC1)	CRUm_ICnIPMC_C1	001E_0066h	0258h	32
CRU Image Converter Main Control Register (SVC2)	CRUm_ICnIPMC_C2	001E_0066h	025Ch	32
CRU Image Converter Main Control Register (SVC3)	CRUm_ICnIPMC_C3	001E_0066h	0260h	32
CRU Non Image Process Data Type Code Select Lower Register (SVC1)	CRUm_ICnNIPDT_C1L	0000_0000h	0264h	32
Reserve	-	-	0268h to 026Bh	-
CRU Non Image Process Data Type Code Select Lower Register (SVC2)	CRUm_ICnNIPDT_C2L	0000_0000h	026Ch	32
Reserve	-	-	0270h to 0273h	-
CRU Non Image Process Data Type Code Select Lower Register (SVC3)	CRUm_ICnNIPDT_C3L	0000_0000h	0274h	32
Reserve	-	-	0278h to 029Bh	-
CRU Scan Line Interrupt Register (SVC1)	CRUm_ICnSI_C1	0000_0000h	029Ch	32
Reserve	-	-	02A0h to 02AFh	-
CRU Scan Line Interrupt Register (SVC2)	CRUm_ICnSI_C2	0000_0000h	02B0h	32
Reserve	-	-	02B4h to 02C3h	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
CRU Scan Line Interrupt Register (SVC3)	CRUm_ICnSI_C3	0000_0000h	02C4h	32
Reserve	-	-	02C8h to 02D7h	-
CRU Module Status Register	CRUm_ICnMS	0000_0000h	02D8h	32
CRU Frame Subsampling Control Register	CRUm_ICnDEC	0000_0000h	02DCh	32
CRU Line Count Register (SVC0)	CRUm_ICnLC_C0	0000_0000h	02E0h	32
CRU Line Count Register (SVC1)	CRUm_ICnLC_C1	0000_0000h	02E4h	32
CRU Line Count Register (SVC2)	CRUm_ICnLC_C2	0000_0000h	02E8h	32
CRU Line Count Register (SVC3)	CRUm_ICnLC_C3	0000_0000h	02ECh	32
Reserve	-	-	02F0h to 02FFh	-
CRU Frame Subsampling Interrupt Control Register	CRUm_ICnINTCTRL	0000_0000h	0300h	32
CRU Output Image Format Register	CRUm_ICnDMR	0000_0000h	0304h	32
CRU YCbCr → RGB Color Space Conversion Coefficient 1 Register	CRUm_ICnCSCC1	0000_129Fh	0308h	32
CRU YCbCr → RGB Color Space Conversion Coefficient 2 Register	CRUm_ICnCSCC2	0100_0800h	030Ch	32
CRU YCbCr → RGB Color Space Conversion Coefficient 3 Register	CRUm_ICnCSCC3	1989_0D02h	0310h	32
CRU YCbCr → RGB Color Space Conversion Coefficient 4 Register	CRUm_ICnCSCC4	0645_2045h	0314h	32
CRU RGB → YCbCr Color Space Conversion Y Coefficient 1 Register	CRUm_ICnYCCR1	0000_0107h	0318h	32
CRU RGB → YCbCr Color Space Conversion Y Coefficient 2 Register	CRUm_ICnYCCR2	0064_0204h	031Ch	32
CRU RGB → YCbCr Color Space Conversion Y Coefficient 3 Register	CRUm_ICnYCCR3	0A00_0100h	0320h	32
CRU RGB → YCbCr Color Space Conversion Cb Coefficient 1 Register	CRUm_ICnCBCCR1	0000_1F68h	0324h	32
CRU RGB → YCbCr Color Space Conversion Cb Coefficient 2 Register	CRUm_ICnCBCCR2	01C2_1ED6h	0328h	32
CRU RGB → YCbCr Color Space Conversion Cb Coefficient 3 Register	CRUm_ICnCBCCR3	0A00_0800h	032Ch	32
CRU RGB → YCbCr Color Space Conversion Cr Coefficient 1 Register	CRUm_ICnCRCCR1	0000_01C2h	0330h	32
CRU RGB → YCbCr Color Space Conversion Cr Coefficient 2 Register	CRUm_ICnCRCCR2	1FB7_1E87h	0334h	32
CRU RGB → YCbCr Color Space Conversion Cr Coefficient 3 Register	CRUm_ICnCRCCR3	0A00_0800h	0338h	32
Reserve	-	-	033Ch to 034Bh	-
CRU Lookup Table Control Register	CRUm_ICnLUT	0000_0000h	034Ch	32
CRU Lookup Table Status Register	CRUm_ICnLUTS	0000_0000h	0350h	32
CRU Lookup Table Pointer Register	CRUm_ICnLUTP	0000_0000h	0354h	32
CRU Lookup Table Data Register	CRUm_ICnLUTD	0000_0000h	0358h	32
CRU Test Image Generation Control 1 Register	CRUm_ICnTICTRL1	0000_0000h	035Ch	32
CRU Test Image Generation Control 2 Register	CRUm_ICnTICTRL2	0000_0000h	0360h	32
CRU Test Image Size Setting 1 Register	CRUm_ICnTISIZE1	0000_0000h	0364h	32
CRU Test Image Size Setting 2 Register	CRUm_ICnTISIZE2	0000_0000h	0368h	32

9.2.4.2.2 Register Descriptions

(1) Bypass Setting for Each Function

Figure 9.2-10 shows where the bypass settings by the ICnIPMC_C0/1/2/3.*THR and CRUnCTRL.ATH registers work.

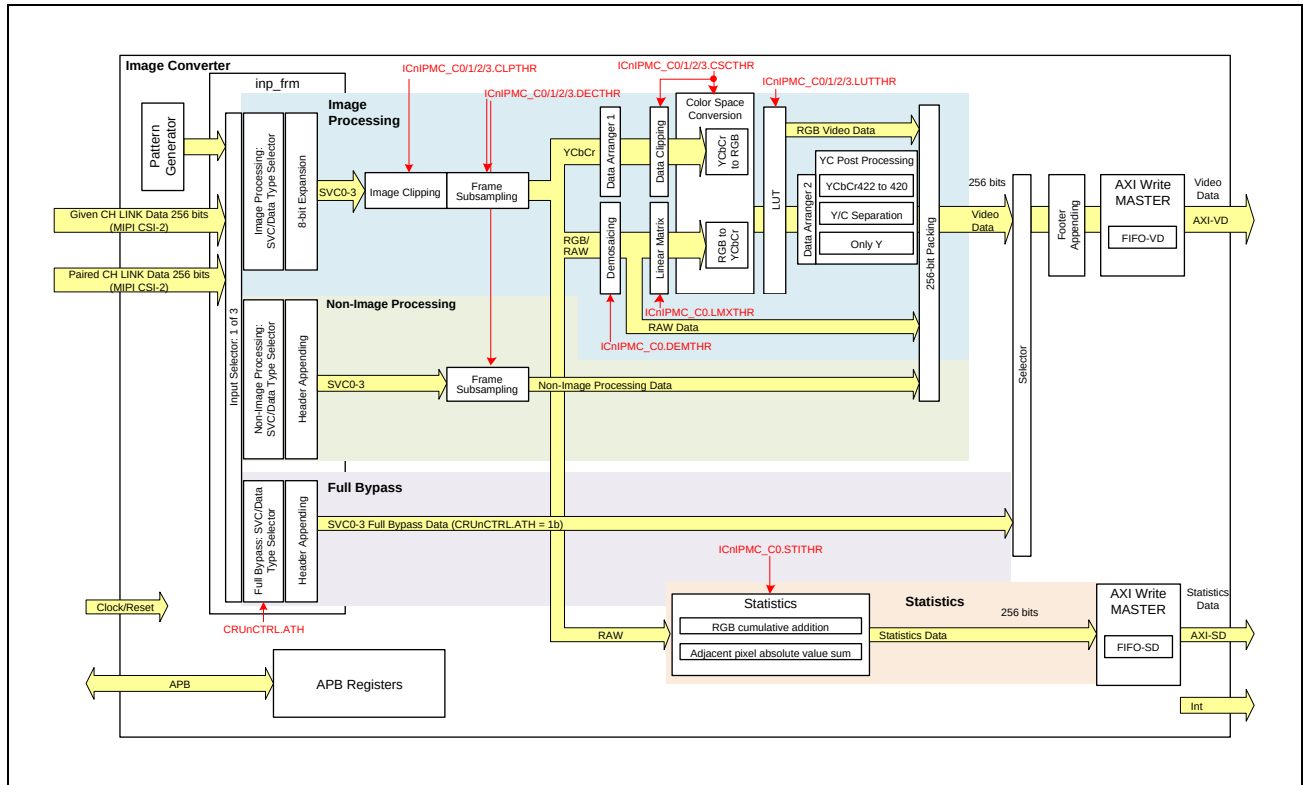


Figure 9.2-10 Bypass Setting for Each Function

(2) CRU Control Register (CRUm_CRUnCTRL) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0000h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ATH	VINSEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	ATH	0h	RW	Full bypass setting for the image converter 0b: Normal 1b: Full bypass is performed and the data from the LINK is output as is to the AXI-VD (the virtual channels and data types set by registers ICnSVC and ICnNIPDT_C0~3L/H are bypassed). Note: See Figure 9.2-10 Bypass Setting for Each Function .
0	VINSEL	0h	RW	MIPI data input selection (This bit is only valid for CH1 and 3. For CH0 and 2, do not change the value of this bit from 0.) 0b: Given CH LINK input 1b: Paired CH (CH0 for CH1, CH2 for CH3) LINK input Note: This bit is valid when ICnTICTRL1.TIEN = 0b.

(3) CRU Interrupt Enable Register1 (CRUm_CRUnIE1) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0004h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SDDEC EE	SDSLV EE	SDFOE	-	-	-	-	-	DECEE	SLVEE	FOE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
30 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
10	SDDECEE	0h	RW	AXI-SD DECERR (statistics data-related) interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated. Note: DECERR occurs when the space where the forwarding address is not allowed is accessed. If an error occurs, check that the forwarding address is correct.
9	SDSLVEE	0h	RW	AXI-SD SLVERR (statistics data-related) interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated. Note: SLVERR occurs by the response from the other party as the destination for transfer from this bus. If this happens, check the destination settings.
8	SDFOE	0h	RW	FIFO-SD overflow (statistics data-related) interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
2	DECEE	0h	RW	AXI-VD DECERR (video data-related) interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated. Note: DECERR occurs when the space where the forwarding address is not allowed is accessed. If an error occurs, check that the forwarding address is correct.
1	SLVEE	0h	RW	AXI-VD SLVERR (video data-related) interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated. Note: SLVERR occurs by the response from the other party as the destination for transfer from this bus. If this happens, check the destination settings.
0	FOE	0h	RW	FIFO-VD overflow (video data-related) interrupt enable control (for debugging) 0b: No interrupts are generated. 1b: Interrupts are generated.

(4) CRU Interrupt Enable Register2 (CRUm_CRUnIE2) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0008h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SDADR1WE	SDADR0WE	-	-	-	VDADR4WE	VDADR3WE	VDADR2WE	VDADR1WE	VDADR0WE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	RW	RW	R	R	R	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	FE3E	FS3E	SL2E	FE2E	FS2E	SL1E	FE1E	FS1E	SL0E	FE0E	FS0E
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
30	-	0h	RW	Do not change the value of this bit from 0h.
29 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25	SDADR1WE	0h	RW	AXI-SD bus write completion interrupt enable control (video data-related) for designated address 1 (35-bit memory address set by AMnSDIVT1ADDRH and AMnSDIVT1ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
24	SDADR0WE	0h	RW	AXI-SD bus write completion interrupt enable control (statistics data-related) for designated address 0 (35-bit memory address set by AMnSDIVT0ADDRH and AMnSDIVT0ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
23 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	VDADR4WE	0h	RW	AXI-VD bus write completion interrupt enable control (video data-related) for designated address 4 (35-bit memory address set by AMnIVT4ADDRH and AMnIVT4ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
19	VDADR3WE	0h	RW	AXI-VD bus write completion interrupt enable control (video data-related) for designated address 3 (35-bit memory address set by AMnIVT3ADDRH and AMnIVT3ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
18	VDADR2WE	0h	RW	AXI-VD bus write completion interrupt enable control (video data-related) for designated address 2 (35-bit memory address set by AMnIVT2ADDRH and AMnIVT2ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
17	VDADR1WE	0h	RW	AXI-VD bus write completion interrupt enable control (video data-related) for designated address 1 (35-bit memory address set by AMnIVT1ADDRH and AMnIVT1ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
16	VDADR0WE	0h	RW	AXI-VD bus write completion interrupt enable control (video data-related) for designated address 0 (35-bit memory address set by AMnIVT0ADDRH and AMnIVT0ADDRH) 0b: No interrupts are generated. 1b: Interrupts are generated.
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
10	FE3E	0h	RW	SVC3 Frame end interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.

Bit	Bit Name	Initial Value	R/W	Description
9	FS3E	0h	RW	SVC3 Frame start interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
8	SL2E	0h	RW	SVC2 Scan line interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
7	FE2E	0h	RW	SVC2 Frame end interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
6	FS2E	0h	RW	SVC2 Frame start interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
5	SL1E	0h	RW	SVC1 Scan line interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
4	FE1E	0h	RW	SVC1 Frame end interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
3	FS1E	0h	RW	SVC1 Frame start interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
2	SL0E	0h	RW	SVC0 Scan line interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
1	FE0E	0h	RW	SVC0 Frame end interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.
0	FS0E	0h	RW	SVC0 Frame start interrupt enable control 0b: No interrupts are generated. 1b: Interrupts are generated.

(5) CRU Interrupt Status Register1 (CRUm_CRUnINTS1) (m = 0, 1)

See **Figure 9.2-9** for where each bit of the interrupt is detected.

Access Size :		32 bits														
Address :		<CRUm_base> + 000Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SDDECES	SDSLVES	SDFOS	-	-	-	-	-	DECES	SLVES	FOS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW1	RW1	RW1	R	R	R	R	RW	RW1	RW1	RW1

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved Whenever it is read, 0 is read. The written value will be ignored.
30 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11	-	0h	RW	Reserved Whenever it is read, 0 is read. The written value will be ignored.
10	SDDECES	0h	RW1	AXI-SD DECERR (statistics data-related) interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE1.SDDECEE = 1b. Note: DECERR occurs when the space where the forwarding address is not allowed is accessed. If an error occurs, check that the forwarding address is correct. Note: This status is detected in the AXI master of statistics data.
9	SDSLVES	0h	RW1	AXI-SD SLVERR (statistics data-related) interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE1.SDSLVEE = 1b. Note: SLVERR occurs by the response from the other party as the destination for transfer from this bus. If this happens, check the destination settings. Note: This status is detected in the AXI master of statistics data.
8	SDFOS	0h	RW1	FIFO-SD overflow (statistics data-related) interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE1.SDFOE = 1b. Note: This status is detected in the AXI master of statistics data.
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	-	0h	RW	Reserved Whenever it is read, 0 is read. The written value will be ignored.
2	DECES	0h	RW1	AXI-VD DECERR (video data-related) interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE1.DECEE = 1b. Note: DECERR occurs when the space where the forwarding address is not allowed is accessed. If an error occurs, check that the forwarding address is correct. Note: This status is detected in the AXI master of video data.

Bit	Bit Name	Initial Value	R/W	Description
1	SLVES	0h	RW1	AXI-VD SLVERR (video data-related) interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit only is valid when CRUnIE1.SLVEE = 1b. Note: SLVERR occurs by the response from the other party as the destination for transfer from this bus. If this happens, check the destination settings. Note: This status is detected in the AXI master of video data.
0	FOS	0h	RW1	FIFO-VD overflow (video data-related) interrupt status (for debugging) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE1.FOE = 1b. Note: This status is detected in the AXI master of video data.

(6) CRU Interrupt Status Register2 (CRUm_CRUnINTS2) (m = 0, 1)

See **Figure 9.2-9** for where each bit of the interrupt is detected.

Access Size :		32 bits														
Address :		<CRUm_base> + 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SDADR1WS	-	-	-	-	VDADR4WS	VDADR3WS	VDADR2WS	VDADR1WS	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	RW1	R	R	R	R	RW1	RW1	RW1	RW1	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SL3S	FE3S	FS3S	SL2S	FE2S	FS2S	SL1S	FE1S	FS1S	SL0S	FE0S	FS0S
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW1	RW1	RW1	RW1	RW1	RW1	RW1	RW1	RW1	RW1	RW1	RW1

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	0h	RW	Reserved Whenever it is read, 0 is read. The written value will be ignored.
29 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25	SDADR1WS	0h	RW1	AXI-SD bus write completion interrupt status (statistics data-related) for designated address 1 (35 bits) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.SDADR1WE = 1b. Note: This status is detected in the AXI master of statistics data.
24 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	VDADR4WS	0h	RW1	AXI-VD bus write completion interrupt status (video data-related) for designated address 4 (35 bits) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.VDADR4WE = 1b. Note: This status is detected in the AXI master of video data.
19	VDADR3WS	0h	RW1	AXI-VD bus write completion interrupt status (video data-related) for designated address 3 (35 bits) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.VDADR3WE = 1b. Note: This status is detected in the AXI master of video data.
18	VDADR2WS	0h	RW1	AXI-VD bus write completion interrupt status (video data-related) for designated address 2 (35 bits) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.VDADR2WE = 1b. Note: This status is detected in the AXI master of video data.
17	VDADR1WS	0h	RW1	AXI-VD bus write completion interrupt status (video data-related) for designated address 1 (35 bits) Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.VDADR1WE = 1b. Note: This status is detected in the AXI master of video data.
16 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
11	SL3S	0h	RW1	SVC3 Scan line interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.SL3E = 1b. Note: This status is detected in SVC3 image clipping.
10	FE3S	0h	RW1	SVC3 Frame end interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FE3E = 1b. Note: This status is detected in SVC3 image clipping.
9	FS3S	0h	RW1	SVC3 Frame start interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FS3E = 1b. Note: This status is detected in SVC3 image clipping.
8	SL2S	0h	RW1	SVC2 Scan line interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.SL2E = 1b. Note: This status is detected in SVC2 image clipping.
7	FE2S	0h	RW1	SVC2 Frame end interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FE2E = 1b. Note: This status is detected in SVC2 image clipping.
6	FS2S	0h	RW1	SVC2 Frame start interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FS2E = 1b. Note: This status is detected in SVC2 image clipping.
5	SL1S	0h	RW1	SVC1 Scan line interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.SL1E = 1b. Note: This status is detected in SVC1 image clipping.
4	FE1S	0h	RW1	SVC1 Frame end interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FE1E = 1b. Note: This status is detected in SVC1 image clipping.
3	FS1S	0h	RW1	SVC1 Frame start interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FS1E = 1b. Note: This status is detected in SVC1 image clipping.
2	SL0S	0h	RW1	SVC0 Scan line interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.SL0E = 1b. Note: This status is detected in SVC0 image clipping.
1	FE0S	0h	RW1	SVC0 Frame end interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FE0E = 1b. Note: This status is detected in SVC0 image clipping.
0	FS0S	0h	RW1	SVC0 Frame start interrupt status Write 0b: Invalid, 1b: Clears the interrupt & status Read 0b: No interrupts generated, 1b: Interrupts generated Note: This bit is only valid when CRUnIE2.FS0E = 1b. Note: This status is detected in SVC0 image clipping.

(7) CRU Reset Register (CRUm_CRUnRST) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0018h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	VRESETN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name															
31 to 1	-															
	All 0															
	R															
	Reserved															
	Whenever it is read, 0b is read. The written value will be ignored.															
0	VRESETN															
	0h															
	RW															
	0b: Resets the Image Converter module.															
	1b: Releases the Image Converter module from the reset state.															

(8) CRU General Read/Write Register (CRUm_CRUnCOM) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 002Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMMON[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMMON[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 0	COMMON [31:0]															
	Initial Value															
	0h															
	R/W															
	Description															
	General register															
	The value written to this register can be read.															

(9) Memory Bank 1 Base Address Lower Register for Video Data (CRUm_AMnMB1ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0040h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB1ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB1ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB1ADDRL [31:9]	0h	RW	Video data transfer destination base address 1 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(10) Memory Bank 1 Base Address Higher Register for Video Data (CRUm_AMnMB1ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0044h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB1ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB1ADDRH [34:32]		0h		RW		Video data transfer destination base address 1 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(11) Memory Bank 2 Base Address Lower Register for Video Data (CRUm_AMnMB2ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0048h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB2ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB2ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB2ADDRL [31:9]	0h	RW	Video data transfer destination base address 2 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(12) Memory Bank 2 Base Address Higher Register for Video Data (CRUm_AMnMB2ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 004Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB2ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB2ADDRH [34:32]		0h		RW		Video data transfer destination base address 2 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(13) Memory Bank 3 Base Address Lower Register for Video Data (CRUm_AMnMB3ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0050h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB3ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB3ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB3ADDRL [31:9]	0h	RW	Video data transfer destination base address 3 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(14) Memory Bank 3 Base Address Higher Register for Video Data (CRUm_AMnMB3ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0054h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB3ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB3ADDRH [34:32]		0h		RW		Video data transfer destination base address 3 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(15) Memory Bank 4 Base Address Lower Register for Video Data (CRUm_AMnMB4ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0058h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB4ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB4ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name		Initial Value	R/W	Description												
31 to 9	MB4ADDRL [31:9]		0h	RW	Video data transfer destination base address 4 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)												
8 to 0	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

(16) Memory Bank 4 Base Address Higher Register for Video Data (CRUm_AMnMB4ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 005Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB4ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB4ADDRH [34:32]		0h		RW		Video data transfer destination base address 4 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(17) Memory Bank 5 Base Address Lower Register for Video Data (CRUm_AMnMB5ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0060h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB5ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB5ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB5ADDRL [31:9]	0h	RW	Video data transfer destination base address 5 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(18) Memory Bank 5 Base Address Higher Register for Video Data (CRUm_AMnMB5ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0064h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB5ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB5ADDRH [34:32]		0h		RW		Video data transfer destination base address 5 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(19) Memory Bank 6 Base Address Lower Register for Video Data (CRUm_AMnMB6ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0068h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB6ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB6ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB6ADDRL [31:9]	0h	RW	Video data transfer destination base address 6 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(20) Memory Bank 6 Base Address Higher Register for Video Data (CRUm_AMnMB6ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 006Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB6ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB6ADDRH [34:32]		0h		RW		Video data transfer destination base address 6 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(21) Memory Bank 7 Base Address Lower Register for Video Data (CRUm_AMnMB7ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0070h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB7ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB7ADDRL[15:9]							-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	MB7ADDRL [31:9]	0h	RW	Video data transfer destination base address 7 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(22) Memory Bank 7 Base Address Higher Register for Video Data (CRUm_AMnMB7ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0074h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB7ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB7ADDRH [34:32]		0h		RW		Video data transfer destination base address 7 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(23) Memory Bank 8 Base Address Lower Register for Video Data (CRUm_AMnMB8ADDRL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0078h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MB8ADDRL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MB8ADDRL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	MB8ADDRL [31:9]	0h	RW	Video data transfer destination base address 8 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(24) Memory Bank 8 Base Address Higher Register for Video Data (CRUm_AMnMB8ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 007Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MB8ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	MB8ADDRH [34:32]		0h		RW		Video data transfer destination base address 8 (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)									

(25) UV Data Address Offset Lower Register for Video Data (CRUm_AMnUVAOFL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0080h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
UVAOFL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
UVAOFL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name																
31 to 9	UVAOFL [31:9]																
	Initial Value 0h																
	R/W RW																
Description																	
UV data address offset for image processing data YUV output (valid bits: [31:9])																	
Set it as the base address in units of 512 bytes.																	
Note: The 9 lower-order bits cannot be written.																	
(These bits cannot be modified while the AXI-VD is operating.)																	
8 to 0	-																
	Initial Value All 0																
	R/W R																
Description																	
Reserved																	
Whenever it is read, 0b is read. The written value will be ignored.																	

(26) UV Data Address Offset Higher Register for Video Data (CRUm_AMnUVAOFH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0084h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	UVAOFH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 3	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
2 to 0	UVAOFH [34:32]		0h		RW	UV data address offset for image processing data YUV output (valid bits: [34:32]) (These bits cannot be modified while the AXI-VD is operating.)										

(27) Memory Bank Enable Register for Video Data (CRUm_AMnMBVALID) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0088h														
Initial Value :		0000_0001h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	MBVALID[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7 to 0	MBVALID[7:0]	1h	RW	Video data transfer destination valid address (the details are described in 9.2.4.3.3 Transfer to DRAM.) [0]: 0: MB1 invalid, 1: MB1 valid [1]: 0: MB2 invalid, 1: MB2 valid [2]: 0: MB3 invalid, 1: MB3 valid [3]: 0: MB4 invalid, 1: MB4 valid [4]: 0: MB5 invalid, 1: MB5 valid [5]: 0: MB6 invalid, 1: MB6 valid [6]: 0: MB7 invalid, 1: MB7 valid [7]: 0: MB8 invalid, 1: MB8 valid Note: Set the valid destination address from the LSB continuously. Data transferred from the LSB are written sequentially only to valid memory banks, and the processing is repeated from the start (LSB) when the last address is reached. (These bits cannot be modified while the AXI-VD is operating.)

(28) Memory Address Lower Status Register for Video Data (CRUm_AMnMADRSL) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 008Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MADRSL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADRSL[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MADRSL [31:0]	0h	R	Memory address for video data (lower) Reads the memory address (lower) to which the current video data was written. Note: When this register is read, AMnMADRSH of the higher-order address also latches the address. Note: Be sure to read AMnMADRSH after AMnMADRSL has been read.

(29) Memory Address Higher Status Register for Video Data (CRUm_AMnMADRSR) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0090h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	MADRSR[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	MADRSR [34:32]	0h	R	Memory address for video data (higher) Read the memory address (higher) to which the current video data was written. Note: When the AMnMADRSL register is read, the higher address AMnMADRSR also latches the address. Note: Be sure to lead AMnMADRSR after AMnMADRSL lead.

(30) Non Image Process Address Offset Lower Register (CRUm_AMnNIAOFL) (m = 0, 1)

Access Size :		32 bits															
Address :		<CRUm_base> + 0094h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
NIAOFL[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
NIAOFL[15:9]								-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 9	NIAOFL[31:9]	0h	RW	Non-image processing address offset (lower) for image processing for video data (significant bits [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)													
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

(31) Non Image Process Address Offset Higher Register (CRUm_AMnNIAOFH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0098h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	NIAOFH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 3	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
2 to 0	NIAOFH [34:32]		0h		RW	Non-image processing address offset (higher) for image processing for video data (valid bits [34:32]) (These bits cannot be modified while the AXI-VD is operating.)										

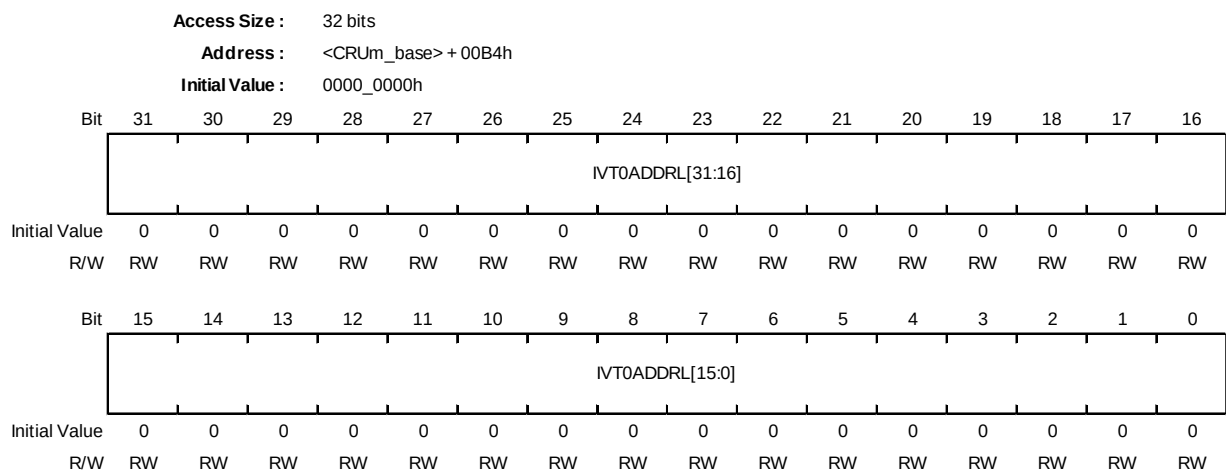
(32) SVC Data Address Offset Lower Register for Video Data (CRUm_AMnSVCAOFL) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 009Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVCAOFL[31:16]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVCAOFL[15:9]							-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name	Initial Value	R/W	Description												
31 to 9	SVCAOFL [31:9]	0h	RW	SVC data address offset (lower) for video data (valid bits [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)												
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

(33) SVC Data Address Offset Higher Register for Video Data (CRUm_AMnSVCAOFH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00A0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SVCAOFH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 3	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
2 to 0	SVCAOFH [34:32]		0h		RW	SVC data address offset (higher) for video data (valid bits [34:32]) (These bits cannot be modified while the AXI-VD is operating.)										

(34) AXI-VD Bus Transfer Completion Event Address 0 Lower Register (CRUm_AMnIVT0ADDRL) (m = 0, 1)

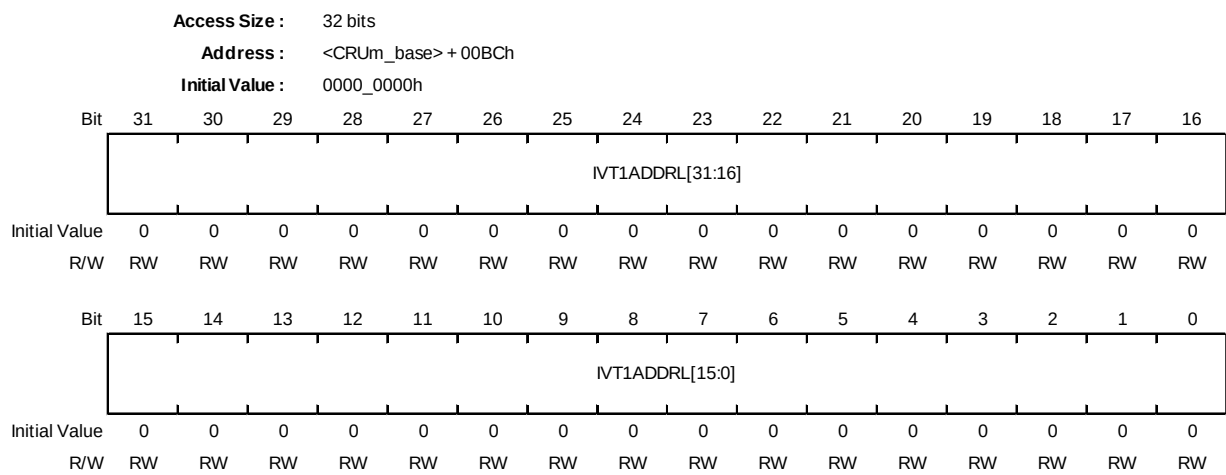


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	IVT0ADDRL [31:0]	0h	RW	AXI-VD bus transfer completion event address 0 for video data (lower) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.

(35) AXI-VD Bus Transfer Completion Event Address 0 Higher Register (CRUm_AMnIVT0ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00B8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IVT0ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	IVT0ADDRH [34:32]		0h		RW		AXI-VD bus transfer completion event address 0 for video data (higher) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.									

(36) AXI-VD Bus Transfer Completion Event Address 1 Lower Register (CRUm_AMnIVT1ADDR1) (m = 0, 1)

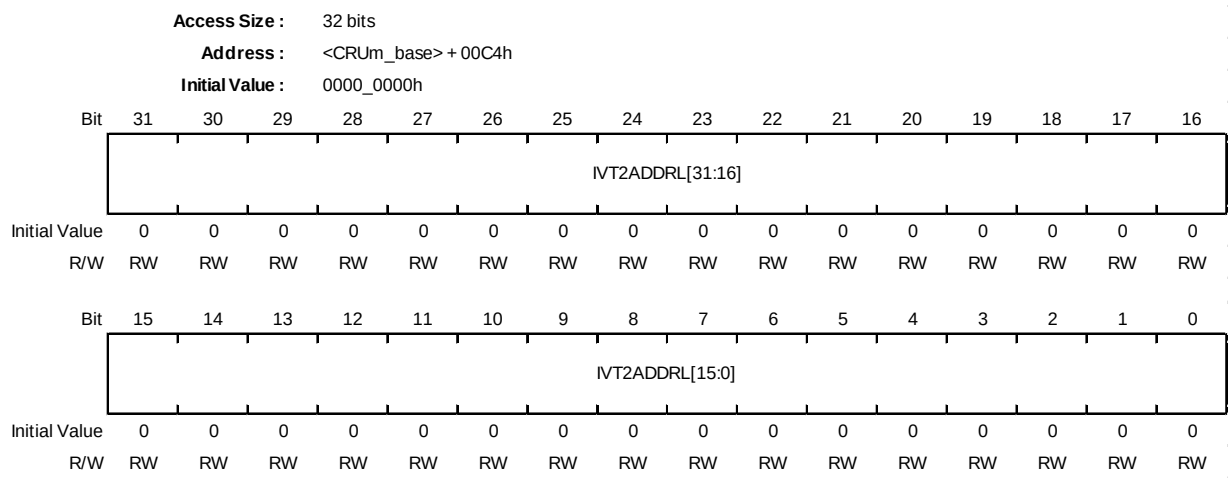


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	IVT1ADDR1 [31:0]	0h	RW	AXI-VD bus transfer completion event address 1 for video data (lower) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.

(37) AXI-VD Bus Transfer Completion Event Address 1 Higher Register (CRUm_AMnIVT1ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00C0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IVT1ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	IVT1ADDRH [34:32]		0h		RW		AXI-VD bus transfer completion event address 1 for video data (higher) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.									

(38) AXI-VD Bus Transfer Completion Event Address 2 Lower Register (CRUm_AMnIVT2ADDRL) (m = 0, 1)

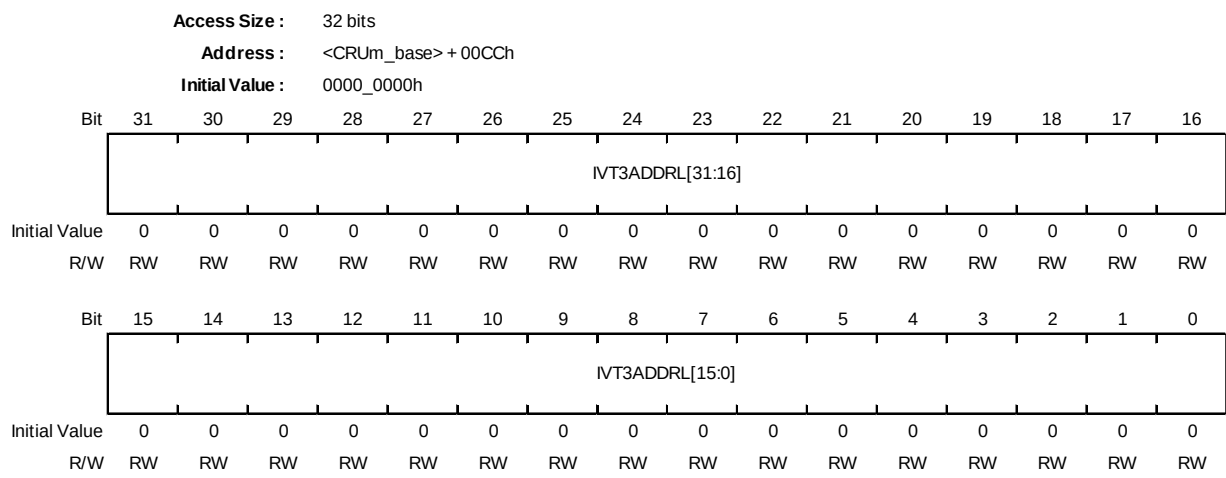


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	IVT2ADDRL [31:0]	0h	RW	AXI-VD bus transfer completion event address 2 for video data (lower) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.

(39) AXI-VD Bus Transfer Completion Event Address 2 Higher Register (CRUm_AMnIVT2ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00C8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IVT2ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	IVT2ADDRH [34:32]		0h		RW		AXI-VD bus transfer completion event address 2 for video data (higher) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.									

(40) AXI-VD Bus Transfer Completion Event Address 3 Lower Register (CRUm_AMnIVT3ADDRL) (m = 0, 1)

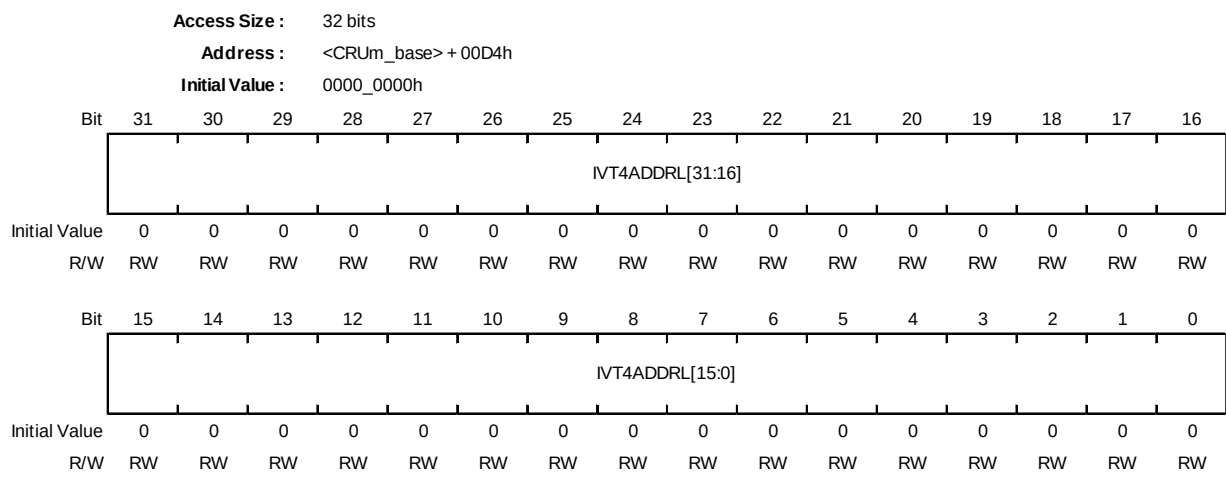


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	IVT3ADDRL [31:0]	0h	RW	AXI-VD bus transfer completion event address 3 for video data (lower) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.

(41) AXI-VD Bus Transfer Completion Event Address 3 Higher Register (CRUm_AMnIVT3ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00D0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IVT3ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	IVT3ADDRH [34:32]		0h		RW		AXI-VD bus transfer completion event address 3 for video data (higher) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.									

(42) AXI-VD Bus Transfer Completion Event Address 4 Lower Register (CRUm_AMnIVT4ADDRL) (m = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	IVT4ADDRL [31:0]	0h	RW	AXI-VD bus transfer completion event address 4 for video data (lower) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.

(43) AXI-VD Bus Transfer Completion Event Address 4 Higher Register (CRUm_AMnIVT4ADDRH) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 00D8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	IVT4ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	IVT4ADDRH [34:32]		0h		RW		AXI-VD bus transfer completion event address 4 for video data (higher) Note: The completion of transfer of video data to the address set in this register is notified by an interrupt.									

(44) AXI-VD Bus Master Transfer Setting Register (CRUm_AMnAXIATTR) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 00ECh

Initial Value : 0020_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	AXILEN[3:0]			
Initial Value	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 24	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
23	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22 to 16	-	20h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9, 8	-	1h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 4	-	5h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
3 to 0	AXILEN[3:0]	0h	RW	Maximum value setting of AXI-VD burst length for video data (changing the value during AXI-VD operation is prohibited) Note: If the transfer speed is given priority, set AXILEN[3:0] = H'F. When YUV is not separated (ICnDMR.YCMODE[2:0] = 3'b010): 0h: Up to 1 burst (initial value) 1h: Up to 2 bursts 2h: Up to 3 bursts 3h: Up to 4 bursts 4h: Up to 5 bursts 5h: Up to 6 bursts ⋮ Fh: Up to 16 bursts When YUV is separated (ICnDMR.YCMODE[2:0] = 3'b010): 0h: Up to 1 burst 1h: Up to 1 burst 2h: Up to 2 bursts 3h: Up to 2 bursts 4h: Up to 3 bursts 5h: Up to 3 bursts ⋮ Eh: Up to 8 bursts Fh: Up to 8 bursts

(45) AXI-VD Bus Master FIFO Setting Register (m = 0, 1) (CRUm_AMnFIFO)

Access Size : 32 bits

Address : <CRUm_base> + 00F0h

Initial Value : 0000_0001h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	RETSVC[2:0]		-	-	-	-	FIFO VFREC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	RW	RW	RW	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 4	RETSVC[2:0]	0h	RW	SVC number setting to resume when FIFO-VD overflow 000b : Resume from the beginning of SVC0 frame (initial value) 001b : Resume from the beginning of the SVC1 frame 010b : Resume from the beginning of the SVC2 frame 011b : Resume from the beginning of the SVC3 frame 1xxb : Resume from the VC first frame that came, without SVC designation
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	FIFOVFREC	1h	RW	Mode setting when image data FIFO-VD overflows When a FIFO-VD overflows, the FIFO-VD is cleared and the AXI-VD master stops transferring to memory. There are two methods as shown below. 0b: Recovery mode: The AXI-VD Master performs the following operations after stopped. - Resume import from SVC selected by AMnFIFO.RETSVC[2:0] - Resume writing from the beginning of the overflow MB 1b: Stop mode (default): The FIFO remains stopped after the overflow occurs. [How to resume] When the AXI-VD master stops, it updates the FOS bit in the CRUnINTS1 register and reg_axi_mst_err_int Interrupts are generated. Set the AXI_TRANS_START bit of the AMnFIFOTRST register to 1 after detection. After this, the operation resumes with the same behavior as in recovery mode. Note 1: The image data being processed will be corrupted during FIFO overflow. FIFO overflow may be caused by bus congestion. Please take measures to alleviate bus congestion. Note 2: The data for the first horizontal line after FIFO-VD overflow return is not guaranteed. Note 3: If set Footer appending ON (ICnDTVP.FTON=1), only if no FIFO-VD overflow occurs. Set to Footer appending OFF (ICnDTVP.FTON=0), when FIFO-VD will overflow occurs.

(46) AXI-VD Bus Master Transfer Resume Register (m = 0, 1) (CRUm_AMnFIFOTRST)

Access Size : 32 bits																
Address : <CRUm_base> + 00F4h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AXI_TRANS_START
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	AXI_TRANS_START	0h	RW1	AXI-VD transfer control for image data (valid when AMnFIFO.FIFOVFREC = 1). 0b: Writing prohibited. Write 1 to this bit to recover the image-related AXI-VD (transfer resume) (This bit returns to 0 when transfer is resumed.)

(47) AXI-VD Bus Master FIFO Pointer Register (CRUm_AMnFIFOPNTR) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 00F8h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFORPNTR_B1[7:0]								FIFORPNTR_B0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFOWPNTR_B1[7:0]								FIFOWPNTR_B0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	FIFORPNTR_B1 [7:0]	0h	R	FIFO-VD read pointer (bank 1) The read pointer and write pointer comparison allows checking if there are data in the FIFO. Bank 1 FIFO-VD read pointer [7:0] Real RAM address for FIFO-VD (Max.: FFh) Note: When (AMnFIFOPNTR.FIFOWPNTR_B0[7:0]≠AMnFIFOPNTR.FIFORPNTR_B0[7:0]) (AMnFIFOPNTR.FIFOWPNTR_B1[7:0]≠AMnFIFOPNTR.FIFORPNTR_B1[7:0]), data are present in the FIFO-VD.
23 to 16	FIFORPNTR_B0 [7:0]	0h	R	FIFO-VD read pointer (bank 0) The read pointer and write pointer comparison allows checking if there are data in the FIFO. Bank 0 FIFO-VD read pointer [7:0] Real RAM address for FIFO-VD (Max.: FFh) Note: When (AMnFIFOPNTR.FIFOWPNTR_B0[7:0]≠AMnFIFOPNTR.FIFORPNTR_B0[7:0]) (AMnFIFOPNTR.FIFOWPNTR_B1[7:0]≠AMnFIFOPNTR.FIFORPNTR_B1[7:0]), data are present in the FIFO-VD.
15 to 8	FIFOWPNTR_B1 [7:0]	0h	R	FIFO-VD write pointer (bank 1) The read pointer and write pointer comparison allows checking if there are data in the FIFO. Bank 1 FIFO-VD write pointer [7:0] Real RAM address for FIFO-VD (Max.: FFh) Note: When (AMnFIFOPNTR.FIFOWPNTR_B0[7:0]≠AMnFIFOPNTR.FIFORPNTR_B0[7:0]) (AMnFIFOPNTR.FIFOWPNTR_B1[7:0]≠AMnFIFOPNTR.FIFORPNTR_B1[7:0]), data are present in the FIFO-VD.
7 to 0	FIFOWPNTR_B0 [7:0]	0h	R	FIFO-VD write pointer (bank 0) The read pointer and write pointer comparison allows checking if there are data in the FIFO. Bank 0 FIFO-VD write pointer [7:0] Real RAM address for FIFO-VD (Max.: FFh) Note: When (AMnFIFOPNTR.FIFOWPNTR_B0[7:0]≠AMnFIFOPNTR.FIFORPNTR_B0[7:0]) (AMnFIFOPNTR.FIFOWPNTR_B1[7:0]≠AMnFIFOPNTR.FIFORPNTR_B1[7:0]), data are present in the FIFO-VD.

(48) AXI-VD Bus Master Transfer Stop Register (CRUm_AMnAXISTP) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0110h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AXI_STOP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	AXI_STOP		0h		RW		AXI-VD bus transfer stop (used to stop the CRU receiving data. Follow the procedure for stopping reception described in 9.2.5.2 Stopping Reception for the MIPI CSI-2 Input.) 0b: Does not stop the AXI-VD master transferring data. 1b: Stops the AXI-VD master transferring data.									

(49) AXI-VD Bus Master Transfer Stop Status Register (CRUm_AMnAXISTPACK) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0114h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AXI_STOP_ACK
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	AXI_STOP_ACK	0h	R	AXI-VD bus transfer stop response (used to stop the CRU receiving data. Follow the procedure for stopping reception described in 9.2.5.2 Stopping Reception for the MIPI CSI-2 Input .) 0b: The AXI-VD master is transferring data. 1b: The AXI-VD master has finished transferring data (valid when AXI_STOP = 1)

(50) Image Stride Setting Register (CRUm_AMnIS) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0128h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IS[7:0]								-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14 to 7	IS[7:0]	0h	RW	Line start address offset setting for image stride You can set 128 to 32640 bytes (with a 128-byte step) by setting IS[7:0]. IS[7:0] = 00h: Setting prohibited IS[7:0] = 01h: 128 bytes IS[7:0] = 02h: 256 bytes IS[7:0] = 03h: 384 bytes IS[7:0] = 04h: 512 bytes : IS[7:0] = 80h: 16384 bytes : IS[7:0] = FFh: 32640 bytes Note: For output format RGB888 (24 bits), the setting must be in units of 384 bytes (multiples of 3 for IS [7:0]). Note: If the output format is other than RGB888 (24 bits), the setting must be in units of 128 bytes. Note: The setting must be at least the number of bytes per line. Note: When adding a header and footer, set the value including a header and footer. Note: Only data output from the video data bus are supported. Statistics data output is not supported.
6 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(51) Memory Bank 1 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDB1ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 013Ch																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMB1ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMB1ADDRL[15:9]								-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 9	SDMB1ADDRL [31:9]															
	Initial Value 0h															
	R/W RW															
Statistics data transfer destination base address 1 (valid bits: [31:9])																
Set it as the base address in units of 512 bytes.																
Note: The 9 lower-order bits cannot be written.																
(These bits cannot be modified while the AXI-VD is operating.)																
8 to 0	-															
	Initial Value All 0															
	R/W R															
Reserved																
Whenever it is read, 0b is read. The written value will be ignored.																

(52) Memory Bank 1 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDBM1ADDRH) (m = 0, 1)

Access Size : 32 bits																		
Address : <CRUm_base> + 0140h																		
Initial Value : 0000_0000h																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMB1ADDRH[34:32]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW		
Bit	Bit Name															Initial Value	R/W	Description
31 to 3	-															All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	SDMB1ADDRH [34:32]															0h	RW	Statistics data transfer destination base address 1 (valid bits: [34:32]) (These bits cannot be modified while the AXI-SD is operating.)

(53) Memory Bank 2 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB2ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0144h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMB2ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMB2ADDRL[15:9]								-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 9	SDMB2ADDRL [31:9]															
	Initial Value 0h															
	R/W RW															
Statistics data transfer destination base address 2 (valid bits: [31:9])																
Set it as the base address in units of 512 bytes.																
Note: The 9 lower-order bits cannot be written.																
(These bits cannot be modified while the AXI-VD is operating.)																
8 to 0	-															
	Initial Value All 0															
	R															
Reserved																
Whenever it is read, 0b is read. The written value will be ignored.																

(54) Memory Bank 2 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDBM2ADDRH) (m = 0, 1)

Access Size : 32 bits															
Address : <CRUm_base> + 0148h															
Initial Value : 0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDBM2ADDRH[34:32]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name														
31 to 3	-														
		Initial Value		R/W		Description									
31 to 3		-		All 0		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0		SDBM2ADDRH [34:32]		0h		RW		Statistics data transfer destination base address 2 (valid bits: [34:32]) (These bits cannot be modified while the AXI-SD is operating.)							

(55) Memory Bank 3 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB3ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 014Ch																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMB3ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMB3ADDRL[15:9]								-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 9	SDMB3ADDRL [31:9]															
	Initial Value 0h															
	R/W RW															
Statistics data transfer destination base address 3 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)																
8 to 0	-															
	Initial Value All 0															
	R/W R															
Reserved Whenever it is read, 0b is read. The written value will be ignored.																

(56) Memory Bank 3 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDBM3ADDRH) (m = 0, 1)

Access Size : 32 bits																		
Address : <CRUm_base> + 0150h																		
Initial Value : 0000_0000h																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMB3ADDRH[34:32]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW		
Bit	Bit Name															Initial Value	R/W	Description
31 to 3	-															All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	SDMB3ADDRH [34:32]															0h	RW	Statistics data transfer destination base address 3 (valid bits: [34:32]) (These bits cannot be modified while the AXI-SD is operating.)

(57) Memory Bank 4 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB4ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0154h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDMB4ADDRL[31:16]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDMB4ADDRL[15:9]							-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 9	SDMB4ADDRL [31:9]															
	Initial Value 0h															
	R/W RW															
	Statistics data transfer destination base address 4 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)															
8 to 0	-															
	All 0															
	R															
	Reserved Whenever it is read, 0b is read. The written value will be ignored.															

(58) Memory Bank 4 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDBM4ADDRH) (m = 0, 1)

Access Size : 32 bits																															
Address : <CRUm_base> + 0158h																															
Initial Value : 0000_0000h																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16															
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDBM4ADDRH[34:32]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW															
Bit	Bit Name																														
31 to 3	-																														
	All 0																														
	R																														
Reserved																															
Whenever it is read, 0b is read. The written value will be ignored.																															
2 to 0	SDBM4ADDRH [34:32]																														
	0h																														
	RW																														
Statistics data transfer destination base address 4 (valid bits: [34:32])																															
(These bits cannot be modified while the AXI-SD is operating.)																															

(59) Memory Bank 5 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDB5ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 015Ch																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDB5ADDRL[31:16]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDB5ADDRL[15:9]								-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
	Initial Value															
	R/W															
	Description															
31 to 9	SDB5ADDRL [31:9]	0h	RW	Statistics data transfer destination base address 5 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)												
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

(60) Memory Bank 5 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDB5ADDRH) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0160h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDB5ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3	-															
	All 0															
	R															
	Reserved															
	Whenever it is read, 0b is read. The written value will be ignored.															
2 to 0	SDB5ADDRH 0h															
	[34:32]															
	RW															
	Statistics data transfer destination base address 5 (valid bits: [34:32])															
	(These bits cannot be modified while the AXI-SD is operating.)															

(61) Memory Bank 6 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB6ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0164h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMB6ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMB6ADDRL[15:9]								-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 9	SDMB6ADDRL [31:9]															
	Initial Value 0h															
	R/W RW															
Statistics data transfer destination base address 6 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)																
8 to 0	-															
	Initial Value All 0															
	R/W R															
Reserved Whenever it is read, 0b is read. The written value will be ignored.																

(62) Memory Bank 6 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDB6ADDRH) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0168h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMB6ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3	-															
	All 0															
	R															
	Reserved															
	Whenever it is read, 0b is read. The written value will be ignored.															
2 to 0	SDMB6ADDRH 0h															
	[34:32]															
	RW															
	Statistics data transfer destination base address 6 (valid bits: [34:32])															
	(These bits cannot be modified while the AXI-SD is operating.)															

(63) Memory Bank 7 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB7ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 016Ch																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDMB7ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMB7ADDRL[15:9]								-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
Bit	Initial Value															
Bit	R/W															
Bit	Description															
31 to 9	SDMB7ADDRL [31:9]	0h	RW	Statistics data transfer destination base address 7 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)												
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

(64) Memory Bank 7 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDB7ADDRH) (m = 0, 1)

Access Size : 32 bits																				
Address : <CRUm_base> + 0170h																				
Initial Value : 0000_0000h																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMB7ADDRH[34:32]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW				
Bit	Bit Name															Initial Value	R/W	Description		
31 to 3	-																			
All 0																		Reserved Whenever it is read, 0b is read. The written value will be ignored.		
2 to 0	SDMB7ADDRH [34:32]															0h	RW			
Statistics data transfer destination base address 7 (valid bits: [34:32]) (These bits cannot be modified while the AXI-SD is operating.)																				

(65) Memory Bank 8 Base Address Lower Register for CRU Statistics Data
(CRUm_AMnSDMB8ADDRL) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0174h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDMB8ADDRL[31:16]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDMB8ADDRL[15:9]								-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
	Initial Value															
	R/W															
	Description															
31 to 9	SDMB8ADDRL [31:9]	0h	RW	Statistics data transfer destination base address 8 (valid bits: [31:9]) Set it as the base address in units of 512 bytes. Note: The 9 lower-order bits cannot be written. (These bits cannot be modified while the AXI-VD is operating.)												
8 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

(66) Memory Bank 8 Base Address Higher Register for CRU Statistics Data
(CRUm_AMnSDBM8ADDRH) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0178h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMB8ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	SDMB8ADDRH [34:32]		0h		RW		Statistics data transfer destination base address 8 (valid bits: [34:32]) (These bits cannot be modified while the AXI-SD is operating.)									

(67) Memory Bank Enable Register for CRU Statistics Data (CRUm_AMnSDMBVALID) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 018Ch

Initial Value : 0000_0001h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	SDMBVALID[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7 to 0	SDMBVALID [7:0]	1h	RW	Statistics data transfer destination valid address (the details are described in 9.2.4.3.3 Transfer to DRAM.) [0]: 0: SDMB1 invalid, 1: SDMB1 valid [1]: 0: SDMB2 invalid, 1: SDMB2 valid [2]: 0: SDMB3 invalid, 1: SDMB3 valid [3]: 0: SDMB4 invalid, 1: SDMB4 valid [4]: 0: SDMB5 invalid, 1: SDMB5 valid [5]: 0: SDMB6 invalid, 1: SDMB6 valid [6]: 0: SDMB7 invalid, 1: SDMB7 valid [7]: 0: SDMB8 invalid, 1: SDMB8 valid Note: Set the valid destination address from the LSB continuously. Data transferred from the LSB are written sequentially only to valid memory banks, and the processing is repeated from the start (LSB) when the last address is reached. (These bits cannot be modified while the AXI-SD is operating.)

(68) Memory Bank Status Register for CRU Statistics Data (CRUm_AMnSDMBS) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0190h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDMBSTS[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	SDMBSTS [2:0]		0h		R		Indicates which memory bank from SDMB1 to SDMB8 the statistics data are being transferred to. (0h: SDMB1, 1h: SDMB2 ... 6h: SDMB7, 7h: SDMB8)									

(69) AXI-SD Bus Transfer Completion Event Address 0 Lower Register (CRUm_AMnSDIVT0ADDRL)
(m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0194h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDIVT0ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDIVT0ADDRL[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 0	SDIVT0ADDRL [31:0]															
	Initial Value															
	0h															
	R/W															
	Description															
	AXI-SD bus transfer completion event address 0 for statistics data (lower)															
	Note: The completion of transfer of statistics data to the address set in this register is notified by an interrupt.															

(70) AXI-SD Bus Transfer Completion Event Address 0 Higher Register (CRUm_AMnSDIVT0ADDRH)
(m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0198h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDIVT0ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	SDIVT0ADDRH [34:32]		0h		RW		AXI-SD bus transfer completion event address 0 for statistics data (higher) Note: The completion of transfer of statistics data to the address set in this register is notified by an interrupt.									

(71) AXI-SD Bus Transfer Completion Event Address 1 Lower Register (CRUm_AMnSDIVT1ADDRL)
(m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 019Ch																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDIVT1ADDRL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDIVT1ADDRL[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 0	SDIVT1ADDRL [31:0]															
	Initial Value															
	0h															
	R/W															
	Description															
	AXI-SD bus transfer completion event address 1 for statistics data (lower)															
	Note: The completion of transfer of statistics data to the address set in this register is notified by an interrupt.															

(72) AXI-SD Bus Transfer Completion Event Address 1 Higher Register (CRUm_AMnSDIVT1ADDRH)
(m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 01A0h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SDIVT1ADDRH[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	SDIVT1ADDRH [34:32]		0h		RW		AXI-SD bus transfer completion event address 1 for statistics data (higher) Note: The completion of transfer of statistics data to the address set in this register is notified by an interrupt.									

(73) AXI-SD Bus Master FIFO Pointer Register (CRUm_AMnSDFIFOPNTR) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 01C0h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SDFIFORPNTR[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	SDFIFOWPNTR[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21 to 16	SDFIFORPNTR [5:0]	0h	R	FIFO-SD read pointer for statistics data The read pointer and write pointer comparison allows checking if there are data in the FIFO. FIFO-SD read pointer [5:0] Real RAM address for FIFO-SD (Max.: 3Fh) Note: When (AMnSDFIFOPNTR.SDFIFOWPNTR[5:0] ≠ AMnSDFIFOPNTR.SDFIFORPNTR[5:0]), data are present in the FIFO-SD.
15 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	SDFIFOWPNTR [5:0]	0h	R	FIFO-SD write pointer for statistics data The read pointer and write pointer comparison allows checking if there are data in the FIFO. Bank 1 FIFO-SD write pointer [5:0] Real RAM address for FIFO-SD (Max.: 3Fh) Note: When (AMnSDFIFOPNTR.SDFIFOWPNTR[5:0] ≠ AMnSDFIFOPNTR.SDFIFORPNTR[5:0]), data are present in the FIFO-SD.

(74) AXI-SD Bus Master Transfer Stop Register (CRUm_AMnSDAXISTP) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUM_base> + 01D8h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SD_AXI_STOP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 1	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
0	SD_AXI_STOP		0h	RW	Statistics data AXI-SD transfer stop (used to stop the CRU receiving data. Follow the procedure for stopping reception described in 9.2.5.2 Stopping Reception for the MIPI CSI-2 Input .) 0b: Does not stop the AXI-SD master transferring data. 1b: Stops the AXI-SD master transferring data.											

(75) AXI-SD Bus Master Transfer Stop Status Register (CRUm_AMnSDAXISTPACK) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 01DCh																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SD_AXI_STOP_ACK
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	SD_AXI_STOP_0h ACK		R	Statistics data AXI-SD transfer stop response (used to stop the CRU receiving data. Follow the procedure for stopping reception described in 9.2.5.2 Stopping Reception for the MIPI CSI-2 Input. 0b: The AXI-SD master is transferring data. 1b: The AXI-SD master has finished transferring data (valid when AXI_STOP = 1)

(76) CRU Image Converter Enable Register (CRUm_ICnEN) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 01F0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ICEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	ICEN		0h		RW		Enables the operation of the Image Converter module. (Enable or disable the operation according to 9.2.5 Operation .) 0b: Disables the Image Converter module. 1b: Enables the Image Converter module. Note: If this bit is set to 0b, a frame start is not detected. If this bit is changed from 1b to 0b while a frame is being processed, the data at that time is processed as the frame end and then the next frames are not processed.									

(77) CRU Data Value Processing (CRUm_ICnDTVP) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 01F4h

Initial Value : 0000_0003h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	FTON	HDON	-	IBINSEL	CLP[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5	FTON	0h	RW	Footer appending ON/OFF 0b: Do not append a footer 1b: Append a footer Note: The footers of both video data and statistics data are turned on or off. Note: For the video data footer, 256 bits (256'h1/256'h0) are written at the end of each line. Initially, 1h is written as a footer. The next time the memory bank is overwritten after one round of writing, 0h is written as the footer. After that, 1h and 0h are written alternately every time the memory bank is overwritten. Note: For the footer of statistics data, 512 bits ({higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} or 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0}) are written at the end of each frame unit. The higher-order 256 bits of 512 bits are reserved (indefinite value), so skip reading them. Initially, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} are written as a footer. Next, when the memory bank is overwritten after one round of writing, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0} are written as a footer. After that, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} and 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0} are written alternately each time the memory bank is overwritten.
4	HDON	0h	RW	Header appending ON/OFF ON/OFF selection for whether to append a header to the start of a line for data transfer from the AXI-VD bus to DRAM 0b: Do not append a header 1b: Append a header Note: The data type, byte number, and virtual channel number are added to the header (256 bits). Note: See 9.2.4.3.3 (5) Frame Structure.
3	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
2	IBINSEL	0h	RW	Data Arranger 1 UV(CbCr) binary type selection for YUV422(YCbCr) data 0b: Offset binary (straight binary) 1b: Two's complement binary Note: Valid when the input is YUV422/420.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	CLP[1:0]	3h	RW	Data clipping setting for the YCbCr422 (YUV422) format To make the input data compliant with the ITU-R BT.601 standard, specify whether to clip the values below 16 and above 240 (when YCbCr is 8 bits). When CRUm_ICnIPMC_C0/X.CSCTHR is set to 1, CLP[1:0] are set to 00b. 00b: No clipping for Y When CbCr is 8 bits: Values below 16 and above 240 are clipped for CbCr When CbCr is 10 bits: Values below 64 and above 960 are clipped for CbCr 01b: When YCbCr is 8 bits: Values below 16 and above 240 are clipped for Y : Values below 16 and above 240 are clipped for CbCr When YCbCr is 10 bits: Values below 64 and above 960 are clipped for Y : Values below 64 and above 960 are clipped for CbCr 10b: No clipping for Y When CbCr is 8 bits: Values below 16 are increased to 128 and values above 240 are clipped to 128 for CbCr When CbCr is 10 bits: Values below 64 are increased to 512 and values above 960 are clipped to 512 for CbCr 11b: No clipping for Y and CbCr (initial value) Note: See 9.2.4.3.11 Data Clipping.

(78) CRU SVC Number Register (CRUm_ICnSVCNUM) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 01F8h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SVCNUM [1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	SVCNUM[1:0]	0h	RW	Set how many channels are used from among SVC0 to 3. 00b: Only SVC0 enabled; SVC1 to 3 disabled (initial value) 01b: SVC0 and 1 enabled; SVC2 and 3 disabled 10b: SVC0 to 2 enabled; SVC3 disabled 11b: SVC0 to 3 enabled Note: When changing the frame synchronization register, only the setting of ICnSVCNUM.SVCNUM [1:0] = 00 can be used. Note: Select virtual channels for SVC0 to 3 by the ICnSVC register setting. Note: When using demosaicing, linear matrix, or statistics, the setting of this must be 00b.

(79) CRU VC Select Register (CRUm_ICnSVC) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 01FCh

Initial Value : 0000_3210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVC3[3:0]				SVC2[3:0]				SVC1[3:0]				SVC0[3:0]			
Initial Value	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 12	SVC3[3:0]	3h	RW	Selected Virtual Channel 3 (SVC3) The maximum number of channels that can be used by the image converter is 4 (SVC0~3). Assign one of VC0~15 to SVC3. 0h: Virtual channel 0 selected 1h: Virtual channel 1 selected 2h: Virtual channel 2 selected : Eh: Virtual channel 14 selected Fh: Virtual channel 15 selected Note: Do not set the same value as other SVC0/1/2[3:0].
11 to 8	SVC2[3:0]	2h	RW	Selected Virtual Channel 2 (SVC2) The maximum number of channels that can be used by the image converter is 4 (SVC0~3). Assign one of VC0~15 to SVC2. 0h: Virtual channel 0 selected 1h: Virtual channel 1 selected 2h: Virtual channel 2 selected : Eh: Virtual channel 14 selected Fh: Virtual channel 15 selected Note: Do not set the same value as other SVC0/1/3[3:0].
7 to 4	SVC1[3:0]	1h	RW	Selected Virtual Channel 1 (SVC1) The maximum number of channels that can be used by the image converter is 4 (SVC0~3). Assign one of VC0~15 to SVC1. 0h: Virtual channel 0 selected 1h: Virtual channel 1 selected 2h: Virtual channel 2 selected : Eh: Virtual channel 14 selected Fh: Virtual channel 15 selected Note: Do not set the same value as other SVC0/2/3[3:0].
3 to 0	SVC0[3:0]	0h	RW	Selected Virtual Channel 0 (SVC0) The maximum number of channels that can be used by the image converter is 4 (SVC0~3). Assign one of VC0~15 to SVC0. 0h: Virtual channel 0 selected 1h: Virtual channel 1 selected 2h: Virtual channel 2 selected : Eh: Virtual channel 14 selected Fh: Virtual channel 15 selected Note: Do not set the same value as other SVC1/2/3[3:0].

(80) CRU Image Converter Main Control Register (SVC0) (CRUm_ICnIPMC_C0) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0200h

Initial Value : 001E_00FEh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	RAWSTTYP [1:0]		-	-	INF[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
R/W	R	R	R	R	R	R	RW	RW	R	R	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	STITHR	LUTTHR	CSCTH	LMXTH	DEMTHT	CLPTH	DECTHT	-
Initial Value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25, 24	RAWSTTYP [1:0]	0h	RW	Specify the color component of the first pixel of the first line of RAW data for SVC0 00b: R (RGrGrGr...) 01b: Gr (GrGrGrR...) 10b: Gb (GbBgBgB...) 11b: B (BgBgBgB...)
23, 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21 to 16	INF[5:0]	1Eh	RW	Specify the MIPI input image formats for SVC0 to be processed by the image converter module. For details about the formats, see Data Type Codes in the MIPI CSI-2 V2.1 specification. Of the formats specified here, only one format is processed. Note 1. Initial value: YUV422 8-bit Note 2. The following values can be set. Other settings are prohibited. 13h to 16h: Generic Long Packet Data Type 18h to 1Ah, 1Ch to 1Fh: YUV Data 20h to 24h: RGB Data 28h to 2Fh: RAW Data 30h to 37h: User-Defined Byte-based Data The data types specified here must be set in CSI2nDTEL.DTEN and CSI2nDTEH.DTEN of MIPI CSI2.
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7	STITHR	1h	RW	Bypassing statistics processing for SVC0 (the details are described in 9.2.4.3.19 Statistics .) 0b: Statistics processing is performed. 1b: Statistics processing is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing .
6	LUTTHR	1h	RW	Bypassing LUT-based image color conversion for SVC0 Regardless of whether the setting of this bit is 0b or 1b, the output of the LUT is 8 bits for 10-bit input. (the details are described in 9.2.4.3.15 LUT (Lookup Table) .) 0b: Image color conversion is performed. 1b: Image color conversion is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing .

Bit	Bit Name	Initial Value	R/W	Description
5	CSCTHR	1h	RW	Bypassing color space conversion and Data clipping for SVC0 (YCbCr(YUV) → RGB, RGB → YCbCr(YUV)) (the details are described in 9.2.4.3.14 Color Space Conversion.) 0b: Color space conversion and Data clipping are performed. 1b: Color space conversion and Data clipping are not performed (bypassed). Note 1. The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing. Note 2. Even when this bit is 1, the data is expanded from 8 bits to 10 bits.
4	LMXTHR	1h	RW	Bypassing liner matrix processing for SVC0 (the details are described in 9.2.4.3.13 Liner Matrix.) 0b: Liner matrix processing is performed. 1b: Liner matrix processing is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing1.
3	DEMTHR	1h	RW	Bypassing demosaicing of RAW data for SVC0 (the details are described in Section 9.2.4.3.12 Demosaicing.) 0b: Demosaicing is performed. 1b: Demosaicing is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing.
2	CLPTHR	1h	RW	Bypassing image clipping for SVC0 (the details are described in 9.2.4.3.8 Image Clipping.) 0b: Image clipping is performed. 1b: Image clipping is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing.
1	DECTHR	1h	RW	Bypassing frame subsampling for SVC0 (the details are described in 9.2.4.3.9 Frame Subsampling.) 0b: Subsampling is performed. 1b: Subsampling is not performed (bypassed). Note: Image processing, non-image processing, and statistics data are supported. Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing.
0	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(81) CRU Image Converter Main Control Register (SVCx) (CRUm_ICnIPMC_Cx) (m = 0, 1)

Note: x = 1 to 3 CRU Non Image Process Data Type Code Select Lower Register

Access Size : 32 bits

Address : <CRUm_base> + 0258h (x = 1)
<CRUm_base> + 025Ch (x = 2)
<CRUm_base> + 0260h (x = 3)

Initial Value : 001E_0066h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	INF[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
R/W	R	R	R	R	R	R	RW	RW	R	R	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	LUTTH R	CSCTH R	-	-	CLPTH R	DECTH R	-
Initial Value	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25, 24	-	All 0	RW	Reserved These bits are read as 0b. The write value should always be 0b.
23, 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21 to 16	INF[5:0]	1Eh	RW	Specify the MIPI input image formats for SVCx to be processed by the image converter module. For details about the formats, see Data Type Codes in the MIPI CSI-2 V2.1 specification. Of the formats specified here, only one format is processed. Note 1. Initial value: YUV422 8-bit Note 2. The following values can be set. Other settings are prohibited. 13h to 16h: Generic Long Packet Data Type 18h to 1Ah, 1Ch to 1Fh: YUV Data 20h to 24h: RGB Data 28h to 2Fh: RAW Data 30h to 37h: User-Defined Byte-based Data The data types specified here must be set in CSI2nDTEL.DTEN and CSI2nDTEH.DTEN of MIPI CSI2.
15 to 7	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6	LUTTHR	1h	RW	Bypassing LUT-based image color conversion for SVCx (the details are described in 9.2.4.3.15 LUT (Lookup Table) .) Regardless of whether the setting of this bit is 0b or 1b, the output of the LUT is 8 bits for 10-bit input. 0b: Image color conversion is performed. 1b: Image color conversion is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing .
5	CSCTHR	1h	RW	Bypassing color space conversion and Data clipping for SVCx (YCbCr(YUV) → RGB, RGB → YCbCr(YUV)) (the details are described in 9.2.4.3.14 Color Space Conversion .) 0b: Color space conversion and data clipping are performed. 1b: Color space conversion and data clipping are not performed (bypassed). Note1. The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing . Note2. Even when this bit is 1, the data is expanded from 8 bits to 10 bits.
4, 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
2	CLPTHR	1h	RW	Bypassing image clipping for SVCx (the details are described in 9.2.4.3.8 Image Clipping .) 0b: Image clipping is performed. 1b: Image clipping is not performed (bypassed). Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing .
1	DECTHR	1h	RW	Bypassing frame subsampling for SVCx (the details are described in 9.2.4.3.9 Frame Subsampling .) 0b: Subsampling is performed. 1b: Subsampling is not performed (bypassed). Note: Image processing, non-image processing, and statistics data are supported. Note: The setting 0b only applies for the corresponding formats shown in 9.2.4.3.1 Input Formats and Image Processing .
0	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(82) CRU Non Image Process Data Type Code Select Lower Register (SVCx) (CRUm_ICnNIPDT_CxL)
(m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0204h (x = 0) <CRUm_base> + 0264h (x = 1) <CRUm_base> + 026Ch (x = 2) <CRUm_base> + 0274h (x = 3)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NIPDT_CxL[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	NIPDT_CxL [31:16]	0h	RW	Non-image processing/full-bypass data type code selection (lower) selected by SVCx Each bit number in NIPDT_CxL [31:16] matches the decimal value of the data type number specified in MIPI CSI-2 v2.1. Select whether to send the data type corresponding to that number. NIPDT_CxL Corresponding data types [16] 10h [17] 11h ⋮ ⋮ [31] 1Fh When each bit is 0b: Do not send data of the corresponding data type as SVCx. When each bit is 1b: Data of the corresponding data type is transmitted as SVCx.
15 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(83) CRU Non Image Process Data Type Code Select Higher Register (SVCx) (CRUm_ICnNIPDT_CxH)
(m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0208h (x = 0) <CRUm_base> + 0268h (x = 1) <CRUm_base> + 0270h (x = 2) <CRUm_base> + 0278h (x = 3)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NIPDT_CxH[63:48]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIPDT_CxH[47:32]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	NIPDT_CxH [63:32]	0h	RW	Non-image processing/full-bypass data type code selection (higher) selected by SVCx Each bit number in NIPDT_CxH [63:32] matches the decimal value of the data type number specified in MIPI CSI-2 v2.1. Select whether to send the data type corresponding to that number. NIPDT_CxH Corresponding data types [32] 20h [33] 21h ⋮ ⋮ [63] 3Fh When each bit is 0b: Do not send data of the corresponding data type as SVCx. When each bit is 1b: Data of the corresponding data type is transmitted as SVCx.

(84) CRU Image Clipping Start Line Register (SVCx) (CRUm_ICnSLPrC_Cx) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 020Ch (x = 0)
 <CRUm_base> + 028Ch (x = 1)
 <CRUm_base> + 02A0h (x = 2)
 <CRUm_base> + 02B4h (x = 3)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SLPrC[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	SLPrC[12:0]	0h	RW	Start line for SVCx in the vertical direction when clipping the input image to the specified area. Set a value of 1 or greater because the frame start is processed as the first line. The image is output from the specified line. (If 0 is specified, the same operation as for setting of 1 is performed.) Note: Set a value such that $1 \leq \text{ICnSLPrC\_Cx.SLPrC} \leq \text{ICnELPrC\_Cx.ELPrC}$. Note: Only image processing data are supported.

(85) CRU Image Clipping End Line Register (SVCx) (CRUm_ICnELPrC_Cx) (m = 0, 1)**Access Size :** 32 bits

Address : <CRUm_base> + 0210h (x = 0)
 <CRUm_base> + 0290h (x = 1)
 <CRUm_base> + 02A4h (x = 2)
 <CRUm_base> + 02B8h (x = 3)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	ELPrC[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	ELPrC[12:0]	0h	RW	End line for SVCx in the vertical direction when clipping the input image to the specified area. Set a value of 1 or greater because the frame start is processed as the first line. The image is output up to the specified line. Note: Set a value such that $1 \leq \text{ICnSLPrC\_Cx.SLPrC} \leq \text{ICnELPrC\_Cx.ELPrC}$. Note: Only image processing data are supported.

(86) CRU Image Clipping Start Pixel Register (SVCx) (CRUm_ICnSPPrC_Cx) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0214h (x = 0)
 <CRUm_base> + 0294h (x = 1)
 <CRUm_base> + 02A8h (x = 2)
 <CRUm_base> + 02BCh (x = 3)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SPPrC[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	SPPrC[12:0]	0h	RW	Start pixel for SVCx in the horizontal direction when clipping the input image to the specified area. Set a value of 1 or greater because the line start is processed as the first pixel. The image is output from the specified pixel data. (If 0 is specified, the same operation as for setting of 1 is performed.) Note 1. Set a value such that $1 \leq \text{ICnSPPrC\_Cx.SPPrC} \leq (\text{ICnEPPrC\_Cx.EPPrC} + 1 - 320)$. Note 2. Specify an odd value when the input format is YUV422 (ICnIPMC_Cx.INF[5:0] = 1Eh to 1Fh). If an even number of 2 or greater is specified, the CRU operates as if an odd number (the specified number - 1) is specified. (Example: When 34 is specified, the data is output from the 33rd pixel.) Note 3. Even when the input format is RGB, etc., set a value that satisfies the following formula so that a line consists of an even number of pixels if the RGBtoYCbCr output is YUV422/420. $\text{ICnEPPrC\_Cx.EPPrC} - \text{ICnSPPrC\_Cx.SPPrC} = 2N - 1$ (N = 160, 161, 162,...) Note: Only image processing data are supported.

(87) CRU Image Clipping End Pixel Register (SVCx) (CRUm_ICnEPPrC_Cx) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0218h (x = 0)
 <CRUm_base> + 0298h (x = 1)
 <CRUm_base> + 02ACh (x = 2)
 <CRUm_base> + 02C0h (x = 3)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	EPPrC[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	EPPrC[12:0]	0h	RW	End pixel for SVCx in the horizontal direction when clipping the input image to the specified area. Set a value of 1 or greater because the line start is processed as the first pixel. The image is output up to the specified image data. Note 1. Set a value such that $1 \leq \text{ICnSPPrC\_Cx.SPPrC} \leq (\text{ICnEPPrC\_Cx.EPPrC} + 1 - 320)$. Note 2. Specify an even value when the input format is YUV422 (ICnIPMC_Cx.INF[5:0] = 1Eh to 1Fh). If an odd number is specified, the CRU operates as if an even number (the specified number + 1) is specified. (Example: When 127 is specified, the data is output up to the 128th pixel.) Note 3. Even when the input format is RGB, etc., set a value that satisfies the following formula so that a line consists of an even number of pixels if the RGBtoYCbCr output is YUV422/420. $\text{ICnEPPrC\_Cx.EPPrC} - \text{ICnSPPrC\_Cx.SPPrC} = 2N - 1$ (N = 160, 161, 162,...) Note: Only image processing data are supported.

(88) CRU Scan Line Interrupt Register (SVCx) (CRUm_ICnSI_Cx) (m = 0, 1)**Access Size :** 32 bits

Address : <CRUm_base> + 021Ch (x = 0)
 <CRUm_base> + 029Ch (x = 1)
 <CRUm_base> + 02B0h (x = 2)
 <CRUm_base> + 02C4h (x = 3)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SI[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	SI[12:0]	0h	RW	Line specification for generating a scan line interrupt (for SVCx) When the line number being processed in a frame of the Image Converter module matches the line setting value specified by this bit, an image_conv_int interrupt can be generated. Setting range: 1h to 1000h Note: By using the ICnINTCTRL.DECINTE bit, whether to generate interrupts in subsampled frames can be specified. Note: Only image processing data are supported.

(89) CRU Linear Matrix offset Register (SVC0 only) (CRUm_ICnLMXOF) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0220h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	BOF[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GOF[7:0]								ROF[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	BOF[7:0]	0h	RW	B offset value of linear matrix calculation for SVC0 (bit7: sign) Note: Setting range: -128 (80h) to 127 (7Fh) Note: Only image processing data are supported.
15 to 8	GOF[7:0]	0h	RW	G offset value of linear matrix calculation for SVC0 (bit7: sign) Note: Setting range: -128 (80h) to 127 (7Fh) Note: Only image processing data are supported.
7 to 0	ROF[7:0]	0h	RW	R offset value of linear matrix calculation for SVC0 (bit7: sign) Note: Setting range: -128 (80h) to 127 (7Fh) Note: Only image processing data are supported.

(90) CRU Linear Matrix R coefficient 1 Register (SVC0 only) (CRUm_ICnLMXRC1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0224h														
Initial Value :		0000_0400h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	RR[12:0]												
Initial Value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	RR[12:0]		400h	RW	Coefficient R of R data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.											

(91) CRU Linear Matrix R coefficient 2 Register (SVC0 only) (CRUm_ICnLMXRC2) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0228h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	RB[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	RG[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	RB[12:0]	0h	RW	Coefficient B of R data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	RG[12:0]	0h	RW	Coefficient G of R data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.

(92) CRU Linear Matrix G coefficient 1 Register (SVC0 only) (CRUm_ICnLMXGC1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 022Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	GR[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	GR[12:0]		0h	RW	Coefficient R of G data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.											

(93) CRU Linear Matrix G coefficient 2 Register (SVC0 only) (CRUm_ICnLMXGC2) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0230h														
Initial Value :		0000_0400h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	GB[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	GG[12:0]												
Initial Value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	GB[12:0]	0h	RW	Coefficient B of G data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	GG[12:0]	400h	RW	Coefficient G of G data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.

(94) CRU Linear Matrix B coefficient 1 Register (SVC0 only) (CRUm_ICnLMXBC1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0234h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BR[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	BR[12:0]	0h	RW	Coefficient R of B data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.

(95) CRU Linear Matrix B coefficient 2 Register (SVC0 only) (CRUm_ICnLMXBC2) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0238h														
Initial Value :		0400_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BB[12:0]												
Initial Value	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BG[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	BB[12:0]		400h	RW	Coefficient B of B data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.											
15 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	BG[12:0]		0h	RW	Coefficient G of B data for SVC0 linear matrix calculation (bit12: sign) Note: Specify a signed 13-bit integer by multiplying the desired coefficient value (-4 to approx. 4) by 1024 Setting range: -4096 (1000h) to 4095 (0FFFh) Note: Only image processing data are supported.											

(96) CRU Statistics Control 1 Register (SVC0 only) (CRUm_ICnSTIC1) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 023Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	STSADPOS[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STUNIT[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 16	STSADPOS [3:0]	0h	RW	Input data bit position selection (0-8) for the sum of absolute values for adjacent pixels for statistics processing of SVC0 Use the 8 lower-order bits from the MSB-"n" bit for each type of RAW. Example: For RAW 16, use input data [15-n:8-n]. Note: Comply with the restrictions described in ICnSTIC1.STUNIT. Note: Only image processing data are supported.
15 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	STUNIT[1:0]	0h	RW	Selecting an arithmetic processing unit for SVC0 statistics processing 0: 16x16 pixels 1: 32x32 pixels 2: 64x64 pixels 3: 128x128 pixels Note: Only image processing data are supported.

(97) CRU Image Converter Register Setting Change Control Register (CRUm_ICnREGC) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0244h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	REFEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	REFEN	0h	RW1	Allows register setting changes to be reflected. By using this register for an MIPI CSI-2 input, the registers at offset addresses 01FCh to 023Ch can be changed through frame synchronization during operation. (Allow or disallow this according to 9.2.5.3 Frame Synchronization Settings .) 0b: Does not allow register setting changes to be reflected. 1b: Allows register setting changes to be reflected. Note: If this bit is set to 1 while the Image Converter module is operating (ICnEN.ICEN = 1), the register setting values (at offset addresses 01FCh to 023Ch) are reflected at a frame start, and then the bit is cleared to 0 automatically. Note: If the Image Converter module is stopped (ICnEN.ICEN = 0), the register setting values can be reflected irrespective of this register. Note that registers other than the above cannot be changed during operation. Note: Only image processing data are supported.

(98) CRU Module Status Register (CRUm_ICnMS) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 02D8h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IA3	AV3	CA3	-	IA2	AV2	CA2	-	IA1	AV1	CA1	-	IA0	AV0	CA0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14	IA3	0h	R	Indicates whether the Image Converter block (the module converting the video data input up to the pre-FIFO-VD/SD stage) is processing data. (for SVC3) 0b: Video/statistics data have not been received. 1b: Video/statistics data are being converted.
13	AV3	0h	R	Indicates whether processing is performed within the image-clipped area. (for SVC3) (For details, see 9.2.4.3.8 Image Clipping .) 0b: Processing is performed outside the image-clipped area. 1b: Processing is performed within the image-clipped area. Note: Only image processing data are supported.
12	CA3	0h	R	Indicates whether the frame is subsampled. (for SVC3) (For details, see 9.2.4.3.9 Frame Subsampling .) 0b: A subsampled frame is being processed. 1b: A frame which is not subsampled is being processed. Note: This bit is 0 when CRUnRST.VRESETN is 0. This bit is set to 1 two cycles of PCLK after CRUnRST.VRESETN has been changed from 0 to 1. Note: Only image processing data are supported.
11	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
10	IA2	0h	R	Indicates whether the Image Converter block (the module converting the video data input up to the pre-FIFO-VD/SD stage) is processing data. (for SVC2) 0b: Video/statistics data have not been received. 1b: Video/statistics data are being converted.
9	AV2	0h	R	Indicates whether processing is performed within the image-clipped area. (for SVC2) (For details, see 9.2.4.3.8 Image Clipping .) 0b: Processing is performed outside the image-clipped area. 1b: Processing is performed within the image-clipped area. Note: Only image processing data are supported.
8	CA2	0h	R	Indicates whether the frame is subsampled. (for SVC2) (For details, see 9.2.4.3.9 Frame Subsampling .) 0b: A subsampled frame is being processed. 1b: A frame which is not subsampled is being processed. Note: This bit is 0 when CRUnRST.VRESETN is 0. This bit is set to 1 two cycles of PCLK after CRUnRST.VRESETN has been changed from 0 to 1.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6	IA1	0h	R	Indicates whether the Image Converter block (the module converting the video data input up to the pre-FIFO-VD/SD stage) is processing data. (for SVC1) 0b: Video/statistics data have not been received. 1b: Video/statistics data are being converted. Note: Only image processing data are supported.

Bit	Bit Name	Initial Value	R/W	Description
5	AV1	0h	R	Indicates whether processing is performed within the image-clipped area. (for SVC1) (For details, see 9.2.4.3.8 Image Clipping .) 0b: Processing is performed outside the image-clipped area. 1b: Processing is performed within the image-clipped area.
4	CA1	0h	R	Indicates whether the frame is subsampled. (for SVC1) (For details, see 9.2.4.3.9 Frame Subsampling .) 0b: A subsampled frame is being processed. 1b: A frame which is not subsampled is being processed. Note: This bit is 0 when CRUnRST.VRESETN is 0. This bit is set to 1 two cycles of PCLK after CRUnRST.VRESETN has been changed from 0 to 1.
3	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	IA0	0h	R	Indicates whether the Image Converter block (the module converting the video data input up to the pre-FIFO-VD/SD stage) is processing data. (for SVC0) 0b: Video/statistics data have not been received. 1b: Video/statistics data are being converted.
1	AV0	0h	R	Indicates whether processing is performed within the image-clipped area. (for SVC0) (For details, see 9.2.4.3.8 Image Clipping .) 0b: Processing is performed outside the image-clipped area. 1b: Processing is performed within the image-clipped area. Note: Only image processing data are supported.
0	CA0	0h	R	Indicates whether the frame is subsampled. (for SVC0) (For details, see 9.2.4.3.9 Frame Subsampling .) 0b: A subsampled frame is being processed. 1b: A frame which is not subsampled is being processed. Note: This bit is 0 when CRUnRST.VRESETN is 0. This bit is set to 1 two cycles of PCLK after CRUnRST.VRESETN has been changed from 0 to 1.

(99) CRU Frame Subsampling Control Register (CRUm_ICnDEC) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 02DCh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	FRMDEC[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	FRMDEC[3:0]	0h	RW	Subsampling interval in frame subsampling (For details, see 9.2.4.3.9 Frame Subsampling.) 0h: All images are passed. 1h: One pass every two frames 2h: One pass every three frames Omitted 14h: One pass every 15 frames 15h: One pass every 16 frames Note: Image processing and non-image processing and statistics data are supported.

(100) CRU Line Count Register (SVCx) (CRUm_ICnLC_Cx) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 02E0h (x = 0) <CRUm_base> + 02E4h (x = 1) <CRUm_base> + 02E8h (x = 2) <CRUm_base> + 02ECh (x = 3)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	LC[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	LC[12:0]	0h	R	Indicate the line counter value before image clipping and subsampling by the Image Converter module. (for SVCx) Note: Only image processing data are supported. Note: The maximum line value is 1000h.

(101) CRU Frame Subsampling Interrupt Control Register (CRUm_ICnINTCTRL) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0300h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DECINT E
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	DECINTE	0h	RW	Specifies whether an interrupt (CRUnINTS2.FS0/1/2/3S, CRUnINTS2.FE0/1/2/3S, or CRUnINTS2.SL0/1/2/3S) is generated for a subsampled frame. 0b: No interrupt is generated for a subsampled frame. (Initial value) 1b: An interrupt is generated for a subsampled frame. Note: Image processing and non-image processing and statistics data are supported. Note: The maximum line value is 1000h.

(102) CRU Output Image Format Register (CRUm_ICnDMR) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0304h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	A8BIT[7:0]								-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	OBINSEL	-	YCMODE[2:0]			-	-	RGBMODE[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	RW	RW	RW	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	A8BIT[7:0]	0h	RW	Alpha bit for AXI-VD bus Alpha value in the ARGB8888 format output Note: The default value is 0h, so it is transparent. Note: This bit is valid when ICnDMR.RGBMODE[1:0]=10b,11b.
23 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	OBINSEL	0h	RW	Data Arranger 2 UV(CbCr) binary type selection for YUV422(YCbCr) data 0b: Offset binary (straight binary) 1b: Two's complement binary Note: Valid when the output is YUV422/420.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 4	YCMODE[2:0]	0h	RW	YC output data format selection for AXI-VD bus 000b: YCbCr422, 8-bit multiplexed (YUYV format (Y 1st)) 001b: YCbCr422, 8-bit multiplexed (UYVY format (Cb 1st)) 010b: YC separation from YCbCr422 011b: Y component extraction from YCbCr422 100b: YCbCr420, 8-bit multiplexed (YUYV format (Y 1st)) 101b: YCbCr420, 8-bit multiplexed (UYVY format (Cb 1st)) 110b: YC Separation of YCbCr420. *The CbCr(U/V) side is Image Stride (AMnIS) ×2. 111b: Y component extraction of YCbCr420 (equivalent to YUV422)
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	RGBMODE[1:0]	0h	RW	RGB output data format selection for AXI-VD bus 00b: RGB888 (24-bit) 01b: RGB888 (32-bit) (The eight higher-order bits are padded with 00h.) 10b: ARGB888 (32-bit, B First A Last) 11b: ARGB888 (32-bit, A First B Last)

(103) CRU YCbCr → RGB Color Space Conversion Coefficient 1 Register (CRUm_ICnSCC1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0308h														
Initial Value :		0000_129Fh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	YMUL[13:0]													
Initial Value	0	0	0	1	0	0	1	0	1	0	0	1	1	1	1	1
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 14	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
13 to 0	YMUL[13:0]		129Fh	RW	Y data multiplication coefficient These bits specify a Y data multiplication coefficient in YCbCr → RGB color space conversion. (Initial value: 1.164) Specify an unsigned 14-bit integer which is the desired coefficient value multiplied by 4096.											

(104) CRU YCbCr → RGB Color Space Conversion Coefficient 2 Register (CRUm_ICnSCC2) (m = 0, 1)

Access Size :		32 bits																
Address :		<CRUm_base> + 030Ch																
Initial Value :		0100_0800h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	YSUB[11:0]													
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	CSUB[11:0]													
Initial Value	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 16	YSUB[11:0]	100h	RW	Y data subtraction coefficient These bits specify a Y data subtraction coefficient in YCbCr → RGB color space conversion. Specify an unsigned 12-bit integer. Calculate on 12 bits extended from YCbCr422 8-/10-bit.
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	CSUB[11:0]	800h	RW	CbCr data subtraction coefficient These bits specify a Cb and Cr data subtraction coefficient in YCbCr → RGB color space conversion. Specify an unsigned 12-bit integer. Calculate on 12 bits extended from YCbCr422 8-/10-bit.

(105) CRU YCbCr → RGB Color Space Conversion Coefficient 3 Register (CRUm_ICnSCC3) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0310h														
Initial Value :		1989_0D02h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	RCRMUL[13:0]													
Initial Value	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	GCRMUL[13:0]													
Initial Value	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 16	RCRMUL [13:0]	1989h	RW	Cr multiplication coefficient in R data calculation These bits specify a Cr multiplication coefficient for the R data equation in YCbCr → RGB color space conversion. (Initial value: 1.596) Specify an unsigned 14-bit integer which is the desired coefficient value multiplied by 4096.
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 0	GCRMUL [13:0]	0D02h	RW	Cr multiplication coefficient in G data calculation These bits specify a Cr multiplication coefficient for the G data equation in YCbCr → RGB color space conversion. (Initial value: 0.813) Specify an unsigned 14-bit integer which is the desired coefficient value multiplied by 4096.

(106) CRU YCbCr → RGB Color Space Conversion Coefficient 4 Register (CRUm_ICnSCC4) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0314h														
Initial Value :		0645_2045h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	GCBMUL[13:0]													
Initial Value	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	BCBMUL[13:0]													
Initial Value	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 16	GCBMUL[13:0]	0645h	RW	Cb multiplication coefficient in G data calculation These bits specify a Cb multiplication coefficient for the G data equation in YCbCr → RGB color space conversion. (Initial value: 0.392) Specify an unsigned 14-bit integer which is the desired coefficient value multiplied by 4096.
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 0	BCBMUL[13:0]	2045h	RW	Cb multiplication coefficient in B data calculation These bits specify a Cb multiplication coefficient for the B data equation in YCbCr → RGB color space conversion. (Initial value: 2.017) Specify an unsigned 14-bit integer which is the desired coefficient value multiplied by 4096.

(107) CRU RGB → YCbCr Color Space Conversion Y Coefficient 1 Register (CRUm_ICnYCCR1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0318h														
Initial Value :		0000_0107h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	YCLRP[12:0]												
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	YCLRP[12:0]	0107h	RW	R multiplication coefficient in Y calculation (R multiplication coefficient for the Y data equation in RGB → YCbCr color space conversion)

(108) CRU RGB → YCbCr Color Space Conversion Y Coefficient 2 Register (CRUm_ICnYCCR2) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 031Ch														
Initial Value :		0064_0204h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	YCLBP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	YCLGP[12:0]												
Initial Value	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	YCLBP[12:0]	0064h	RW	B multiplication coefficient in Y calculation (B multiplication coefficient for the Y data equation in RGB → YCbCr color space conversion)
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	YCLGP[12:0]	0204h	RW	G multiplication coefficient in Y calculation (G multiplication coefficient for the Y data equation in RGB → YCbCr color space conversion)

(109) CRU RGB → YCbCr Color Space Conversion Y Coefficient 3 Register (CRUm_ICnYCCR3) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0320h

Initial Value : 0A00_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	YCLSFT[4:0]					-	-	-	-	-	-	-	YCLCEN
Initial Value	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW

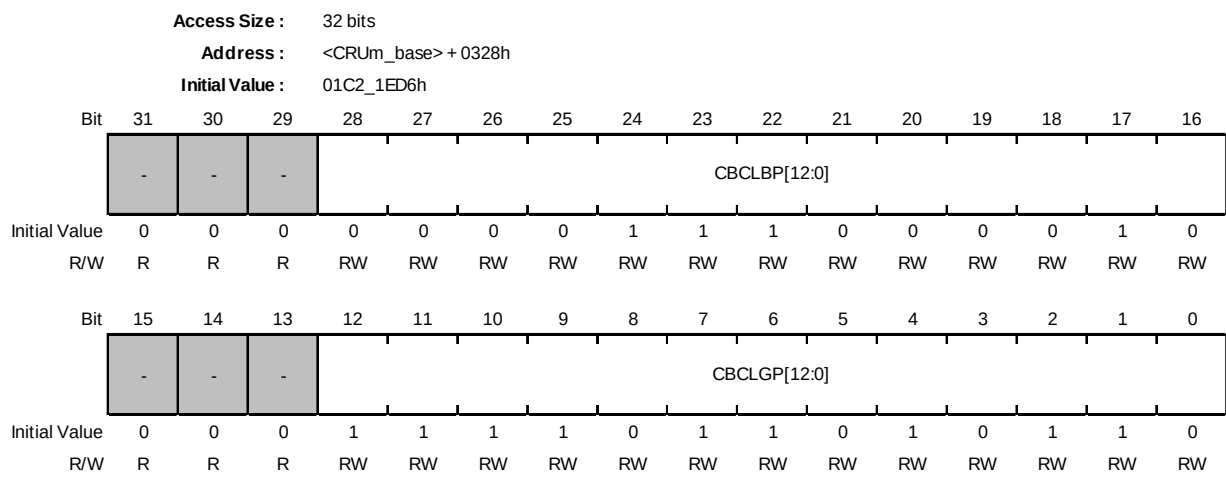
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	YCLAP[11:0]											
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
30, 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 24	YCLSFT[4:0]	0Ah	RW	Amount of downward shift in Y calculation (specification of the amount of downward shift for Y calculation in RGB → YCbCr color space conversion)
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	YCLCEN	0h	RW	Y Calculation Data Clip Enable (Data Clipping Process Control for Y Data Calculation in RGB → YCbCr Color Space Conversion) 0b: Data clipping is not performed 1b: Data clipping is performed
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	YCLAP[11:0]	100h	RW	Value added for normalization of Y calculation data (specification of Y data addition constant in RGB → YCbCr color space conversion)

(110) CRU RGB → YCbCr Color Space Conversion Cb Coefficient 1 Register (CRUm_ICnCBCCR1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0324h														
Initial Value :		0000_1F68h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	CBCLRP[12:0]												
Initial Value	0	0	0	1	1	1	1	1	0	1	1	0	1	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 13	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
12 to 0	CBCLRP[12:0]		1F68h		RW		R multiplication coefficient in Cb calculation (R multiplication coefficient for the Cb data equation in RGB → YCbCr color space conversion)									

(111) CRU RGB → YCbCr Color Space Conversion Cb Coefficient 2 Register (CRUm_ICnCBCCR2) (m = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	CBCLBP[12:0]	01C2h	RW	B multiplication coefficient in Cb calculation (B multiplication coefficient for the Cb data equation in RGB → YCbCr color space conversion)
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	CBCLGP[12:0]	1ED6h	RW	G multiplication coefficient in Cb calculation (G multiplication coefficient for the Cb data equation in RGB → YCbCr color space conversion)

(112) CRU RGB → YCbCr Color Space Conversion Cb Coefficient 3 Register (CRUm_ICnCBCCR3) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 032Ch

Initial Value : 0A00_0800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	CBCLSFT[4:0]					-	-	-	-	-	-	-	CBCLCEN
Initial Value	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	CBCLAP[11:0]											
Initial Value	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
30, 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 24	CBCLSFT[4:0]	0Ah	RW	Amount of downward shift in Cb calculation (specification of the amount of downward shift for Cb calculation in RGB → YCbCr color space conversion)
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	CBCLCEN	0h	RW	Cb Calculation Data Clip Enable (Data Clipping Process Control for Cb Data Calculation in RGB → YCbCr Color Space Conversion) 0b: Data clipping is not performed 1b: Data clipping is performed
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	CBCLAP[11:0]	800h	RW	Value added for normalization of Cb calculation data (specification of Cb data addition constant in RGB → YCbCr color space conversion)

(113) CRU RGB → YCbCr Color Space Conversion Cr Coefficient 1 Register (CRUm_ICnCRCCR1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0330h														
Initial Value :		0000_01C2h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	CRCLRP[12:0]												
Initial Value	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	CRCLRP[12:0]	01C2h	RW	R multiplication coefficient in Cr calculation (R multiplication coefficient for the Cr data equation in RGB → YCbCr color space conversion)

(114) CRU RGB → YCbCr Color Space Conversion Cr Coefficient 2 Register (CRUm_ICnCRCCR2) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0334h														
Initial Value :		1FB7_1E87h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	CRCLBP[12:0]												
Initial Value	0	0	0	1	1	1	1	1	1	0	1	1	0	1	1	1
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	CRCLGP[12:0]												
Initial Value	0	0	0	1	1	1	1	0	1	0	0	0	0	1	1	1
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	CRCLBP[12:0]	1FB7h	RW	B multiplication coefficient in Cr calculation (B multiplication coefficient for the Cr data equation in RGB → YCbCr color space conversion)
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	CRCLGP[12:0]	1E87h	RW	G multiplication coefficient in Cr calculation (G multiplication coefficient for the Cr data equation in RGB → YCbCr color space conversion)

(115) CRU RGB → YCbCr Color Space Conversion Cr Coefficient 3 Register (CRUm_ICnCRCCR3) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0338h

Initial Value : 0A00_0800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	CRCLSFT[4:0]					-	-	-	-	-	-	-	CRCLC EN
Initial Value	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	CRCLAP[11:0]											
Initial Value	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	RW	Reserved A register that can be read and written. Do not change the value of this bit from the initial value.
30, 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 24	CRCLSFT[4:0]	0Ah	RW	Amount of downward shift in Cr calculation (specification of the amount of downward shift for Cr calculation in RGB → YCbCr color space conversion)
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	CRCLCEN	0h	RW	Cr Calculation Data Clip Enable (Data Clipping Process Control for Cr Data Calculation in RGB → YCbCr Color Space Conversion) 0b: Data clipping is not performed 1b: Data clipping is performed
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	CRCLAP[11:0]	800h	RW	Value added for normalization of Cr calculation data (specification of Cr data addition constant in RGB → YCbCr color space conversion)

(116) CRU Lookup Table Control Register (CRUm_ICnLUT) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 034Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LUTWR	LUTSEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	LUTWR	0h	RW	Enables APB read/write access to the lookup table (LUT). 0b: Disables APB access (set this bit to 0 in operation with the normal path). 1b: Enables APB access (normal access is unavailable in this case). Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.
0	LUTSEL	0h	RW	Specifies the access path of the lookup table (LUT). 0b: Normal path (image conversion by using the LUT) 1b: APB I/F (read/write from registers) Used to write (and confirm) data to the LUT in initial setting. Set this bit to 0 after finishing writing and reading data. Note that the LUT data is retained even after a reset. Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.

(117) CRU Lookup Table Status Register (CRUm_ICnLUTS) (m = 0, 1)

Access Size : 32 bits																
Address : <CRUm_base> + 0350h																
Initial Value : 0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LUTREGSEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	LUTREGSEL	0h	R	Indicates that the lookup table (LUT) is writable and readable from the APB. The LUTWR register value is reflected in this register after a while. Access the lookup table after confirming this bit. 0b: Neither writable nor readable. 1b: Writable and readable. Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.

(118) CRU Lookup Table Pointer Register (CRUm_ICnLUTP) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0354h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	LTCRPR[9:0]										LTCBPR[9:6]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTCBPR[5:0]						LTYPR[9:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 20	LTCRPR[9:0]	0h	RW	Lookup table (LUT) Cr/B pointer Specify the current pointer (access point) to the lookup table Cr/B in APB access. The pointer is incremented automatically when ICnLUTD.LTCRDT[7:0] are written. It returns to 0 upon an overflow. It is not incremented when reading these bits. Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.
19 to 10	LTCBPR[9:0]	0h	RW	Lookup table (LUT) Cb/G pointer Specify the current pointer (access point) to lookup table Cb/G in APB access. The pointer is incremented automatically when ICnLUTD.LTCBDT[7:0] are written. It returns to 0 upon an overflow. It is not incremented when reading these bits. Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.
9 to 0	LTYPR[9:0]	0h	RW	Lookup table (LUT) Y/R pointer Specify the current pointer (access point) to lookup table Y/R in APB access. The pointer is incremented automatically when ICnLUTD.LTYDT[7:0] are written. It returns to 0 upon an overflow. It is not incremented when reading these bits. Note: See Figure 9.2-60 Storing LUT Data for the LUT setting procedure.

(119) CRU Lookup Table Data Register (CRUm_ICnLUTD) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 0358h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	LTCRDT[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTCBDT[7:0]								LTYDT[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	LTCRDT[7:0]	0h	RW	Lookup table (LUT) Cr/B data Read and write data to the location indicated by the ICnLUTP.LTCRPR[9:0] pointer in lookup table Cr/B through the APB interface.
15 to 8	LTCBDT[7:0]	0h	RW	Lookup table (LUT) Cb/G data Read and write data to the location indicated by the ICnLUTP.LTCBPR[9:0] pointer in lookup table Cb/G through the APB interface.
7 to 0	LTYDT[7:0]	0h	RW	Lookup table (LUT) Y/R data Read and write data to the location indicated by the ICnLUTP.LTYPR[9:0] pointer in lookup table Y/R through the APB interface.

(120) CRU Test Image Generation Control 1 Register (CRUm_ICnTICTRL1) (m = 0, 1)

Access Size : 32 bits

Address : <CRUm_base> + 035Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	TIRATE[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIPTNV1[3:0]				TIPTNU1[3:0]				TIPTNY1[3:0]				-	TIROTS SEL	TIMOD E	TIEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20 to 16	TIRATE[4:0]	0h	RW	Specify the test image generation rate. 16 pixels / (18 + TIRATE) cycles
15 to 12	TIPTNV1[3:0]	0h	RW	Specify the pixel (V) counter bits for the test image pattern. 0000b: Selects bits [3:0]. 0001b: Selects bits [4:1]. 0010b: Selects bits [5:2]. Omitted 1100b: Selects bits [15:12]. 1101b to 1111b: Setting prohibited
11 to 8	TIPTNU1[3:0]	0h	RW	Specify the pixel (U) counter bits for the test image pattern. 0000b: Selects bits [3:0]. 0001b: Selects bits [4:1]. 0010b: Selects bits [5:2]. Omitted 1100b: Selects bits [15:12]. 1101b to 1111b: Setting prohibited
7 to 4	TIPTNY1[3:0]	0h	RW	Specify the pixel (Y) counter bits for the test image pattern. 0000b: Selects bits [3:0]. 0001b: Selects bits [4:1]. 0010b: Selects bits [5:2]. Omitted 1100b: Selects bits [15:12]. 1101b to 1111b: Setting prohibited
3	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TIROTSSEL	0h	RW	Image processing/non-image processing route selection for test image output 0b: Input for image processing 1b: Input for non-image processing
1	TIMODE	0h	RW	Specifies a test image generation pattern. 0b: Selects a counter value for YUV output. 1b: Same-color YUV output for the entire screen
0	TIEN	0h	RW	Enables generation of a test image pattern. This test image works with VC0 only. 0b: Does not generate a test image pattern (processes the data type from the LINK). 1b: Generates a test image pattern (process test image pattern data). Note: When this bit is set to 1b, test data is forcibly selected regardless of CRUCTRL.VINSEL.

(121) CRU Test Image Generation Control 2 Register (CRUm_ICnTICTRL2) (m = 0, 1)

Access Size :		32 bits																	
Address :		<CRUm_base> + 0360h																	
Initial Value :		0000_0000h																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
	-	-	-	-	-	-	-	-	TIPTNV2[7:0]										
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	TIPTNU2[7:0]								TIPTNY2[7:0]										
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	TIPTNV2[7:0]	0h	RW	Specify V for the entire-screen same-color test image pattern. Note: This bit is valid when ICnTICTRL1.TIMODE=1b.
15 to 8	TIPTNU2[7:0]	0h	RW	Specify U for the entire-screen same-color test image pattern. Note: This bit is valid when ICnTICTRL1.TIMODE=1b.
7 to 0	TIPTNY2[7:0]	0h	RW	Specify Y for the entire-screen same-color test image pattern. Note: This bit is valid when ICnTICTRL1.TIMODE=1b.

(122) CRU Test Image Size Setting 1 Register (CRUm_ICnTISIZE1) (m = 0, 1)

Access Size :		32 bits														
Address :		<CRUm_base> + 0364h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	TIPPL[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 12	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 0	TIPPL[11:0]		0h		RW	Number of valid pixels per line in the test image pattern Note 1. If a value from 0 to 15 is specified, operation is for the setting of 16. Note 2. The four lower-order bits are invalid.										

(123) CRU Test Image Size Setting 2 Register (CRUm_ICnTISIZE2) (m = 0, 1)

Access Size :		32 bits																
Address :		<CRUm_base> + 0368h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	TIM[11:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	TIN[11:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 16	TIM[11:0]	0h	RW	Invalid period (number of lines) per frame in the test image pattern Note: If 0 is specified, operation is for the setting of 1.
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	TIN[11:0]	0h	RW	Valid period (number of lines) per frame in the test image pattern Note: If 0 is specified, operation is for the setting of 1.

9.2.4.3 Function Description

The prefix “CRUm (m = 0, 1)” of the register names is omitted in the following subsections.

9.2.4.3.1 Input Formats and Image Processing

Table 9.2-11, **Table 9.2-12**, and **Table 9.2-13** list the input formats in which images can be processed by this module and the corresponding available forms of image processing.

Table 9.2-11 Input Formats and Corresponding Forms of Image Processing (for Image Processing) (1/3)

Input Format			Full Bypass (CNCTRL.ATH = 1) 8-bit Expansion (RGB888, RAW8)	Processing Indicator												Interrupt		Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
				COMMON				RAW		YUV			RGB																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Data Type	Data Type Classes	Data Type Name		Header Appending	Footer Appending	Frame Subsampling	Image Clipping	LUT	Statistics	Demosaicing	Linear Matrix	Data Arranger 1	Data Clipping	YCbCr to RGB	YCbCr422 to 420	Data Arranger 2	RGB to YCbCr	Frame Start/End, Scan line	ERROR (DECERR/SLVERR/FIFO Overflow)	AXI Specified Address Forwarding Completed	YCbCr 422	YCbCr 420	RGB			RAW						MIPI Recommended Memory Storage Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
Input settings (ICnIPMC_C0/1/2/3.INF[5:0] register)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
00h	Synchronization short packet data types	Frame start code																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											</

Table 9.2-11 Input Formats and Corresponding Forms of Image Processing (for Image Processing) (2/3)

Input Format					Processing Indicator														Interrupt		Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
					COMMON				RAW		YUV				RGB																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
					Header Appending																Demosaicing		Data Arranger 1		Data Clipping		YCbCr to RGB		YCbCr422 to 420		Data Arranger 2		RGB to YCbCr		Frame Start/End, Scan line		ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed		8-bit multiplexed		YCbCr 422		YCbCr 420		RGB		RAW				MIPI Recommended Memory Storage Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
					Footer Appending				Linear Matrix		Data Clipping		YCbCr to RGB		YCbCr422 to 420		Data Arranger 2				Frame Start/End, Scan line		ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed		8-bit multiplexed		YCbCr 422		YCbCr 420		RGB		RAW																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
Data Type	Data Type Classes	Data Type Name	Full Bypass (CnCTRLATH = 1)														8-bit Expansion (RGB888, RAW8)		Header Appending				Footer Appending		Frame Subsampling				Image Clipping		LUT		Statistics		Demosaicing		Linear Matrix		Data Arranger 1		Data Clipping		YCbCr to RGB		YCbCr422 to 420		Data Arranger 2		RGB to YCbCr		Frame Start/End, Scan line		ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed		8-bit multiplexed		YCbCr 422		YCbCr 420		RGB		RAW				MIPI Recommended Memory Storage Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
10h	Generic long packet data types	Null	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—			—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 9.2-11 Input Formats and Corresponding Forms of Image Processing (for Image Processing) (3/3)

Input Format			Full Bypass (ICnCTRL.ATH = 1)	Processing Indicator																Interrupt	Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
Data Type	Data Type Classes	Data Type Name		COMMON				RAW				YUV				RGB	Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
				Header Appending	Footer Appending	Frame Subsampling	Image Clipping	LUT	Statistics	Demosaicing	Linear Matrix	Data Arranger 1	Data Clipping	YCbCr to RGB	YCbCr422 to 420	Data Arranger 2	RGB to YCbCr	Frame Start/End, Scan line	ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed	YCbCr 422	YCbCr 420	RGB	RAW												MIPI Recommended Memory Storage Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
																						8-bit multiplexed	YCbCr 422	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420		YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420	YCbCr 422 to 420	YCbCr 420

Note: ✓: Supports SVC0 to SVC3

⊙: Only SVC0 is supported

—: Not supported

Table 9.2-12 Input Formats and Corresponding Forms of Image Processing (for Non-Image Processing) (1/3)

[illegible]

Table 9.2-12 Input Formats and Corresponding Forms of Image Processing (for Non-Image Processing) (2/3)

Input Format			Full Bypass (CnCTRLATH = 1)	Processing Indicator																Interrupt	Output Format															
				COMMON						RAW			YUV				RGB																			
				Header Appending	Footer Appending	Frame Subsampling	Image Clipping	LUT	Statistics	Demosaicing	Linear Matrix	Data Arranger 1	Data Clipping	YCbCr to RGB	YCbCr422 to 420	Data Arranger 2	RGB to YCbCr	Frame Start/End, Scan line	ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed	YCbCr 422	YCbCr 420	RGB	RAW										MIPI Recommended Memory Storage Format	
																						8-bit multiplexed	Y component extraction	8-bit multiplexed	YC separation	RGB888 (24 bits)	RGB888 (32 bits)	ARGB888 (32 bits)	RAW6	RAW7	RAW8	RAW10	RAW12	RAW14		RAW16
Data Type	Data Type Classes	Data Type Name	18h	YUV data	YUV420 8-bit	✓	✓	✓	✓											✓	✓														✓	
19h		YUV420 10-bit	✓		✓	✓	✓													✓	✓													✓		
1Ah		Legacy YUV420 8-bit	✓		✓	✓	✓													✓	✓													✓		
1Bh		Reserved	—	—	—	—	—													—	—													—		
1Ch		YUV420 8-bit (chroma shifted pixel sampling)	✓		✓	✓	✓													✓	✓													✓		
1Dh		YUV420 10-bit (chroma shifted pixel sampling)	✓		✓	✓	✓													✓	✓													✓		
1Eh		YUV422 8-bit	✓		✓	✓	✓													✓	✓													✓		
1Fh		YUV422 10-bit	✓		✓	✓	✓													✓	✓													✓		
20h	RGB data	RGB444	✓		✓	✓	✓													✓	✓													✓		
21h		RGB555	✓		✓	✓	✓													✓	✓													✓		
22h		RGB565	✓		✓	✓	✓													✓	✓													✓		
23h		RGB666	✓		✓	✓	✓													✓	✓													✓		
24h		RGB888	✓		✓	✓	✓													✓	✓													✓		
25h to 27h		Reserved	—	—	—	—	—	—												—	—														—	
28h	RAW data	RAW6	✓		✓	✓	✓													✓	✓													✓		
29h		RAW7	✓		✓	✓	✓													✓	✓													✓		
2Ah		RAW8	✓		✓	✓	✓													✓	✓													✓		
2Bh		RAW10	✓		✓	✓	✓													✓	✓													✓		
2Ch		RAW12	✓		✓	✓	✓													✓	✓													✓		
2Dh		RAW14	✓		✓	✓	✓													✓	✓													✓		
2Eh		RAW16	✓		✓	✓	✓													✓	✓													✓		
2Fh		RAW20	✓		✓	✓	✓													✓	✓													✓		
30h	User defined byte-based data	User defined 8-bit data type 1	✓		✓	✓	✓													✓	✓													✓		
31h		User defined 8-bit data type 2	✓		✓	✓	✓													✓	✓													✓		
32h		User defined 8-bit data type 3	✓		✓	✓	✓													✓	✓													✓		
33h		User defined 8-bit data type 4	✓		✓	✓	✓													✓	✓													✓		
34h		User defined 8-bit data type 5	✓		✓	✓	✓													✓	✓													✓		
35h		User defined 8-bit data type 6	✓		✓	✓	✓													✓	✓													✓		
36h		User defined 8-bit data type 7	✓		✓	✓	✓													✓	✓													✓		
37h		User defined 8-bit data type 8	✓		✓	✓	✓													✓	✓													✓		

Table 9.2-12 Input Formats and Corresponding Forms of Image Processing (for Non-Image Processing) (3/3)

Input Format			Full Bypass (ICnCTRL.ATH = 1)	Processing Indicator																Interrupt	Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
Data Type	Data Type Classes	Data Type Name		COMMON				RAW		YUV				RGB	Output Format																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
				Header Appending	Footer Appending	Frame Subsampling	Image Clipping	LUT	Statistics	Demosaicing	Linear Matrix	Data Arranger 1	Data Clipping	YCbCr to RGB	YCbCr422 to 420	Data Arranger 2	RGB to YCbCr	Frame Start/End, Scan line	ERROR (DECERR/SLVERR/FIFO Overflow)		AXI Specified Address Forwarding Completed	YCbCr 422	YCbCr 420	RGB	RAW																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
																						8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420	8-bit multiplexed	YCbCr 422	YCbCr 420

Note: ✓: Supports SVC0 to SVC3
 —: Not supported

Table 9.2-13 Input Formats and Corresponding Forms of Image Processing (for Statistics)

Input Format			Full Bypass (ICnCTRLATH = 1)	8-bit Expansion (RGB888, RAW8)	Processing Indicator														Interrupt			Output Format
Data Type	Data Type Classes	Data Type Name			COMMON					RAW		YUV				RGB						
					Header Appending	Footer Appending	Frame Subsampling	Image Clipping	LUT	Statistics	Demosaicing	Linear Matrix	Data Arranger 1	Data Clipping	YCbCr to RGB	YCbCr422 to 420	Data Arranger 2	RGB to YCbCr				
					Input settings (ICnIPMC_C0/1/2/3.INF[5:0] register)																	
00h to 27h		Not supports	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
28h	RAW data	RAW6	—	○	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
29h		RAW7	—	○	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Ah		RAW8	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Bh		RAW10	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Ch		RAW12	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Dh		RAW14	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Eh		RAW16	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
2Fh		RAW20	—	—	—	○	○	○	—	○	—	—	—	—	—	—	—	○	○	○	○	
30h to 3Fh		Not supports	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
Pattern generator (ICnTICTRL1.TIEN register)																						
1b	Test pattern	YCbCr422 8-bit	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

Note: ✓: Supports SVC0 to SVC3

⊙: Only SVC0 is supported

—: Not supported

9.2.4.3.2 **Output Formats**

Table 9.2-14 and **Table 9.2-15** list the video output formats and the statistics output format, respectively.

Table 9.2-14 Video Output Formats (1) (1/2)

Format	Line# ⁴	AMnUVA OFH/L address offset ⁵	Registers	Data[255:0]															
				byte 31	byte 30	byte 29	byte 28	byte 27	byte 26	byte 25	byte 24	byte 23	byte 22	byte 21	byte 20	byte 19	byte 18	byte 17	byte 16
YUV (YCbCr) 422	Cb 1st	Both	Non-use ICnDMR.YCMODE = 001b CRUnCTRL.ATH = 0	Y15 [7:0]	Cr14 [7:0]	Y14 [7:0]	Cb14 [7:0]	Y13 [7:0]	Cr12 [7:0]	Y12 [7:0]	Cb12 [7:0]	Y11 [7:0]	Cr10 [7:0]	Y10 [7:0]	Cb10 [7:0]	Y9 [7:0]	Cr8 [7:0]	Y8 [7:0]	Cb8 [7:0]
	Y 1st	Both	Non-use ICnDMR.YCMODE = 000b CRUnCTRL.ATH = 0	Cr14 [7:0]	Y15 [7:0]	Cb14 [7:0]	Y14 [7:0]	Cr12 [7:0]	Y13 [7:0]	Cb12 [7:0]	Y12 [7:0]	Cr10 [7:0]	Y11 [7:0]	Cb10 [7:0]	Y10 [7:0]	Cr8 [7:0]	Y9 [7:0]	Cb8 [7:0]	Y8 [7:0]
	YC separation	Both	Non-use ICnDMR.YCMODE = 010b	Y31 [7:0]	Y30 [7:0]	Y29 [7:0]	Y28 [7:0]	Y27 [7:0]	Y26 [7:0]	Y25 [7:0]	Y24 [7:0]	Y23 [7:0]	Y22 [7:0]	Y21 [7:0]	Y20 [7:0]	Y19 [7:0]	Y18 [7:0]	Y17 [7:0]	Y16 [7:0]
		Both	Use CRUnCTRL.ATH = 0	Cr30 [7:0]	Cb30 [7:0]	Cr28 [7:0]	Cb28 [7:0]	Cr26 [7:0]	Cb26 [7:0]	Cr24 [7:0]	Cb24 [7:0]	Cr22 [7:0]	Cb22 [7:0]	Cr20 [7:0]	Cb20 [7:0]	Cr18 [7:0]	Cb18 [7:0]	Cr16 [7:0]	Cb16 [7:0]
YUV (YCbCr) 420	Cb 1st	Odd	Non-use ICnDMR.YCMODE = 101b	Y15 [7:0]	Cr14 [7:0]	Y14 [7:0]	Cb14 [7:0]	Y13 [7:0]	Cr12 [7:0]	Y12 [7:0]	Cb12 [7:0]	Y11 [7:0]	Cr10 [7:0]	Y10 [7:0]	Cb10 [7:0]	Y9 [7:0]	Cr8 [7:0]	Y8 [7:0]	Cb8 [7:0]
		Even	Non-use CRUnCTRL.ATH = 0	Y31 [7:0]	Y30 [7:0]	Y29 [7:0]	Y28 [7:0]	Y27 [7:0]	Y26 [7:0]	Y25 [7:0]	Y24 [7:0]	Y23 [7:0]	Y22 [7:0]	Y21 [7:0]	Y20 [7:0]	Y19 [7:0]	Y18 [7:0]	Y17 [7:0]	Y16 [7:0]
	Y 1st	Odd	Non-use ICnDMR.YCMODE = 100b/111b	Cr14 [7:0]	Y15 [7:0]	Cb14 [7:0]	Y14 [7:0]	Cr12 [7:0]	Y13 [7:0]	Cb12 [7:0]	Y12 [7:0]	Cr10 [7:0]	Y11 [7:0]	Cb10 [7:0]	Y10 [7:0]	Cr8 [7:0]	Y9 [7:0]	Cb8 [7:0]	Y8 [7:0]
		Even	Non-use CRUnCTRL.ATH = 0	Y31 [7:0]	Y30 [7:0]	Y29 [7:0]	Y28 [7:0]	Y27 [7:0]	Y26 [7:0]	Y25 [7:0]	Y24 [7:0]	Y23 [7:0]	Y22 [7:0]	Y21 [7:0]	Y20 [7:0]	Y19 [7:0]	Y18 [7:0]	Y17 [7:0]	Y16 [7:0]
RGB-888	24 bit/pixel	Both	Non-use ICnDMR.RGBMODE = 00b CRUnCTRL.ATH=0	G10 [7:0]	B10 [7:0]	R9 [7:0]	G9 [7:0]	B9 [7:0]	R8 [7:0]	G8 [7:0]	B8 [7:0]	R7 [7:0]	G7 [7:0]	B7 [7:0]	R6 [7:0]	G6 [7:0]	B6 [7:0]	R5 [7:0]	G5 [7:0]
		Both	Non-use ICnDMR.RGBMODE = 01b CRUnCTRL.ATH = 0	00h	R7 [7:0]	G7 [7:0]	B7 [7:0]	00h	R6 [7:0]	G6 [7:0]	B6 [7:0]	00h	R5 [7:0]	G5 [7:0]	B5 [7:0]	00h	R4 [7:0]	G4 [7:0]	B4 [7:0]
	32 bit/pixel	Both	Non-use ICnDMR.RGBMODE = 01b CRUnCTRL.ATH = 0	00h	R7 [7:0]	G7 [7:0]	B7 [7:0]	00h	R6 [7:0]	G6 [7:0]	B6 [7:0]	00h	R5 [7:0]	G5 [7:0]	B5 [7:0]	00h	R4 [7:0]	G4 [7:0]	B4 [7:0]
		Both	Use CRUnCTRL.ATH = 0	Cr30 [7:0]	Cb30 [7:0]	Cr28 [7:0]	Cb28 [7:0]	Cr26 [7:0]	Cb26 [7:0]	Cr24 [7:0]	Cb24 [7:0]	Cr22 [7:0]	Cb22 [7:0]	Cr20 [7:0]	Cb20 [7:0]	Cr18 [7:0]	Cb18 [7:0]	Cr16 [7:0]	Cb16 [7:0]
ARGB-8888	B 1st	Both	Non-use ICnDMR.RGBMODE = 10b CRUnCTRL.ATH = 0	A7 [7:0]	R7 [7:0]	G7 [7:0]	B7 [7:0]	A8 [7:0]	R6 [7:0]	G6 [7:0]	B6 [7:0]	A5 [7:0]	R5 [7:0]	G5 [7:0]	B5 [7:0]	A4 [7:0]	R4 [7:0]	G4 [7:0]	B4 [7:0]
	A 1st	Both	Non-use ICnDMR.RGBMODE = 11b CRUnCTRL.ATH = 0	B7 [7:0]	G7 [7:0]	R7 [7:0]	A7 [7:0]	B6 [7:0]	G6 [7:0]	R6 [7:0]	A6 [7:0]	B5 [7:0]	G5 [7:0]	R5 [7:0]	A5 [7:0]	B4 [7:0]	G4 [7:0]	R4 [7:0]	A4 [7:0]
RAW	RAW6	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 28h ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P39[5:0], P38[5:0], P37[5:0], P36[5:0], P35[5:0], P34[5:0], P33[5:0], P32[5:0], P31[5:0], P30[5:0]}								{Xh, P29[5:0], P28[5:0], P27[5:0], P26[5:0], P25[5:0], P24[5:0], P23[5:0], P22[5:0], P21[5:0], P20[5:0]}							
	RAW7	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 29h ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P35[6:0], P34[6:0], P33[6:0], P32[6:0], P31[6:0], P30[6:0], P29[6:0], P28[6:0], P27[6:0]}								{Xh, P26[6:0], P25[6:0], P24[6:0], P23[6:0], P22[6:0], P21[6:0], P20[6:0], P19[6:0], P18[6:0]}							
	RAW8	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Ah ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	P31 [7:0]	P30 [7:0]	P29 [7:0]	P28 [7:0]	P27 [7:0]	P26 [7:0]	P25 [7:0]	P24 [7:0]	P23 [7:0]	P22 [7:0]	P21 [7:0]	P20 [7:0]	P19 [7:0]	P18 [7:0]	P17 [7:0]	P16 [7:0]
	RAW10	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Bh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P23[9:0], P22[9:0], P21[9:0], P20[9:0], P19[9:0], P18[9:0]}								{Xh, P17[9:0], P16[9:0], P15[9:0], P14[9:0], P13[9:0], P12[9:0]}							
	RAW12	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Ch ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P19[11:0], P18[11:0], P17[11:0], P16[11:0], P15[11:0]}								{Xh, P14[11:0], P13[11:0], P12[11:0], P11[11:0], P10[11:0]}							

Table 9.2-14 Video Output Formats (1) (2/2)

Format		Line*4	AMnUVA OFH/L address offset*5	Registers	Data[255:0]																	
					byte 31	byte 30	byte 29	byte 28	byte 27	byte 26	byte 25	byte 24	byte 23	byte 22	byte 21	byte 20	byte 19	byte 18	byte 17	byte 16		
RAW	RAW14	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Dh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	00h	{P15[13:0], P14[13:0], P13[13:0], P12[13:0]}								00h	{P11[13:0], P10[13:0], P9[13:0], P8[13:0]}							
	RAW16	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Eh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	P14[15:0]		P13[15:0]		P12[15:0]		P11[15:0]		P10[15:0]		P9[15:0]		P8[15:0]		P7[15:0]			
	RAW20	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Fh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P11[19:0], P10[19:0], P9[19:0]}								{Xh, P8[19:0], P7[19:0], P6[19:0]}									
MIPI recommended memory storage format (full bypass)*2		Both	Non-use	CRUnCTRL.ATH = 1	Byte 32 [7:0]*1	Byte 31 [7:0]*1	Byte 30 [7:0]*1	Byte 29 [7:0]*1	Byte 28 [7:0]*1	Byte 27 [7:0]*1	Byte 26 [7:0]*1	Byte 25 [7:0]*1	Byte 24 [7:0]*1	Byte 23 [7:0]*1	Byte 22 [7:0]*1	Byte 21 [7:0]*1	Byte 20 [7:0]*1	Byte 19 [7:0]*1	Byte 18 [7:0]*1	Byte 17 [7:0]*1		

Note 1. Data from MIPI CSI-2 LINK is output in 8-bit units. However, it is output in the data array of the Recommended Memory Storage of the MIPI CSI-2 V2.1 standard.

Note 2. Only data types marked with ✓ in "MIPI Recommended Memory Storage Format" in **Table 9.2-12** are output in this format. Otherwise, the data will be output in one of the above formats.

Note 3. The CbCr(U/V) side is Image Stride(AMnIS) × 2. See **Image Stride** in **9.2.4.3.3(5)(a) Video data frame** for details.

Note 4. The entries in this column have the following meanings.
 "Odd" means the image has an odd number of horizontal lines.
 "Even" means the image has an even number of horizontal lines.
 "Both" means the image has an odd and even number of horizontal lines.

Note 5. When transferring data to memory, there are cases where Cb and Cr data are transferred to one area as Y, or data is divided into two independent areas.

The entries in this column represent the following.

Non-use: Cb and Cr data are transferred to the same area as Y when transferred to memory.

Use: When Cb and Cr data are transferred to memory, they are transferred to an area separate from Y. This separate area is transferred to the area specified in the AMnUVAOFH/L register for the Y data storage destination.

Table 9.2-15 Video Output Formats (2) (1/2)

Format	Line# ⁴	AMnUVA OFH/L address offset ⁵	Registers	Data[255:0]															
				byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
YUV (YCbCr) 422	Cb 1st	Both	Non-use ICnDMR.YCMODE = 001b CRUnCTRL.ATH = 0	Y7 [7:0]	Cr6 [7:0]	Y6 [7:0]	Cb6 [7:0]	Y5 [7:0]	Cr4 [7:0]	Y4 [7:0]	Cb4 [7:0]	Y3 [7:0]	Cr2 [7:0]	Y2 [7:0]	Cb2 [7:0]	Y1 [7:0]	Cr0 [7:0]	Y0 [7:0]	Cb0 [7:0]
	Y 1st	Both	Non-use ICnDMR.YCMODE = 000b CRUnCTRL.ATH=0	Cr6 [7:0]	Y7 [7:0]	Cb6 [7:0]	Y6 [7:0]	Cr4 [7:0]	Y5 [7:0]	Cb4 [7:0]	Y4 [7:0]	Cr2 [7:0]	Y3 [7:0]	Cb2 [7:0]	Y2 [7:0]	Cr0 [7:0]	Y1 [7:0]	Cb0 [7:0]	Y0 [7:0]
	YC separation	Both	Non-use ICnDMR.YCMODE = 010b	Y15 [7:0]	Y14 [7:0]	Y13 [7:0]	Y12 [7:0]	Y11 [7:0]	Y10 [7:0]	Y9 [7:0]	Y8 [7:0]	Y7 [7:0]	Y6 [7:0]	Y5 [7:0]	Y4 [7:0]	Y3 [7:0]	Y2 [7:0]	Y1 [7:0]	Y0 [7:0]
		Both	Use CRUnCTRL.ATH = 0	Cr14 [7:0]	Cb14 [7:0]	Cr12 [7:0]	Cb12 [7:0]	Cr10 [7:0]	Cb10 [7:0]	Cr8 [7:0]	Cb8 [7:0]	Cr6 [7:0]	Cb6 [7:0]	Cr4 [7:0]	Cb4 [7:0]	Cr2 [7:0]	Cb2 [7:0]	Cr0 [7:0]	Cb0 [7:0]
	Y component extraction	Both	Non-use ICnDMR.YCMODE = 011b CRUnCTRL.ATH = 0	Y15 [7:0]	Y14 [7:0]	Y13 [7:0]	Y12 [7:0]	Y11 [7:0]	Y10 [7:0]	Y9 [7:0]	Y8 [7:0]	Y7 [7:0]	Y6 [7:0]	Y5 [7:0]	Y4 [7:0]	Y3 [7:0]	Y2 [7:0]	Y1 [7:0]	Y0 [7:0]
YUV (YCbCr) 420	Cb 1st	Odd	Non-use ICnDMR.YCMODE = 101b	Y7 [7:0]	Cr6 [7:0]	Y6 [7:0]	Cb6 [7:0]	Y5 [7:0]	Cr4 [7:0]	Y4 [7:0]	Cb4 [7:0]	Y3 [7:0]	Cr2 [7:0]	Y2 [7:0]	Cb2 [7:0]	Y1 [7:0]	Cr0 [7:0]	Y0 [7:0]	Cb0 [7:0]
		Even	Non-use CRUnCTRL.ATH = 0	Y15 [7:0]	Y14 [7:0]	Y13 [7:0]	Y12 [7:0]	Y11 [7:0]	Y10 [7:0]	Y9 [7:0]	Y8 [7:0]	Y7 [7:0]	Y6 [7:0]	Y5 [7:0]	Y4 [7:0]	Y3 [7:0]	Y2 [7:0]	Y1 [7:0]	Y0 [7:0]
	Y 1st	Odd	Non-use ICnDMR.YCMODE = 100b/111b	Cr6 [7:0]	Y7 [7:0]	Cb6 [7:0]	Y6 [7:0]	Cr4 [7:0]	Y5 [7:0]	Cb4 [7:0]	Y4 [7:0]	Cr2 [7:0]	Y3 [7:0]	Cb2 [7:0]	Y2 [7:0]	Cr0 [7:0]	Y1 [7:0]	Cb0 [7:0]	Y0 [7:0]
		Even	Non-use CRUnCTRL.ATH = 0	Y15 [7:0]	Y14 [7:0]	Y13 [7:0]	Y12 [7:0]	Y11 [7:0]	Y10 [7:0]	Y9 [7:0]	Y8 [7:0]	Y7 [7:0]	Y6 [7:0]	Y5 [7:0]	Y4 [7:0]	Y3 [7:0]	Y2 [7:0]	Y1 [7:0]	Y0 [7:0]
	YC separation*3	Both	Non-use ICnDMR.YCMODE = 110b	Y15 [7:0]	Y14 [7:0]	Y13 [7:0]	Y12 [7:0]	Y11 [7:0]	Y10 [7:0]	Y9 [7:0]	Y8 [7:0]	Y7 [7:0]	Y6 [7:0]	Y5 [7:0]	Y4 [7:0]	Y3 [7:0]	Y2 [7:0]	Y1 [7:0]	Y0 [7:0]
		Both	Use CRUnCTRL.ATH = 0	Cr14 [7:0]	Cb14 [7:0]	Cr12 [7:0]	Cb12 [7:0]	Cr10 [7:0]	Cb10 [7:0]	Cr8 [7:0]	Cb8 [7:0]	Cr6 [7:0]	Cb6 [7:0]	Cr4 [7:0]	Cb4 [7:0]	Cr2 [7:0]	Cb2 [7:0]	Cr0 [7:0]	Cb0 [7:0]
RGB-888	24 bit/pixel	Both	Non-use ICnDMR.RGBMODE = 00b CRUnCTRL.ATH = 0	B5 [7:0]	R4 [7:0]	G4 [7:0]	B4 [7:0]	R3 [7:0]	G3 [7:0]	B3 [7:0]	R2 [7:0]	G2 [7:0]	B2 [7:0]	R1 [7:0]	G1 [7:0]	B1 [7:0]	R0 [7:0]	G0 [7:0]	B0 [7:0]
	32 bit/pixel	Both	Non-use ICnDMR.RGBMODE = 01b CRUnCTRL.ATH = 0	00h	R3 [7:0]	G3 [7:0]	B3 [7:0]	00h	R2 [7:0]	G2 [7:0]	B2 [7:0]	00h	R1 [7:0]	G1 [7:0]	B1 [7:0]	00h	R0 [7:0]	G0 [7:0]	B0 [7:0]
ARGB-8888	B 1st	Both	Non-use ICnDMR.RGBMODE = 10b CRUnCTRL.ATH = 0	A3 [7:0]	R3 [7:0]	G3 [7:0]	B3 [7:0]	A2 [7:0]	R2 [7:0]	G2 [7:0]	B2 [7:0]	A1 [7:0]	R1 [7:0]	G1 [7:0]	B1 [7:0]	A0 [7:0]	R0 [7:0]	G0 [7:0]	B0 [7:0]
	A 1st	Both	Non-use ICnDMR.RGBMODE = 11b CRUnCTRL.ATH = 0	B3 [7:0]	G3 [7:0]	R3 [7:0]	A3 [7:0]	B2 [7:0]	G2 [7:0]	R2 [7:0]	A2 [7:0]	B1 [7:0]	G1 [7:0]	R1 [7:0]	A1 [7:0]	B0 [7:0]	G0 [7:0]	R0 [7:0]	A0 [7:0]
RAW	RAW6	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 28h ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P19[5:0], P18[5:0], P17[5:0], P16[5:0], P15[5:0], P14[5:0], P13[5:0], P12[5:0], P11[5:0], P10[5:0]}								{Xh, P9[5:0], P8[5:0], P7[5:0], P6[5:0], P5[5:0], P4[5:0], P3[5:0], P2[5:0], P1[5:0], P0[5:0]}							
	RAW7	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 29h ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P17[6:0], P16[6:0], P15[6:0], P14[6:0], P13[6:0], P12[6:0], P11[6:0], P10[6:0], P9[6:0]}								{Xh, P8[6:0], P7[6:0], P6[6:0], P5[6:0], P4[6:0], P3[6:0], P2[6:0], P1[6:0], P0[6:0]}							
	RAW8	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Ah ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	P15 [7:0]	P14 [7:0]	P13 [7:0]	P12 [7:0]	P11 [7:0]	P10 [7:0]	P9 [7:0]	P8 [7:0]	P7 [7:0]	P6 [7:0]	P5 [7:0]	P4 [7:0]	P3 [7:0]	P2 [7:0]	P1 [7:0]	P0 [7:0]
	RAW10	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Bh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P11[9:0], P10[9:0], P9[9:0], P8[9:0], P7[9:0], P6[9:0]}								{Xh, P5[9:0], P4[9:0], P3[9:0], P2[9:0], P1[9:0], P0[9:0]}							
	RAW12	Both	Non-use ICnIPMC_C0/1/2/3.IN F = 2Ch ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P9[11:0], P8[11:0], P7[11:0], P6[11:0], P5[11:0]}								{Xh, P4[11:0], P3[11:0], P2[11:0], P1[11:0], P0[11:0]}							

Table 9.2-15 Video Output Formats (2) (2/2)

Format		Line*4	AMnUVA OFH/L address offset*5	Registers	Data[255:0]															
					byte 15	byte 14	byte 13	byte 12	byte 11	byte 10	byte 9	byte 8	byte 7	byte 6	byte 5	byte 4	byte 3	byte 2	byte 1	byte 0
RAW	RAW14	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Dh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	00h	{P7[13:0], P6[13:0], P5[13:0], P4[13:0]}							00h	{P3[13:0], P2[13:0], P1[13:0], P0[13:0]}						
	RAW16	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Eh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	P7[15:0]		P6[15:0]		P5[15:0]		P4[15:0]		P3[15:0]		P2[15:0]		P1[15:0]		P0[15:0]	
	RAW20	Both	Non-use	ICnIPMC_C0/1/2/3.IN F = 2Fh ICnIPMC_C0/1/2/3.DE MTHR = 1 CRUnCTRL.ATH = 0	{Xh, P5[19:0], P4[19:0], P3[19:0]}							{Xh, P2[19:0], P1[19:0], P0[19:0]}								
MIPI recommended memory storage format (full bypass)*2		Both	Non-use	CRUnCTRL.ATH = 1	Byte 16 [7:0]*1	Byte 15 [7:0]*1	Byte 14 [7:0]*1	Byte 13 [7:0]*1	Byte 12 [7:0]*1	Byte 11 [7:0]*1	Byte 10 [7:0]*1	Byte 9 [7:0]*1	Byte 8 [7:0]*1	Byte 7 [7:0]*1	Byte 6 [7:0]*1	Byte 5 [7:0]*1	Byte 4 [7:0]*1	Byte 3 [7:0]*1	Byte 2 [7:0]*1	Byte 1 [7:0]*1

Note 1. Data from MIPI CSI-2 LINK is output in 8-bit units. However, it is output in the data array of the Recommended Memory Storage of the MIPI CSI-2 V2.1 standard.

Note 2. Only data types marked with ✓ in "MIPI Recommended Memory Storage Format" in **Table 9.2-12** are output in this format. Otherwise, the data will be output in one of the above formats.

Note 3. The CbCr(U/V) side is Image Stride(AMnIS) × 2. See Image Stride in **9.2.4.3.3(5)(a) Video data frame** for details.

Note 4. The entries in this column have the following meanings.
 "Odd" means the image has an odd number of horizontal lines.
 "Even" means the image has an even number of horizontal lines.
 "Both" means the image has an odd and even number of horizontal lines.

Note 5. When transferring data to memory, there are cases where Cb and Cr data are transferred to one area as Y, or data is divided into two independent areas.

The entries in this column represent the following.

Non-use: Cb and Cr data are transferred to the same area as Y when transferred to memory.

Use: When Cb and Cr data are transferred to memory, they are transferred to an area separate from Y. This separate area is transferred to the area specified in the AMnUVAOFH/L register for the Y data storage destination.

Table 9.2-16 Statistics Output Format

Format	Data[255:0]															
	byte31	byte30	byte29	byte28	byte27	byte26	byte25	byte24	byte23	byte22	byte21	byte20	byte19	byte18	byte17	byte16
Statistics data	In-block sum of absolute difference data		In-block accumulated data				In-block accumulated data				In-block accumulated data				In-block sum of absolute difference data	
	D_G7 [7:0]	S_R7 [7:0]	S_G7 [7:0]	S_B7 [7:0]	D_G6 [7:0]	S_R6 [7:0]	S_G6 [7:0]	S_B6 [7:0]	D_G5 [7:0]	S_R5 [7:0]	S_G5 [7:0]	S_B5 [7:0]	D_G4 [7:0]	S_R4 [7:0]	S_G4 [7:0]	S_B4 [7:0]

Format	Data[255:0]															
	byte15	byte14	byte13	byte12	byte11	byte10	byte9	byte8	byte7	byte6	byte5	byte4	byte3	byte2	byte1	byte0
Statistics data	In-block sum of absolute difference data		In-block accumulated data				In-block accumulated data				In-block accumulated data				In-block sum of absolute difference data	
	D_G3 [7:0]	S_R3 [7:0]	S_G3 [7:0]	S_B3 [7:0]	D_G2 [7:0]	S_R2 [7:0]	S_G2 [7:0]	S_B2 [7:0]	D_G1 [7:0]	S_R1 [7:0]	S_G1 [7:0]	S_B1 [7:0]	D_G0 [7:0]	S_R0 [7:0]	S_G0 [7:0]	S_B0 [7:0]

9.2.4.3.3 Transfer to DRAM

In data transfer from the CRU to the DRAM, data are transferred frame by frame separately based on the concept of memory banks. The memory banks can be set to operate with from 1 to 8 planes. As the transfer destination address for each memory bank, the transfer destination start address of the memory bank must be set such that it does not overlap with memory for the previous frames of data. Operation takes a cyclic form whereby the frame data returns to 1 when it reaches the last plane (see **Figure 9.2-11**). Accordingly, data on the DRAM must have been taken for next processing before overwriting. Set the number of memory banks so that this can be done before overwriting.

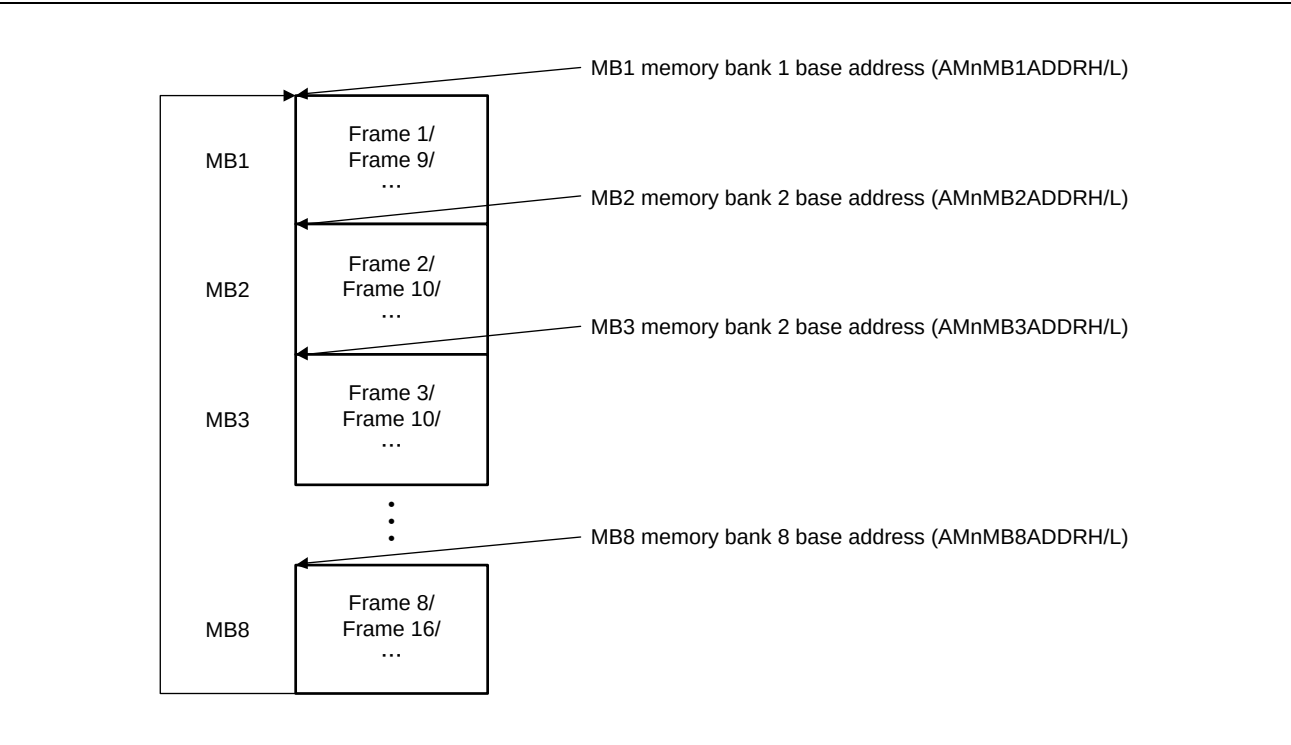


Figure 9.2-11 Image of Data Transfer to DRAM (Example: Eight Destinations in DRAM for the Transfer of Image-Processing Data)

Table 9.2-17 lists the memory bank setting addresses required for the DRAM. In this module, the transfer destination start addresses for the transfer of image data can be specified by setting the offsets listed below to proceed with transfer of up to four planes for SVC, transfer of the UV plane at the time of Y/UV separation, and transfer of non-image processing data or full-bypass data.

Furthermore, the transfer destination start addresses can be specified independently of those for statistics data that are transferred through the statistics bus.

Table 9.2-17 Memory Bank Setting Addresses

x = 0 to 7 (number of memory banks); y = 0 to 3 (number of SVC channels)

Transfer Bus	Processing	Transfer Destination Start Address	Description	Related Section
Video data bus	Image processing data	AMnMBxADDRH/L + AMnSVCAOFH/Lxy	Image data transfer destination. Transfer is possible for SVC0 to 3.	(1) Video Data Transfer Destination Address (Video Data Storage Destination)
		AMnMBxADDRH/L + AMnSVCAOFH/Lxy + AMnUVAOFH/L	Transfer destination of the CbCr (UV) plane at the time of YCbCr separation in image data. Transfer is possible for SVC0 to 3.	(2) Video Data Transfer Destination Address (Storage in Y/UV Separation)
	Non-Image processing data/full bypass	AMnMBxADDRH/L + AMnSVCAOFH/Lxy + MnNIAOFH/L	Non-image data/full-bypass transfer destination. Transfer is possible for SVC0 to 3.	(3) Video Data Transfer Destination Address (Image Processing/Non-Image Processing, 4-ch. SVC, Y/UV Separated Data)
Statistics data bus	Statistics processing	AMnSDBxADDRH/L	Statistics data transfer destination. Transfer is only possible for SVC0.	(4) Statistics Data Transfer Destination Address (Statistics Data Storage Destination)

Note: x represents the number of planes of each memory bank and it can be set from one to eight planes by the AMnMBVALID and AMnSDBVALID registers (the number differs between video and statistics).
y represents the channel number of SVC and it can be set from SVC0 to SVC3 by the ICnSVCNUM register.

Data are transferred to the address obtained by adding the given offset to AMnMBxADDRH/L, the image processing data transfer destination start address for SVC0.

1) SVC OFFSET

SVCs from SVC0 to 3 are supported. Data are transferred to the address obtained by adding the offset of SVC1 to 3 to the image data transfer destination start address for SVC0.

SVC1: AMnMBxADDRH/L + AMnSVCAOFH/L

SVC2: AMnMBxADDRH/L + AMnSVCAOFH/L × 2

SVC3: AMnMBxADDRH/L + AMnSVCAOFH/L × 3

2) UV OFFSET

In the separate transfer of Y and CbCr (UV), the CbCr (UV) plane is transferred to the address obtained by adding AMnUVAOFH/L to AMnMBxADDRH/L, the image data transfer destination start address for SVC0.

UV plane for SVC0: AMnMBxADDRH/L + AMnUVAOFH/L

UV plane for SVC1: AMnMBxADDRH/L + AMnSVCAOFH/L + AMnUVAOFH/L

UV plane for SVC2: AMnMBxADDRH/L + AMnSVCAOFH/L × 2 + AMnUVAOFH/L

UV plane for SVC3: AMnMBxADDRH/L + AMnSVCAOFH/L × 3 + AMnUVAOFH/L

3) Non-Image Processing Data

Data are transferred to the address obtained by adding AMnNIAOFH/L to AMnMBxADDRH/L, the image data transfer destination start address for SVC0.

Non-image data plane for SVC0: AMnMBxADDRH/L + AMnNIAOFH/L

Non-image data plane for SVC1: AMnMBxADDRH/L + AMnSVCAOFH/L + AMnNIAOFH/L

Non-image data plane for SVC2: AMnMBxADDRH/L + AMnSVCAOFH/L × 2 + AMnNIAOFH/L

Non-image data plane for SVC3: AMnMBxADDRH/L + AMnSVCAOFH/L × 3 + AMnNIAOFH/L

4) Statistics Data

Statistics data are transferred through the static data (SD) bus, which is independent of the video data bus. Therefore, the destination addresses for these data are defined in the following form, regardless of the address settings for image data. In addition, such data are only transferred to the following set locations, since they are only supported for SVC0.

AMnSDMBxADDRH/L

(1) Video Data Transfer Destination Address (Video Data Storage Destination)

- The start address of each memory bank as the video data transfer destination is determined by the AMnMB1 to 8ADDRH and AMnMB1 to 8ADDRH registers. Valid AMnMB1ADDRH/L, AMnMB2ADDRH/L, ... registers are switched by AMnMBVALID. 1 to 8 can be set for one step of memory-bank transfer (examples for settings of 8, 4, and 1 are shown in **(a) AMnMBVALID[7:0] = 1111_1111b (eight transfer destinations)**, **(b) AMnMBVALID[7:0] = 0000_1111b (four transfer destinations)**, and **(c) AMnMBVALID[7:0] = 0000_0001b (one transfer destination)**, respectively).

Set the following offset addresses as required based on the above. Set the following offset addresses as required based on the above description.

- Set the SVC offset addresses, AMnSVCAOFL and AMnSVCAOFH, when SVC1 to 3 are used.
- Set the non-image processing offset addresses, AMnNIAOFL and AMnNIAOFH, when non-image processing/full bypass is selected.
- Set the YC offset addresses, AMnUVAOFL and AMnUVAOFH, when YC separation (ICnDMR.YCMODE[2:0] = 010b/110b) is selected.

(a) AMnMBVALID[7:0] = 1111_1111b (eight transfer destinations)

Order of transfer destinations: MB1 → MB2 → ... → MB8 → MB1 → MB2 → ...

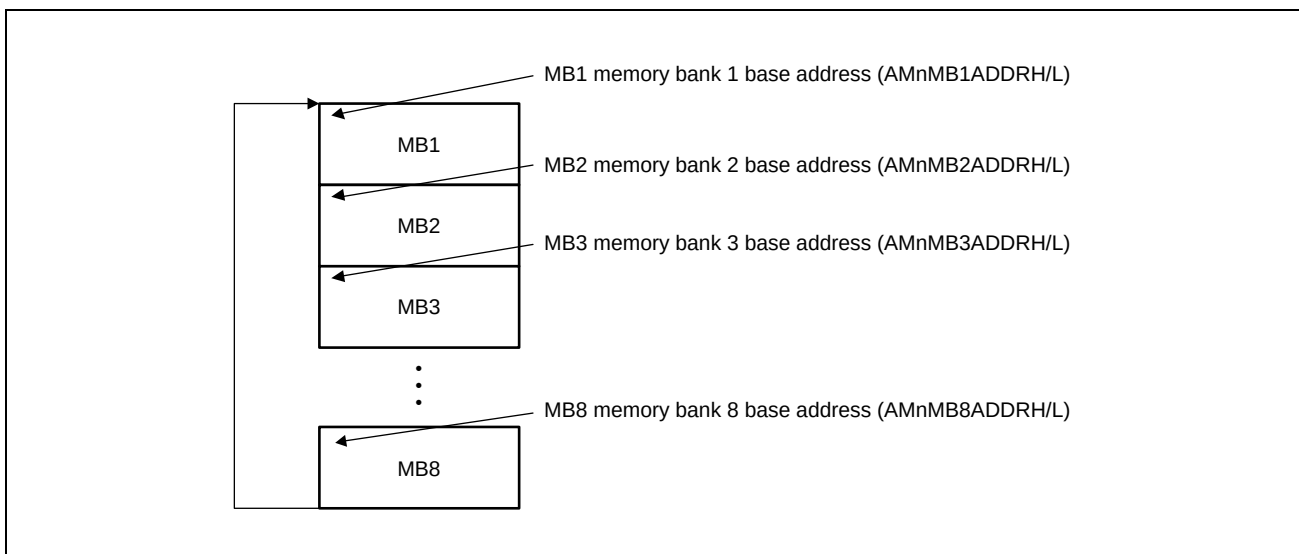


Figure 9.2-12 Image of Memory Bank (Eight Transfer Destinations)

(b) AMnMBVALID[7:0] = 0000_1111b (four transfer destinations)

Order of transfer destinations: MB1 → MB2 → MB3 → MB4 → MB1 → MB2 → ...

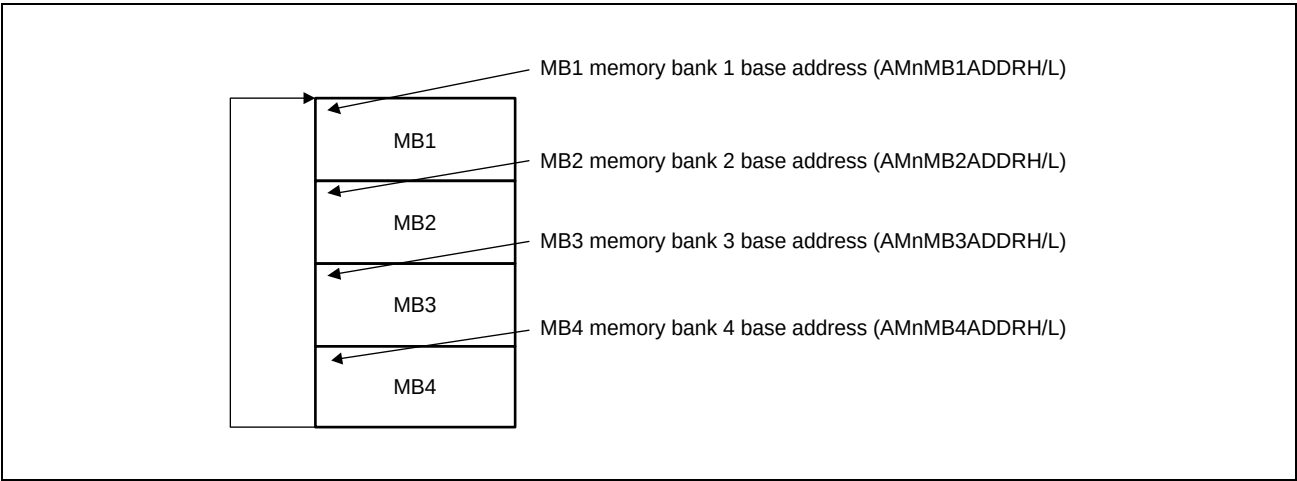


Figure 9.2-13 Image of Memory Bank (Four Transfer Destinations)

(c) AMnMBVALID[7:0] = 0000_0001b (one transfer destination)

Order of transfer destinations: MB1 → MB1 → MB1 → ...

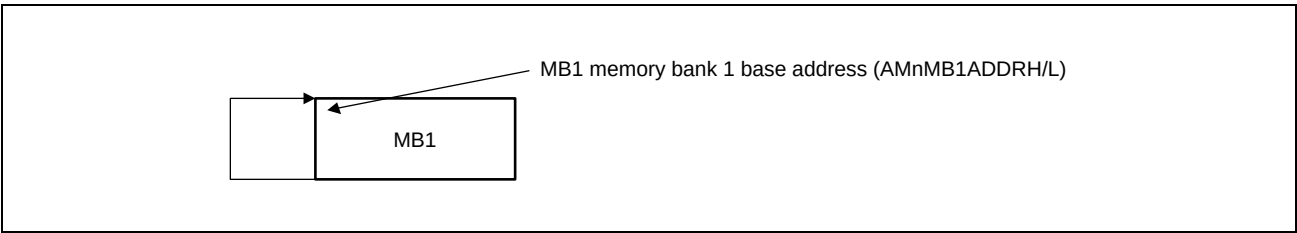


Figure 9.2-14 Image of Memory Bank (One Transfer Destination)

(2) Video Data Transfer Destination Address (Storage in Y/UV Separation)

The order of storing data in the MB1 transfer destination start address (AMnMB1ADDRH/L) in the case of Y-first (YUYV..) is shown below.

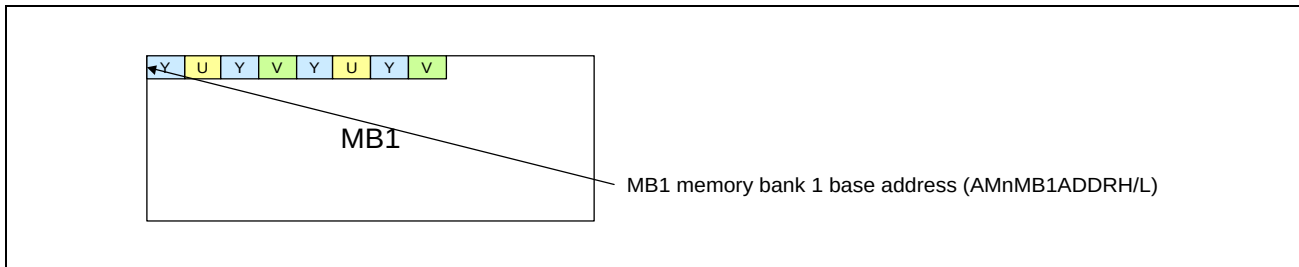
(a) When YUV is not separated (ICnDMR.YCMODE[2:0] = 000b to 001b, 100b to 101b)

Figure 9.2-15 Image of Data Storage (in the case of no separation of YUV)

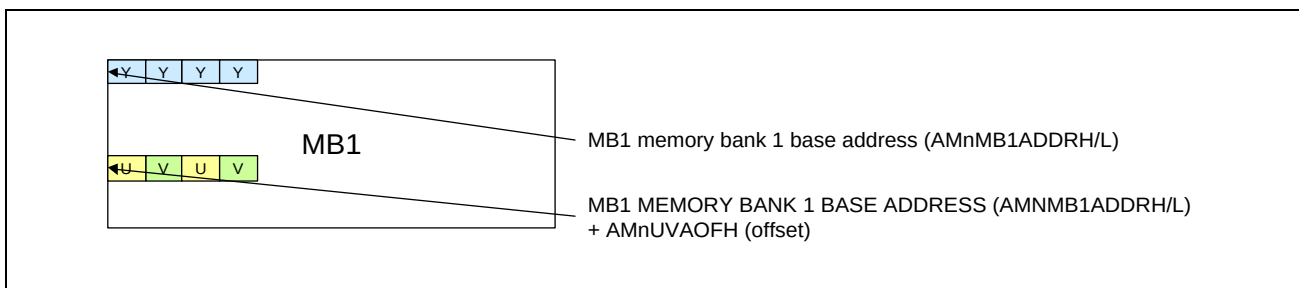
(b) When YUV is separated (ICnDMR.YCMODE[2:0] = 010b)

Figure 9.2-16 Image of Data Storage (in the case of separation of YUV)

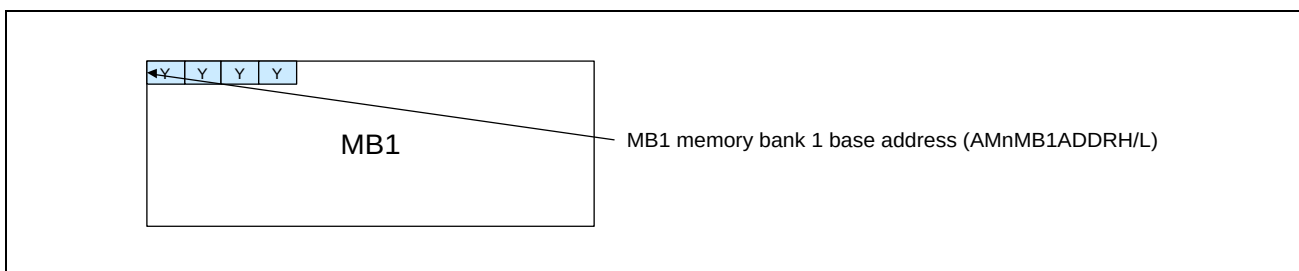
(c) When only Y is separated (ICnDMR.YCMODE[2:0] = 011b)

Figure 9.2-17 Image of Data Storage (in the case of only Y)

(3) Video Data Transfer Destination Address (Image Processing/Non-Image Processing, 4-ch. SVC, Y/UV Separated Data)

Figure 9.2-18 shows an image of memory to illustrate how image data of image processing/non-image processing, 4-ch. SVC, and Y/UV separated data are stored in memory.

● Storage location for SVC0		● Storage location for SVC1		● Storage location for SVC2		● Storage location for SVC3	
Address	Memory Content	Address	Memory Content	Address	Memory Content	Address	Memory Content
AMMB1ADDRHL	SVC0 MB1 Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*1	SVC1 MB1 Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*2	SVC2 MB1 Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*3	SVC3 MB1 Image Processing Data
AMMB1ADDRHL+AMnLVAOFHL	SVC0 MB1 Image Processing Data UV	AMMB1ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB1 Image Processing Data UV	AMMB1ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB1 Image Processing Data UV	AMMB1ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB1 Image Processing Data UV
AMMB1ADDRHL+AMnNIAOFHL	SVC0 MB1 Non-Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB1 Non-Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB1 Non-Image Processing Data	AMMB1ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB1 Non-Image Processing Data
AMMB2ADDRHL	SVC0 MB2 Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*1	SVC1 MB2 Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*2	SVC2 MB2 Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*3	SVC3 MB2 Image Processing Data
AMMB2ADDRHL+AMnLVAOFHL	SVC0 MB2 Image Processing Data UV	AMMB2ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB2 Image Processing Data UV	AMMB2ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB2 Image Processing Data UV	AMMB2ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB2 Image Processing Data UV
AMMB2ADDRHL+AMnNIAOFHL	SVC0 MB2 Non-Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB2 Non-Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB2 Non-Image Processing Data	AMMB2ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB2 Non-Image Processing Data
AMMB3ADDRHL	SVC0 MB3 Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*1	SVC1 MB3 Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*2	SVC2 MB3 Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*3	SVC3 MB3 Image Processing Data
AMMB3ADDRHL+AMnLVAOFHL	SVC0 MB3 Image Processing Data UV	AMMB3ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB3 Image Processing Data UV	AMMB3ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB3 Image Processing Data UV	AMMB3ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB3 Image Processing Data UV
AMMB3ADDRHL+AMnNIAOFHL	SVC0 MB3 Non-Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB3 Non-Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB3 Non-Image Processing Data	AMMB3ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB3 Non-Image Processing Data
AMMB4ADDRHL	SVC0 MB4 Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*1	SVC1 MB4 Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*2	SVC2 MB4 Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*3	SVC3 MB4 Image Processing Data
AMMB4ADDRHL+AMnLVAOFHL	SVC0 MB4 Image Processing Data UV	AMMB4ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB4 Image Processing Data UV	AMMB4ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB4 Image Processing Data UV	AMMB4ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB4 Image Processing Data UV
AMMB4ADDRHL+AMnNIAOFHL	SVC0 MB4 Non-Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB4 Non-Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB4 Non-Image Processing Data	AMMB4ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB4 Non-Image Processing Data
AMMB5ADDRHL	SVC0 MB5 Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*1	SVC1 MB5 Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*2	SVC2 MB5 Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*3	SVC3 MB5 Image Processing Data
AMMB5ADDRHL+AMnLVAOFHL	SVC0 MB5 Image Processing Data UV	AMMB5ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB5 Image Processing Data UV	AMMB5ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB5 Image Processing Data UV	AMMB5ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB5 Image Processing Data UV
AMMB5ADDRHL+AMnNIAOFHL	SVC0 MB5 Non-Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB5 Non-Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB5 Non-Image Processing Data	AMMB5ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB5 Non-Image Processing Data
AMMB6ADDRHL	SVC0 MB6 Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*1	SVC1 MB6 Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*2	SVC2 MB6 Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*3	SVC3 MB6 Image Processing Data
AMMB6ADDRHL+AMnLVAOFHL	SVC0 MB6 Image Processing Data UV	AMMB6ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB6 Image Processing Data UV	AMMB6ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB6 Image Processing Data UV	AMMB6ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB6 Image Processing Data UV
AMMB6ADDRHL+AMnNIAOFHL	SVC0 MB6 Non-Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB6 Non-Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB6 Non-Image Processing Data	AMMB6ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB6 Non-Image Processing Data
AMMB7ADDRHL	SVC0 MB7 Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*1	SVC1 MB7 Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*2	SVC2 MB7 Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*3	SVC3 MB7 Image Processing Data
AMMB7ADDRHL+AMnLVAOFHL	SVC0 MB7 Image Processing Data UV	AMMB7ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB7 Image Processing Data UV	AMMB7ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB7 Image Processing Data UV	AMMB7ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB7 Image Processing Data UV
AMMB7ADDRHL+AMnNIAOFHL	SVC0 MB7 Non-Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB7 Non-Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB7 Non-Image Processing Data	AMMB7ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB7 Non-Image Processing Data
AMMB8ADDRHL	SVC0 MB8 Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*1	SVC1 MB8 Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*2	SVC2 MB8 Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*3	SVC3 MB8 Image Processing Data
AMMB8ADDRHL+AMnLVAOFHL	SVC0 MB8 Image Processing Data UV	AMMB8ADDRHL+AMnSVCAOFHL*1+AMnLVAOFHL	SVC1 MB8 Image Processing Data UV	AMMB8ADDRHL+AMnSVCAOFHL*2+AMnLVAOFHL	SVC2 MB8 Image Processing Data UV	AMMB8ADDRHL+AMnSVCAOFHL*3+AMnLVAOFHL	SVC3 MB8 Image Processing Data UV
AMMB8ADDRHL+AMnNIAOFHL	SVC0 MB8 Non-Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*1+AMnNIAOFHL	SVC1 MB8 Non-Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*2+AMnNIAOFHL	SVC2 MB8 Non-Image Processing Data	AMMB8ADDRHL+AMnSVCAOFHL*3+AMnNIAOFHL	SVC3 MB8 Non-Image Processing Data

Figure 9.2-18 Image of Data Storage (Image Processing/Non-Image Processing, 4-ch. SVC, and Y/UV Separated Data)

(a) Example of Division by VC and by MB

Figure 9.2-19 shows an image of memory when data are to be stored separately by VC and by memory bank.

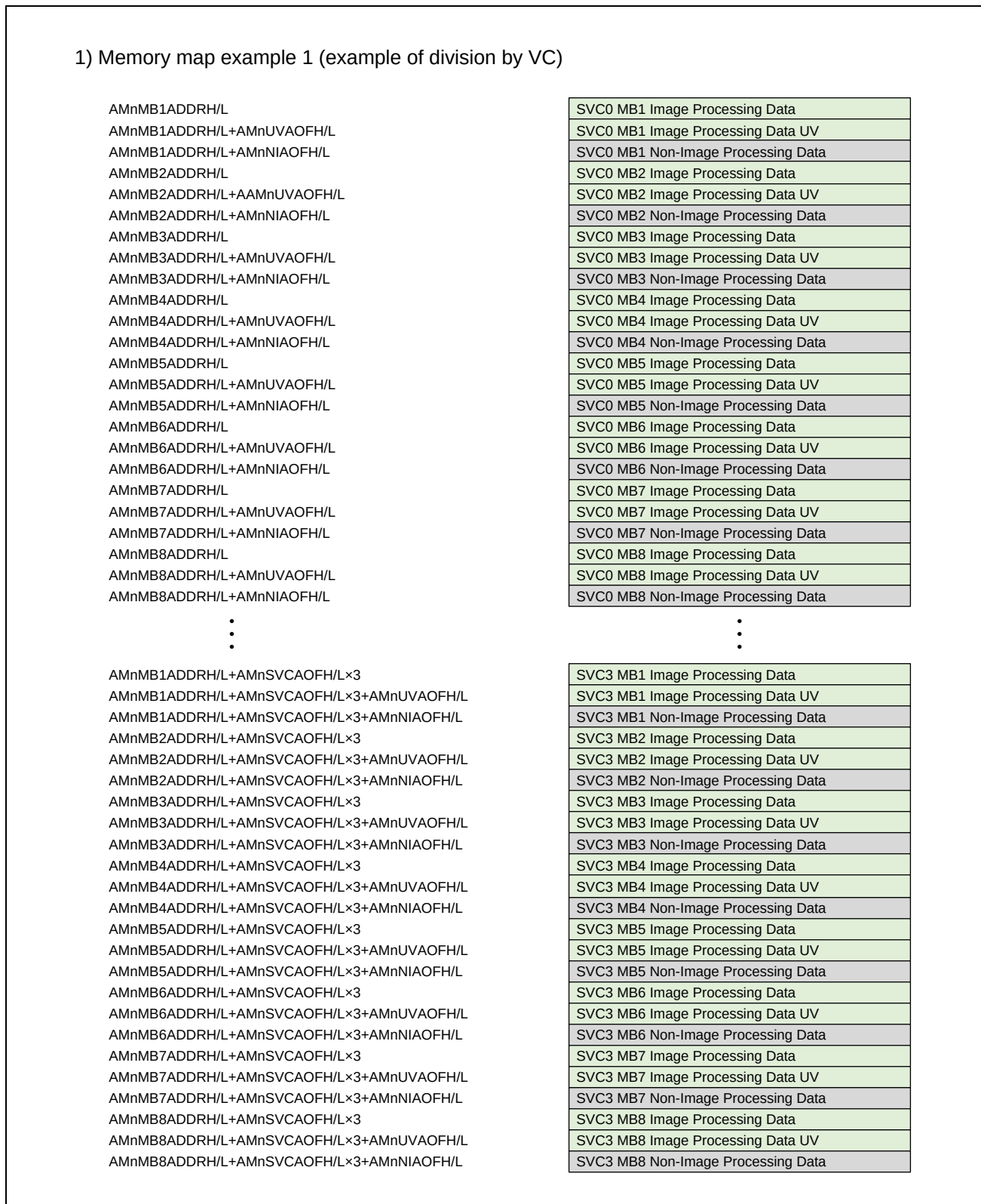


Figure 9.2-19 Image of Data Storage (Example of Division by VC)

2) Memory map example 2 (example of division by MB)

AMnMB1ADDRH/L	SVC0 MB1 Image Processing Data
AMnMB1ADDRH/L+AMnUVAOFH/L	SVC0 MB1 Image Processing Data UV
AMnMB1ADDRH/L+AMnNIAOFH/L	SVC0 MB1 Non-Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×1	SVC1 MB1 Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×1+AMnUVAOFH/L	SVC1 MB1 Image Processing Data UV
AMnMB1ADDRH/L+AMnSVCAOFH/L×1+AMnNIAOFH/L	SVC1 MB1 Non-Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×2	SVC2 MB1 Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×2+AMnUVAOFH/L	SVC2 MB1 Image Processing Data UV
AMnMB1ADDRH/L+AMnSVCAOFH/L×2+AMnNIAOFH/L	SVC2 MB1 Non-Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×3	SVC3 MB1 Image Processing Data
AMnMB1ADDRH/L+AMnSVCAOFH/L×3+AMnUVAOFH/L	SVC3 MB1 Image Processing Data UV
AMnMB1ADDRH/L+AMnSVCAOFH/L×3+AMnNIAOFH/L	SVC3 MB1 Non-Image Processing Data
AMnMB2ADDRH/L	SVC0 MB2 Image Processing Data
AMnMB2ADDRH/L+AMnUVAOFH/L	SVC0 MB2 Image Processing Data UV
AMnMB2ADDRH/L+AMnNIAOFH/L	SVC0 MB2 Non-Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×1	SVC1 MB2 Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×1+AMnUVAOFH/L	SVC1 MB2 Image Processing Data UV
AMnMB2ADDRH/L+AMnSVCAOFH/L×1+AMnNIAOFH/L	SVC1 MB2 Non-Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×2	SVC2 MB2 Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×2+AMnUVAOFH/L	SVC2 MB2 Image Processing Data UV
AMnMB2ADDRH/L+AMnSVCAOFH/L×2+AMnNIAOFH/L	SVC2 MB2 Non-Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×3	SVC3 MB2 Image Processing Data
AMnMB2ADDRH/L+AMnSVCAOFH/L×3+AMnUVAOFH/L	SVC3 MB2 Image Processing Data UV
AMnMB2ADDRH/L+AMnSVCAOFH/L×3+AMnNIAOFH/L	SVC3 MB2 Non-Image Processing Data
⋮	⋮
AMnMB7ADDRH/L	SVC0 MB7 Image Processing Data
AMnMB7ADDRH/L+AMnUVAOFH/L	SVC0 MB7 Image Processing Data UV
AMnMB7ADDRH/L+AMnNIAOFH/L	SVC0 MB7 Non-Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×1	SVC1 MB7 Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×1+AMnUVAOFH/L	SVC1 MB7 Image Processing Data UV
AMnMB7ADDRH/L+AMnSVCAOFH/L×1+AMnNIAOFH/L	SVC1 MB7 Non-Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×2	SVC2 MB7 Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×2+AMnUVAOFH/L	SVC2 MB7 Image Processing Data UV
AMnMB7ADDRH/L+AMnSVCAOFH/L×2+AMnNIAOFH/L	SVC2 MB7 Non-Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×3	SVC3 MB7 Image Processing Data
AMnMB7ADDRH/L+AMnSVCAOFH/L×3+AMnUVAOFH/L	SVC3 MB7 Image Processing Data UV
AMnMB7ADDRH/L+AMnSVCAOFH/L×3+AMnNIAOFH/L	SVC3 MB7 Non-Image Processing Data
AMnMB8ADDRH/L	SVC0 MB8 Image Processing Data
AMnMB8ADDRH/L+AMnUVAOFH/L	SVC0 MB8 Image Processing Data UV
AMnMB8ADDRH/L+AMnNIAOFH/L	SVC0 MB8 Non-Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×1	SVC1 MB8 Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×1+AMnUVAOFH/L	SVC1 MB8 Image Processing Data UV
AMnMB8ADDRH/L+AMnSVCAOFH/L×1+AMnNIAOFH/L	SVC1 MB8 Non-Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×2	SVC2 MB8 Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×2+AMnUVAOFH/L	SVC2 MB8 Image Processing Data UV
AMnMB8ADDRH/L+AMnSVCAOFH/L×2+AMnNIAOFH/L	SVC2 MB8 Non-Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×3	SVC3 MB8 Image Processing Data
AMnMB8ADDRH/L+AMnSVCAOFH/L×3+AMnUVAOFH/L	SVC3 MB8 Image Processing Data UV
AMnMB8ADDRH/L+AMnSVCAOFH/L×3+AMnNIAOFH/L	SVC3 MB8 Non-Image Processing Data

Figure 9.2-20 Image of Data Storage (Example of Division by MB)

(4) Statistics Data Transfer Destination Address (Statistics Data Storage Destination)

The results of statistics processing are transferred from the statistics data bus to external memory. Since this statistics function is limited to SVC0, the external transfer destination start address is determined by AMnSDMB1 to 8ADDRH and AMnSDMB1 to 8ADDRH.

The number of planes of the memory bank is determined by AMnSDMBVALID. The transfer start address (statistics data storage destination) is determined by the MnSDMB1 to 8ADDRH and AMnSDMB1 to 8ADDRH registers.

The valid AMnSDMB1ADDR, AMnSDMB2ADDR, ... registers are switched by AMnSDMBVALID.

(a) AMnSDMBVALID[7:0] = 1111_1111b (eight transfer destinations)

- Order of transfer destinations: SDMB1 → SDMB2 → ... → SDMB8 → SDMB1 → SDMB2 → ...

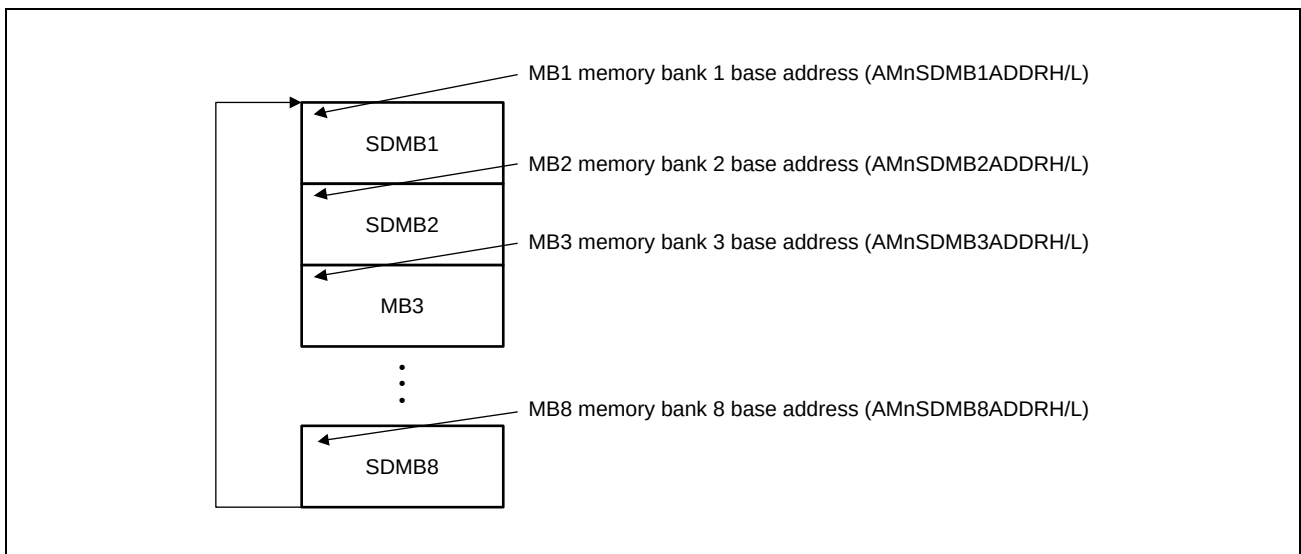


Figure 9.2-21 Image of Memory Bank (Eight Transfer Destinations)

(5) Frame Structure

(a) Video data frame

Image Stride

The image stride function is always handled for image processing and non-image processing from the video data system.

Figure 9.2-22 through **Figure 9.2-24** show the image of the storage of data with the image stride function. The image stride function involves writing from the memory write start address of each line in memory in units of address with the value set by the AMnIS (128-byte Step) register. The end of the line for writing is determined in multiples of 256 bits (32 bytes), so take this into account when setting the address. Furthermore, if a header and footer are to be appended, make the setting in consideration of that amount of data.

The following are the descriptions of storage in memory for each output format.

TYPE1: When the output format is YUV without separation (ICnDMR.YCMODE[2:0] = 000b to 001b, 011b to 101b, 111b), RGB, RAW or MIPI Recommended Memory Storage Format.

As shown in **Figure 9.2-22**, the first address of memory write for each horizontal line is written to memory in address units.

TYPE2: When the output format is YUV422 (ICnDMR.YCMODE[2:0] = 010b)

As shown in **Figure 9.2-23**, the first address of memory write for each horizontal line is written to memory in address units.

TYPE3: When the output format is YUV420 (ICnDMR.YCMODE[2:0] = 110b)

As shown in **Figure 9.2-24**, Y writes the first address of memory write for each line to memory in address units of the value set in the AMnIS register.

Note that UV writes the first address of each line to memory in units of twice the value set in the AMnIS register.

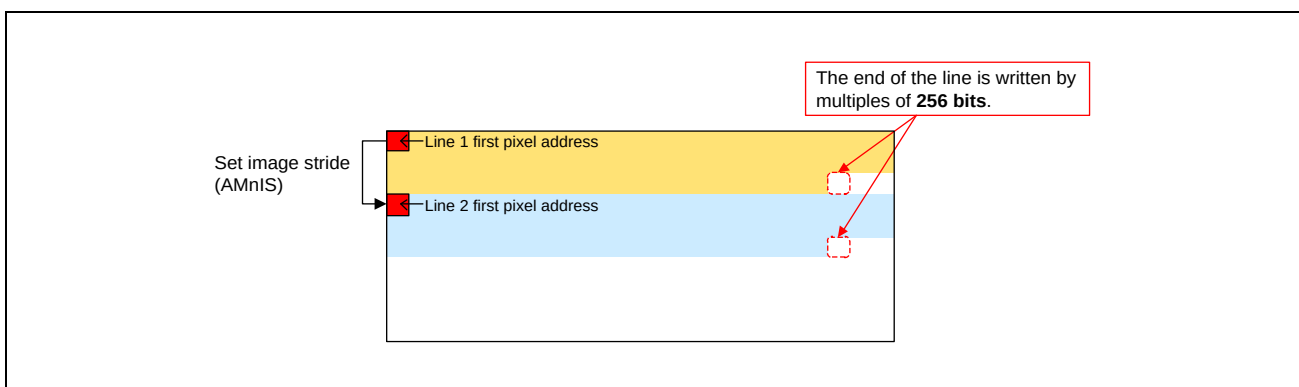


Figure 9.2-22 Image of Data Storage with Image Stride Function (TYPE1)

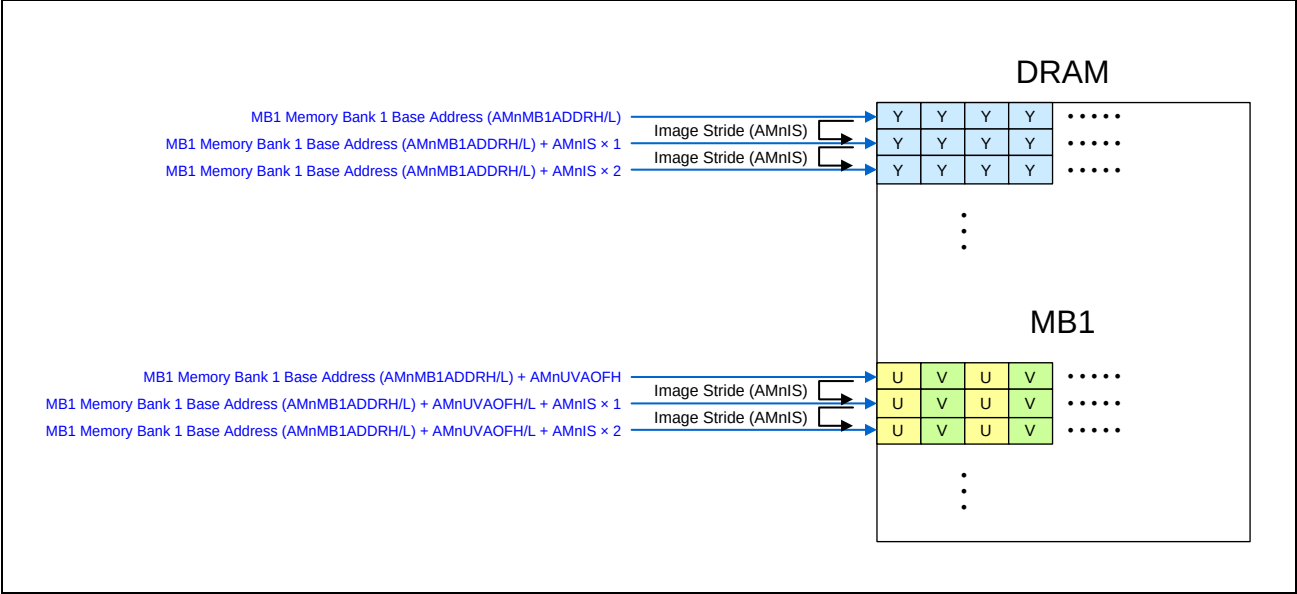


Figure 9.2-23 Image of Data Storage with Image Stride Function (TYPE2)

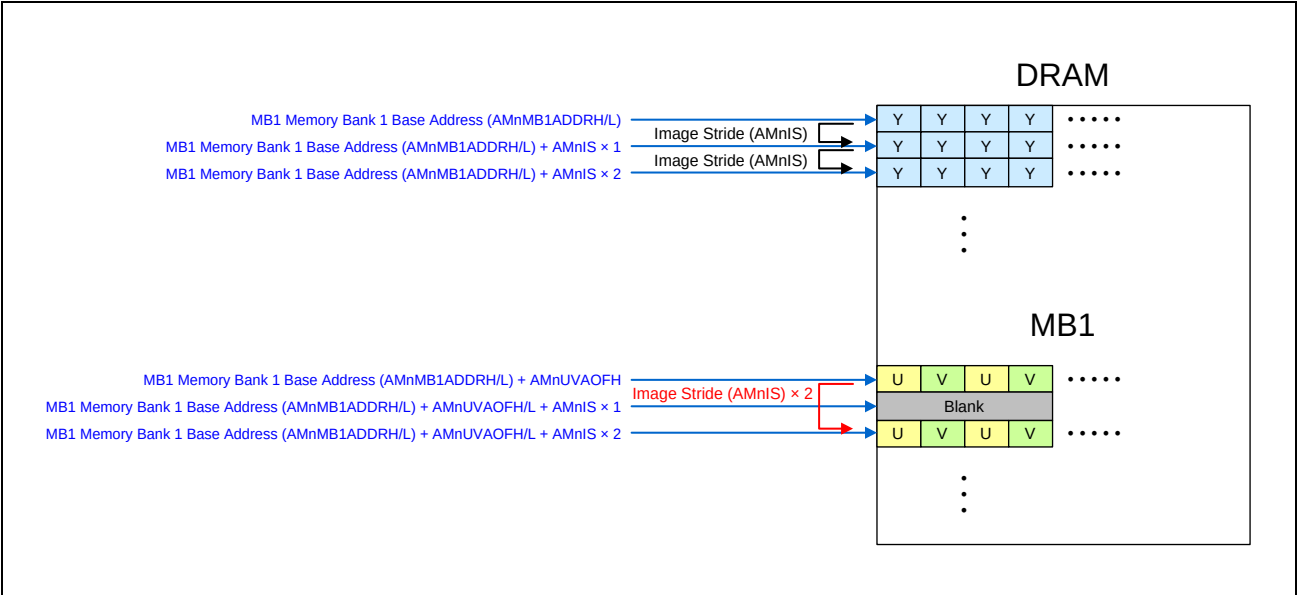


Figure 9.2-24 Image of Data Storage with Image Stride Function (TYPE3)

Image Processing Data Frame

Figure 9.2-25 shows the structure of the image processing data frame.

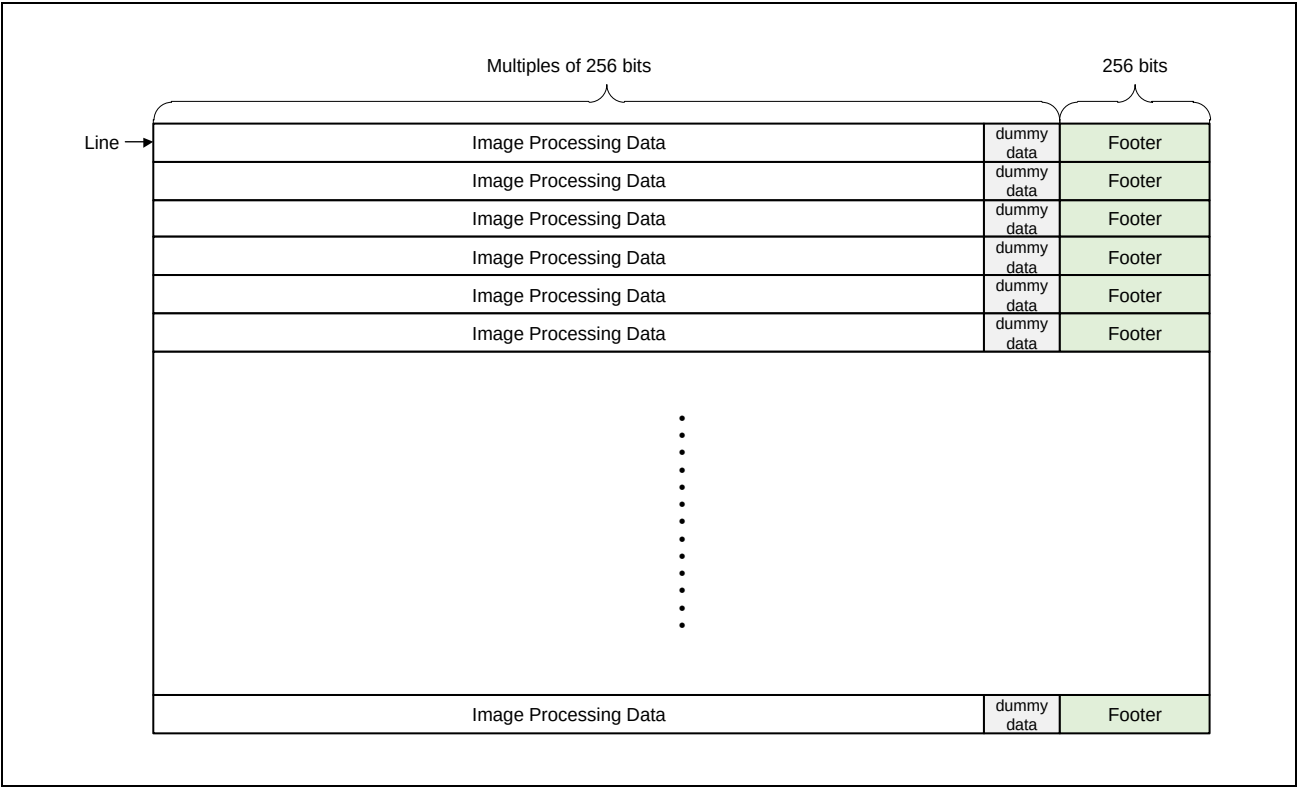


Figure 9.2-25 Image Processing Data Frame Structure

1) Dummy data

Transfer from this module to external memory proceeds in units of 256 bits (32 bytes). Therefore, only the amount of dummy data which is required to reach a 32-byte boundary is transferred. When processing images, ignore the dummy data.

2) Footer

Whether or not to insert footer with the image data can be selected by using the ICnDTVP.FTON register. For the data format of 256 bits of footer data, see **Footer Appending Function**. The value of this footer is updated alternately every time data at the target address are updated. The state of progress can be checked by monitoring which horizontal line has been updated.

This module has functionality for checking the address for transfer from the AXI bus via a register or using an interrupt to indicate the transfer of data to the specified address. Actually, whether or not data have been written to the DRAM can be checked by inspecting the footer to confirm the updating of data.

Non-Image Processing Data/Full-Bypass Data Frame

Figure 9.2-26 shows the structure of the non-image processing data/full-bypass data frame.

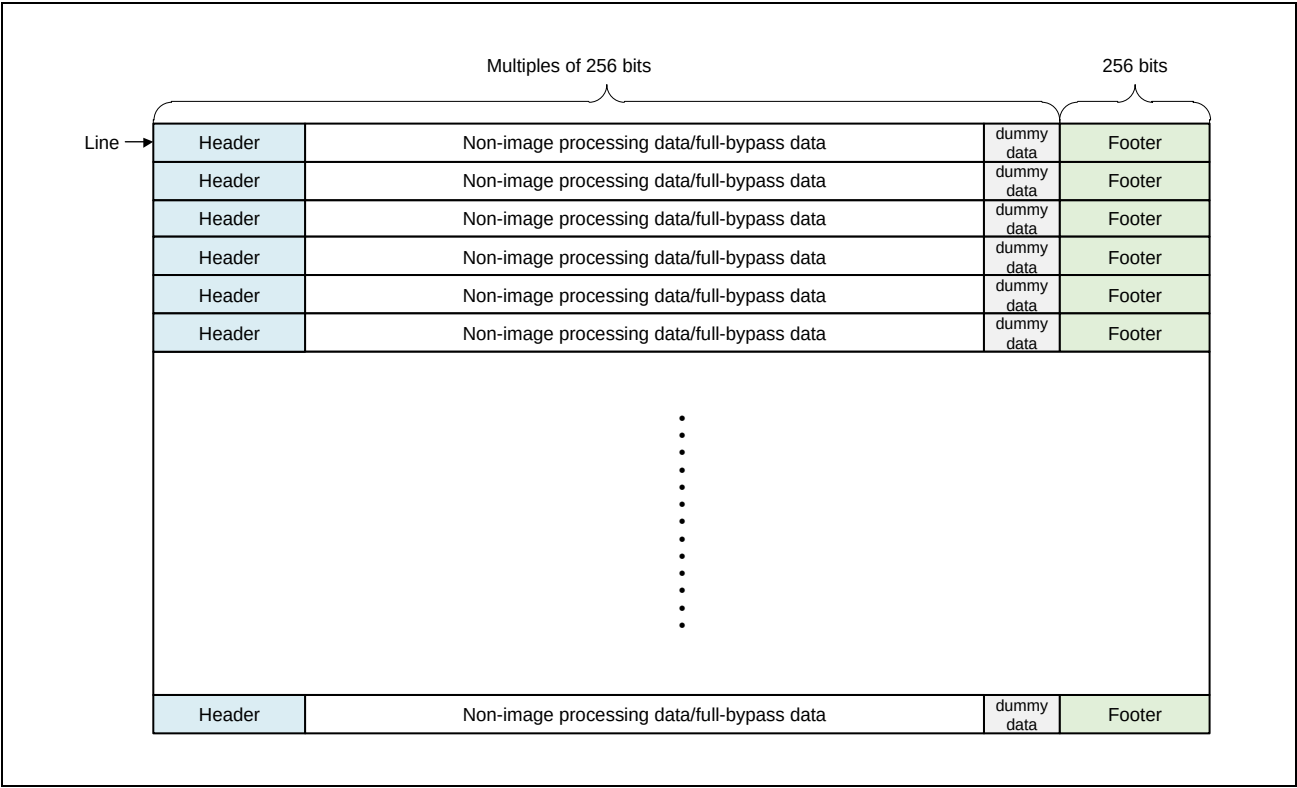


Figure 9.2-26 Non-Image Processing Data/Full-Bypass Data Frame Structure

A difference from an image processing frame is that a header can be inserted for these frame types.

Insertion of the header can be set by using the ICnDTVP.HDON register. For the format of the header, see **Header Appending Function (for Non-Image Processing Data/Full-Bypass Data)**.

Since there are several data types for non-image processing and full-bypass data whereas image processing data are of one data type, the purpose of the header is to allow the CRU to identify the data type in line units. If the CRU has recognized the data type beforehand or a single data type is selected, headers need not be inserted.

Header Appending Function (for Non-Image Processing Data/Full-Bypass Data)

The data types for non-image processing and full-bypass are selectable from among several data types and the data are stored in a state such that they are mixed in the same SVC memory area, so a header can be appended at the start of each line for non-image processing or full-bypass data so that the data type and data length can be identified when read. Since this header appending function is not required for a sensor that can identify the data types by their order, etc., it can be switched on or off.

However, when the pattern generator is selected for output, a header will not be appended regardless of whether the setting for header appending is on or off.

Figure 9.2-27 shows the contents of the header.

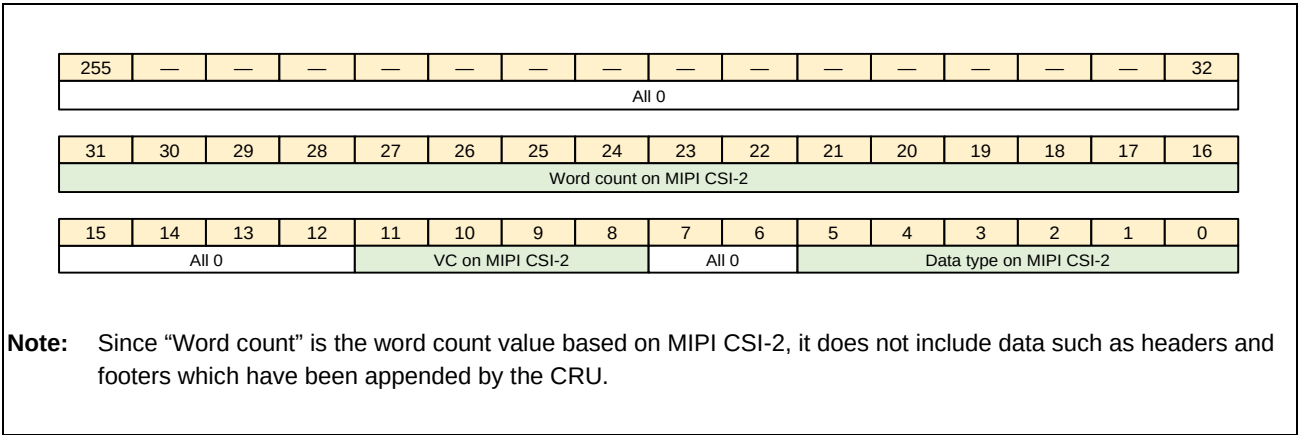


Figure 9.2-27 Header Contents

Footer Appending Function

This function writes footer data to memory so that the user can see up to which horizontal line data has been written in the memory bank set by AMnMB1-8ADDRH/L (for video data).

This function can be switched on or off with ICnDTVP.FTON.

For the video data footer, 256 bits (256'h1/256'h0) are written at the end of each line unit.

At first, 256'h1 is written as a footer. The next time the memory bank is overwritten after round of writing, 256'h0 is written as a footer. Subsequently, 256'h1 and 256'h0 are written alternately each time the memory bank is overwritten.

Accordingly, by checking the value of the footer, it is possible to prevent erroneous use of data before it is overwritten.

A 256-bit footer (256'd1/256'd0) is appended to the end of each line unit from the image system (image processing, non-image processing, or full bypass) and to the end of each frame unit from the statistics system.

(b) Statistics data frame

Figure 9.2-28 shows the structure of the image processing data frame.

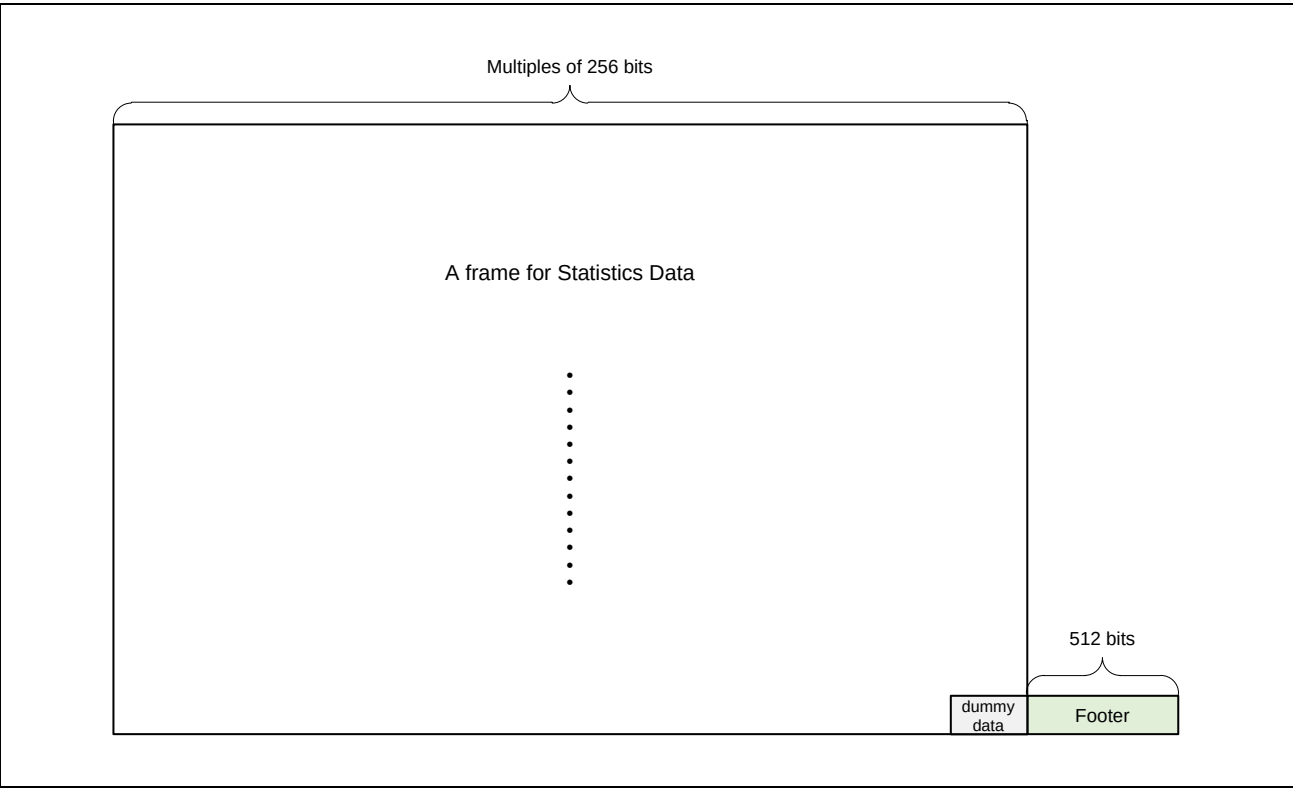


Figure 9.2-28 Statistics Data Frame Structure

Two differences between statistics frames and frames from image processing and non-image processing are that statistics frames have no concept of image striding and that footers are added in frame units. This is because statistics are processed in block units so that line-by-line processing is not possible.

Processing of the Last Line in a Frame (for Statistics Data)

Figure 9.2-29 shows an image of storing frames of statistics data. Statistics data are written to memory with no intervals in the data to reflect line transitions. The end of the frame for writing is determined in multiples of 256 bits (32 bytes), so take this into account when setting the address.

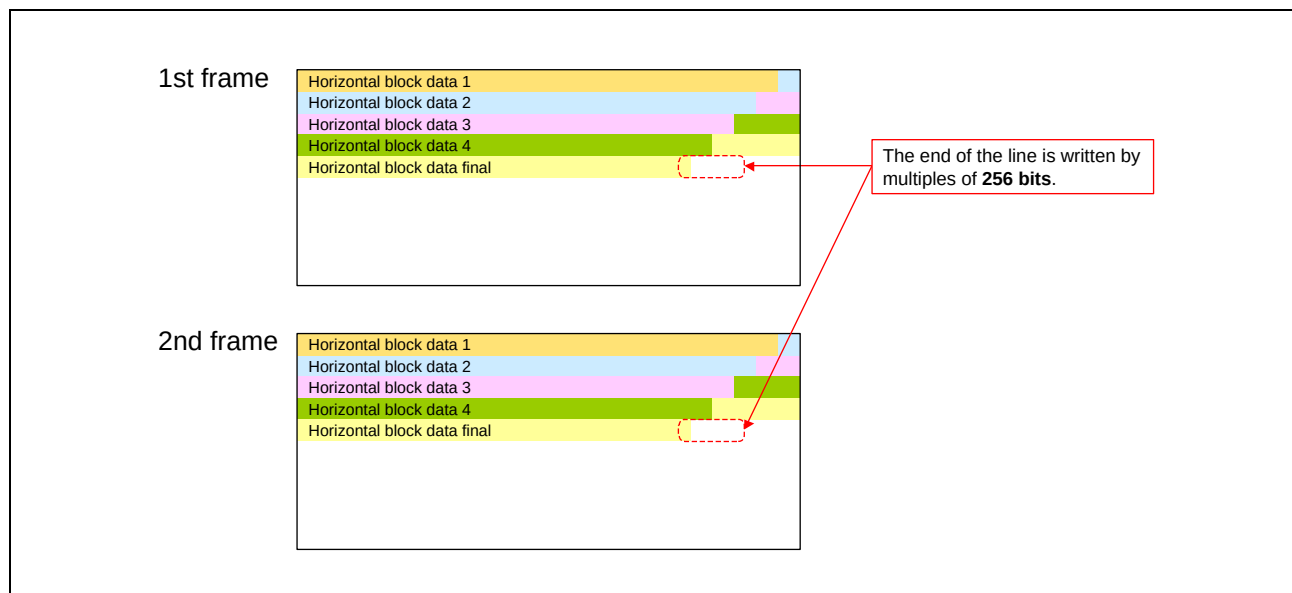


Figure 9.2-29 Image of Storing Frames of Statistics Data

Footer Appending Function

This function writes a footer to the end of each memory so that the user can see which bank number of the memory bank set by MnSDMB1-8ADDRH/L (for statistics data) has been written to the memory.

This function can be switched on or off with ICnDTVP.FTON.

A 512-bit ({higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} or 512-bit {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0}) footer of statistics data is written at the end of each frame unit.

Ignore the higher-order 256 bits of 512 bits because they are reserved (indefinite value).

At first, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} are written as a footer.

The next time the memory bank is overwritten after one round of writing, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0} are written as a footer.

After that, 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h1} and 512 bits {higher-order 256 bits: reserved (indefinite value); lower-order 256 bits: 256'h0} are alternately written each time overwriting is done.

Therefore, by checking the footer value as the video data footer, it is possible to prevent the data from being inadvertently used before overwriting.

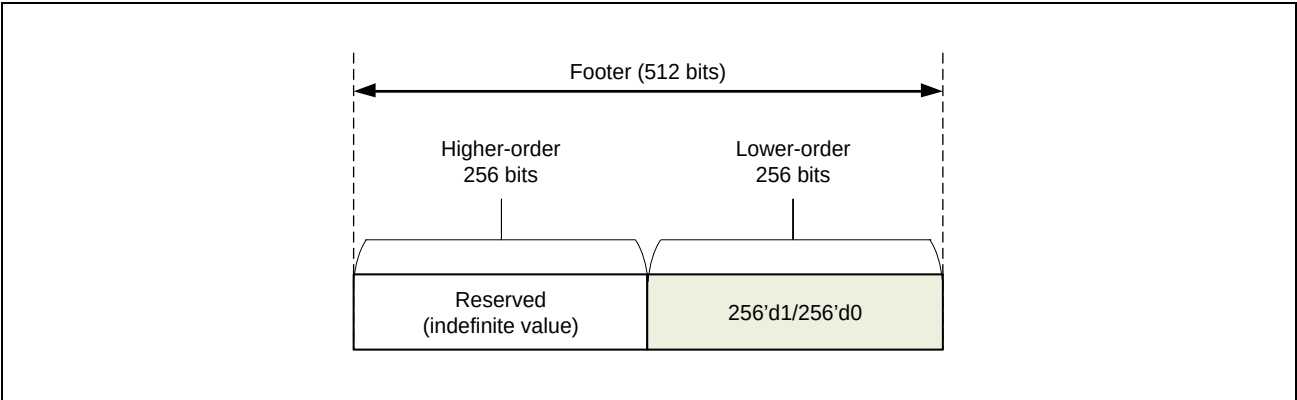


Figure 9.2-30 Statistics Data Footer Structure

9.2.4.3.4 Select Channel MIPI Data Input

This function selects signals for input to the image converter as link data for the given channels, link data for the paired channel, or pattern generator data. Selection of the inputs as link data for the given channel or link data for the paired channel can be set by using the CRUnCTRL.VINSEL registers. However, as shown below, paired channels are only selectable for CRU1 and the given channels are only selectable for CRU0. If you want to select the pattern generator data input, set this by using the ICnTICTRL1.TIEN register (for details of pattern generator data selection, see

9.2.4.3.5 Pattern Generator (for Debugging)).

Here, examples of using the input of MIPI data from the paired channels are shown in **Figure 9.2-31**. This function is used in the following cases that cannot be handled by only one image converter.

- Since a single channel only supports up to four VCs, when five or more VCs are required
- When demosaicing and statistics are to be applied to two raw data sets because only one demosaicing and one statistics section are present.

The examples shown in the figure are for when 8 VCs are to be handled.

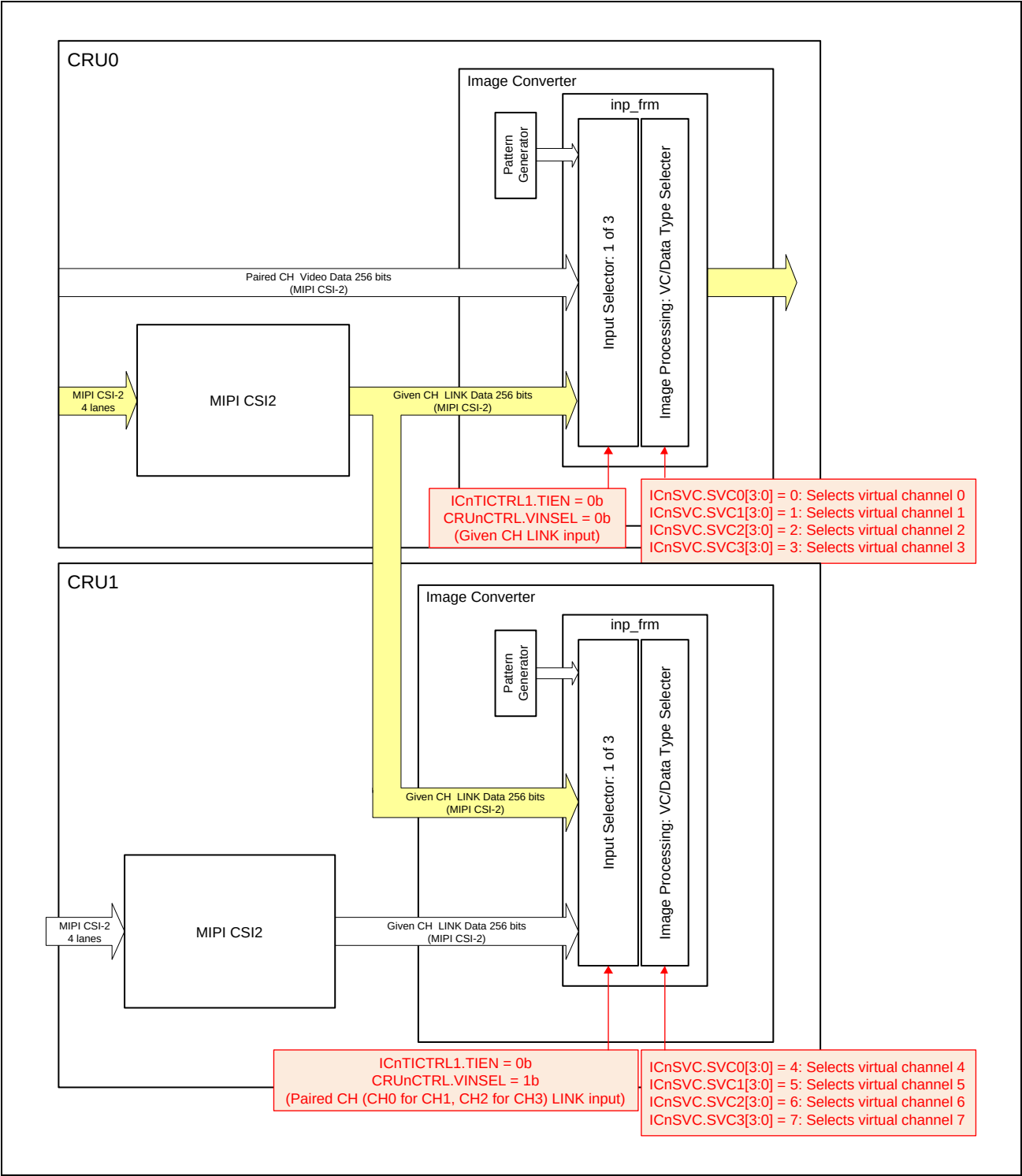


Figure 9.2-31 Example when 8 VCs are to be Handled

9.2.4.3.5 Pattern Generator (for Debugging)

A test image (YUV422 only) can be generated for use as an image converter input by setting the following registers. This function is used for debugging to identify where the problem is originated (in the sensor or in the internal settings). Two modes are implemented for this pattern (monochrome or pattern). This pattern generator operates as VC0. You can select the route of image processing/non-image processing. See **9.2.5 Operation** for the setting procedure.

TIEN = ICnTICTRL1.TIEN: Enables generation of a test image pattern.

TIMODE = ICnTICTRL1.TIMODE: Selects a test image generation pattern (monochrome or pattern).

TIROTSSEL = ICnTICTRL1.TIROTSSEL: Selects the route of image processing/non-image processing.

TIPTNY1[3:0] = ICnTICTRL1.TIPTNY1[3:0]: Bit settings for the pixel (Y) counter

TIPTNU1[3:0] = ICnTICTRL1.TIPTNU1[3:0]: Bit settings for the pixel (U) counter

TIPTNV1[3:0] = ICnTICTRL1.TIPTNV1[3:0]: Bit settings for the pixel (V) counter

TIRATE[4:0] = ICnTICTRL1.TIRATE[4:0]: Specify a test image generation rate.

TIPTNY2[7:0] = ICnTICTRL2.TIPTNY2[7:0]: Specify V for the entire-screen same-color test image pattern.

TIPTNU2[7:0] = ICnTICTRL2.TIPTNU2[7:0]: Specify U for the entire-screen same-color test image pattern.

TIPTNV2[7:0] = ICnTICTRL2.TIPTNV2[7:0]: Specify Y for the entire-screen same-color test image pattern.

TIPPL[11:0] = ICnTISIZE1.TIPPL[11:0]: Number of valid pixels per line in the test image pattern

TIN[11:0] = ICnTISIZE2.TIN[11:0]: Valid period (number of lines) per frame in the test image pattern

TIM[11:0] = ICnTISIZE2.TIM[11:0]: Invalid period (number of lines) per frame in the test image pattern

(1) Image and Settings (Common to Monochrome and Pattern Generation)

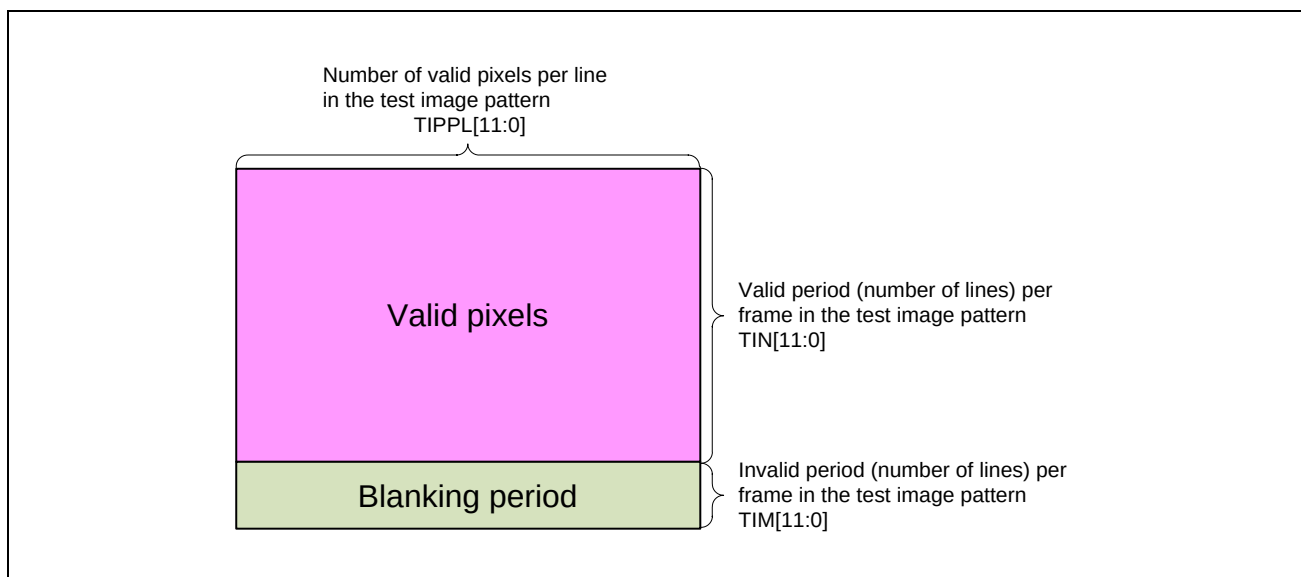


Figure 9.2-32 Image and Settings

(2) Timing Generation and Settings (Common to Monochrome and Pattern Generation)

(Bits TIRATE[4:0] allow the test image generation rate to be set in the range between 3.291 Gbps and 8.960 Gbps with VCLK at 630 MHz.)

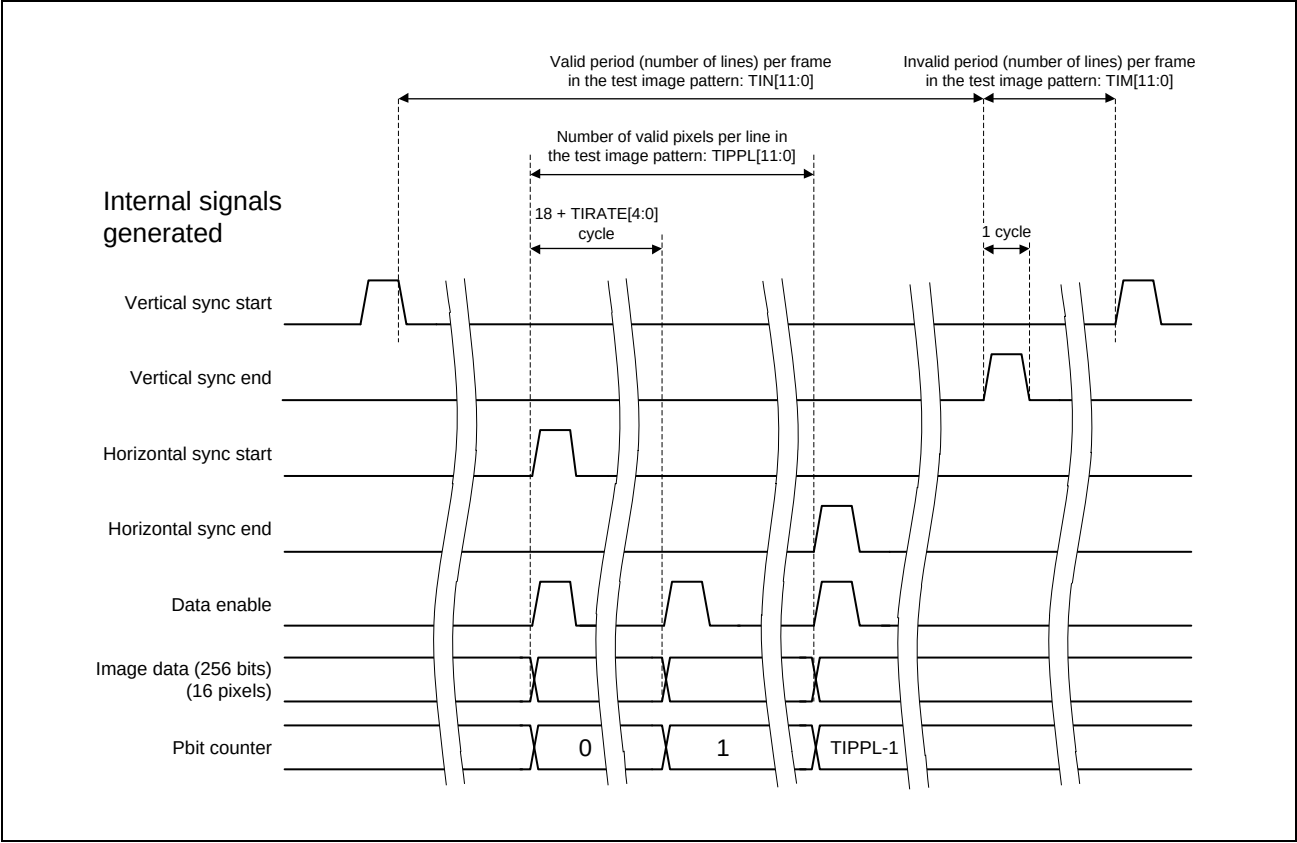


Figure 9.2-33 Timing Generation and Settings

(3) Data Generation Circuit (in the Case of Pattern Generation)

Image YUV422 8-bit data of 256 bits (16 pixels) is generated as follows (other formats cannot be generated). The 16-bit Pbit counter is incremented in $18 + \text{TIRATE}[4:0]$ (cycles at 630 MHz). The bits of this Pbit counter are selected by the TIPTNV1[3:0], TIPTNU1[3:0], and TIPTNY1[3:0] bits, and then they are packed and sent as follows:

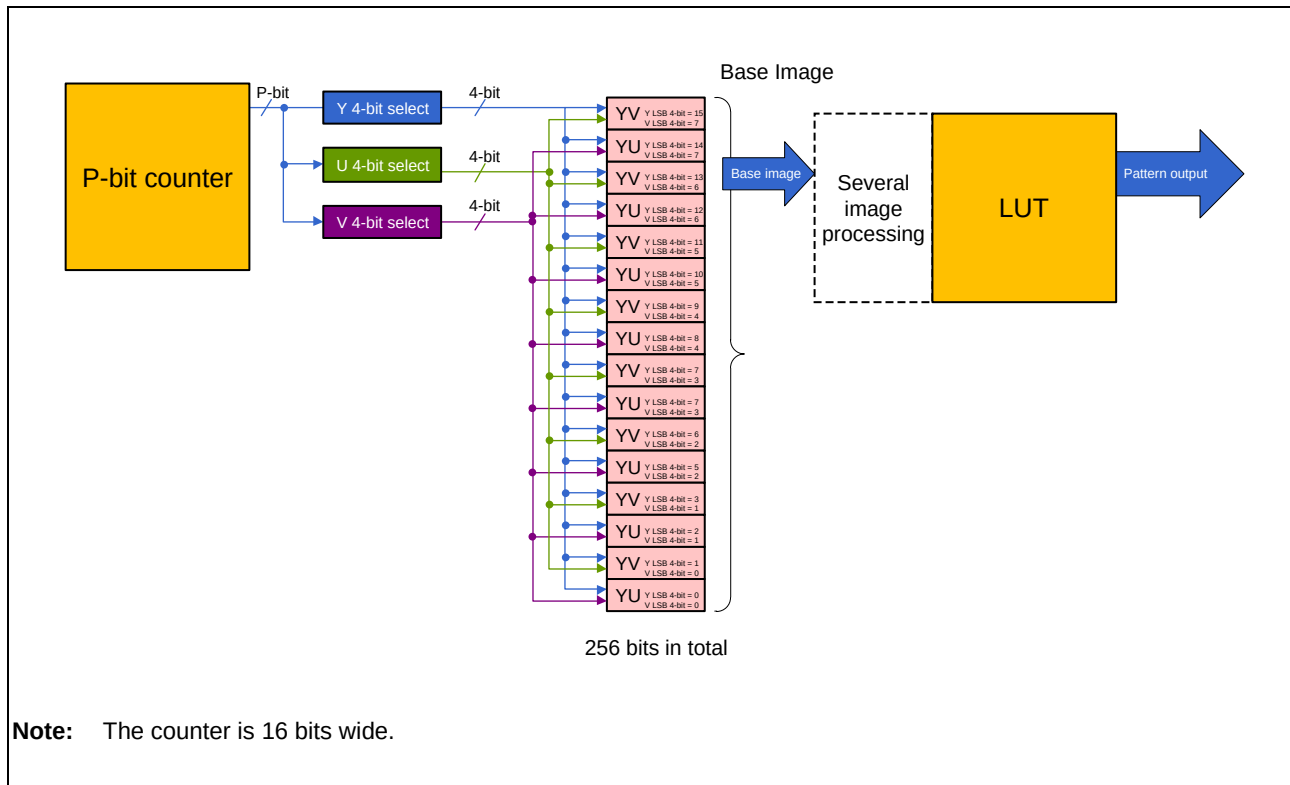


Figure 9.2-34 Data Generation Circuit (in the Case of Pattern Generation)

(4) Data Generation Circuit (in the Case of Monochrome Generation)

Image YUV422 8-bit data of 256 bits (16 pixels) is generated in monochrome as follows (other formats cannot be generated). The values set by the TIPTNV2[7:0], TIPTNU2[7:0], and TIPTNY2[7:0] bits are packed and sent as follows:

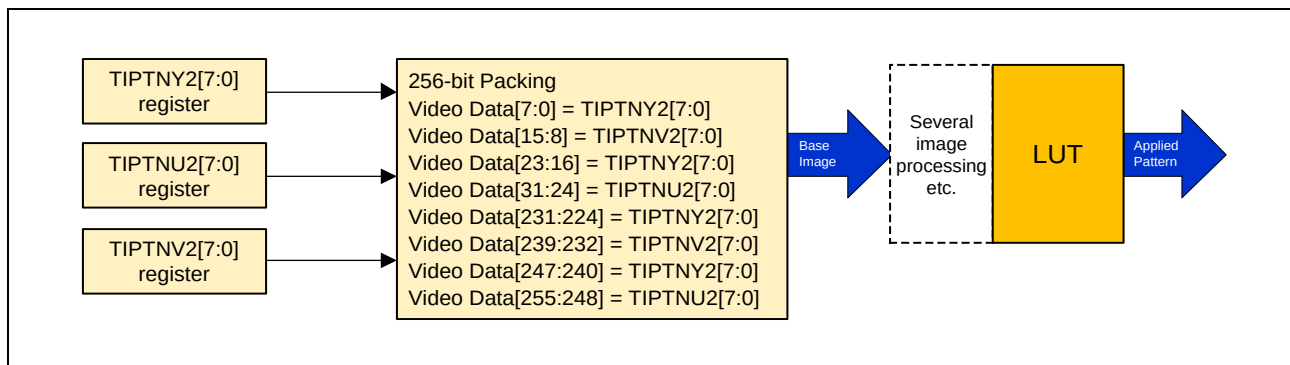


Figure 9.2-35 Data Generation Circuit (in the Case of Monochrome Generation)

(5) Generated Image Data Example (in the Case of Pattern Generation)

VGA (640 × 480): TIEN = 1b, TIMODE = 0b, TIPTNY1[3:0] = 0000b, TIPTNU1[3:0] = 0100b, TIPTNV1[3:0] = 1000b (Offset binary: ICnIPMC_C0/1/2/3.IBINSEL = 0b, ICnDMR.OBINSEL = 0b)

Table 9.2-18 Expected Values

No	Y	U	V
0	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
16	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
32	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
48	30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
64	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
80	50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
96	60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
112	70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
128	80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
144	90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
160	A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, AA, AB, AC, AD, AE, AF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
176	B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, BA, BB, BC, BD, BE, BF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
192	C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, CA, CB, CC, CD, CE, CF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
208	D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, DA, DB, DC, DD, DE, DF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
224	E0, E1, E2, E3, E4, E5, E6, E7, E8, E9, EA, EB, EC, ED, EE, EF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
240	F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, FC, FD, FE, FF,	10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, 13, 14, 15, 16, 17,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
256	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F,	20, 21, 22, 23, 24, 25, 26, 27, 20, 21, 22, 23, 24, 25, 26, 27,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
272	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F,	20, 21, 22, 23, 24, 25, 26, 27, 20, 21, 22, 23, 24, 25, 26, 27,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,
288	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F,	20, 21, 22, 23, 24, 25, 26, 27, 20, 21, 22, 23, 24, 25, 26, 27,	00, 01, 02, 03, 04, 05, 06, 07, 00, 01, 02, 03, 04, 05, 06, 07,

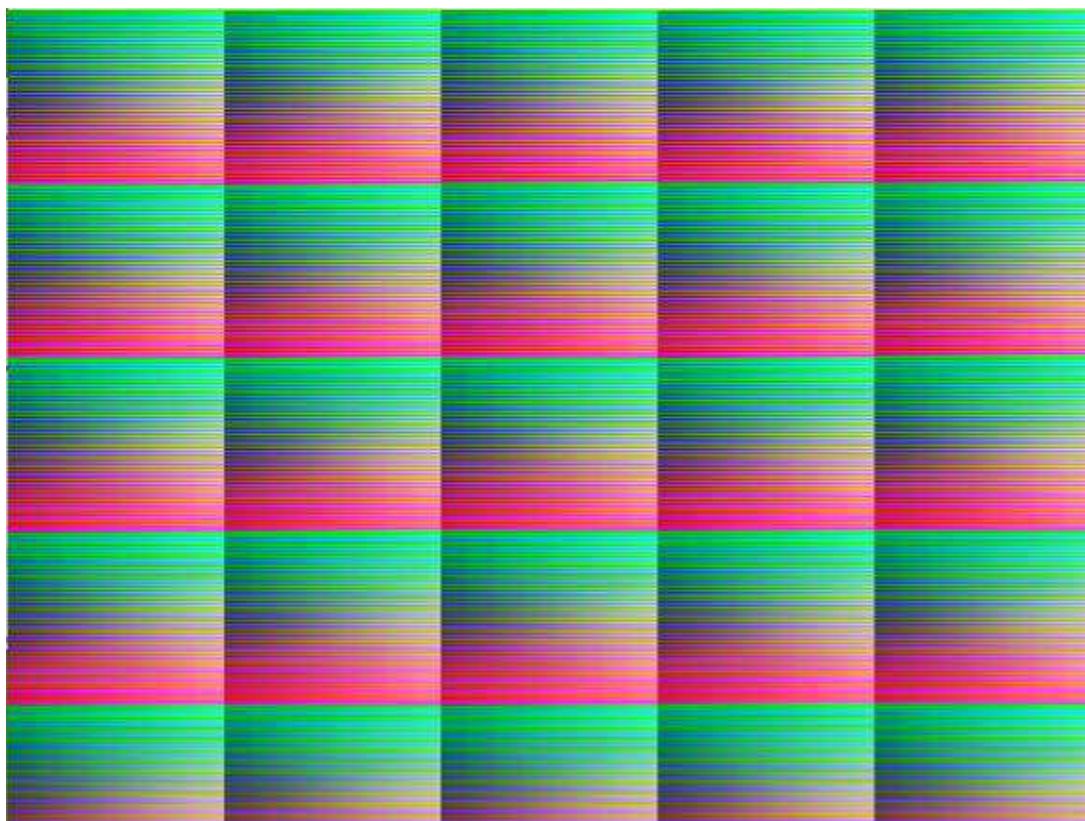


Figure 9.2-36 Generated Image Data Example (in the Case of Pattern Generation)

(6) Generated Image Data Example (in the Case of Monochrome Generation)

VGA (640 × 480): TIEN = 1b, TIMODE = 1b, TIPTNY2[7:0] = 52h, TIPTNU2[7:0] = 5Ah, TIPTNV2[3:0] = F0h
(Offset binary: ICnIPMC_C0/1/2/3.IBINSEL = 0b, ICnDMR.OBINSEL = 0b)

Table 9.2-19 Expected Values

	Y	U	V
Fixed value	52h	5Ah	F0h

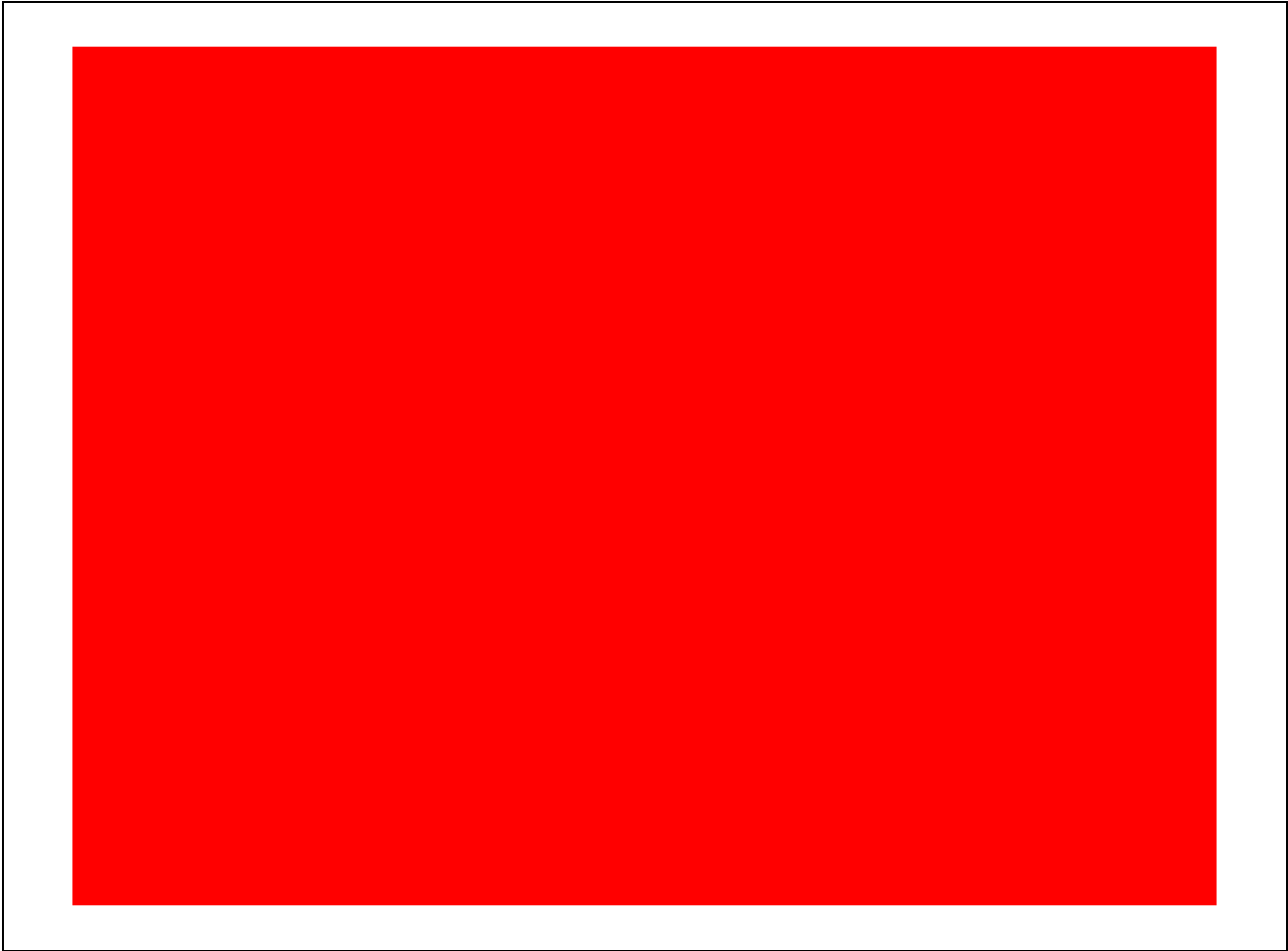


Figure 9.2-37 Generated Image Data Example (in the Case of Monochrome Generation)

9.2.4.3.6 VC/Data Type Selector (for Image Processing)

Virtual channels from VC0 to VC15 are supported for video input data input from the MIPI CSI2 block. On the other hand, the number of channels which can be handed in image processing is up to four. This block handles assignment of the selected virtual channels (SVCs).

Only one data type can be handled by each SVC and the data types are selected in the ICnIPMC_C0 to 3.INF[5:0] registers.

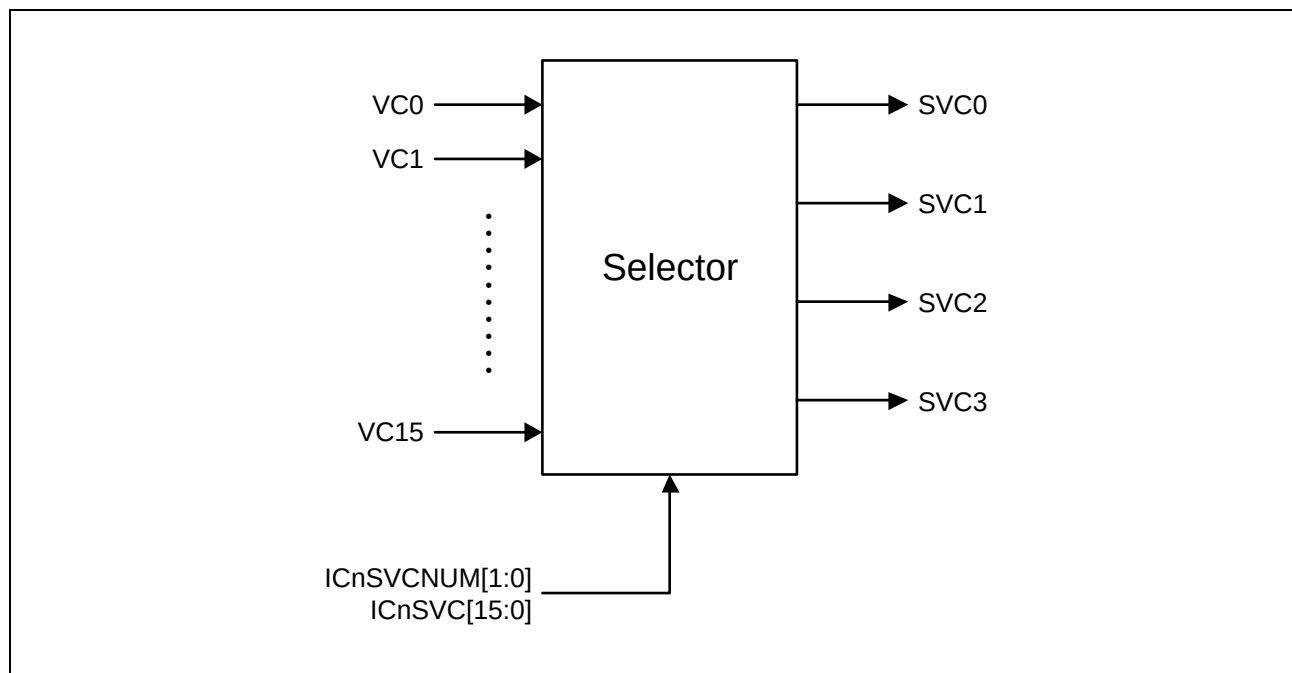


Figure 9.2-38 VC Selector Image

9.2.4.3.7 8-bit Expansion

(1) RGB444/RGB555/RGB565/RGB666 to RGB888

This block handles internal processing of RGB with 8 bits for each color. Inputs with fewer color bits are expanded to 8 bits. The conversion of inputs in the RGB444 and RGB555 and in the RGB565 and RGB666 formats to RGB888 is shown in **Table 9.2-20** and **Table 9.2-21**, respectively.

Table 9.2-20 RGB444/RGB555 to RGB888 Conversion

RGB444	RGB888
R[3:0]	{R[3:0],R[3:0]}
G[3:0]	{G[3:0],G[3:0]}
B[3:0]	{B[3:0],B[3:0]}

RGB555	RGB888
R[4:0]	{R[4:0],R[4:2]}
G[4:0]	{G[4:0],G[4:2]}
B[4:0]	{B[4:0],B[4:2]}

Table 9.2-21 RGB565/RGB666 to RGB888 Conversion

RGB565	RGB888
R[4:0]	{R[4:0],R[4:2]}
G[5:0]	{G[5:0],G[5:4]}
B[4:0]	{B[4:0],B[4:2]}

RGB666	RGB888
R[5:0]	{R[5:0],R[5:4]}
G[5:0]	{G[5:0],G[5:4]}
B[5:0]	{B[5:0],B[5:4]}

(2) RAW6/7 to RAW8

This block handles internal processing of RAW with 8 bits for each color. Inputs with fewer color bits are expanded to 8 bits. The conversion of inputs in the RAW6 and RAW7 formats to RGB888 is shown in **Table 9.2-22**.

Table 9.2-22 RAW6/7 to RAW8 Conversion

RAW6	RAW8
RAW[5:0]	{RAW[5:0],RAW[5:4]}

RAW7	RAW8
RAW[6:0]	{RAW[6:0],RAW[6]}

9.2.4.3.8 Image Clipping

Ranges for the clipping of input images are controlled by register settings. Specifically, the ranges of clipping for each SVC are set through the following register settings.

(Horizontal direction)

- Image clipping start pixel registers: ICnSPPrC_C0~3.SPPrC[12:0]
- Image clipping end pixel registers: ICnEPPrC_C0~3.EPPrC[12:0]

(Vertical direction)

- Image clipping start line registers: ICnSLPrC_C0~3.SLPrC[12:0]
- Image clipping end line registers: ICnELPrC_C0~3.ELPrC[12:0]

While the part of the image with the size after clipping (the gray part below) is being processed, the value of register ICnMS.AV0/1/2/3 = 1b. For the rest of the image, ICnMS.AV0/1/2/3 = 0b. When subsampling is done, operation proceeds in the same way for the subsampled images.

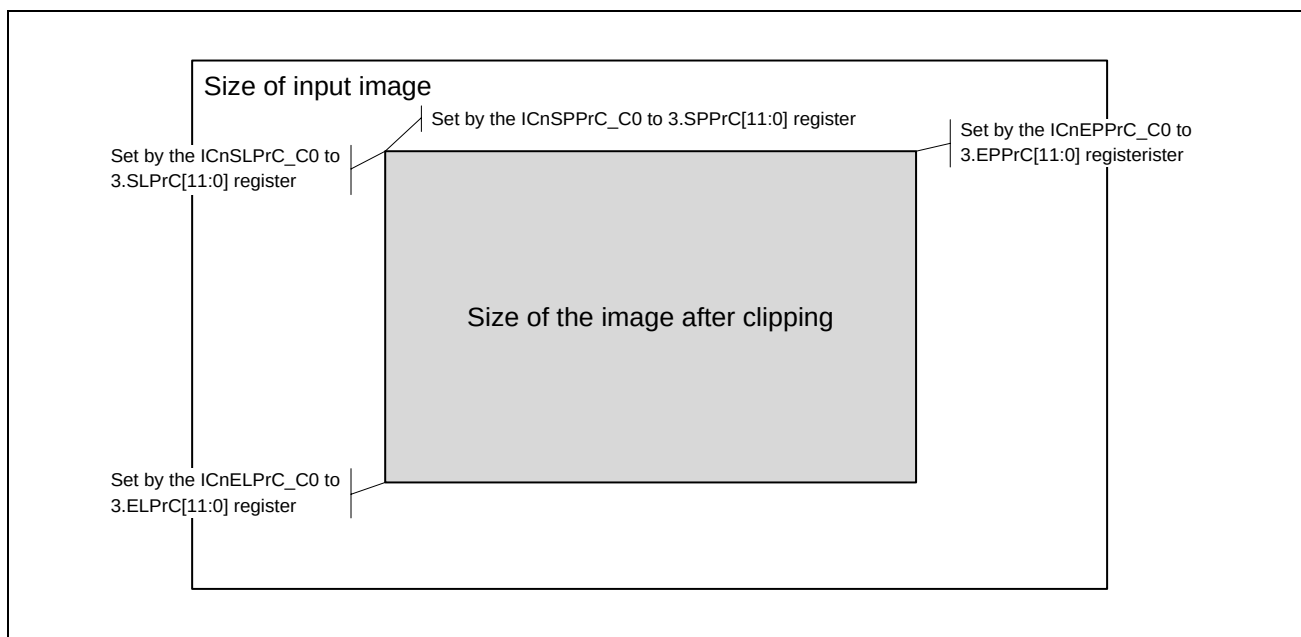


Figure 9.2-39 Image of Image Clipping

9.2.4.3.9 Frame Subsampling

Image frames are subsampled with the use of the registers which set input images. The settings of the following bit fields select the number of frames to be transferred in each round of transfer.

- Frame subsampling control register: ICnDEC.FRMDEC[3:0] (common to all SVCs)

Operation when the setting of ICnDEC.FRMDEC[3:0] = 2h is as follows:

Images are transferred in units of (the above register value + 1) frames. The state of ICnMS.CA0/1/2/3 is 1b when an image which is not to be eliminated by subsampling is being processed, as shown below.

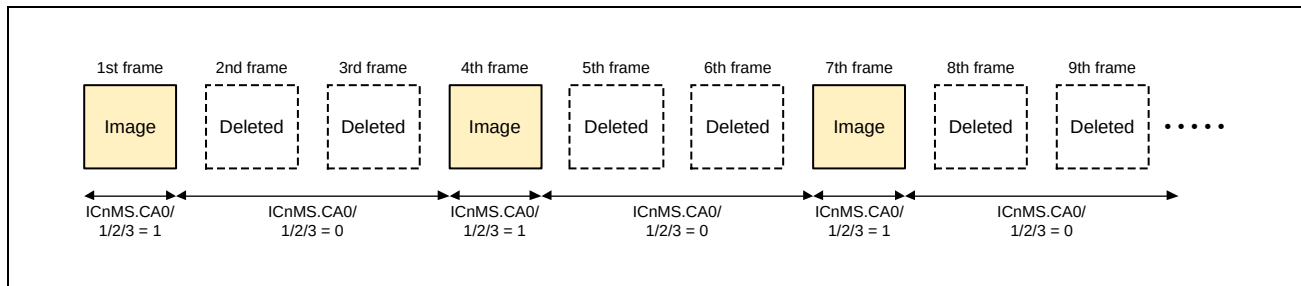


Figure 9.2-40 Frame Subsampling when ICnDEC.FRMDEC[3:0] = 2h

9.2.4.3.10 Data Arrangers 1 and 2

When image data are input and output, UV values expressed in negative form in YUV (YCbCr) data are converted. The data arranger 1 input side and data arranger 2 output side settings can be made by using the ICnDTVP.IBINSEL and ICnDMR.OBINSEL registers, respectively.

Settings for data arranger 1 must suit the specifications of the sensor. Data arranger 1 supports 8 or 10 bits depending on the number of bits of input data. Settings for data arranger 2 must suit the specifications of the destination for output. Data arranger 2 supports 8 bits depending on the output data. YUV of MIPI CSI-2 is usually used in offset binary. However, this depends on the sensor, and it may be output as two's complements. **Table 9.2-23** shows an example of processing when the input is 8 bits. Data arranger 1 also has 10-bit input, and when 10-bit input is used, it can be converted to 10-bit offset binary/2's complement as well as 8-bit.

Table 9.2-23 Input Formats and Corresponding Forms of Image Processing (at 8-bit input)

UV Value (Hex)	Offset Binary (Dec)	Two's Complements (Dec)
	Register Settings: Data Arranger 1: ICnDTVP.IBINSEL = 0b Data Arranger 2: ICnDMR.OBINSEL = 0b	Register Settings: Data Arranger 1: ICnDTVP.IBINSEL = 1b Data Arranger 2: ICnDMR.OBINSEL = 1b
FFh	127	-1
FEh	126	-2
...	...	...
81h	1	-127
80h	0	-128
7Fh	-1	127
...	...	...
01h	-127	1
00h	-128	0

9.2.4.3.11 Data Clipping

Data clipping is performed on the YUV (YCbCr) format. To make input data for RGB conversion compliant with the ITU-R BT.601 standard, whether or not to clip values less than or equal to 16 and greater than or equal to 240 in 8-bit encoding (or less than or equal to 64 and greater than or equal to 960 in 10-bit encoding) is specified. This setting is made by using the ICnDTV.P.CLP[1:0] register.

9.2.4.3.12 Demosaicing

This function is only valid on input data of the RAW type. Its operation is on the assumption that the RAW data are Bayer data and generates pixel data that are interpolated from the RAW data.

The demosaicing function uses a bilinear method. Of the peripheral 3 × 3 pixels at the position to be calculated, 2 or 4 pixels are used for interpolation processing. The interpolation formula is given below.

The following describes the method of calculating the RGB components (R22, Gb22/Gr22, B22) of the pixels at the center position.

In the case of RAW10/12/14/16/20, the eight higher-order bits are used for the calculation.

(1) RAW20 to RAW16

For a RAW20 input, the higher-order bits of RAW20 are extracted and the lower-order bits are truncated in preprocessing for demosaicing and statistics.

RAW20 → RAW8 (RAW[19:4])

(2) Calculating the Gb22 data portion

Figure 9.2-41 shows the method of interpolation calculation for pixel data (R/G/B) of Gb22.

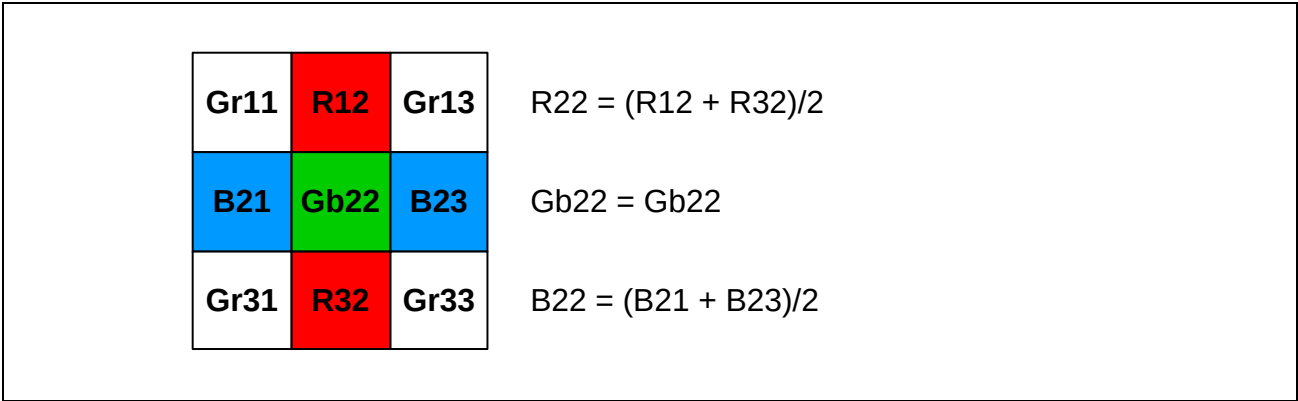


Figure 9.2-41 Calculation of the Gb22 Data Portion

(3) Calculating the B22 data portion

Figure 9.2-42 shows the method of interpolation calculation for pixel data (R/G/B) of B22.

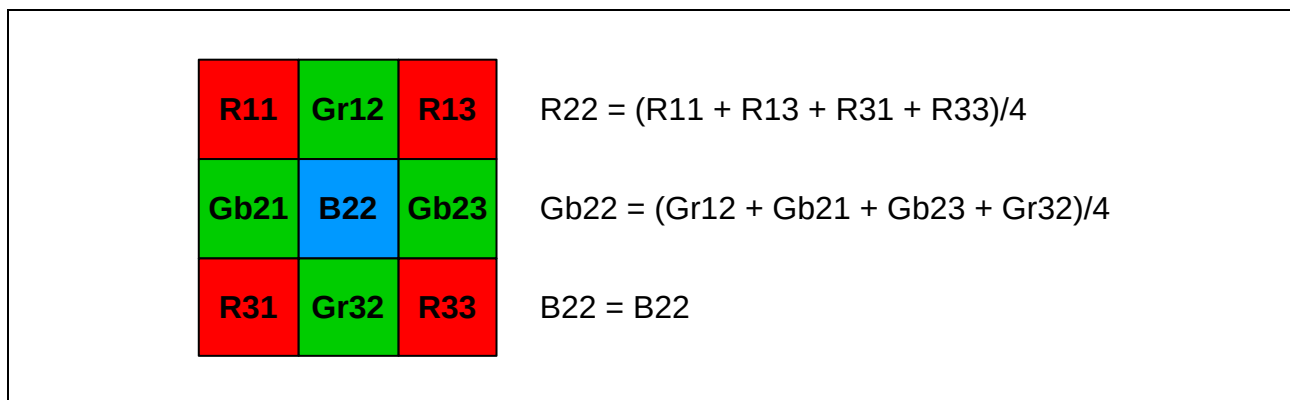


Figure 9.2-42 Calculation of the B22 Data Portion

(4) Calculating the R22 data portion

Figure 9.2-43 shows the method of interpolation calculation for pixel data (R/G/B) of R22.

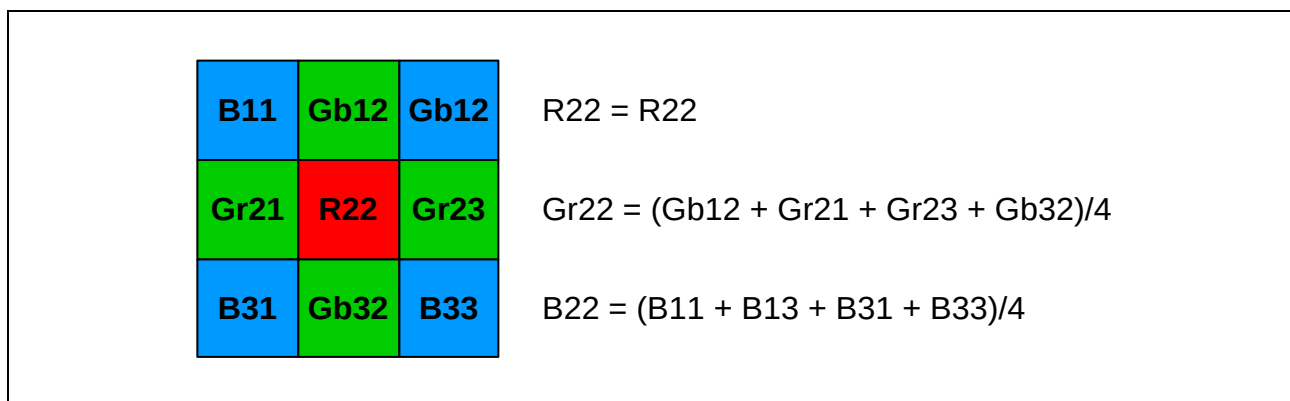


Figure 9.2-43 Calculation of the R22 Data Portion

(5) Calculating the Gr22 data portion

Figure 9.2-44 shows the method of interpolation calculation for pixel data (R/G/B) of G22.

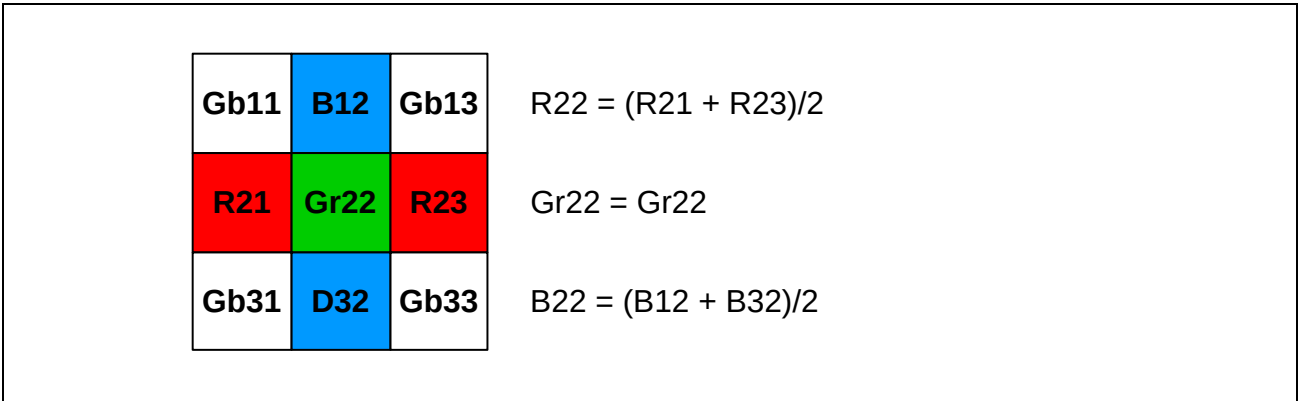


Figure 9.2-44 Calculation of the Gr22 Data Portion

9.2.4.3.13 Linear Matrix

Linear matrix processing involves calculation of the matrix of R', G', and B' (8 bits per color) by using the registers listed below to set the following coefficients for R, G, and B (8 bits per color) after demosaicing as the previous stage.

Gof = Register ICnLMXOF.GOF[7:0]

Rof = Register ICnLMXOF.ROF[7:0]

Bof = Register ICnLMXOF.BOF[7:0]

Rr = Register ICnLMXRC1.RR[12:0]

Rb = Register ICnLMXRC2.RB[12:0]

Rg = Register ICnLMXRC2.RG[12:0]

Gr = Register ICnLMXGC1.GR[12:0]

Gb = Register ICnLMXGC2.GB[12:0]

Gg = Register ICnLMXGC2.GG[12:0]

Br = Register ICnLMXBC1.BR[12:0]

Bb = Register ICnLMXBC2.BB[12:0]

Bg = Register ICnLMXBC2.BG[12:0]

$$R' = \frac{Rr \cdot (R + Rof)^{*1} + Rg \cdot (G + Gof)^{*1} + Rb \cdot (B + Bof)^{*1}}{1024}$$

$$G' = \frac{Gr \cdot (R + Rof)^{*1} + Gg \cdot (G + Gof)^{*1} + Gb \cdot (B + Bof)^{*1}}{1024}$$

$$B' = \frac{Br \cdot (R + Rof)^{*1} + Bg \cdot (G + Gof)^{*1} + Bb \cdot (B + Bof)^{*1}}{1024}$$

Note 1. When the result of addition is a negative value, it is converted to 0.

9.2.4.3.14 Color Space Conversion

(1) RGB to YCbCr

RGB (8/10 bits) data to which image clipping, frame subsampling, or demosaicing has been applied are expanded to 12 bits and then converted to 10-bit YCbCr. Calculations equivalent to those specified in ITU-R BT.601 or ITU-R BT.709 are performed by setting coefficients through registers.

The calculations proceed according to the coefficients and expressions set in the following registers.

YCLRP = Register ICnYCCR1.YCLRP[12:0]

YCLBP = Register ICnYCCR2.YCLBP[12:0]

YCLGP = Register ICnYCCR2.YCLGP[12:0]

YCLSFT = Register ICnYCCR3.YCLSFT[4:0]

YCLAP = Register ICnYCCR3.YCLAP[11:0]

CBCLRP = Register ICnCBCCR1.CBCLRP[12:0]

CBCLBP = Register ICnCBCCR2.CBCLBP[12:0]

CBCLGP = Register ICnCBCCR2.CBCLGP[12:0]

CBCLSFT = Register ICnCBCCR3.CBCLSFT[4:0]

CBCLAP = Register ICnCBCCR3.CBCLAP[11:0]

CRCLRP = Register ICnCRCCR1.CRCLRP[12:0]

CRCLBP = Register ICnCRCCR2.CRCLBP[12:0]

CRCLGP = Register ICnCRCCR2.CRCLGP[12:0]

CRCLSFT = Register ICnCRCCR3.CRCLSFT[4:0]

CRCLAP = Register ICnCRCCR3.CRCLAP[11:0]

$$Y = \left(\frac{YCLRP \times R + YCLGP \times G + YCLBP \times B}{2^{YCLSFT}} \right) + YCLAP$$

$$Cb = \left(\frac{CBCLRP \times R + CBCLGP \times G + CBCLBP \times B}{2^{CBCLSFT}} \right) + CBCLAP$$

$$Cr = \left(\frac{CRCLRP \times R + CRCLGP \times G + CRCLBP \times B}{2^{CRCLSFT}} \right) + CRCLAP$$

For ITU-R BT.601 (8 bits): (Same as the default values)

$$Y = 0.257 \times R + 0.504 \times G + 0.098 \times B + 16$$

$$Cb = -0.148 \times R - 0.291 \times G + 0.439 \times B + 128$$

$$Cr = 0.439 \times R - 0.368 \times G - 0.071 \times B + 128$$

[Register settings]

YCLRP = 0107h

YCLBP = 0064h

YCLGP = 0204h

YCLSFT = 0Ah

YCLAP = 100h

CBCLRP = 1F68h

CBCLBP = 01C2h

CBCLGP = 1ED6h

CBCLSFT = 0Ah
 CBCLAP = 800h
 CRCLRP = 01C2h
 CRCLBP = 1FB7h
 CRCLGP = 1E87h
 CRCLSFT = 0Ah
 CRCLAP = 800h

(2) YCbCr to RGB

YCbCr (8/10 bits) data which have been arranged and clipped are expanded to 12 bits (by appending zeros as the lower-order bits) and then converted to 10-bit RGB. Calculations equivalent to those specified in ITU-R BT.601 or ITU-R BT.709 are performed by setting coefficients through registers.

The calculations proceed according to the coefficients and expressions set in the following registers.

YMUL = Register ICnCSCC1.YMUL[13:0]
 YSUB = Register ICnCSCC2.YSUB[11:0]
 RCRMUL = Register ICnCSCC2.CSUB[11:0]
 GCRMUL = Register ICnCSCC3.RCRMUL[13:0]
 GCBMUL = Register ICnCSCC3.GCRMUL[13:0]
 BCBMUL = Register ICnCSCC4.GCBMUL[13:0]
 CSUB = Register ICnCSCC4.BCBMUL[13:0]

$$R = \frac{(YMUL \times (Y - YSUB)^{*1} + RCRMUL \times (Cr - CSUB))}{4096}$$

$$G = \frac{YMUL \times (Y - YSUB)^{*1} - GCRMUL \times (Cr - CSUB) - GCBMUL \times (Cb - CSUB)}{4096}$$

$$B = \frac{YMUL \times (Y - YSUB)^{*1} + BCBMUL \times (Cb - CSUB)}{4096}$$

Note 1. A negative value is rounded up to 0.

For ITU-R BT.601 (8 bits): (Same as the default values)

$R = 1.164 \times (Y - 16) + 1.596 \times (Cr - 128)$
 $G = 1.164 \times (Y - 16) - 0.813 \times (Cr - 128) - 0.392 \times (Cb - 128)$
 $B = 1.164 \times (Y - 16) + 2.017 \times (Cb - 128)$

[Register settings]

YMUL = 129Fh
 YSUB = 100h
 RCRMUL = 1989h
 GCRMUL = 0D02h
 GCBMUL = 0645h
 BCBMUL = 2045h
 CSUB = 800h

9.2.4.3.15 LUT (Lookup Table)

This is used to convert 10-bit data after color space conversion (YCbCr to RGB or RGB to YCbCr) into 8-bit data.

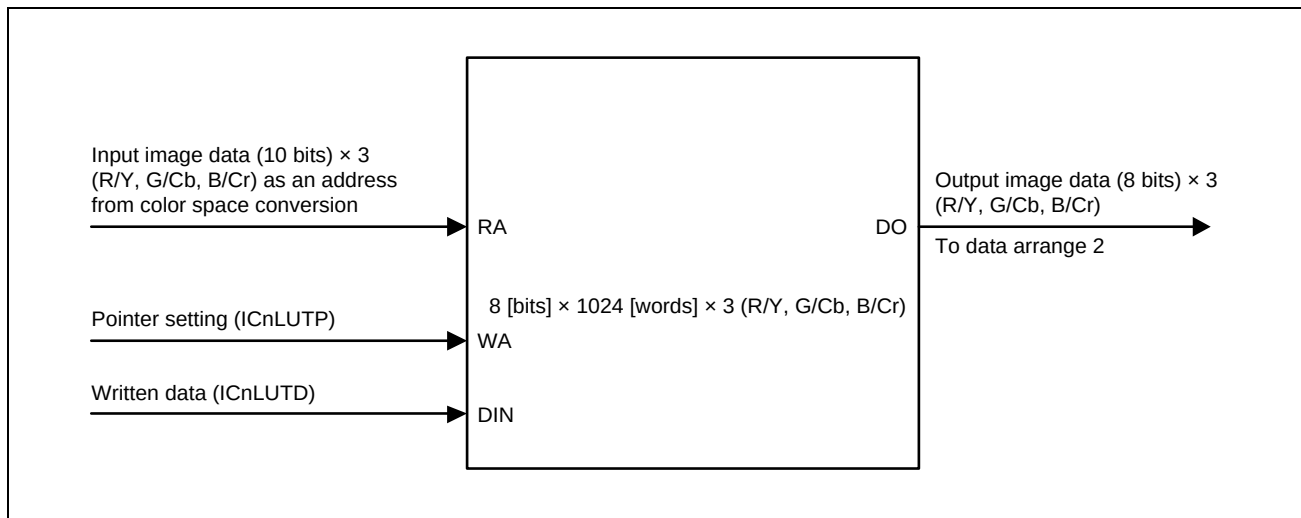


Figure 9.2-45 Lookup Table (LUT)

(1) Pointer Access

The pointer is incremented every time the ICnLUTD register is written.

The incremented value can be read from the ICnLUTP register. When the ICnLUTD register is read, the pointer is not incremented. The pointer must be set immediately before the register is read.

Follow the procedure below to access the lookup table. See **9.2.5.1 Starting Reception for the MIPI CSI-2 Input**.

The lookup table cannot be accessed from the CPU in normal mode.

(2) Usage Notes on LUT

1. Set the lookup table only after the CRU has stopped its operation.
2. Rewrite the entire lookup table before use because the initial values are not guaranteed after the power is turned on.
3. Data to be set in the lookup table must be restricted to the range that is compatible with the input bit width.

9.2.4.3.16 YC Post Processing

(1) YCbCr422 to 420

This function thins out the even-line CbCr data for the image processed in the previous stage and outputs only Y. Averaging or any other calculations are not performed. This conversion can be set by ICnDMR.YCMODE[2:0]=100b/101b/110b.

(2) Y/C Separation

This function separates the Y and C (CbCr) components for storage in two separate memory banks of the DRAM. For details, see **9.2.4.3.3 Transfer to DRAM**.

(3) Y Only

This function changes the method of storage in the DRAM. For details, see **9.2.4.3.3 Transfer to DRAM**.

9.2.4.3.17 VC/Data Type Selector (for Non-Image Processing)

When non-image processing is selected, set the target SVC and data types. When selecting the number of channels of SVC to be used, use the ICnSVCNUM.SVCNUM[1:0] register to set how many channels are used from among SVC0 to 3. Selection of SVCs with the ICnSVC.SVC0~3[3:0] registers determines the allocation of which channels are to be selected from among VC0 to VC15 for SVC0 to 3.

In data type selection, the required type can be selected with the ICnNIPDT_C0~3H/L registers and selection of several types is also possible (see **Figure 9.2-46**).

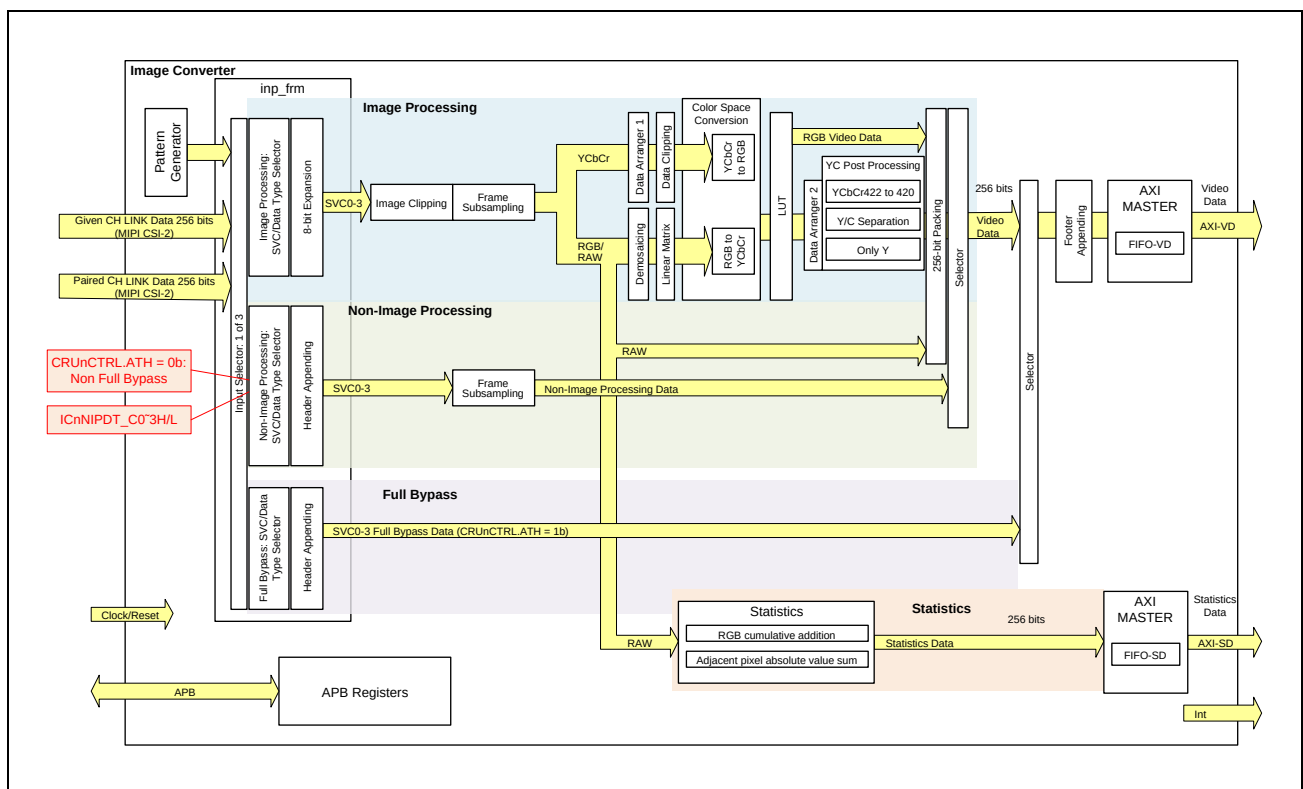


Figure 9.2-46 Example of VC Specification and Selection of Several Data Types for Non-Image Processing

9.2.4.3.18 Full Bypass function

Figure 9.2-47 shows an example of the settings for full bypass. These settings are used when you want to store images in memory without processing them within the CRU (for example, when you want to process images with the unit).

The full bypass function handles the transfer of data of the data type selected for the specified single VC from the video data bus to memory. Data are not corrected at all and are transferred in the recommended memory storage format for MIPI CSI-2.

The data types are not limited to one and several types can be selected by the settings of ICnNIPDT_C0~3H/L.

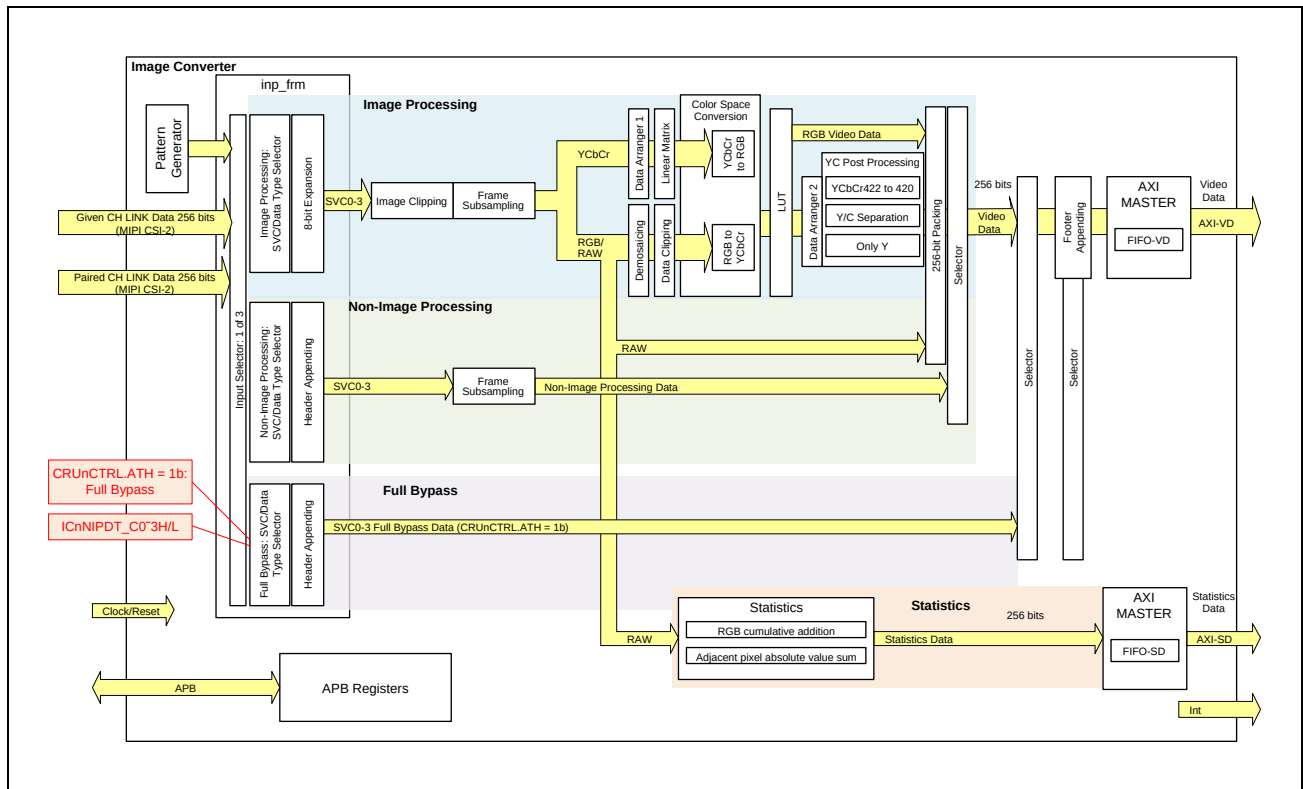


Figure 9.2-47 Example of Settings for Full Bypass

9.2.4.3.19 Statistics

This function is limited to a RAW data input through SVC0 . This module divides one image vertically and horizontally into blocks (see **Figure 9.2-48**). The statistical information within this block is transferred to memory and the image is adjusted by weighting the information of every block by the CPU and providing feedback to the external video state.

The number of block divisions can be determined by the setting of register ICnSTIC1.STUNIT[1:0].

The statistical information of each block has two types and their respective purposes are as follows.

1) In-block accumulation

Accumulation of each color component, R, G, and B, is performed within each block.

On the CPU side, the weighting is applied per block based on the information from each block, and the obtained evaluation values can be used for exposure control, etc. from the next frame.

The weighting could in general be center-weighted allocation, weighted allocation according to the subject, and so on. If the subject is judged to be dark, exposure control is applied to widen the aperture, and if the subject is judged to be bright, the aperture is made narrower.

Additionally, each color component among the R, G, and B components is adjusted and the latter stage performs operations in such a way that the B component is reduced, for example, in video which appears bluish in white regions.

This adjustment of each color includes changing the coefficients of the sensor or the linear matrix as the latter stage.

2) In-block sum of absolute difference values

Focusing on the G component within the block, the overall value for the differences from the adjacent parts of the image is extracted and accumulation is performed within the block.

On the CPU side, the weighting is applied per block based on the information from each block, and the obtained evaluation values can be used for focus adjustment, etc. from the next frame.

The weighing could in general be center-weighted allocation, weighted allocation according to the subject, and so on. If the obtained evaluation value is compared with the previous frame and if the amount continues to be reduced, the evaluation value is increased to adjust the focus, for example, by adjusting the distance between the lens and the sensor toward the opposite direction.

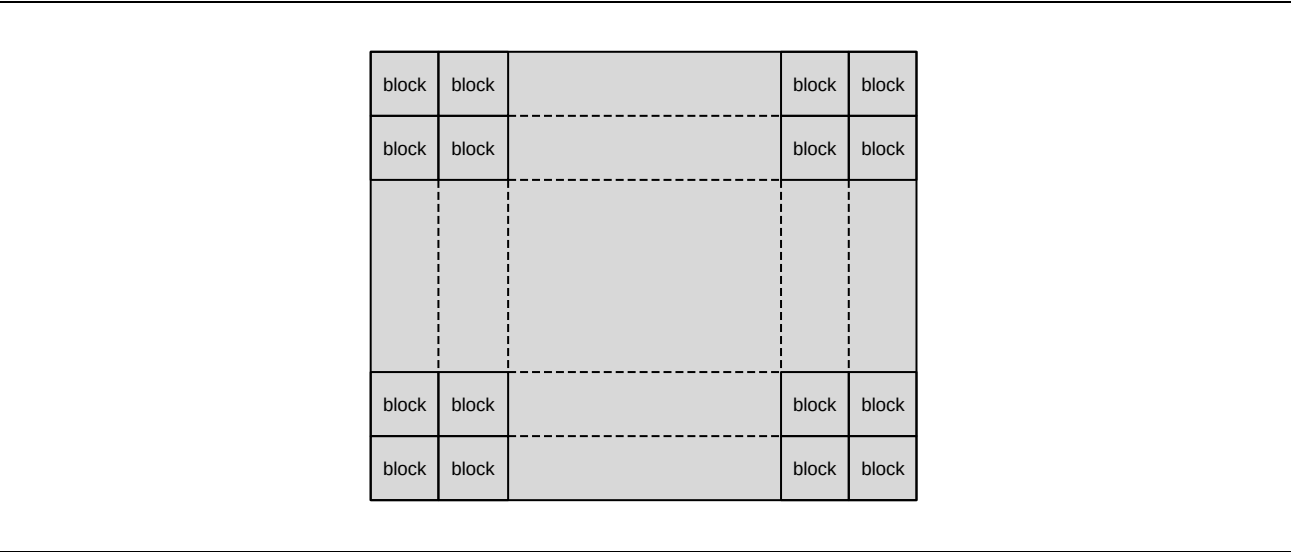


Figure 9.2-48 Image of Dividing an Image into Blocks Vertically and Horizontally

Figure 9.2-49 is a block diagram of statistics. The result of AE/AWB statistical calculations for per block is 8 bits $\times$ 3 = 3 bytes, respectively, and that of the AF statistical calculations is 1 byte for G only, so a total of 4 bytes of data are output.

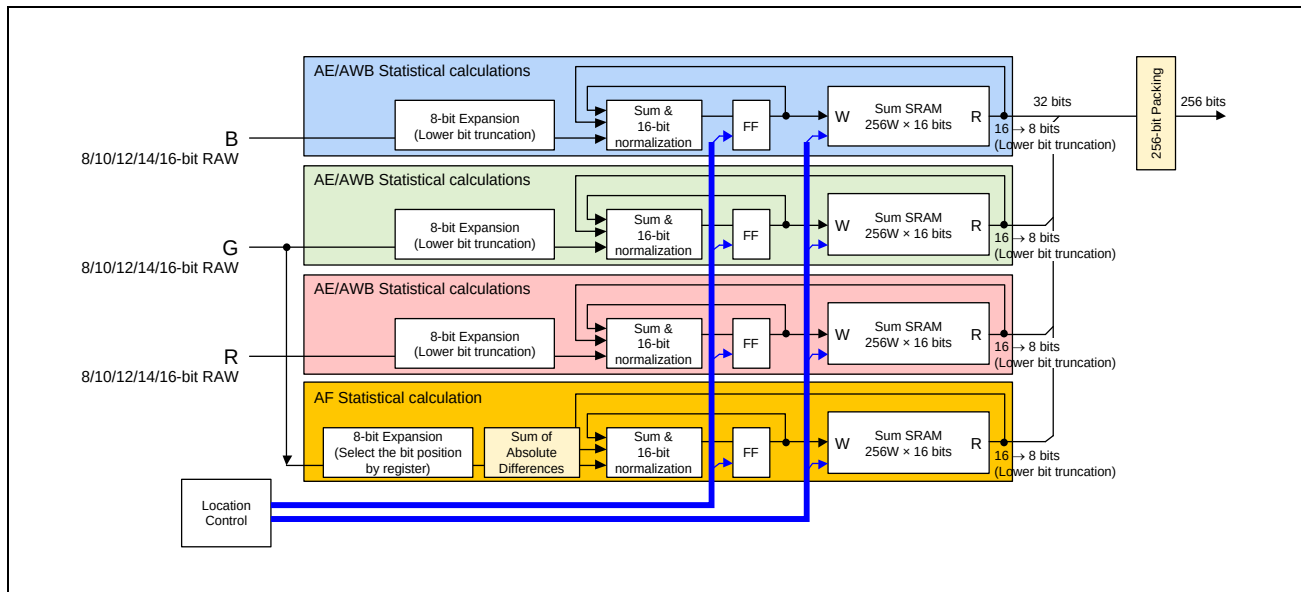


Figure 9.2-49 Statistics Block Diagram

(1) RGB cumulative addition

The STUNIT[1:0] bit fields in each of the ICnSTIC1 registers are used to select the pixel (block) sizes for statistical calculations from among four values, 16 $\times$ 16, 32 $\times$ 32, 64 $\times$ 64, and 128 $\times$ 128. The target pixels when the setting is 16 $\times$ 16 are described in **(a) Target pixels when the block size is selected as 16 $\times$ 16 (ICnSTIC1.STUNIT[1:0] = 0b)** and the methods of calculation for each block-size setting are described in **(b) Methods of calculation for each block-size setting**.

(a) Target pixels when the block size is selected as 16 $\times$ 16 (ICnSTIC1.STUNIT[1:0] = 0b)

G component: 128 pixels are present in the area of 16 $\times$ 16

R and B components: 32 pixels are present in the area of 16 $\times$ 16

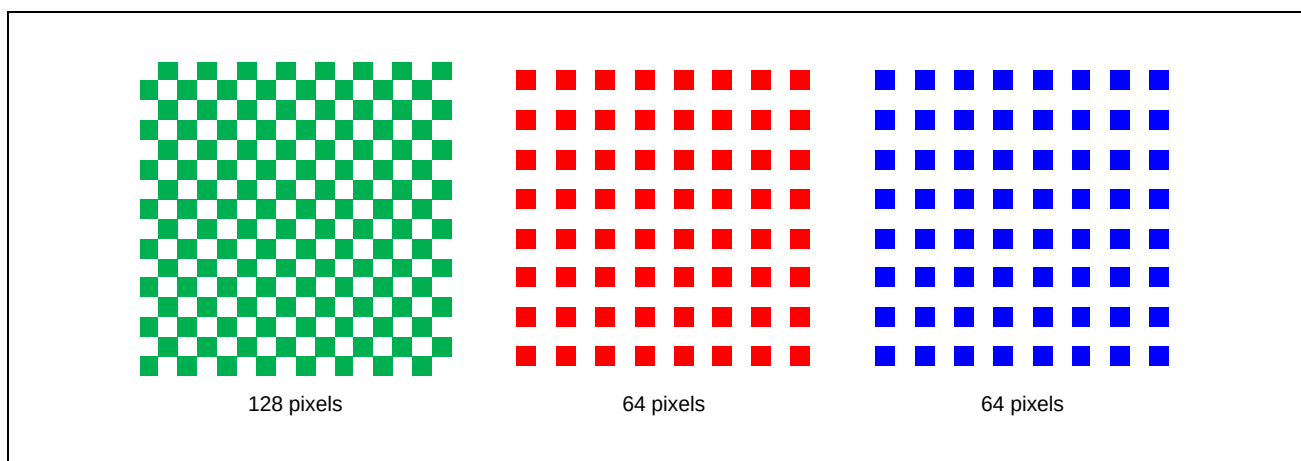


Figure 9.2-50 Target Pixels when the Block Size is Selected as 16 $\times$ 16

(b) Methods of calculation for each block-size setting**[G component]**

- When the block size is 16×16: 8 bits × 128 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 32×32: 8 bits × 512 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 64×64: 8 bits × 2048 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 128×128: 8 bits × 8192 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.

[R and B components]

- When the block size is 16×16: 8 bits × 64 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 32×32: 8 bits × 256 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 64×64: 8 bits × 1024 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 128×128: 8 bits × 4096 pixels are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.

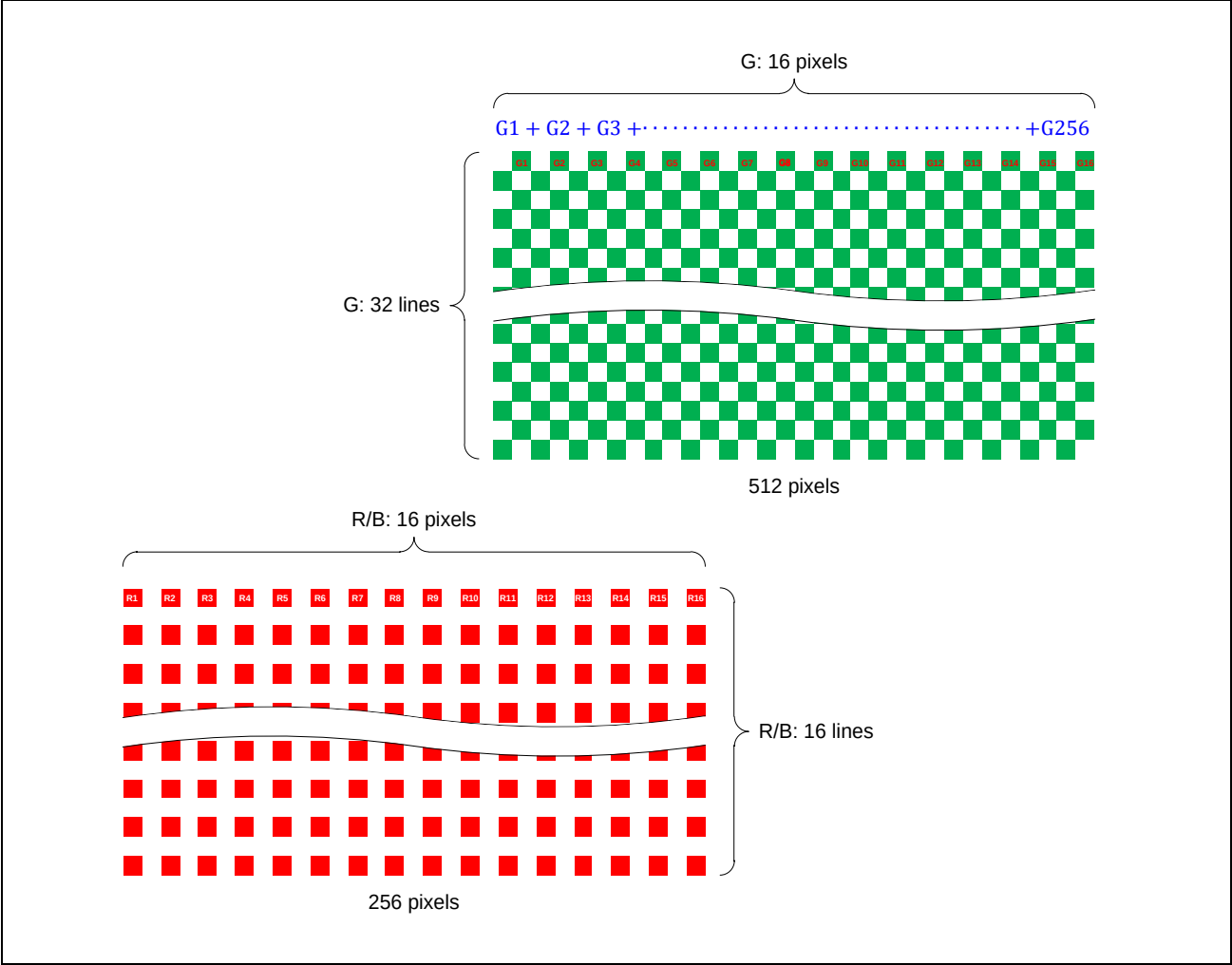


Figure 9.2-51 Illustration Diagram of Accumulation within the Block when the Block Size is 32×32

(2) Sum of absolute values for adjacent pixels

Set the area for which the sum of absolute values for adjacent pixels is taken with the use of register ICnSTIC1.STUNIT[1:0] to make the calculations in the following ways.

● In the case of 16×16 pixel unit operations (ICnSTIC1.STUNIT[1:0] = 0b):

- For 8-bit data: Up to 16 bits
- For 10- to 16-bit data: A sum-of-absolute-differences (SAD) operation is performed with the eight higher-order bits (default).
- Addition can also be performed with the eight lower-order bits by shifting the bits by setting register ICnSTIC1.STSADPOS[3:0].

● Methods of calculation for the setting area:

[G component calculation]

- When the block size is 16×16: Adjacent differences × 112 are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 32×32: Adjacent differences × 480 are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 64×64: Adjacent differences × 1984 are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.
- When the block size is 128×128: Adjacent differences × 8064 are accumulated: The eight higher-order bits are output after accumulation and normalization with a 16-bit precision.

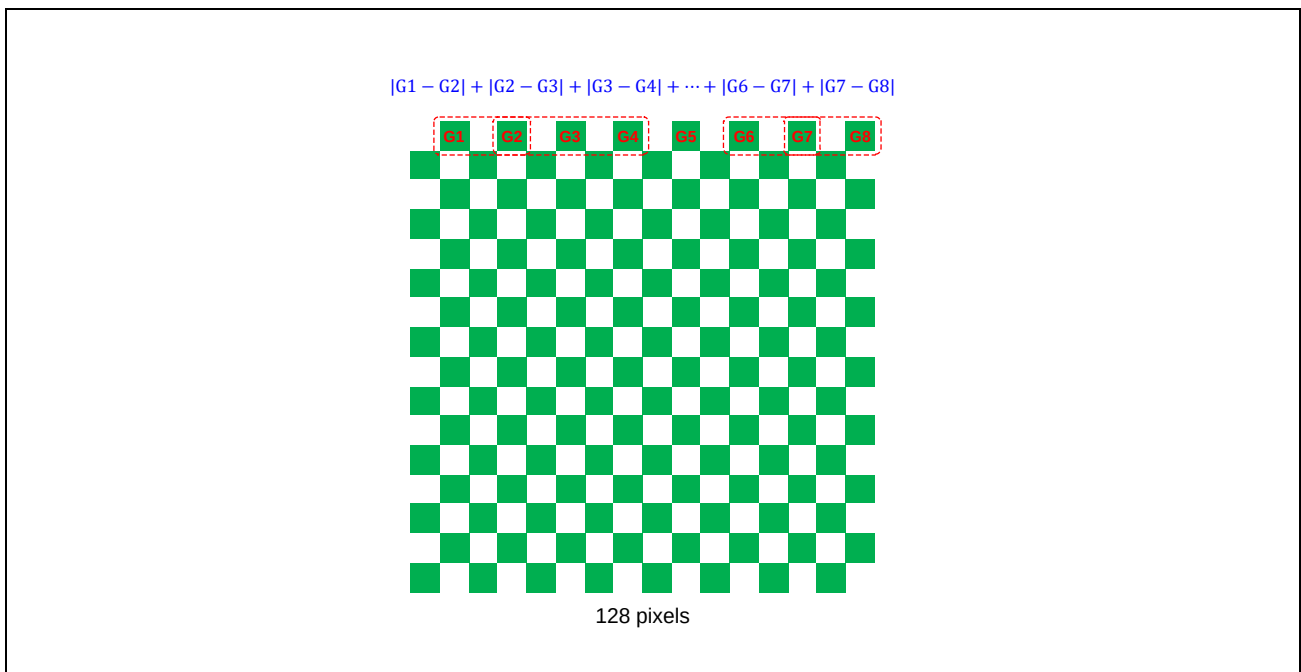


Figure 9.2-52 Illustration Diagram of Sum of Absolute Values for Adjacent Pixels when the Block Size is 16×16

(3) Statistics data generated image

The following describes the amounts of data in the operations when a 1920×1080 pixel area is set as 16×16 blocks and as 128×128 blocks for the statistics function. The numerals in the figures below represent the block numbers. In processing of the right end or bottom of the image, if the input image size is greater than the size specified as the number of blocks in the horizontal direction, that difference does not fall within the target range for arithmetic processing. This also applies in the vertical direction.

● In the case of operations for a 1920×1080 pixel area as 16×16 blocks:

As shown in **Figure 9.2-53**, the calculated values for each of the blocks are transferred sequentially. A total of 31 Kbytes of data are transferred for one frame. For the format of data for transmission, see **Table 9.2-16**.

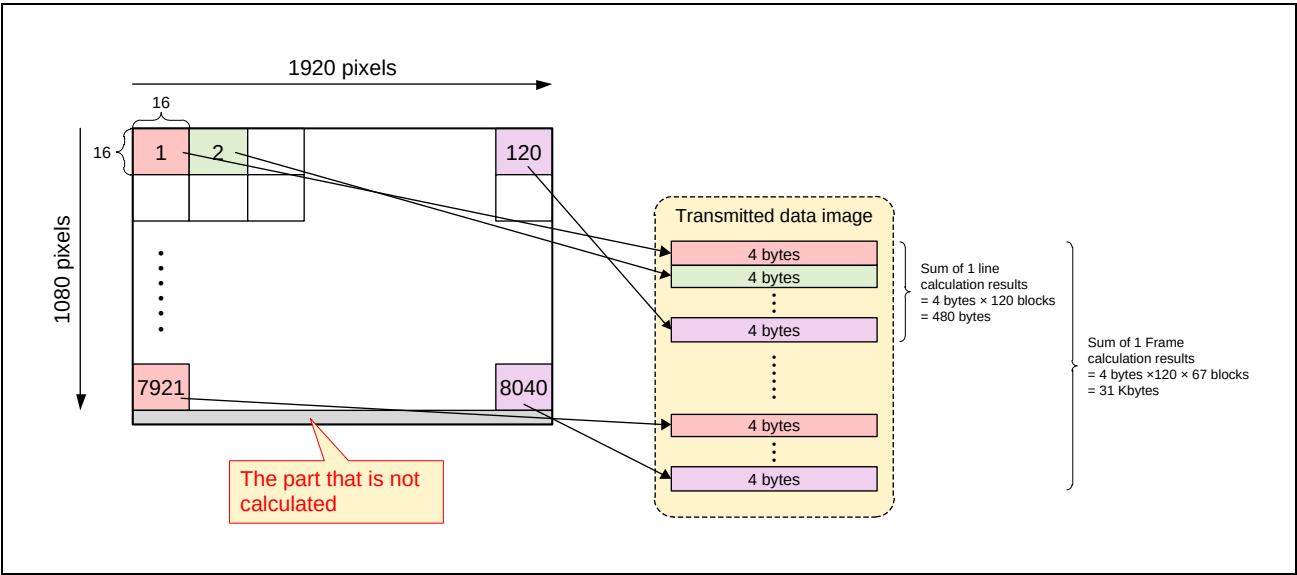


Figure 9.2-53 Illustration Diagram of the Amounts of Data in the Operations when a 1920×1080 Pixel Area is Set as 16×16 Blocks

● In the case of operations for a 1920×1080 pixel area as 128×128 blocks:

The calculated values for the corresponding blocks are sequentially transferred. A total of 480 bytes of data are transferred for one frame.

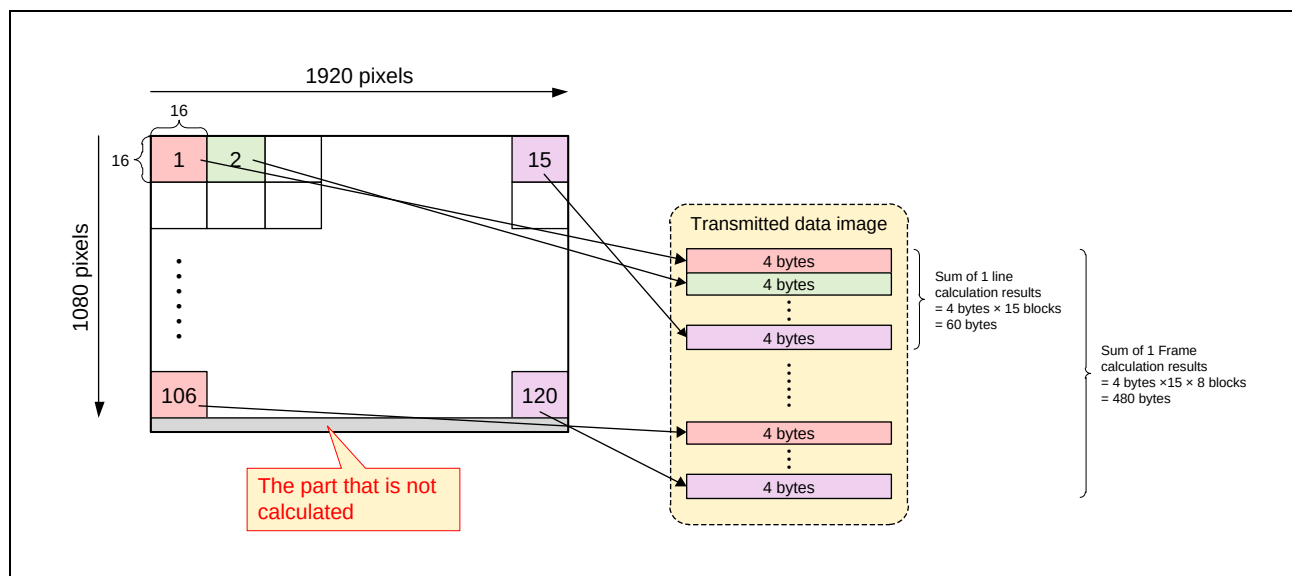


Figure 9.2-54 Illustration Diagram of the Amounts of Data in the Operations when a 1920×1080 Pixel Area is Set as 16×16 Blocks

Confirm the completion of transfer for one frame by setting the specified address interrupt described in **9.2.4.4.2 Designated address forwarding completion interrupt**.

9.2.4.4 Interrupts

9.2.4.4.1 Interrupt list

Table 9.2-24 lists the interrupt sources and registers, and the interrupt signals to the interrupt controller.

Table 9.2-24 Interrupt Sources and Registers, and Interrupt Signals (1/2) [n = 0, 1]

Internal Signal Name	Kind	Item	Interrupt Source Status/Clear	Interrupt Enable Register Field	Generated Blocks	Conditions for Occurrence
axi_mst_err_int* ⁴	Level	FIFO-VD overflow (video data) (for debugging)	CRUnINTS.FOS	CRUnIE1.FOE	FIFO-VD	When Video data FIFO-VD overflows.
		SLVERR error (video data)	CRUnINTS.SLVES	CRUnIE1.SLVEE	AXI-VD	When Video Data AXI-VD SLVERR error occurs.
		DECERR error (video data)	CRUnINTS.DECES	CRUnIE1.DECEE	AXI-VD	When Video Data AXI-VD SLVERR error occurs.
		FIFO-SD overflow* ¹ (statistics data) (for debugging)	CRUnINTS.SDFOS	CRUnIE1.SDFOE	FIFO-SD	When Statistics Data FIFO-SD overflows.
		SLVERR error* ¹ (statistics data)	CRUnINTS.SDSLVES	CRUnIE1.SDSLVEE	AXI-SD	When the SLVERR error of Statistics Data AXI-SD occurs.
		DECERR error* ¹ (statistics data)	CRUnINTS.SDDECES	CRUnIE1.SDDECEE	AXI-SD	When the Statistics Data AXI-SD DECERR error.
image_conv_int* ²	Level	SVC0 frame start	CRUnINTS2.FS0S	CRUnIE2.FS0E	Image Clipping	When SVC0 frame starts.* ³
	Level	SVC0 frame end	CRUnINTS2.FE0S	CRUnIE2.FE0E	Image Clipping	When SVC0 frame ends.* ³
	Level	SVC0 scan line	CRUnINTS2.SL0S	CRUnIE2.SL0E	Image Clipping	When the SVC0 register (ICnSI_C0.SI[11:0]) matches the set number of lines.* ³
	Level	SVC1 frame start	CRUnINTS2.FS1S	CRUnIE2.FS1E	Image Clipping	When SVC1 frame starts.* ³
	Level	SVC1 frame end	CRUnINTS2.FE1S	CRUnIE2.FE1E	Image Clipping	When SVC1 frame ends.* ³
	Level	SVC1 scan line	CRUnINTS2.SL1S	CRUnIE2.SL1E	Image Clipping	When the SVC1 register (ICnSI_C1.SI[11:0]) matches the set number of lines.* ³
	Level	SVC2 frame start	CRUnINTS2.FS2S	CRUnIE2.FS2E	Image Clipping	When SVC2 frame starts.* ³
	Level	SVC2 frame end	CRUnINTS2.FE2S	CRUnIE2.FE2E	Image Clipping	When SVC2 frame ends.* ³
	Level	SVC2 scan line	CRUnINTS2.SL2S	CRUnIE2.SL2E	Image Clipping	When the SVC2 register (ICnSI_C2.SI[11:0]) matches the set number of lines.* ³
	Level	SVC3 frame start	CRUnINTS2.FS3S	CRUnIE2.FS3E	Image Clipping	When SVC3 frame starts.* ³
	Level	SVC3 frame end	CRUnINTS2.FE3S	CRUnIE2.FE3E	Image Clipping	When SVC3 frame ends.* ³
	Level	SVC3 scan line	CRUnINTS2.SL3S	CRUnIE2.SL3E	Image Clipping	When the SVC3 register (ICnSI_C3.SI[11:0]) matches the set number of lines.* ³
cru_vd_addr_wend	Edge	Specified address 0 write completed (video data)	—	CRUnIE2.VDADROWE	AXI-VD	When AXI-VD bus write to the AXI-VD Bus Transfer Completion Event Address 0 Higher/Lower Register (AMnIVT0ADDRH/L) address is completed.
cru_sd_addr_wend	Edge	Specified address 0 write completed* ¹ (statistics data)	—	CRUnIE2.SDADROWE	AXI-SD	When AXI-SD bus write to the AXI-SD Bus Transfer Completion Event Address 0 Higher/Lower Register (AMnSDIVT0ADDRH/L) address is completed.

Table 9.2-24 Interrupt Sources and Registers, and Interrupt Signals (2/2) [n = 0, 1]

Internal Signal Name	Kind	Item	Interrupt Factor Status/Clear	Interrupt Enable Register Field	Generation Blocks	Conditions of Occurrence
cru_vsd_addr_wend	Level	Specified address 1 write completed (video data)	CRUnINTS2.VDADR1WS	CRUnIE2.VDADR1WE	AXI-VD	When AXI-VD bus write to the AXI-VD Bus Transfer Completion Event Address 1 Higher/Lower Register (AMnIVT1ADDRH/L) address is completed.
		Specified address 2 write completed (video data)	CRUnINTS2.VDADR2WS	CRUnIE2.VDADR2WE	AXI-VD	When AXI-VD bus write to the AXI-VD Bus Transfer Completion Event Address 2 Higher/Lower Register (AMnIVT2ADDRH/L) address is completed.
		Specified address 3 write completed (video data)	CRUnINTS2.VDADR3WS	CRUnIE2.VDADR3WE	AXI-VD	When AXI-VD bus write to the AXI-VD Bus Transfer Completion Event Address 3 Higher/Lower Register (AMnIVT3ADDRH/L) address is completed.
		Specified address 4 write completed (video data)	CRUnINTS2.VDADR4WS	CRUnIE2.VDADR4WE	AXI-VD	When AXI-VD bus write to the AXI-VD Bus Transfer Completion Event Address 4 Higher/Lower Register (AMnIVT4ADDRH/L) address is completed.
		Specified address 1 write completed* ¹ (statistics data)	CRUnINTS2.SDADR1WS	CRUnIE2.SDADR1WE	AXI-SD	When AXI-SD bus write to the AXI-SD Bus Transfer Completion Event Address 1 Higher/Lower Register (AMnISDVT1ADDRH/L) address is completed.

Note 1. CRU0 and 1 are valid.

Note 2. An interrupt from SVC is detected in image processing. It is not detected in non-image processing/full bypass.

Note 3. The ICnINTCTRL.DECINT register allows you to set whether or not interrupts are issued for subsampled frames.

Note 4. If an error interrupt occurs, a reset is required after the CRU has been stopped.

9.2.4.4.2 Designated address forwarding completion interrupt

This interrupt function is used to notify the CPU of the timing of fetching video and statistics data by using an interrupt.

Set the desired addresses in the DRAM as the destinations for data transfer by using the AMnIVT0/1/2/3/4ADDRH/L (for video data) and AMnSDIVT0ADDRH/L (for statistics data) registers. Up to four addresses can be set for video data and up to two addresses can be set for statistics data. These must be set according to the number of memory banks and the speed of reading by the CPU. An interrupt is generated upon the completion of writing to the address set by the AXI master (AXI-VD, AXI-SD).

Figure 9.2-55 shows an example when the number of memory banks is 4 for video data and the last addresses of memory banks 2 and 4 are set as the completion addresses.

The CPU reads data from the initial data in bank 1 in response to the first interrupt as a trigger and then reads up to the end of bank 2. When the next interrupt arrives, the CPU reads data from the initial data in bank 3 and then reads up to the end of bank 4. This is thereafter repeated and consecutive data can thus be read.

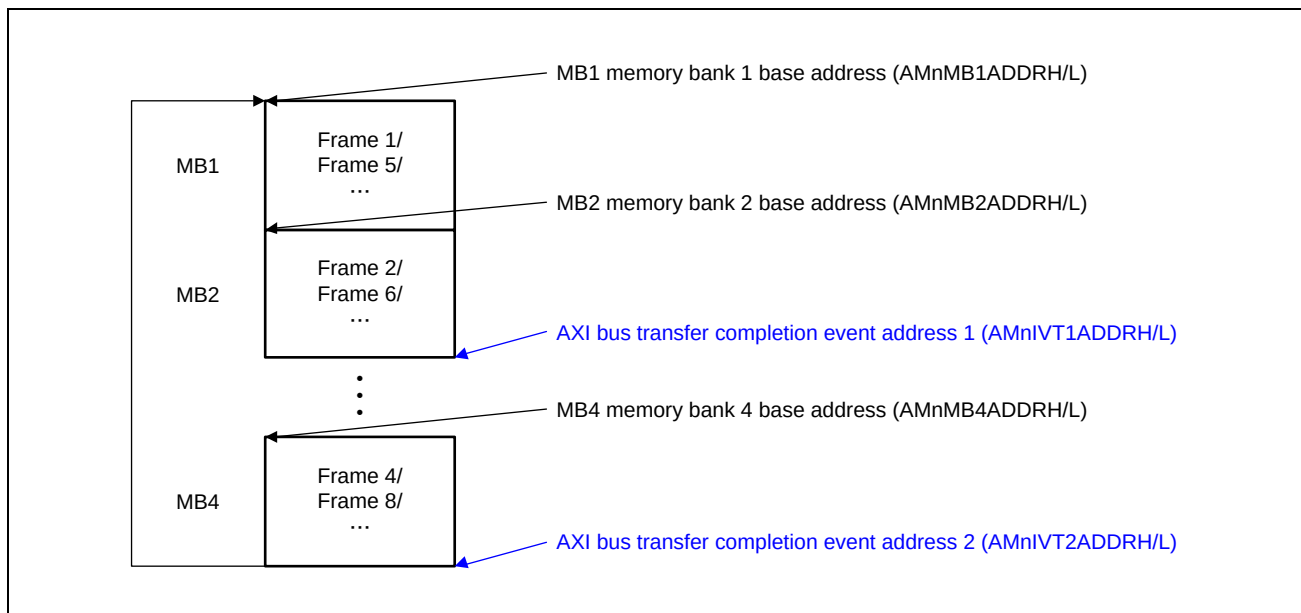


Figure 9.2-55 Example when the Number of Memory Banks is 4 for Video Data and the Last Addresses of Memory Banks 2 and 4 are Set as the Completion Addresses

9.2.4.5 Usage Notes

Use of this CRU requires caution on the following points.

- (1) When the paired channels are used, set VCLK for CRU0 and CRU1 to the same frequency.
- (2) The following functions can only be used with SVC0 (that is, they cannot be used with SVC1 to SVC3).
 - Demosaicing
 - Statistics
 - Changing the frame synchronization
- (3) The same data (the same data type for the same VC) cannot be set for image processing and non-image processing. For example, making the following register settings is prohibited.
 - (Case1) This is an example of setting the same VC data type in Image Process to SVC0 and SVC1.
 - SVC0: VC0, YUV422 8-bit: ICnSVC.SVC0[3:0] = 0h, ICnIPMC_C0.INF[5:0] = 1Eh
 - SVC1: VC0, YUV422 8-bit: ICnSVC.SVC1[3:0] = 0h, ICnIPMC_C1.INF[5:0] = 1Eh
 - (Case2) This is an example of setting the same VC data type in Non Image Process to SVC0 and SVC1.
 - SVC0: VC0, YUV422 8-bit: ICnSVC.SVC0[3:0] = 0h, ICnNIPDT_C0L.INF[5:0] = 1b
 - SVC1: VC0, YUV422 8-bit: ICnSVC.SVC1[3:0] = 0h, ICnNIPDT_C1L.INF[5:0] = 1b
 - (Case3) This is an example of setting the same VC data type in Image Process and Non Image Process on the same SVC0.
 - SVC0: VC0, YUV422 8-bit: ICnSVC.SVC0[3:0] = 0h, ICnIPMC_C0.INF[5:0] = 1Eh
 - SVC1: VC0, YUV422 8-bit: ICnNIPDT_C0L[30]=1b
- (4) Interlacing is not supported.
- (5) The timing for periodic deskewing calibration for the DPHY is 2¹³ to 10 μ s. Be sure to use a sensor that conforms to this specification.
- (6) When FIFO-VD and FIFO-SD are in use, make sure that they do not overflow. In the event that an overflow occurs during operation with footers appended, the footer value following the overflow is not guaranteed. In such cases, reset the CRU after it has been stopped.
- (7) An interrupt will not occur unless the addresses that are set for the specified address 0 to 4 write completed interrupt are set as addresses to which video or statistics data are to be written, so take care on this point. For example, an interrupt will not occur if video or statistics data are not written to the address which has been set for the specified address 0 to 4 write completed interrupt and the address jumps to the start of the next line due to the image-stride setting.
- (8) For the order of the startup of the sensor and CRU, be sure to initialize the CRU before sending data from the sensor, as stated in **9.2.5.1 Starting Reception for the MIPI CSI-2 Input**. If this order is not observed, the CRU may receive incomplete packets of MIPI CSI-2 signals and various errors may occur.

(9) Parameter setting of the SVCs are as follows.

Table 9.2-25 Parameter Setting of the SVCs

Function	SVC0	SVC1	SVC2	SVC3	Note
Image Clipping	(A)	(A)	(A)	(A)	—
Frame Subsampling	(B)	(B)	(B)	(B)	—
Data Arrange 1/2	(B)	(B)	(B)	(B)	—
Data Clip	(B)	(B)	(B)	(B)	—
Color Space Conversion	(B)	(B)	(B)	(B)	—
Demosaic	(C)	(D)	(D)	(D)	When any of the Demosaic, Linear Matrix, and Statistic functions are used, SVC1 to SVC3 cannot be used for other functions.
Linear Matrix	(C)	(D)	(D)	(D)	
Statistics	(C)	(D)	(D)	(D)	
LUT	(B)	(B)	(B)	(B)	—
YCbCr422 to a420	(A)	(A)	(A)	(A)	—
Y/C Separation	(A)	(A)	(A)	(A)	—
Only Y	(A)	(A)	(A)	(A)	—

Note: (A): Can be set independently of other SVCs
 (B): Parameter setting common to SVC0 to SVC3
 (C): Functions only with SVC0
 (D): Cannot be set

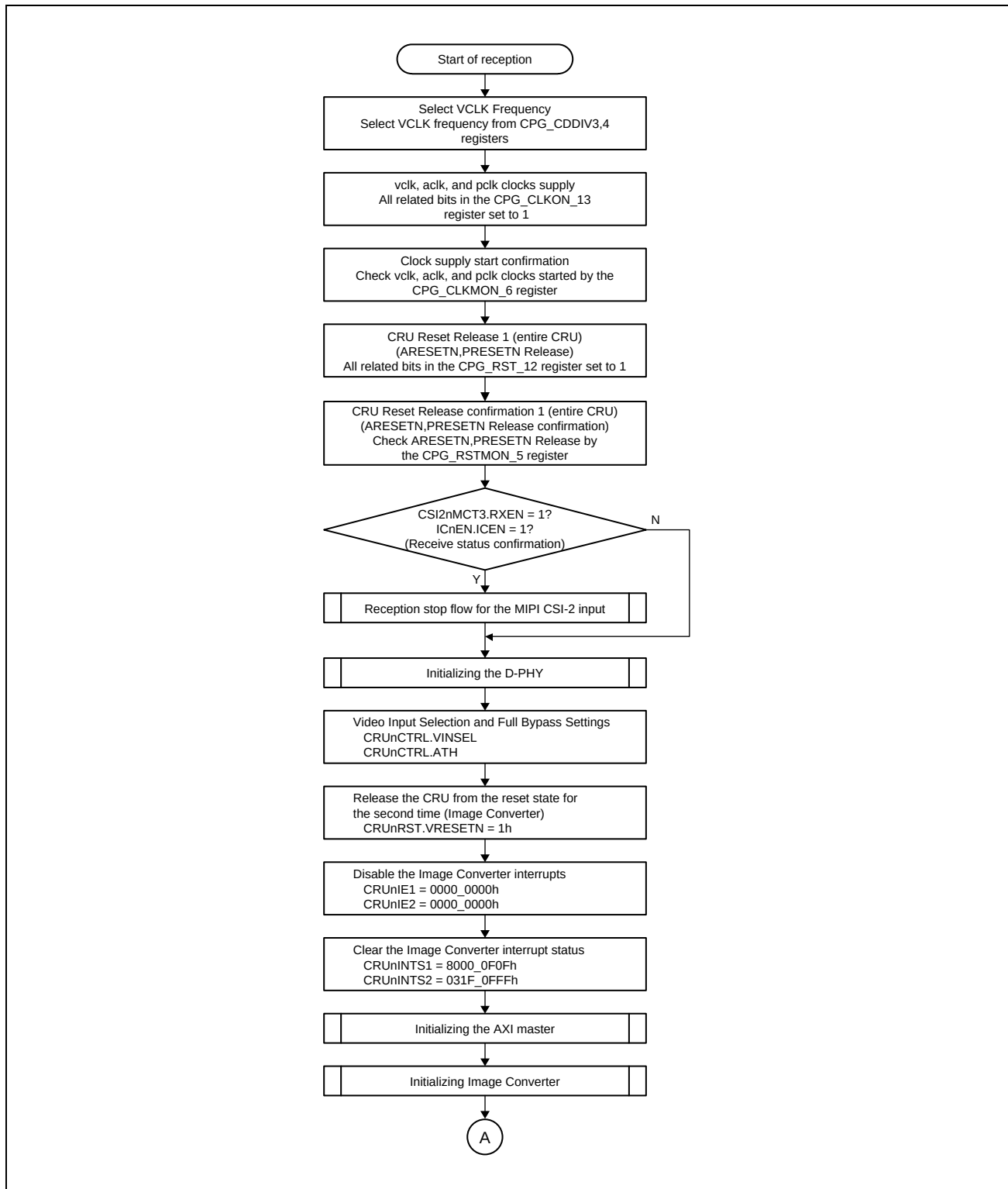
(10) Image Stride when output format is YUV420

Note that when the output format is set to YUV420 (ICnDMR.YCMODE[2:0] = 110b), the U/V side is written to RAM as Image Stride (AMnIS) $\times$ 2.

See **Image Stride** in **9.2.4.3.3(5)(a) Video data frame** for details.

9.2.5 Operation

9.2.5.1 Starting Reception for the MIPI CSI-2 Input



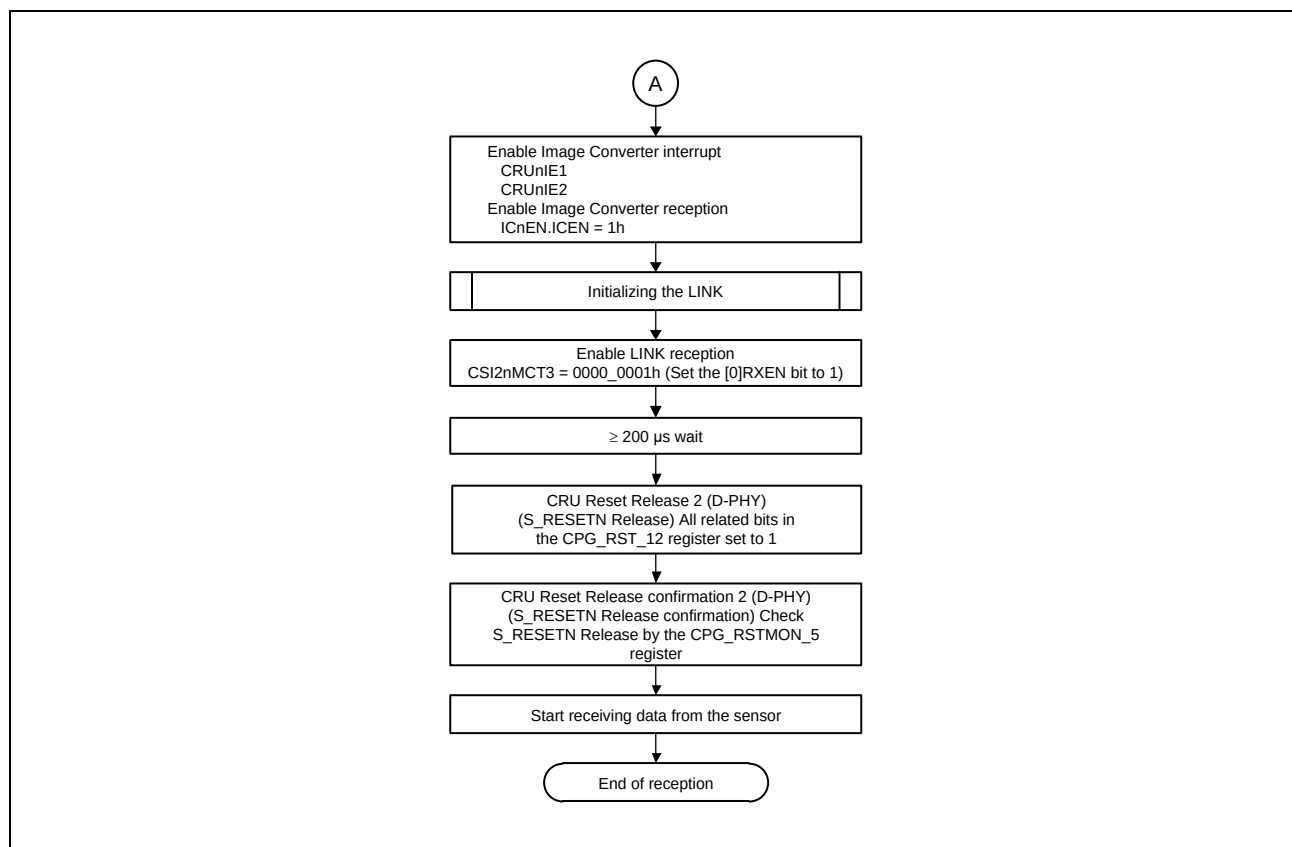


Figure 9.2-56 Reception Start Flow for the MIPI CSI-2 Input

9.2.5.1.1 Initializing the D-PHY

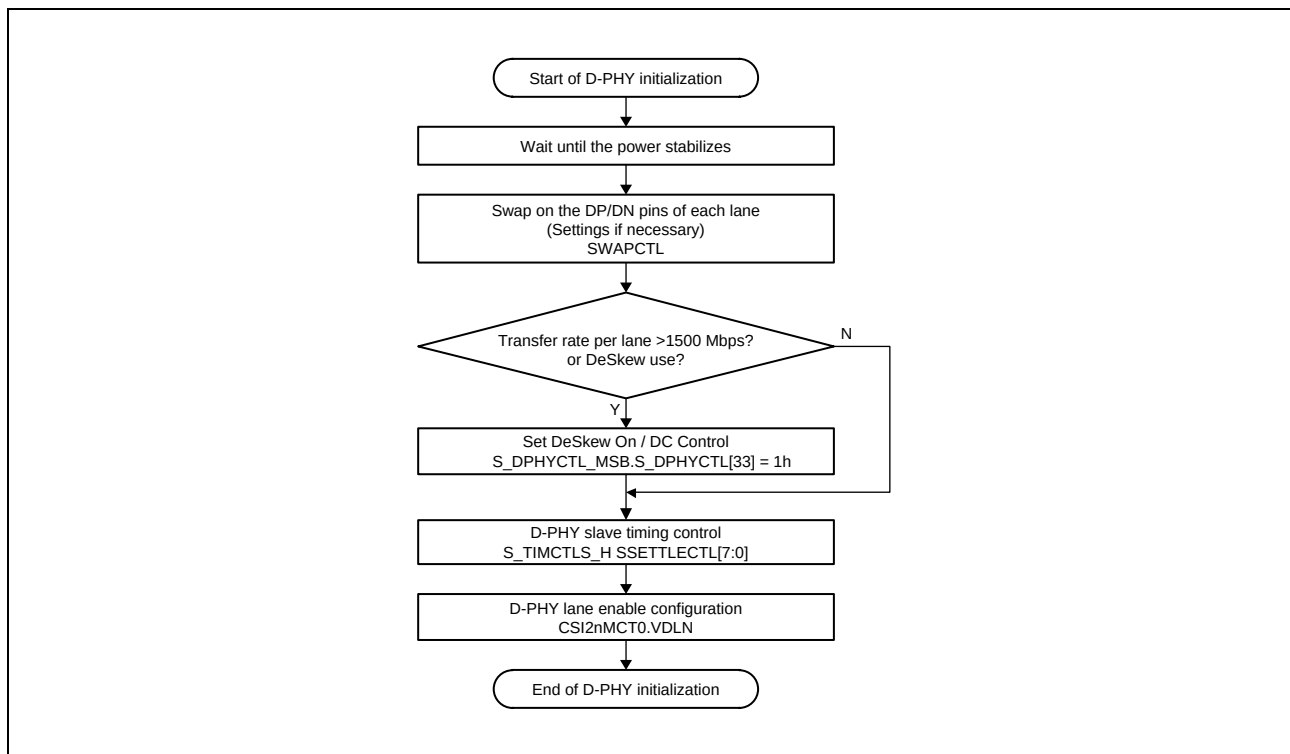


Figure 9.2-57 Initializing the D-PHY

9.2.5.1.2 Initializing the AXI Master

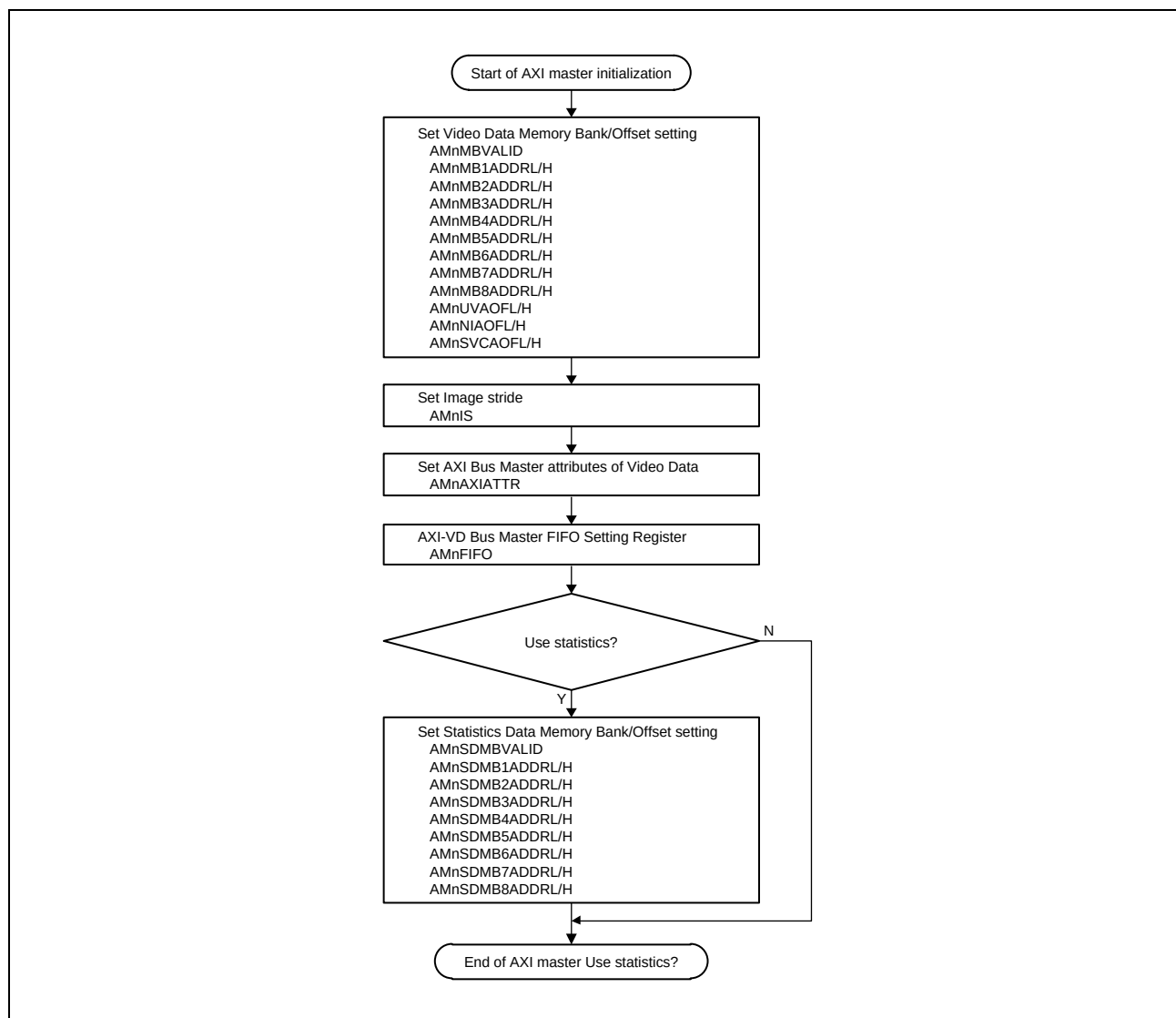
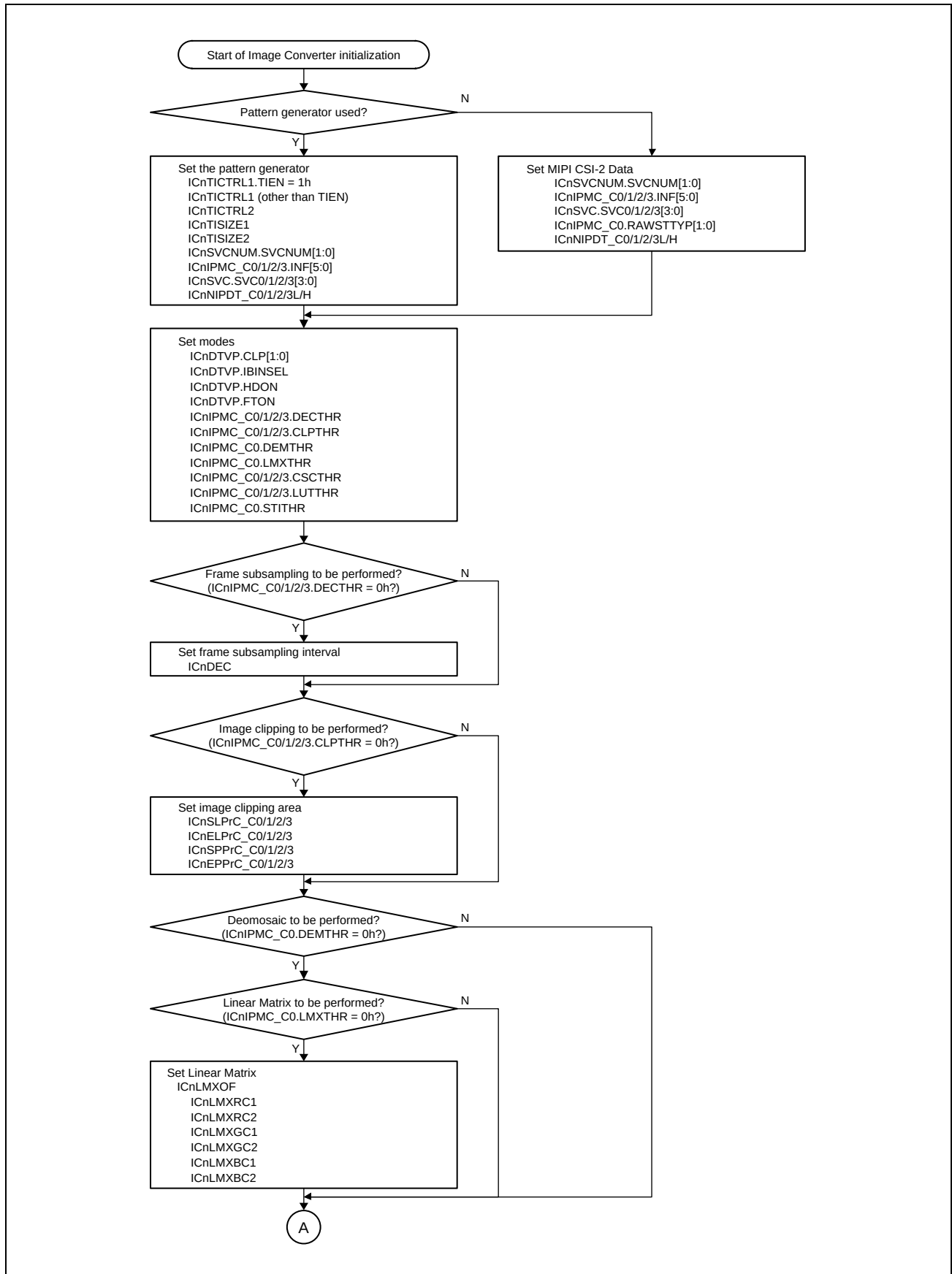


Figure 9.2-58 Initializing the AXI Master

9.2.5.1.3 Initializing the Image Converter



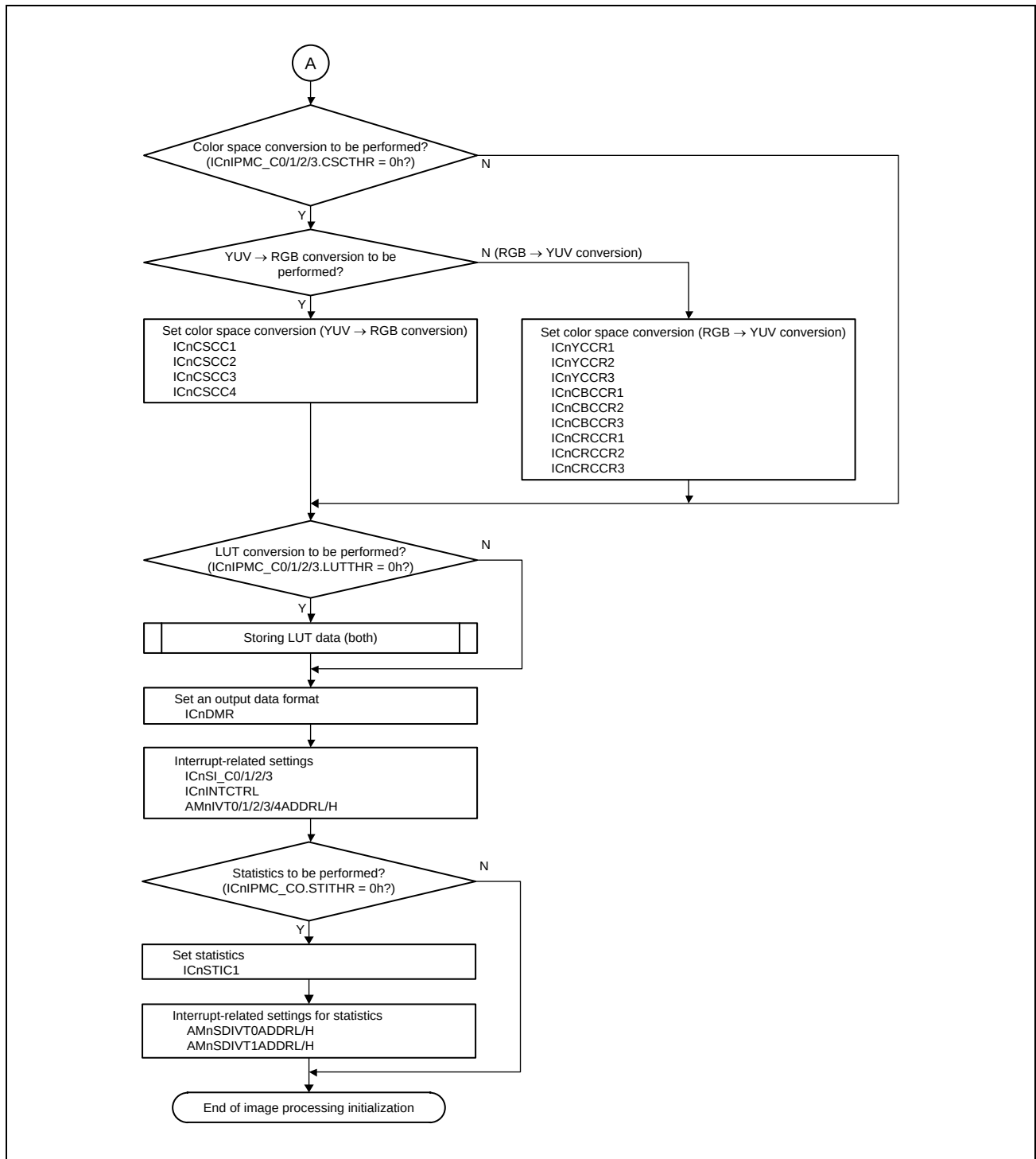


Figure 9.2-59 Initializing Image Processing

9.2.5.1.4 Storing LUT Data

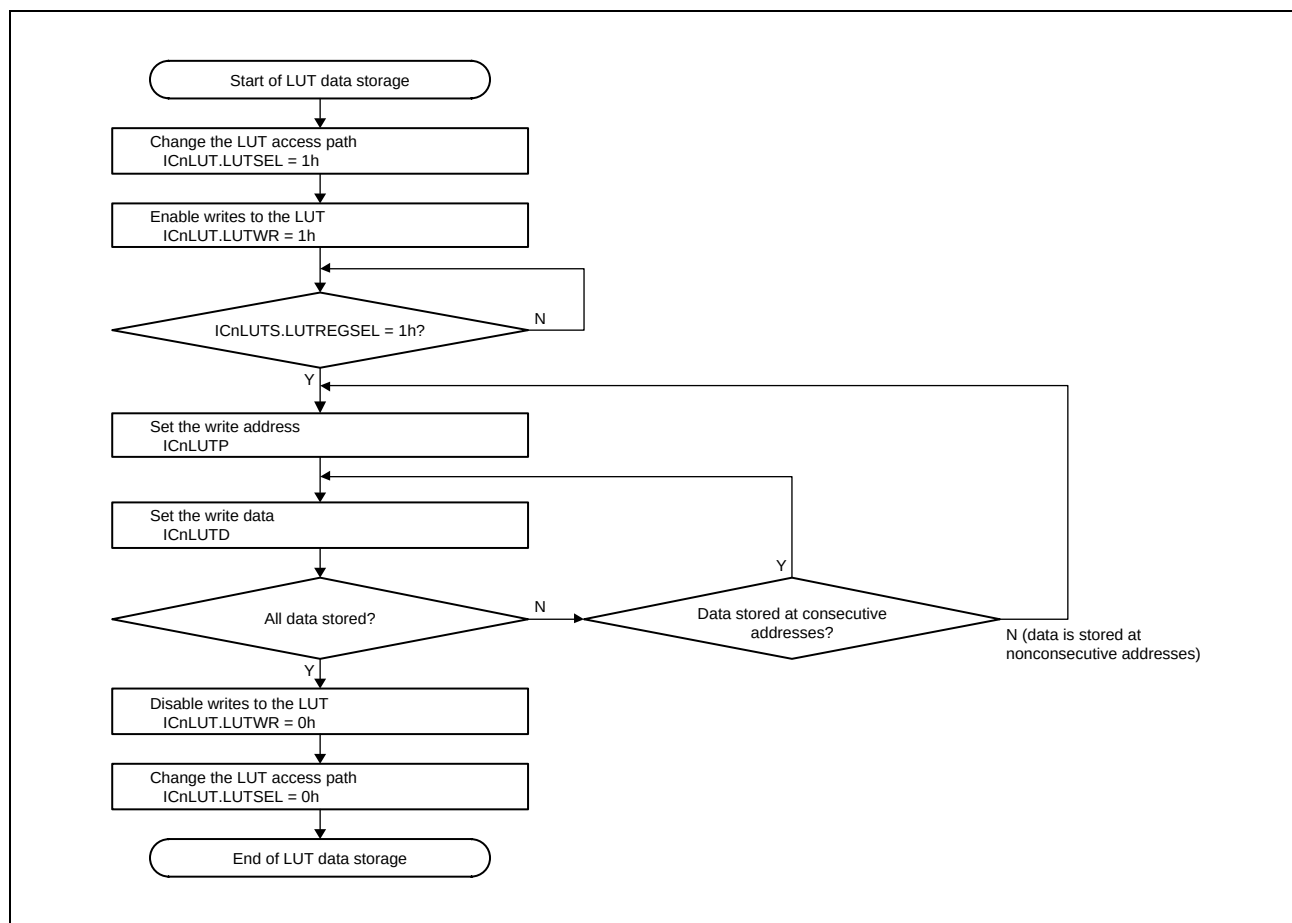


Figure 9.2-60 Storing LUT Data

9.2.5.1.5 Initializing the LINK

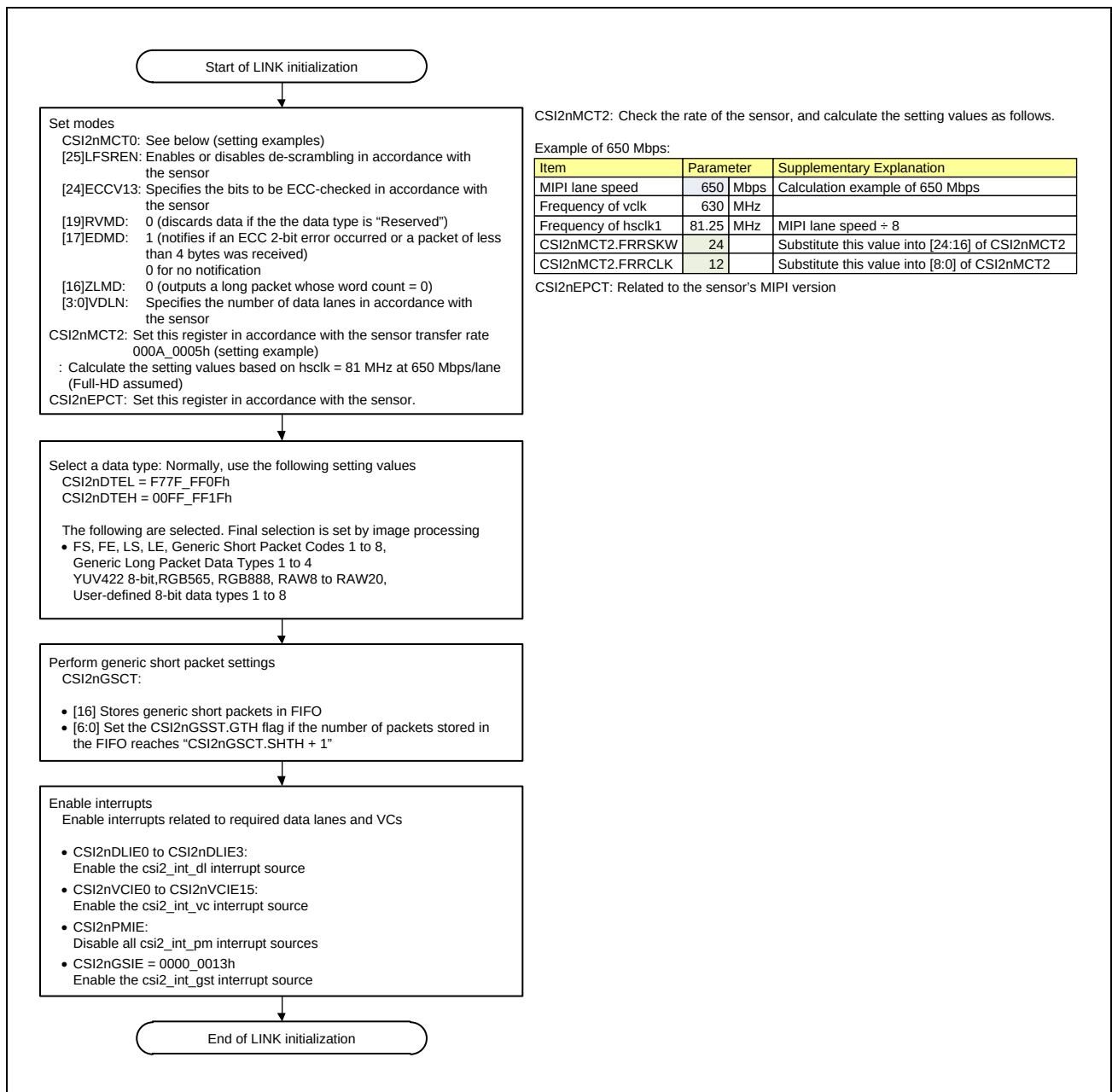


Figure 9.2-61 Initializing the LINK

9.2.5.2 Stopping Reception for the MIPI CSI-2 Input

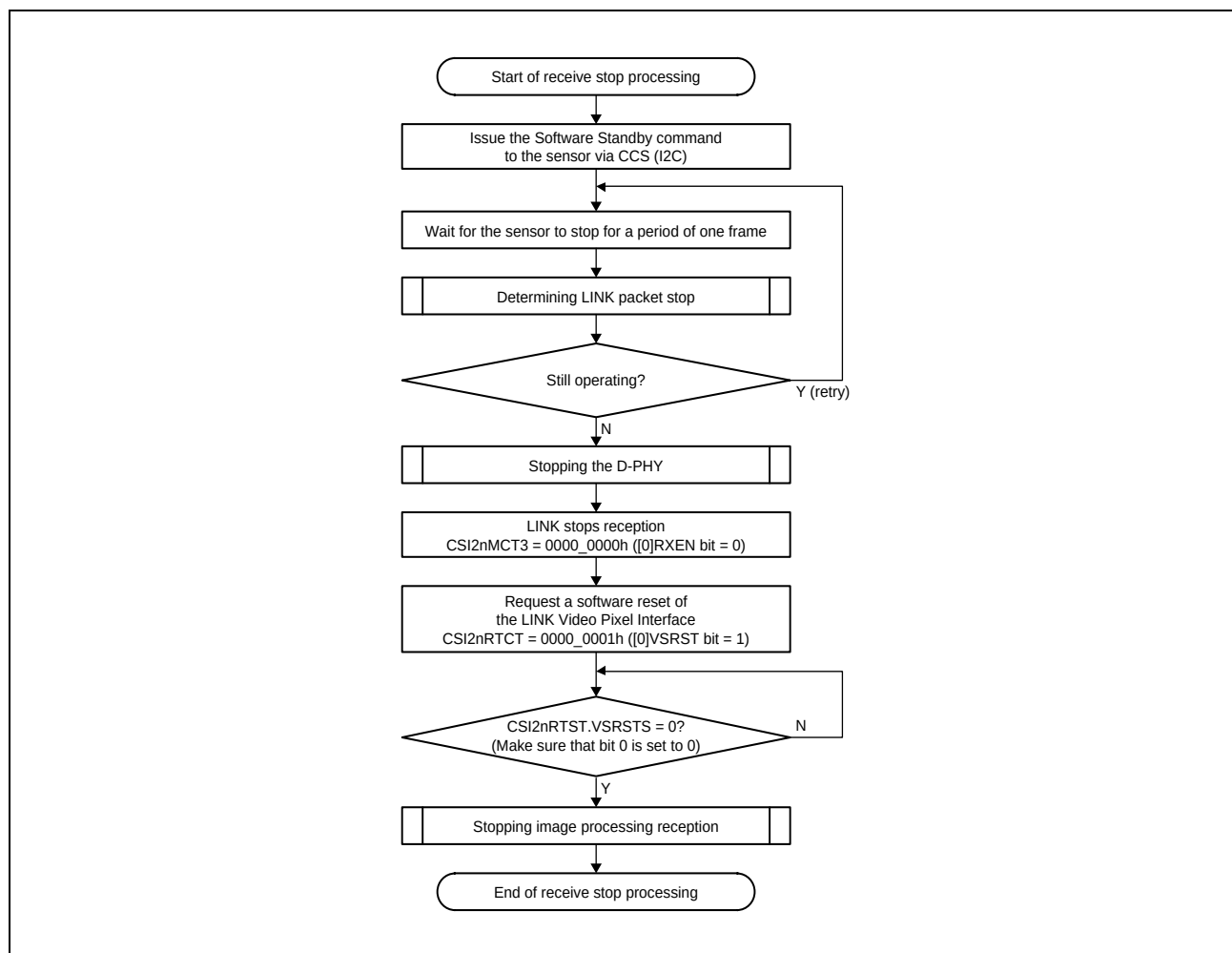


Figure 9.2-62 Flow of Stopping Reception for the MIPI CSI-2 Input

9.2.5.2.1 Determining LINK Packet Stop

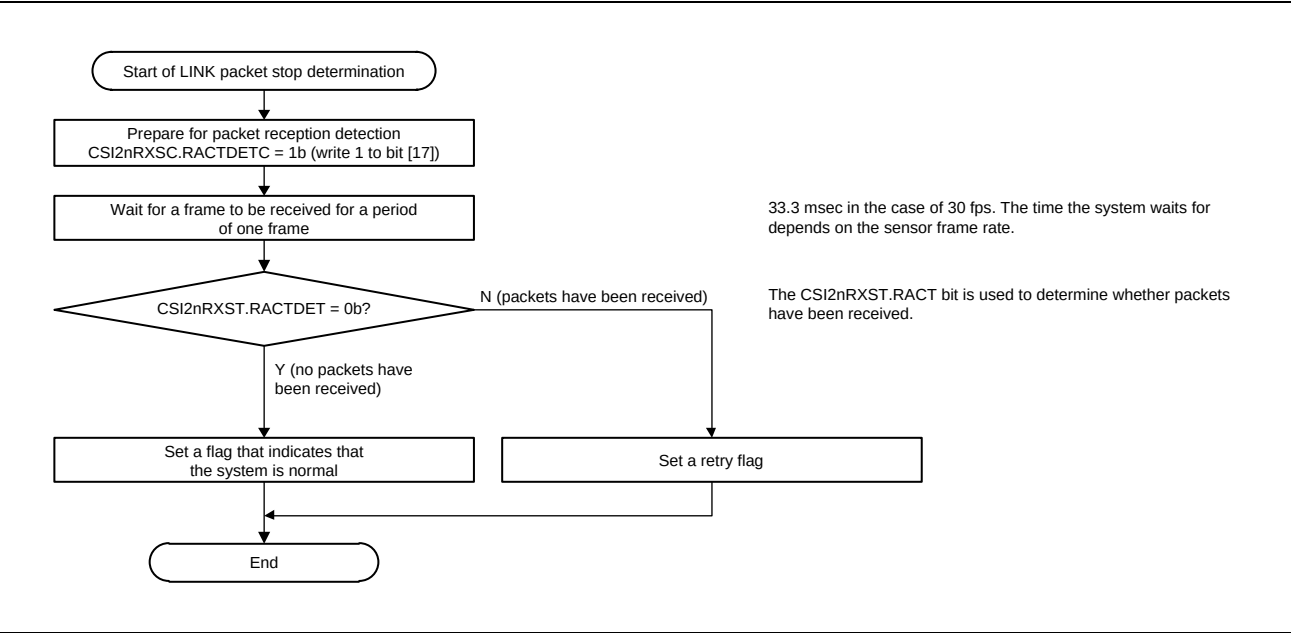


Figure 9.2-63 Determining LINK Packet Stop

9.2.5.2.2 Stopping the D-PHY

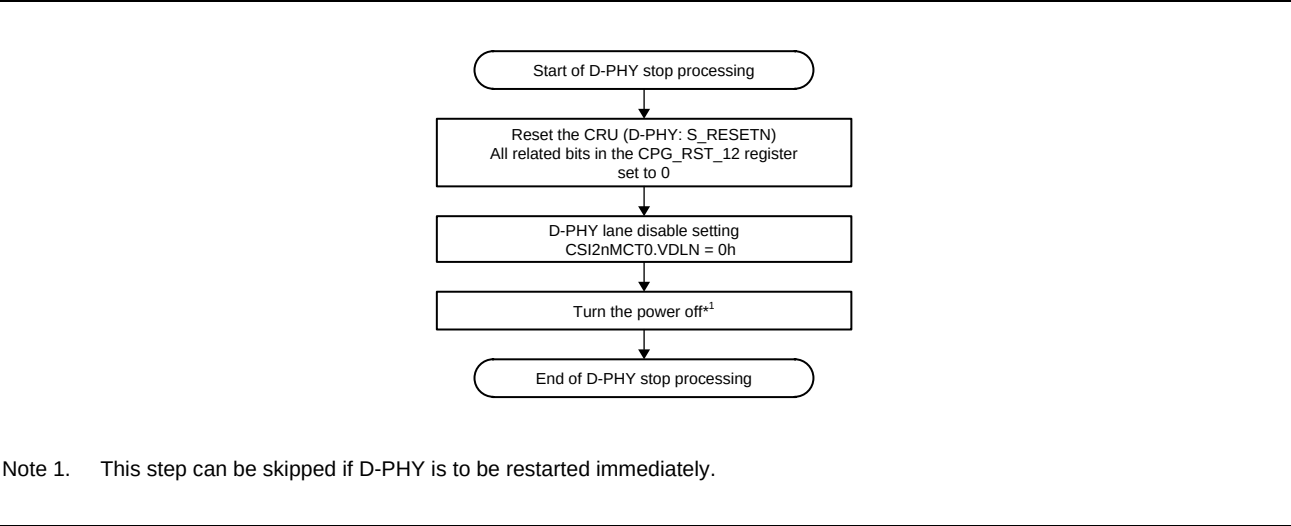
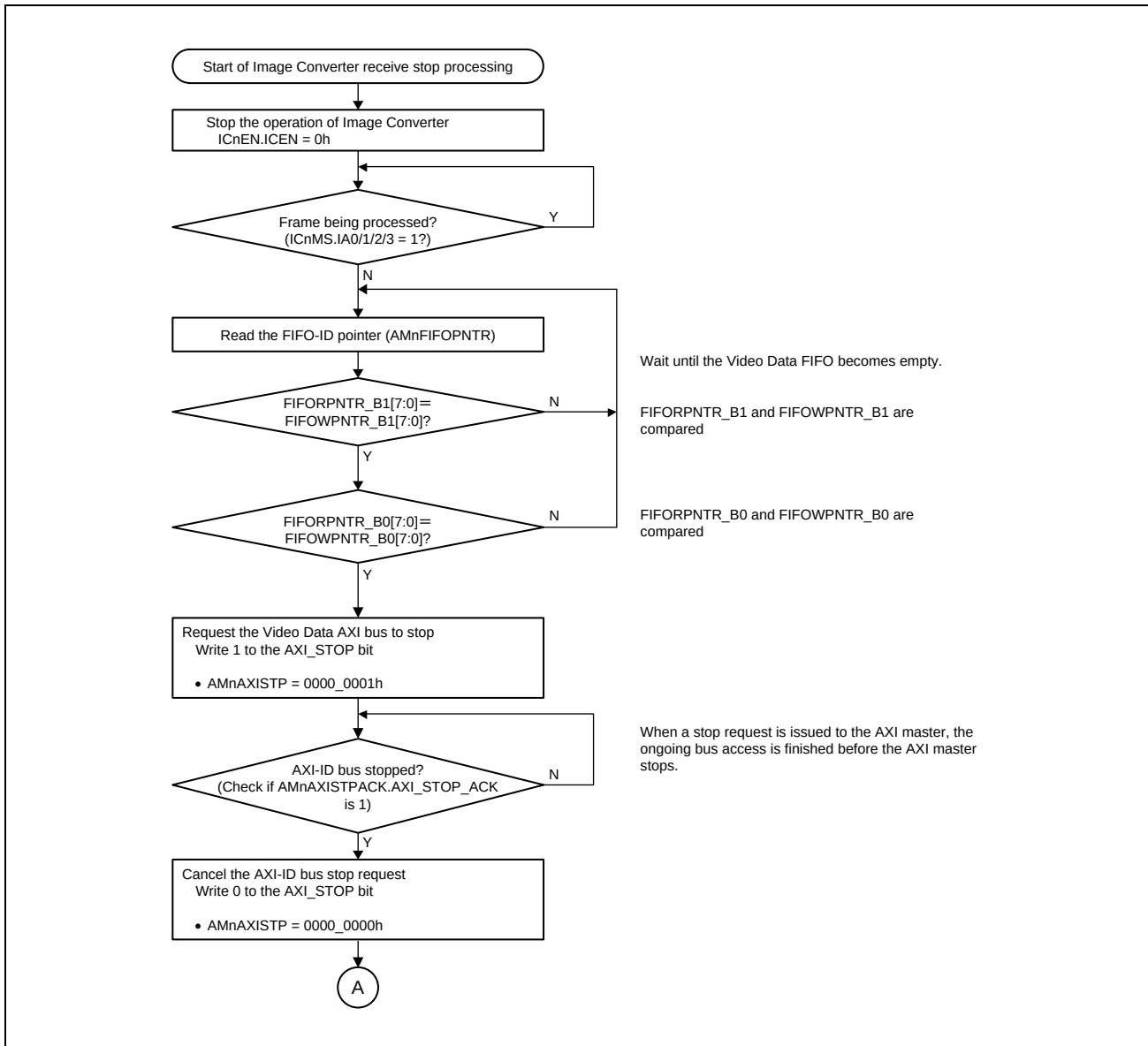


Figure 9.2-64 Stopping the D-PHY

9.2.5.2.3 Stopping Image Converter Reception



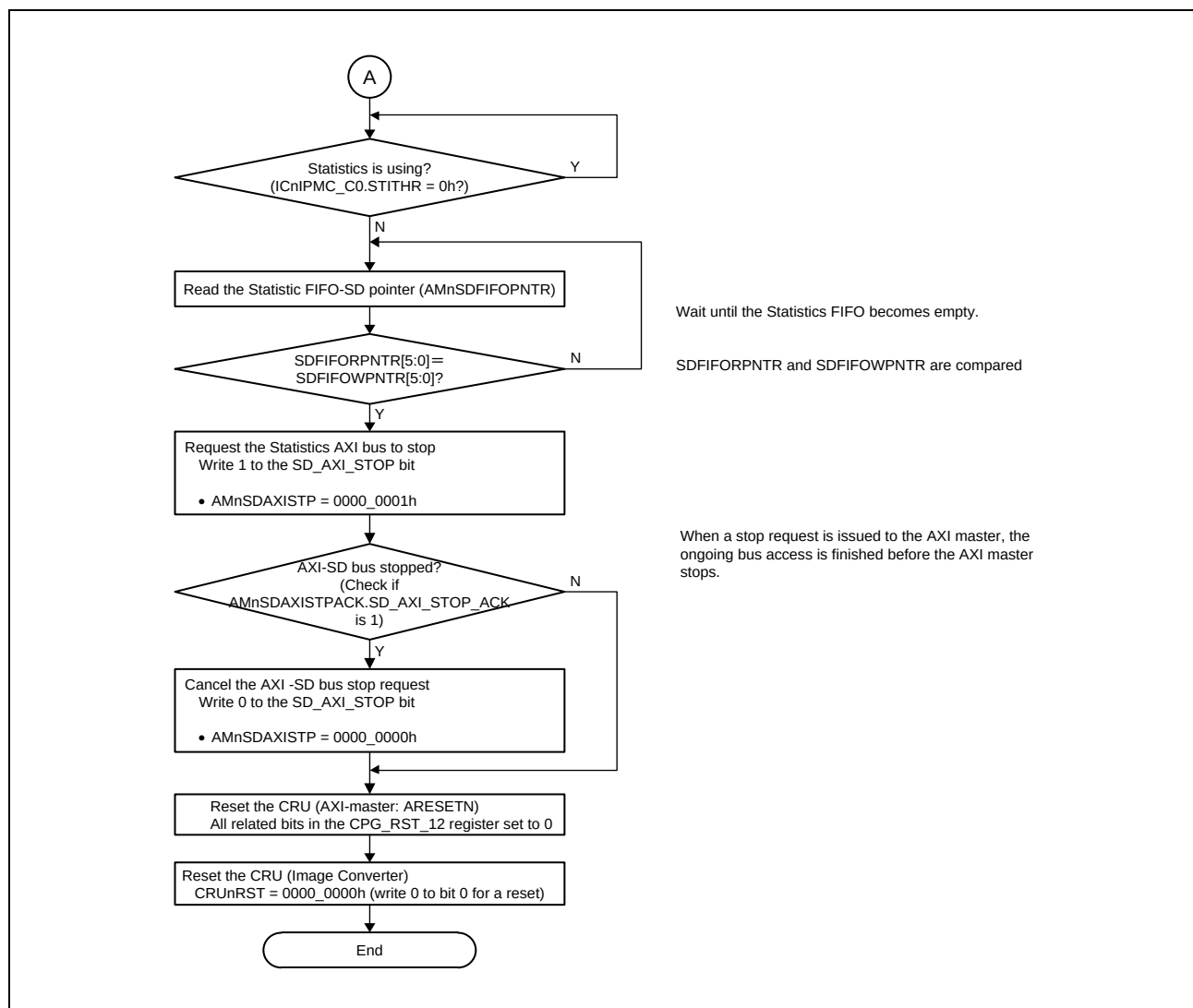


Figure 9.2-65 Stopping Image Processing Reception

9.2.5.3 Frame Synchronization Settings

While the CRU is receiving data using SVC0, the following registers can only be changed with frame synchronization:

- ICnSVC.SVC0[3:0]
- ICnIPMC_C0.STITHR
- ICnNIPDT_C0L
- ICnNIPDT_C0H
- ICnSLPrC_C0
- ICnELPrC_C0
- ICnSPPrC_C0
- ICnEPPrC_C0
- ICnSI_C0
- ICnLMXOF
- ICnLMXRC1
- ICnLMXRC2
- ICnLMXGC1
- ICnLMXGC2
- ICnLMXBC1
- ICnLMXBC2
- ICnSTIC1

9.2.5.3.1 Changing Frame Synchronization Settings (by Setting Registers)

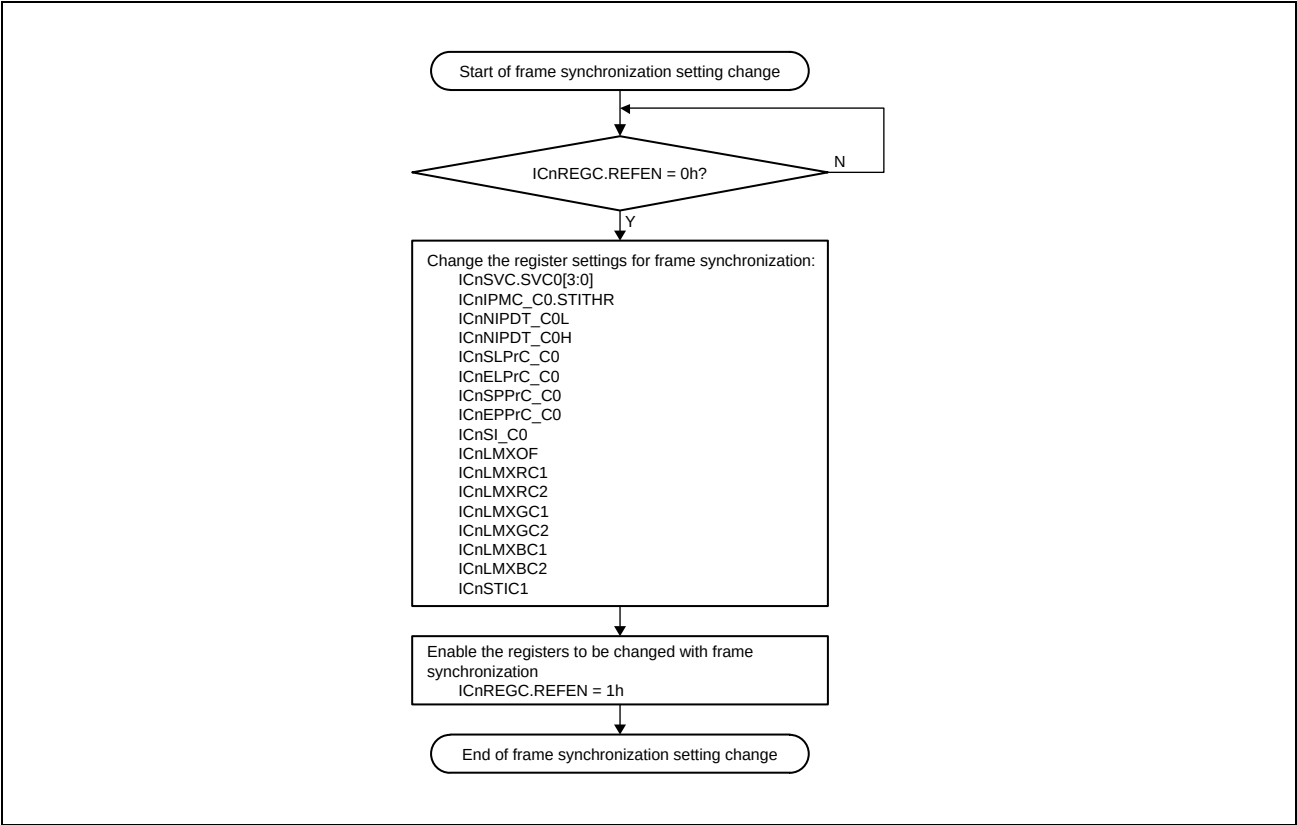


Figure 9.2-66 Flow for Changing the Frame Synchronization Settings (by Setting Registers)

9.2.5.3.2 Changing Frame Synchronization Settings (by Using the DMAC)

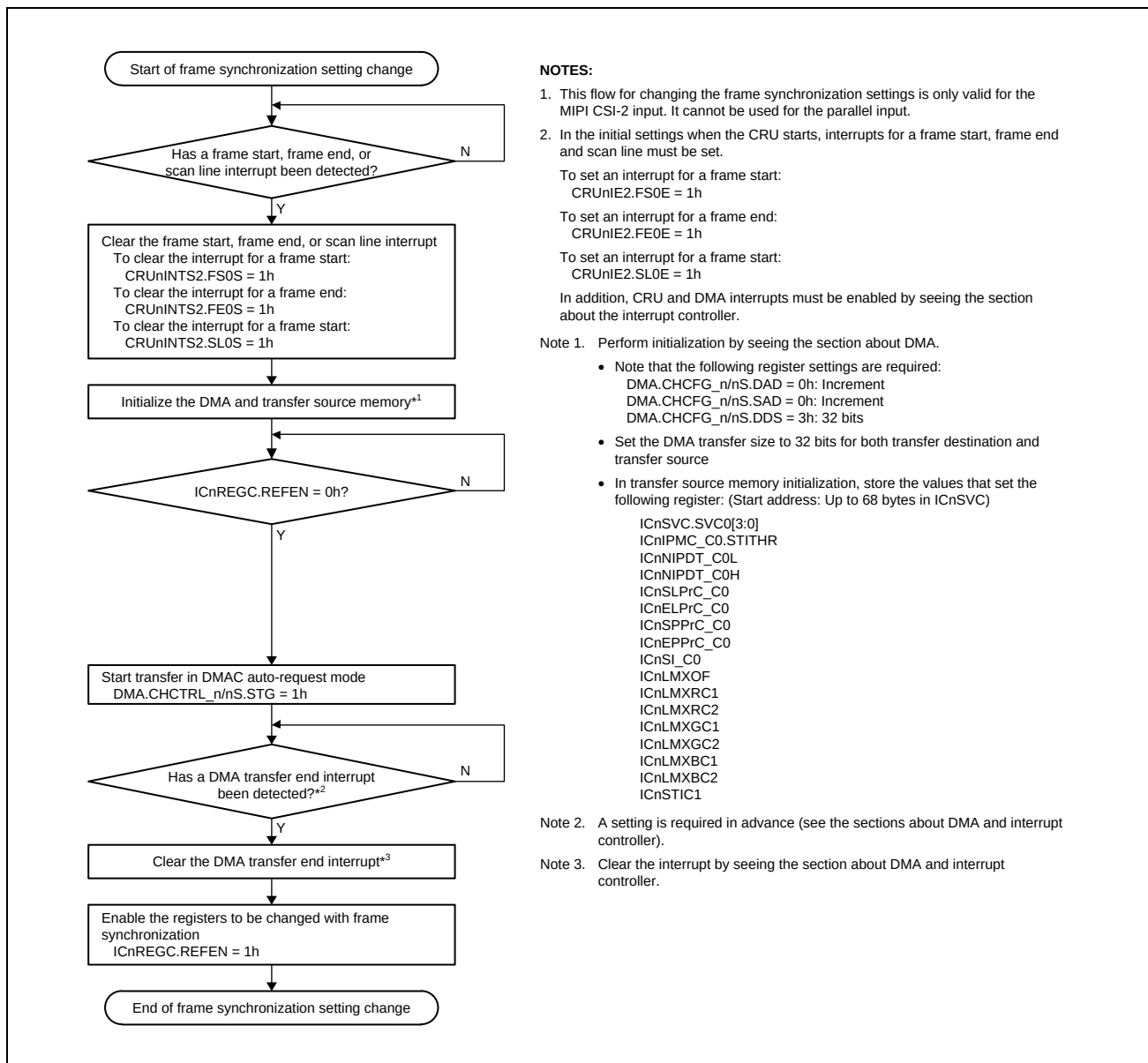


Figure 9.2-67 Flow for Changing the Frame Synchronization Settings (by Using the DMAC)

9.2.5.4 Starting Reception for the Pattern Generator Input

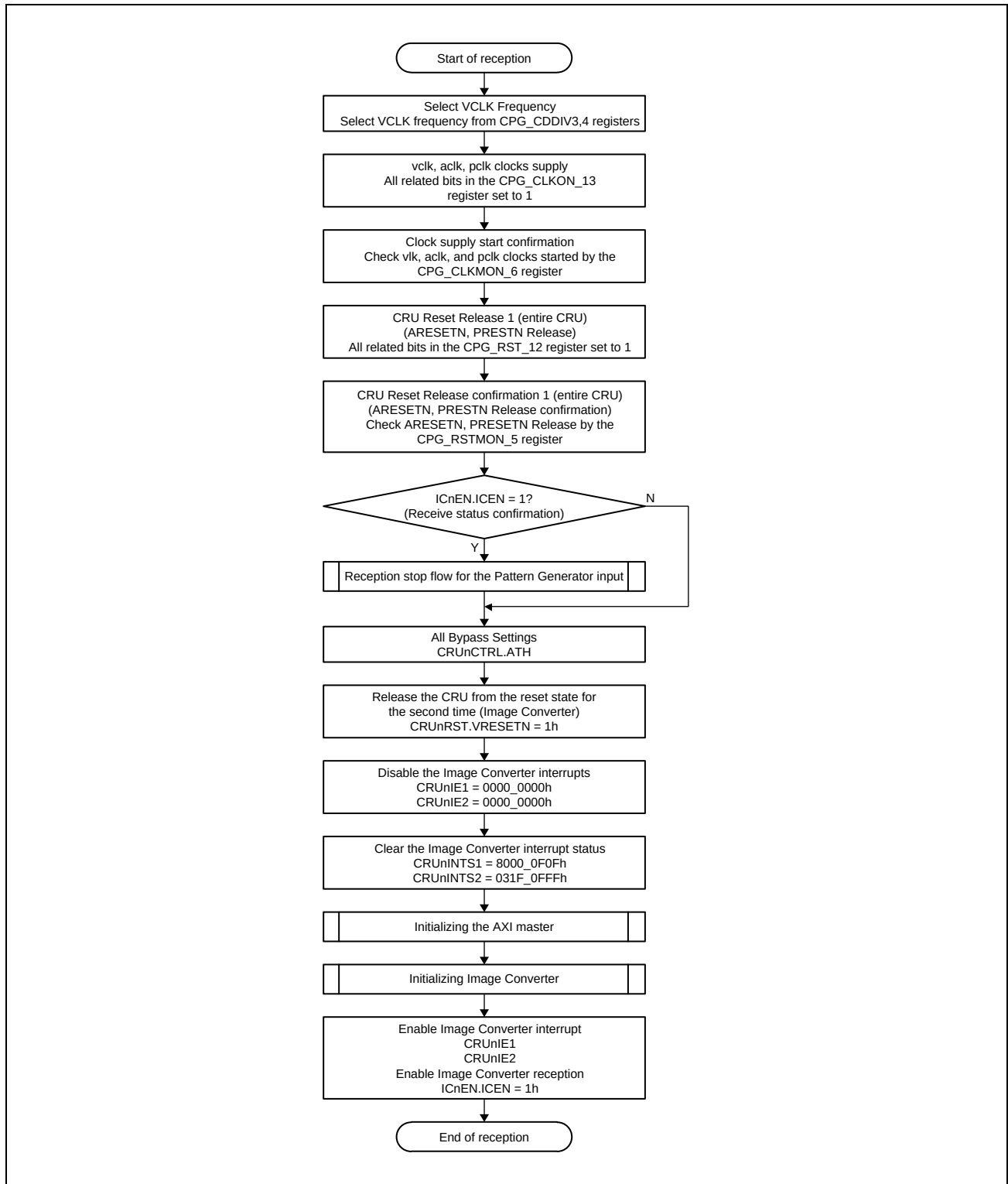


Figure 9.2-68 Reception Start Flow for the Pattern Generator Input

9.2.5.5 Stopping Reception for the Pattern Generator Input

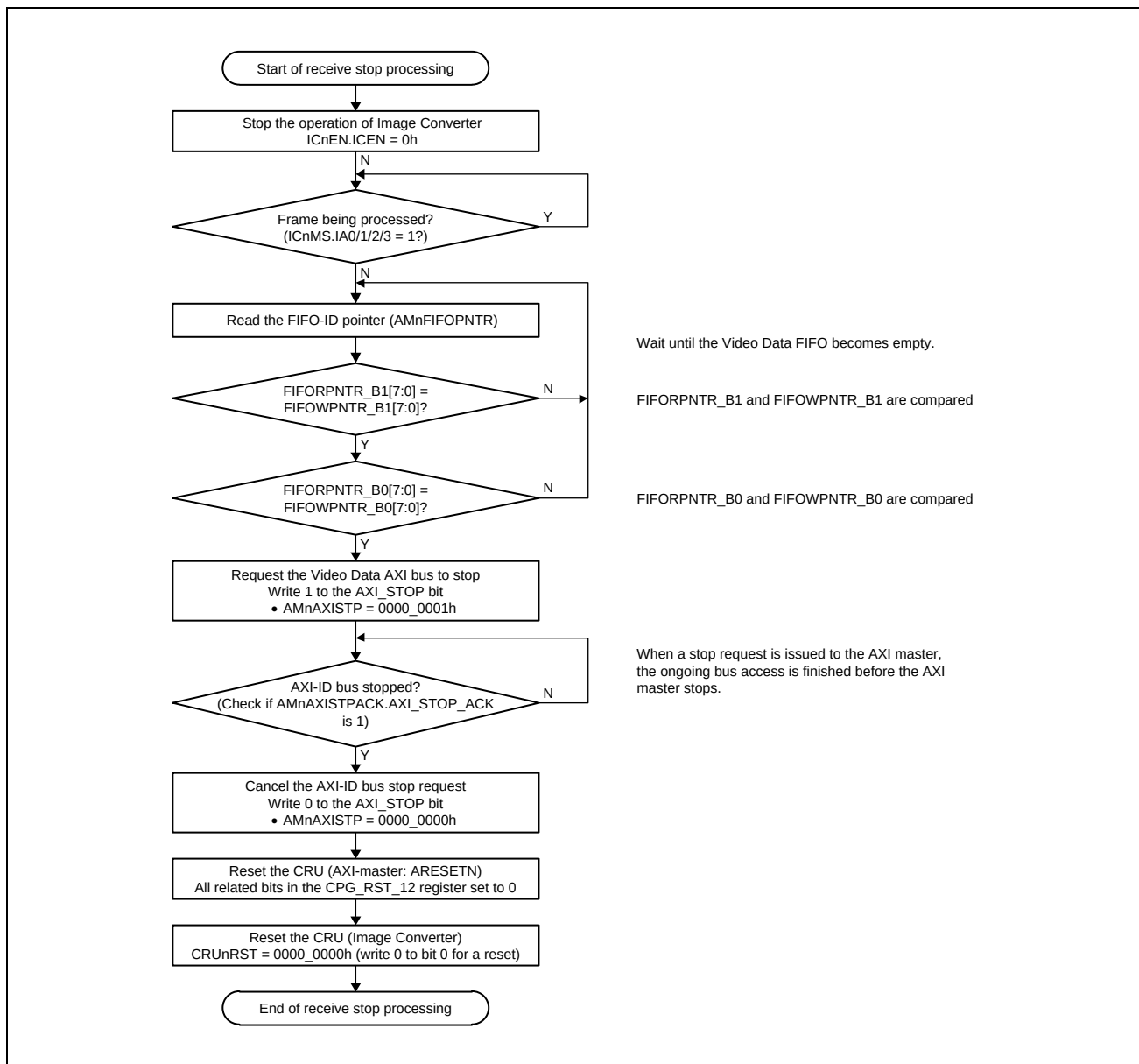


Figure 9.2-69 Flow of Stopping Reception for the Pattern Generator Input

9.2.5.6 **How to Synchronize CRU0 to CRU1**

If CRU0 to 1 are connected with several cameras, they must be synchronized on the sensor side basically in accord with the sensor specification. This section describes the methods for checking synchronization.

Making the interrupt settings described below allows the CPU to adjust the timing of fetching data and to check synchronization with each camera.

(1) CRU0/1 specified address interrupt setting

To obtain synchronization for CRU0/1, by setting the specified address interrupts (AMnIVT1 to 4ADDRH/L) respectively for the video data bus and statistics data bus, the state of synchronization with each CRU can be confirmed. For example, **Figure 9.2-70** illustrates an example in which the given address of the same memory bank is set per CRU. This is an example of setting the specified address interrupt (AMnIVT1ADDRH/L) for the last address of the MB2 frame for each CRU. In this case, if the same amount of data is accumulated in each CRU, interrupts occur in the order in which data have been accumulated, so the state of synchronization can be identified by having the CPU check the occurrence of an interrupt in each CRU.

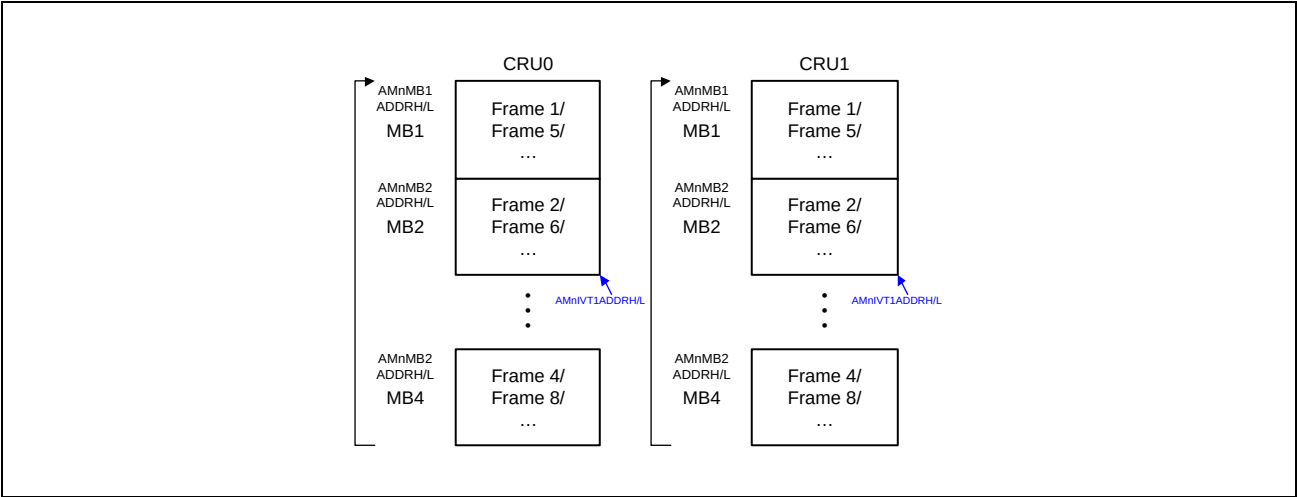


Figure 9.2-70 Example in which the Given Address of the Same Memory Bank is Set per CRU

(2) Footer function of memory bank (option for when stagnation due to congestion is a concern)

If only making the setting (1) above raises a concern with stagnation due to congestion on the buses, whether or not the data have actually been written can be checked by the CPU reading the footers at the above address, which allows judging whether the actual data from the past frames have been updated.

SECTION 9 IMAGE

9.3 Image Scaling Unit (ISU)

The ISU is a module that reads out the image stored in the external DRAM, scales down image size, and outputs the reduced image into the DRAM. The ISU also support a color format conversion and cropping image.

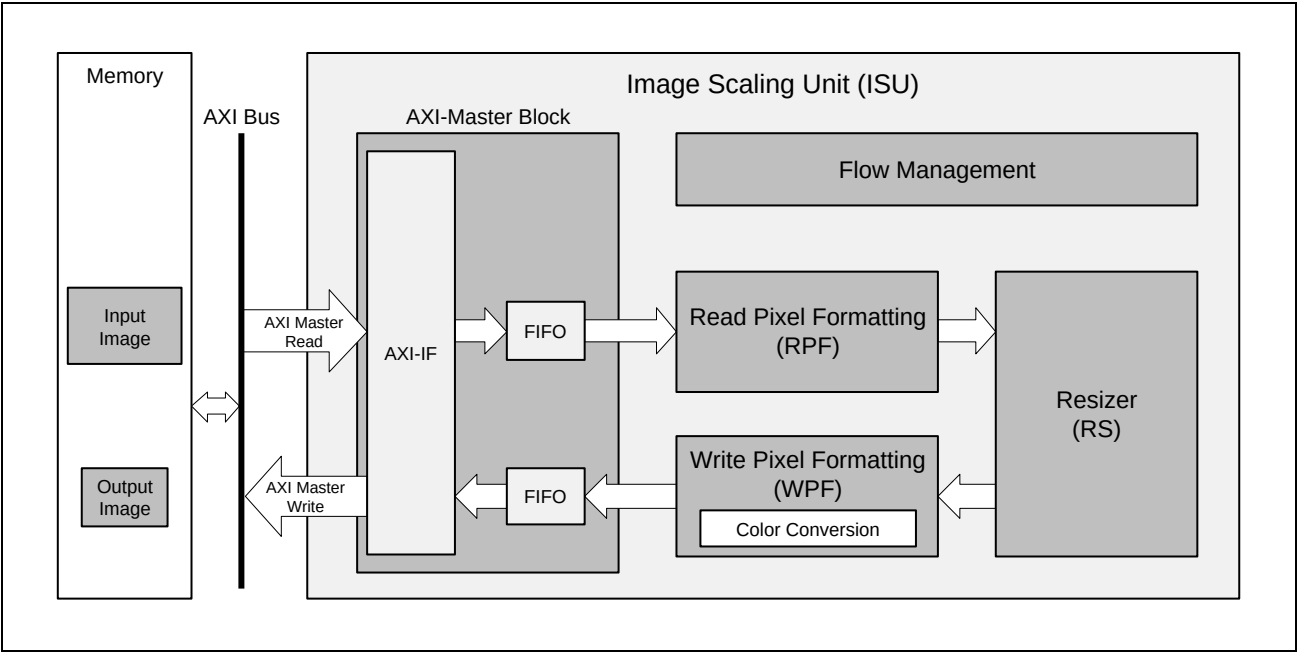


Figure 9.3-1 ISU Block Diagram

9.3.1 Features

The ISU functionality consists of “Resizer” and “Color Format Conversion”. “Color Format Conversion” can convert color space and color format (RGB/ARGB, YCbCr/YUV, RAW).

- Resizer
 - Scaling ratio: $\times 1/1$ to $1/15$ (Scale down)
 - Scaling algorithm: Bilinear
 - Horizontal and Vertical image scaling ratio are adjustable individually.
 - Cropping image
- Color Space and Color Format Conversion
 - Supported color format: RGB/ARGB, YCbCr/YUV, RAW (Grayscale)
A (Alpha) of ARGB format is transparency. Zero is transparent (colorless).
 - Color space conversion by using 3×3 matrix which coefficients can be set arbitrary.
 - Convert color space and color format each other.
 - Endian correction

Table 9.3-1 Supported Image Format

Item	Description
Input	• Color Format – RGB/ARGB: 8 formats RGB565, RGB888, BGR888, BGR666, ARGB8888, ARGB1555, RGBA8888, ABGR8888 – YCbCr/YUV: 4 formats YCbCr422 (Interleave) = UYVY YCbCr422 (Interleave) = YUY2 (YUYV) YCbCr422 (Semi-Planar) = NV16 YCbCr420 (Semi-Planar) = NV12 – RAW: 8 formats RAW6, RAW7, RAW8, RAW10, RAW12, RAW14, RAW16, RAW20 (Grayscale)*1 • Maximum Size 4096 × 4096
Output	• Color Format – RGB/ARGB: 8 formats RGB565, RGB888, BGR888, BGR666, ARGB8888, ARGB1555, RGBA8888, ABGR8888 – YCbCr/YUV: 4 formats YCbCr422 (Interleave) = UYVY YCbCr422 (Interleave) = YUY2 (YUYV) YCbCr422 (Semi-Planar) = NV16 YCbCr420 (Semi-Planar) = NV12 – RAW: 8 formats RAW6, RAW7, RAW8, RAW10, RAW12, RAW14, RAW16, RAW20 (Grayscale)*1 • Maximum Size 4096 × 4096

Note 1. RAW10, RAW12, RAW14, RAW16, and RAW20 are rounded to 8-bit, and RAW6 and RAW7 are expanded to 8-bit internally.

9.3.1.1 Flow Management (FM)

Controls the processing of the entire ISU. The frame processing is controlled by receiving the frame processing start instruction from the register. There are three types of frame processing:

1. Register Mode
2. Descriptor Mode (without automatic start of next frame)
3. Descriptor Mode (with automatic start of next frame)

9.3.1.1.1 Frame Processing Mode

There are the following three types of frame processing methods depending on the register settings.

- Register Setting Mode (1) is one-time processing operation according to the register setting at once.
- Descriptor Mode without Auto-start (2) is also one-time processing operation according to the descriptor, not the register setting. Descriptor Mode with Auto-start (3) automatically reads out the descriptor and continues its operation until all descriptors have been processed.

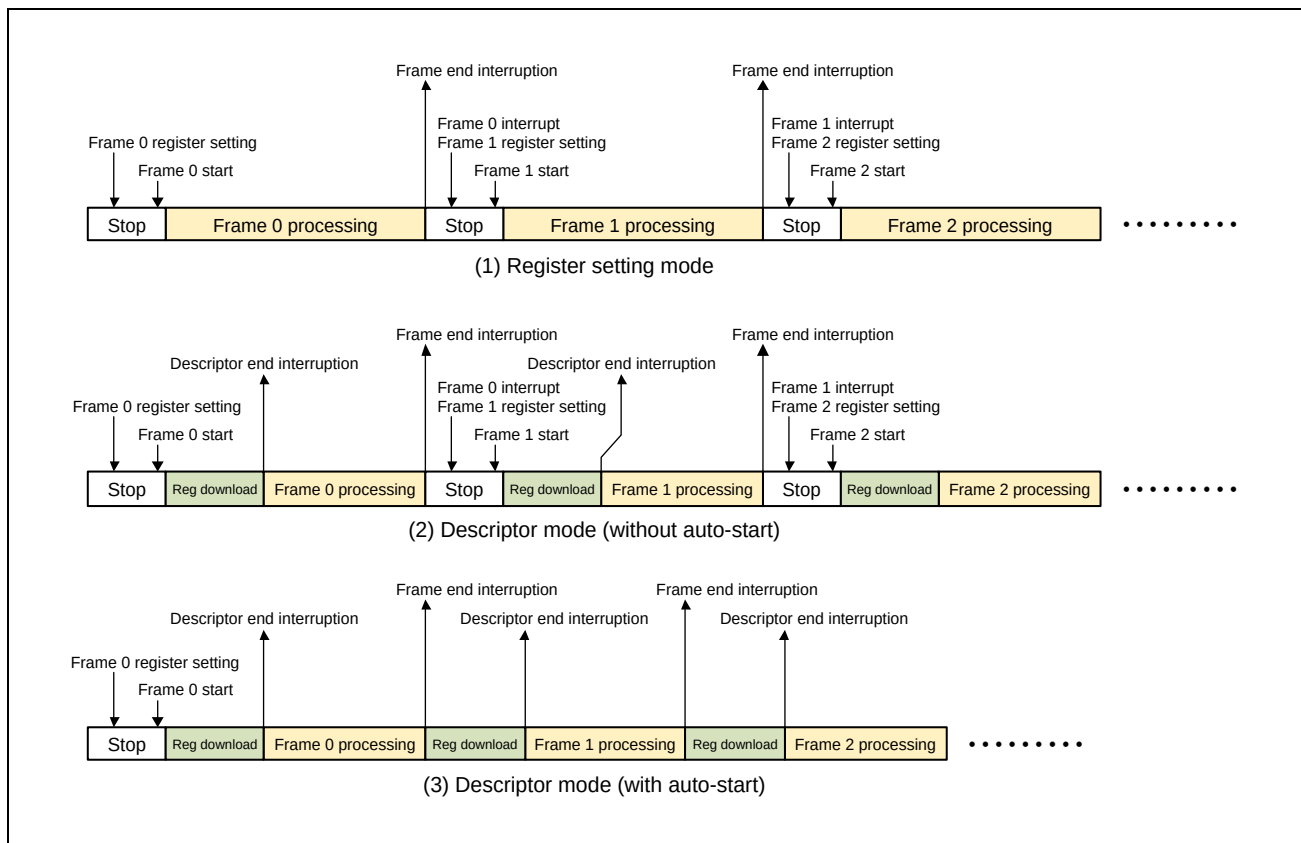


Figure 9.3-2 Frame Processing Mode

9.3.1.1.2 Descriptor

Descriptor Mode is the processing which reads out the descriptor format data stored in the memory and sets the registers automatically. It also includes the Auto-start setting for the next frame.

Flow Management (FM) reads this descriptor setting data while not frame processing.

(1) Descriptor format

The descriptor format is shown below.

Table 9.3-2 Descriptor Format

No.	Category	Item	Byte	Setting Range	Description
1	Header*1,*2	Descriptor Table Byte Count Setting	4 bytes	0000_0000h to 0000_2000h	Up to 8 Kbytes. The total number of bytes of register address and register setting data.
2	Register Setting Main body*3,*4,*5	Register Address 1	4 bytes	0000_0100h to 0000_01F0h	1st Register Setting Address
		Register Setting Data 1	4 bytes	0000_0000h to FFFF_FFFFh	1st Register Setting Data
		Register Address 2	4 bytes	0000_0100h to 0000_01F0h	2nd Register Setting Address
		Register Setting Data 2	4 bytes	0000_0000h to FFFF_FFFFh	2nd Register Setting Data
		Register Address 3	4 bytes	0000_0100h to 0000_01F0h	3rd Register Setting Address
		Register Setting Data 3	4 bytes	0000_0000h to FFFF_FFFFh	3rd Register Setting Data
		Register Address 4	4 bytes	0000_0100h to 0000_01F0h	4th Register Setting Address
		Register Setting Data 4	4 bytes	0000_0000h to FFFF_FFFFh	4th Register Setting Data
		•	•	•	•
		•	•	•	•
		•	•	•	•
3	Footer*2	Next Descriptor Table Address	8 bytes	0000_0000_0000_0000h to 0000_0002_3FFF_FFFFh	The next address is a multiple of 32 bytes. The setting range will be changed according to the product specifications.
		Control identifier register	4 bytes	0000_0000h to 0000_0003h	b31 to 2: All-0 b1: Select bit to generate a frame end interruption. 0: Does not generate a Frame end interruption. 1: Generate a Frame end interruption. b0: Select bit for auto-start of next frame or end at this frame. 0: Disable next frame auto-start. (Exit without next frame operation.) 1: Enable next frame auto-start.

Note 1. The location of the Descriptor List is a 32-byte boundary. (Address that is a multiple of 32-byte) 0,32,64 ...

Note 2. The number of bytes is fixed for the header and footer. The register body is 8 bytes per set (address 4 bytes, data 4 bytes)

Note 3. The registers that can be set in the descriptor are limited to the Common register.

Note 4. Operation when the same register address is described in multiple addresses is not guaranteed.

Note 5. Specify in the offset address to be added to the ISU Base Address. Specifically, the lower 10 bits [9:0] should be the address of the register map, and the higher 22 bits [31:10] should be the address of All 0b. If the higher 22 bits are other than All 0b, it will be out of the register area and a list error will occur.

(2) Descriptor error

An error flag is issued in the following cases.

- a) When the number of bytes in the header exceeds the limit, or when the value is not in units of 8 bytes
 - An error interrupt is generated, and subsequent frame processing is not executed.
 - Normal operation and recovery are not guaranteed unless the system is reset.
- b) When the register address is outside the specified address
 - A decoding error will occur if decoding is performed for items outside the register area.
 - Ignore the write to the register where the error occurred and continue the process itself.

9.3.1.2 AXI-Master

In response to a descriptor read request or input image read request from Flow Management (FM), the descriptor or input image is read from the DRAM. It also writes a reduced image to DRAM in response to a reduced image write request from the WPF.

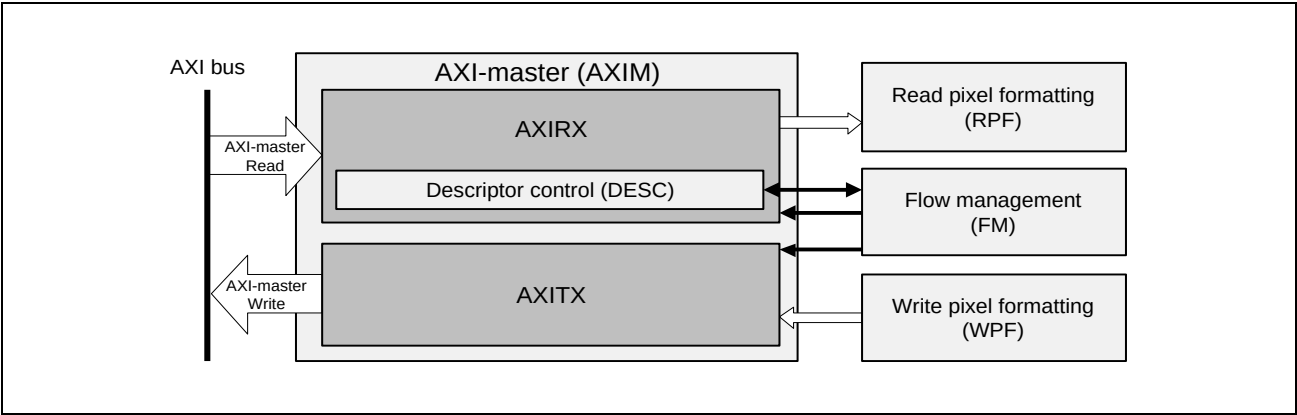


Figure 9.3-3 AXI-Master Block Diagram

9.3.1.2.1 Read input image (AXIRX)

In response to the descriptor read request from FM or the frame transfer start request, the descriptor or input image data is read from DRAM by AXI.

(1) Image input format

AXIRX support to read following formats from memory.

Table 9.3-3 Input Image Format

Read Method	Image Format	Variation
Interleave (Plane Number = 1)	ARGB (8 types)	- The order of colors is different / the number of color bits is different. - There may or may not be A.
	YCbCr422 8-bit (2 types)	The two types differ only in the order of the colors of Cb and Cr.
	RAW (8 types)	Grayscale 6-/7-/8-/10-/12-/14-/16-/20-bit.
Semi-Planner (Plane Number = 2)	YCbCr422 8-bit (1 type)	
	YCbCr420 8-bit (1 type)	

• Interleave

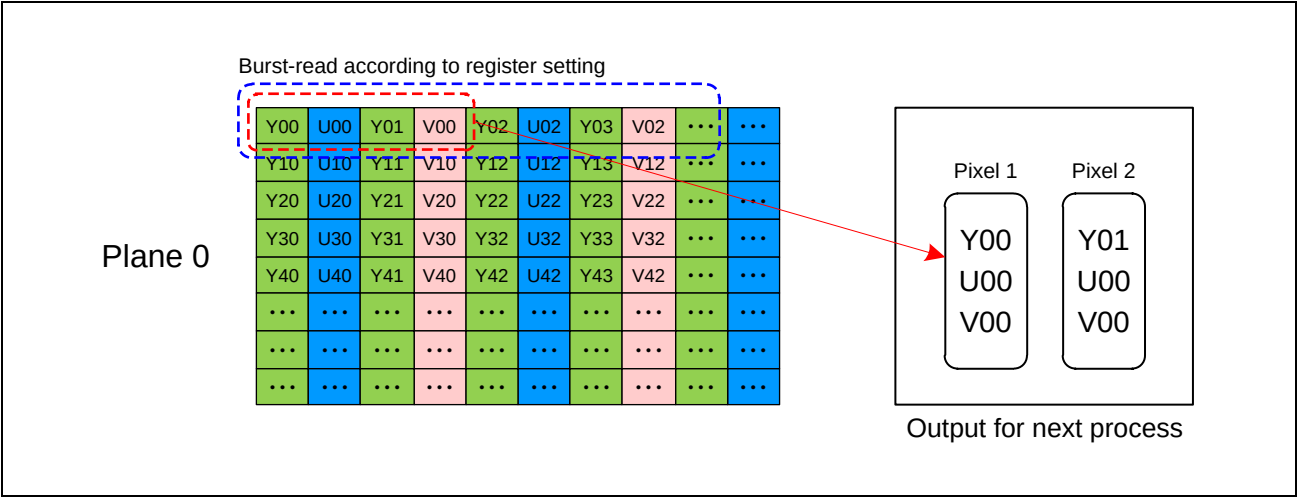


Figure 9.3-4 Example of Interleave Format for YCbCr422 (YUV422)

• Semi-planar

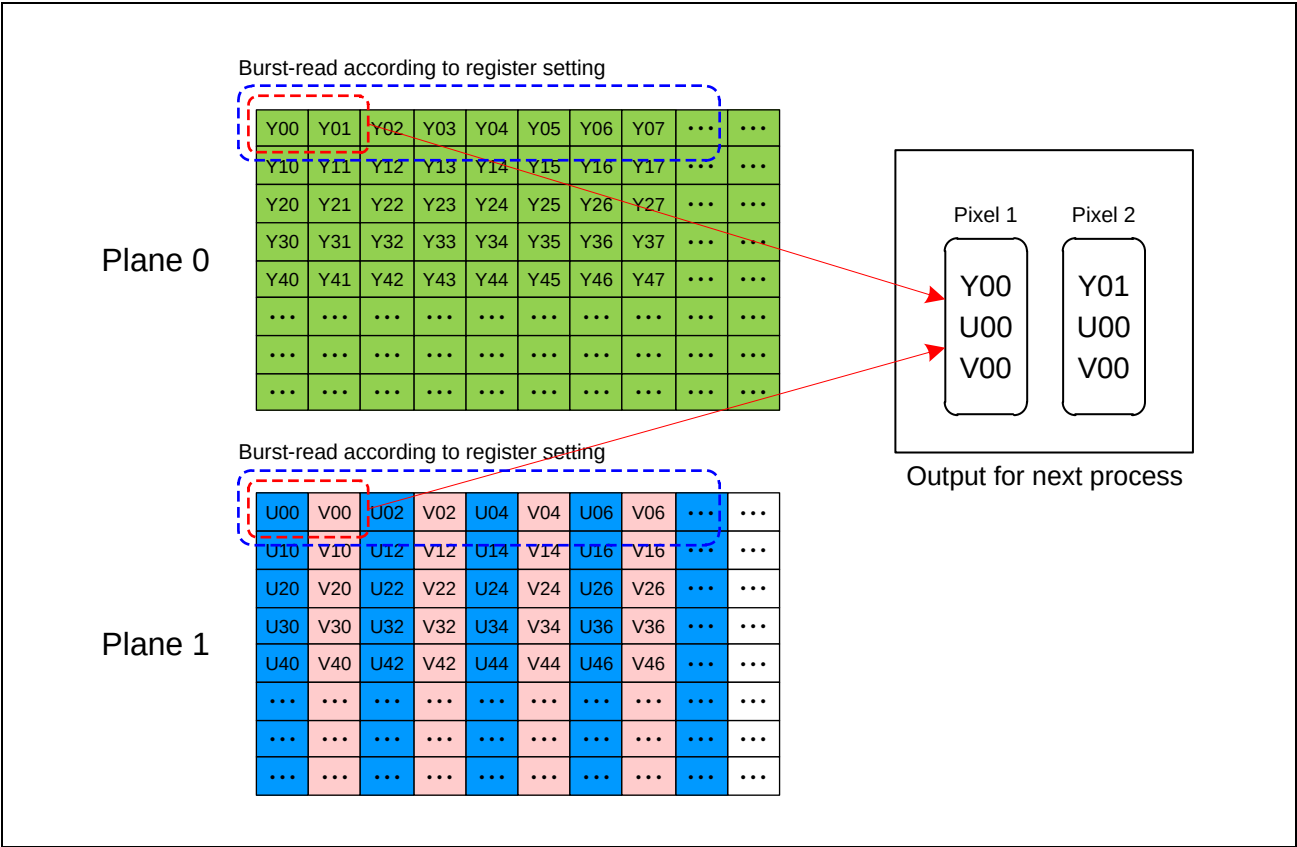


Figure 9.3-5 Example of Semi-planar Format for YCbCr422 (YUV422)

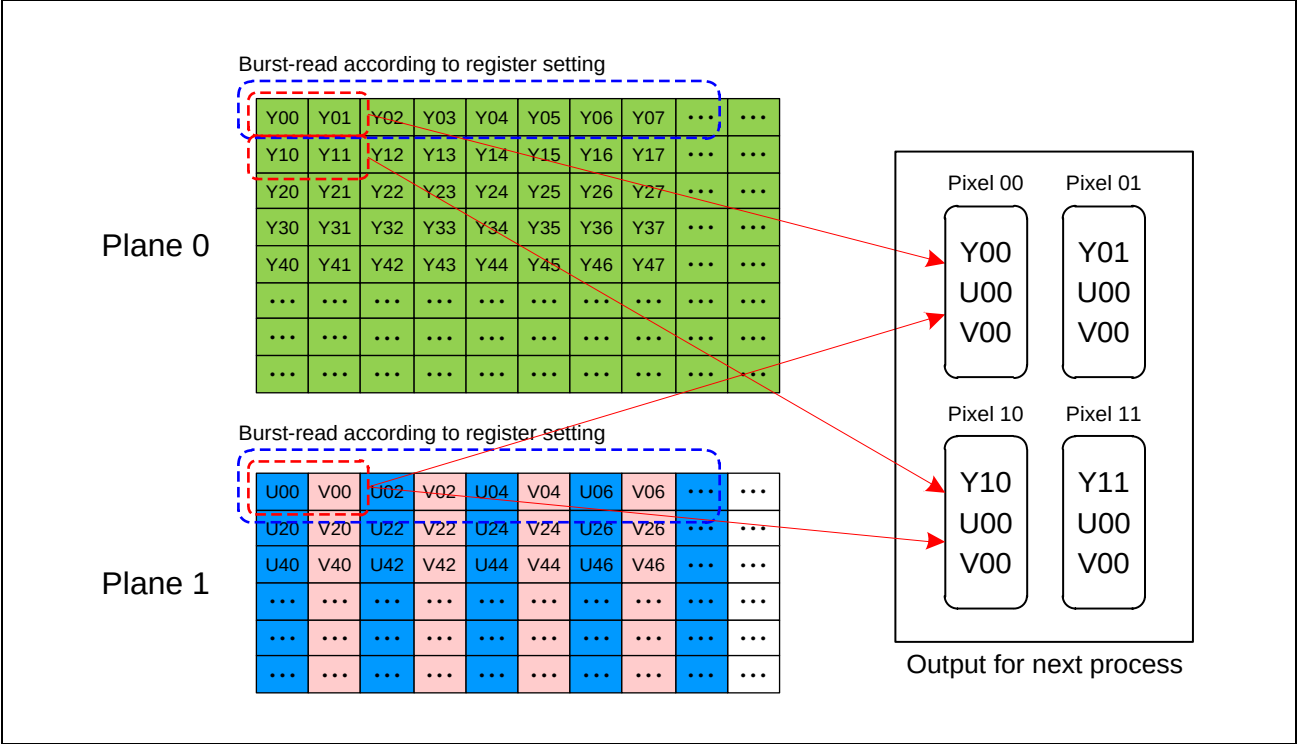


Figure 9.3-6 Example of Semi-planar format for YCbCr420 (YUV420)

(2) Input setting

Data is read from each plane according to the following register setting. Registers are required to set for each plane (Max 2 Plane).

Table 9.3-4 Input Image Setting

No.	Category	Register	Variation
(1)	Start Address	ISU_RPF_SRC_ADDH_PL0.SADD_PL0[2:0]	• Start address for Plane 0 of input image.
		ISU_RPF_SRC_ADDL_PL0.SADD_PL0[31:0]	
		ISU_RPF_SRC_ADDH_PL0.SADD_PL1[2:0]	• Start address for Plane 1 of input image.
		ISU_RPF_SRC_ADDL_PL0.SADD_PL1[31:0]	
(2)	Image Stride	ISU_RPF_SRC_STRD.SSTRPL0[15:0]	• Image stride of Plane 0 [Byte]
		ISU_RPF_SRC_STRD.SSTRPL1[15:0]	• Image stride of Plane 1 [Byte]
(3)	Input Area	ISU_RPF_SRC_SIZE.S_HSIZE[12:0]	• Horizontal size of processing area [Pixel] • Max 4096 pixels
(4)		ISU_RPF_SRC_SIZE.S_VSIZE[12:0]	• Vertical size of processing area [Pixel] • Max 4096 pixels

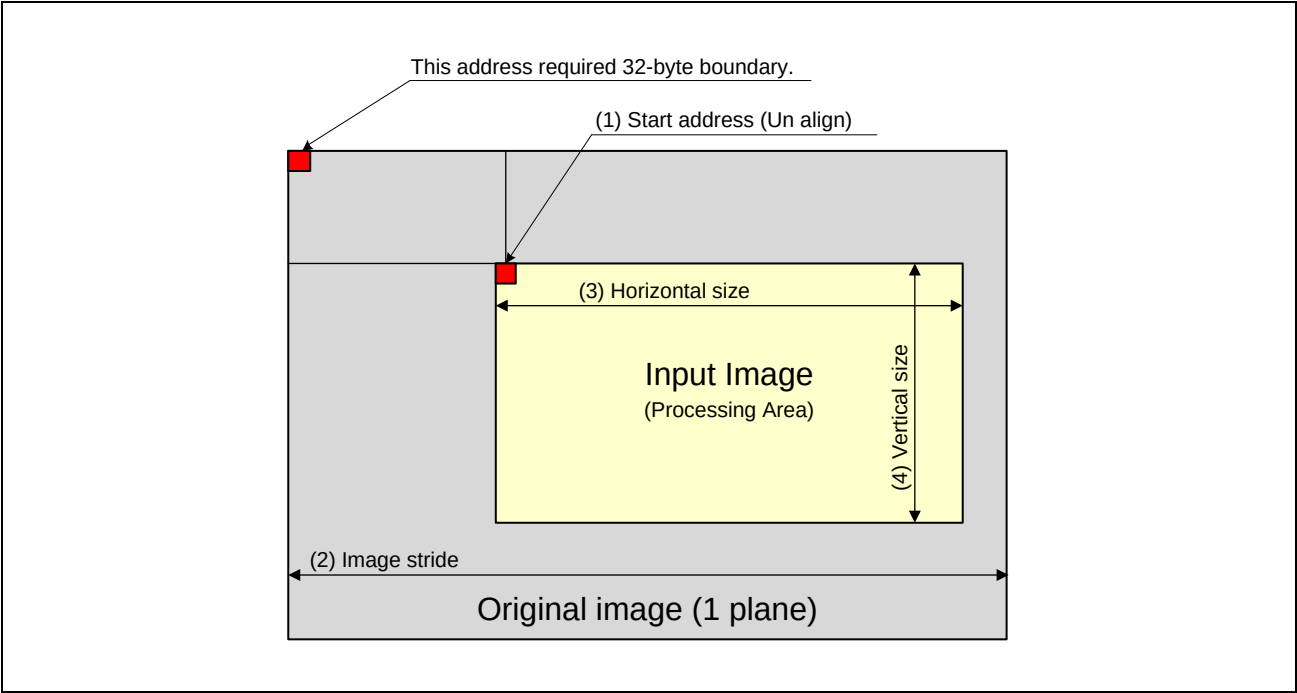


Figure 9.3-7 Input Area (Processing Area) Setting

(3) Write output image (AXITX)

This function write the resized (reduced) image data to DRAM.

(4) Support format

AXITX support the same format as AXIRX.

(5) Output setting

Data is written out for each plane according to the following setting. Registers are required to set for each plane (Max plane number is 2).

Table 9.3-5 Output Image Setting

No.	Category	Register	Variation
(1)	Start Address	ISU_WPF_DST_ADDH_PL0.DADD_PL0[2:0]	• Start address of Plane 0 for output image.
		ISU_WPF_DST_ADDL_PL0.DADD_PL0[31:0]	
		ISU_WPF_DST_ADDH_PL1.DADD_PL1[2:0]	• Start address of Plane 1 for output image.
		ISU_WPF_DST_ADDL_PL1.DADD_PL1[31:0]	
(2)	Image Stride	ISU_WPF_DST_STRD.DSTRPL0[15:0]	• Image stride of Plane 0 [Byte]
		ISU_WPF_DST_STRD.DSTRPL1[15:0]	• Image stride of Plane 1 [Byte]
(3)	Processing Area	ISU_RS_OS_CROP.O_HSIZE[12:0]	• Horizontal crop size [Pixel] • Max 4096 pixels
		ISU_RS_OS_CROP.O_VSIZE[12:0]	• Vertical crop size [Pixel] • Max 4096 pixels
(4)			

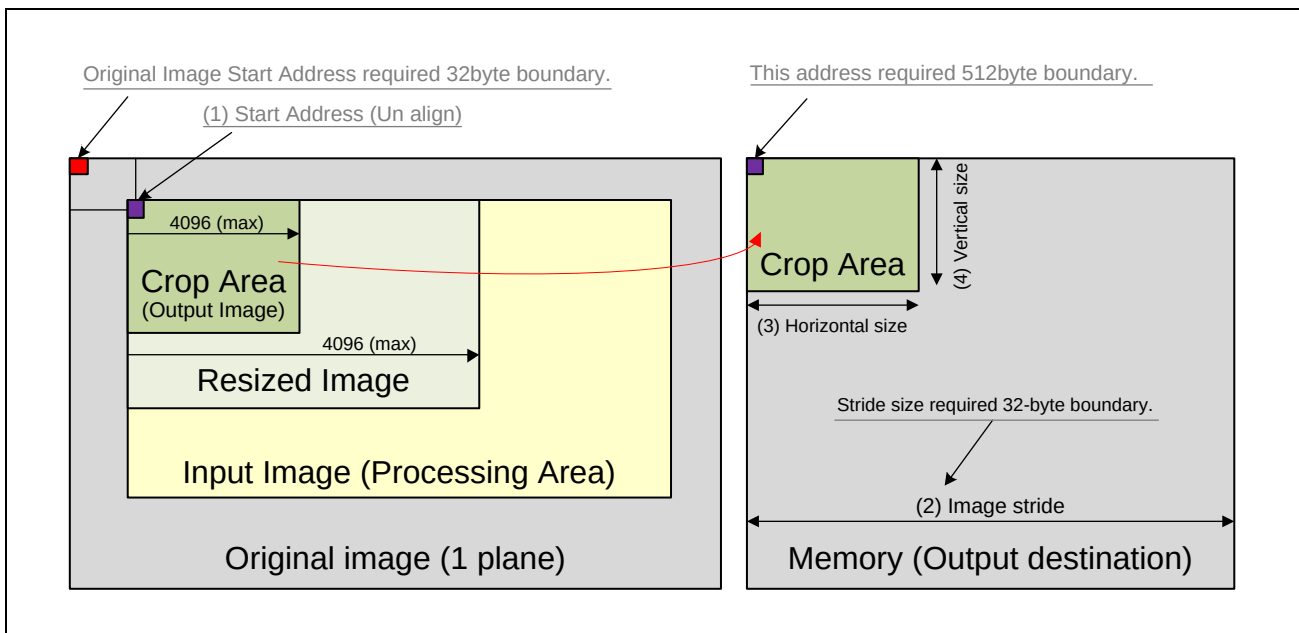


Figure 9.3-8 Output Setting

9.3.1.3 Read Pixel Formatting (RPF)

This function converts the input image data from the AXI-Master and outputs it to the resizer. There are four types of conversion: Data swap, Data extraction, Offset binary conversion (YCbCr only), and pixel normalization. RPF also includes a test pattern generation function.

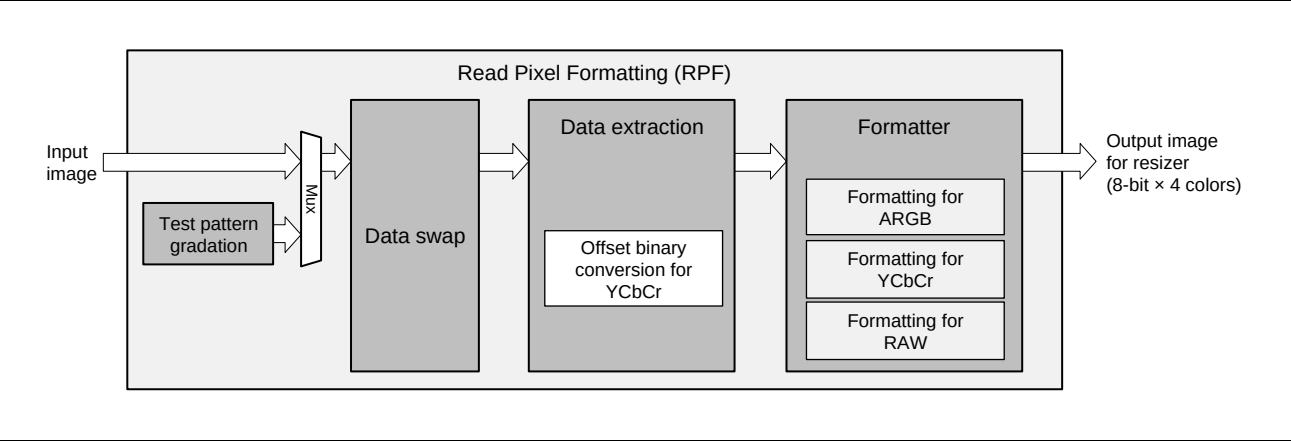


Figure 9.3-9 RPF Block Diagram

9.3.1.3.1 Data swap (endian correction)

This function corrects byte order so that image data can be handled by the ISU. Each bytes can be arranged any order according to the register setting of ISU_RPF_SRC_DSWAP.RD_SWAP[2:0].

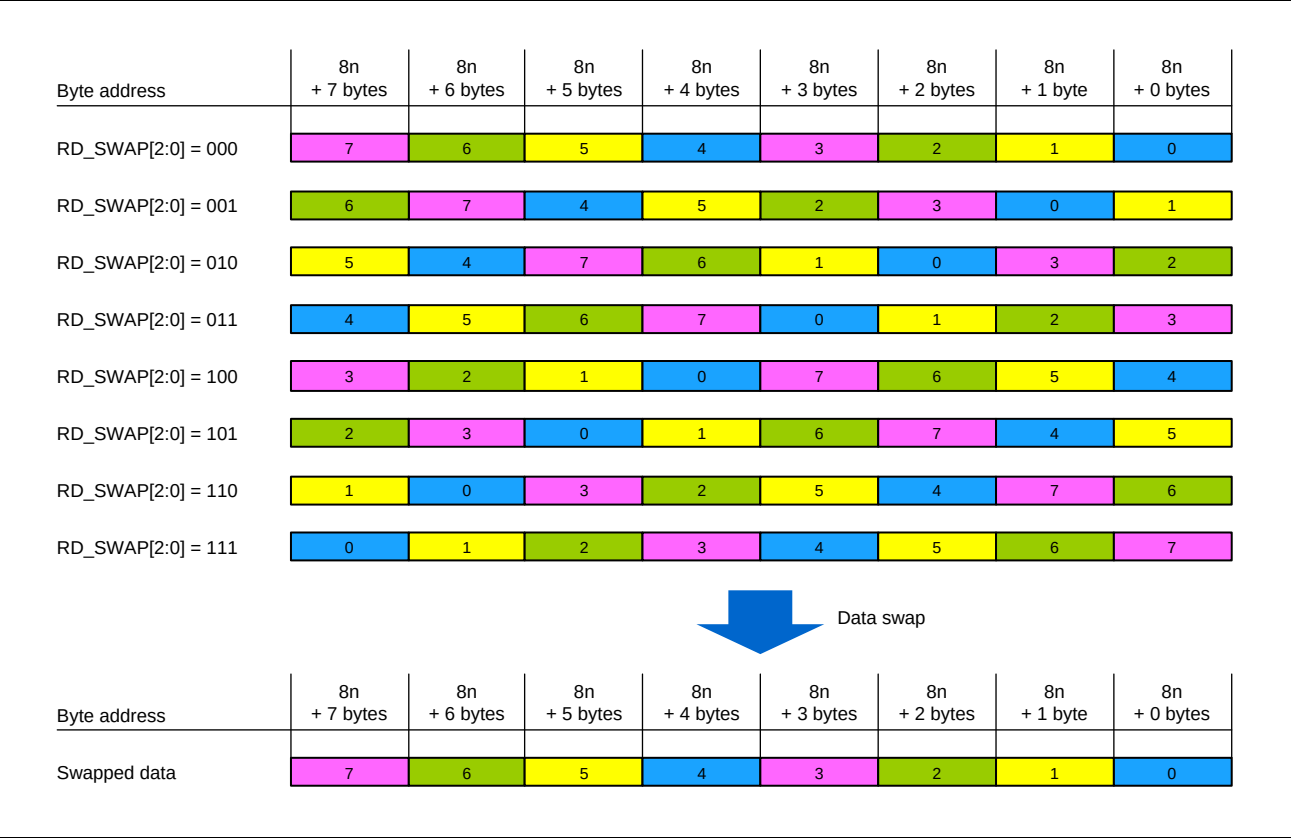


Figure 9.3-10 Data Swap Correction

9.3.1.3.2 RPF Formatter

These functions extract each color components (ex. R, G, B) and normalized as 8-bit for each so that image data can be handled by Resizer. Appropriate processing is performed for each input color format (RGB/ARGB, YCbCr/YUV, RAW) by setting of ISU_RPF_FMT.RDFMT[5:0].

■ RGB/ARGB

This function supports 8 ARGB/RGB formats as shown below. All formats are “Number of planes = 1”.

Table 9.3-6 ARGB Support Format

	Format				n								n + 1								n + 2								n + 3										
RDFMT [5:0]	Color Format	Padding	bit/pixel	phase	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
01h	RGB565	—	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1		
05h	ARGB8888	—	24	—	A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0		
06h	RGBA8888	—		—	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A			
03h	RGB888	—		0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1	R1		
				1	G1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2		
				2	B2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3	B3		
04h	BGR888	—	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1		
				1	G1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2	G2		
				2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3	R3		
00h	ARGB1555	—	15	—	A	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1			
02h	BGR666	BGR upper	18	0			B0	B0	B0	B0	B0	B0				G0	G0	G0	G0	G0				R0	R0	R0	R0	R0				B1	B1	B1	B1	B1	B1		
				1			G1	G1	G1	G1	G1	G1				R1	R1	R1	R1	R1	R1				B2	B2	B2	B2	B2	B2				G2	G2	G2	G2	G2	G2
				2			R2	R2	R2	R2	R2	R2				B3	B3	B3	B3	B3	B3				G3	G3	G3	G3	G3	G3				R3	R3	R3	R3	R3	R3
07h	ABGR8888	—	24	—	A	A	A	A	A	A	A	A	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	R0		

■ YCbCr/YUV

This function supports 4 YCbCr/YUV formats as shown below.

Table 9.3-7 YCbCr/YUV Support Format

Data Format	RDFMT[5:0]	Color Format	Alias	Figure Index
Interleave (Number of Planes = 1)	20h	YCbCr422 8-bit	UYVY	(1)
	21h	YCbCr422 8-bit	YUY2	(2)
Semi-Planar (Number of Planes = 2)	22h	YCbCr422 8-bit	NV16	(3)
	23h	YCbCr420 8-bit	NV12	(4)

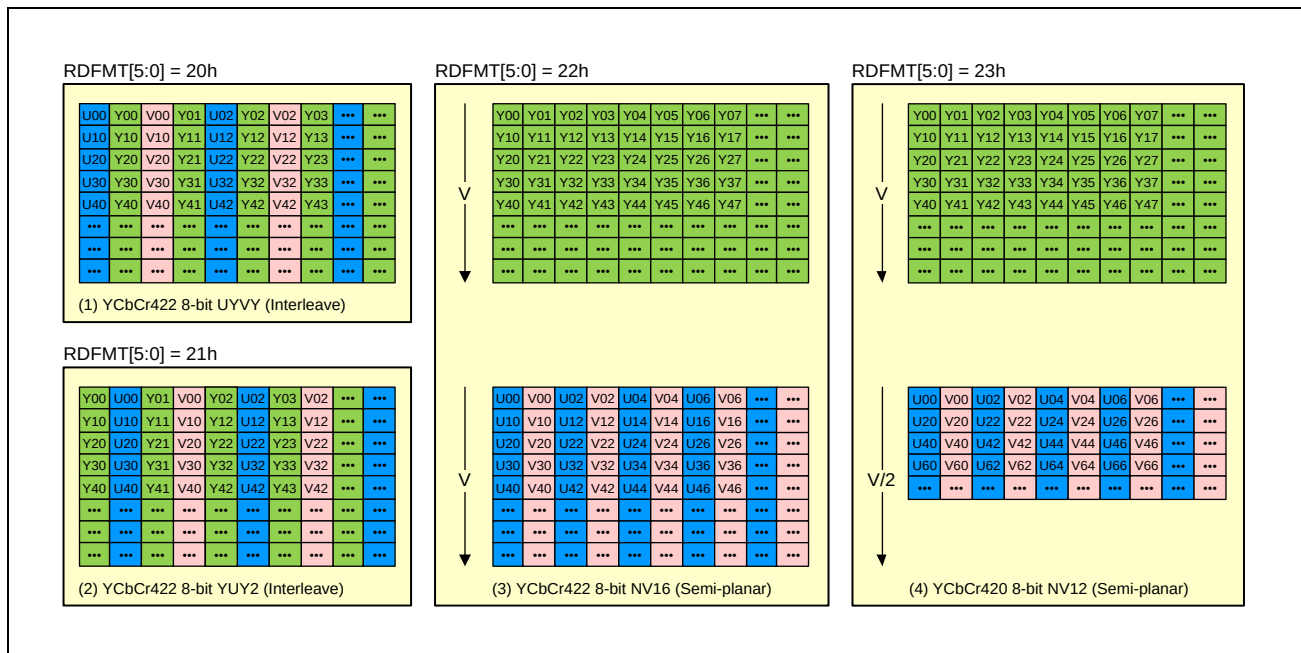


Figure 9.3-11 YCbCr/YUV Support Format

ISU can handle only Offset Binary as the chroma data format (CbCr/UV). Therefore, it is necessary to convert from Two's Complement Binary to Offset Binary.

■ ISU_RPF_UVBIN.UVCHG = 0 (Default)

ISU uses original data on DRAM as chroma value when RDFMT = 20h to 23h.
(CrCb/UV is Offset Binary.)

■ ISU_RPF_UVBIN.UVCHG = 1

ISU convert chroma data (CrCb/UV) from Two's Complement Binary to Offset Binary.

Decimal Number	Offset Binary	Decimal Number	Two's Compliment Binary
127	1111 1111	127	0111 1111
126	1111 1110	126	0111 1110
1	1000 0001	1	0000 0001
0	1000 0000	0	0000 0000
-1	0111 1111	-1	1111 1111
-128	0000 0000	-128	1000 0000

Figure 9.3-12 Types of Binary Expression

■ RAW

This function supports 8 RAW formats as shown below. All formats are “Number of planes = 1”.

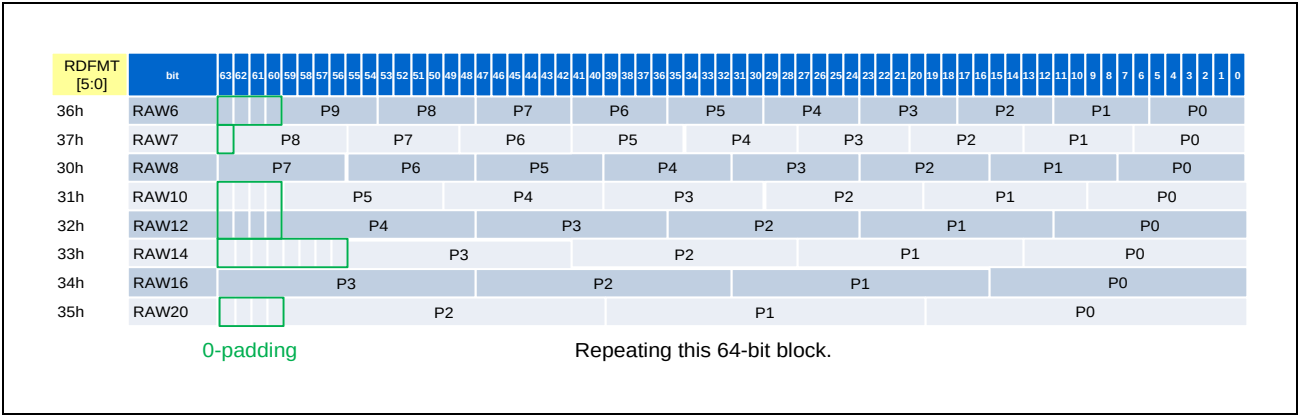


Figure 9.3-13 RAW Support Format

9.3.1.3.3 Data extraction

The 8-bit normalization is required for ARGB/RGB and RAW formats since Resizer can handle the 8-bit format only. On the other hand, although YCbCr/YUV does not need the 8-bit normalization since its components are already 8-bit, it is necessary to exchange into the YCbCr/YUV444 format.

■ RGB/ARGB

For the RGB/ARGB format normalization, higher bits are copied to lower bits.

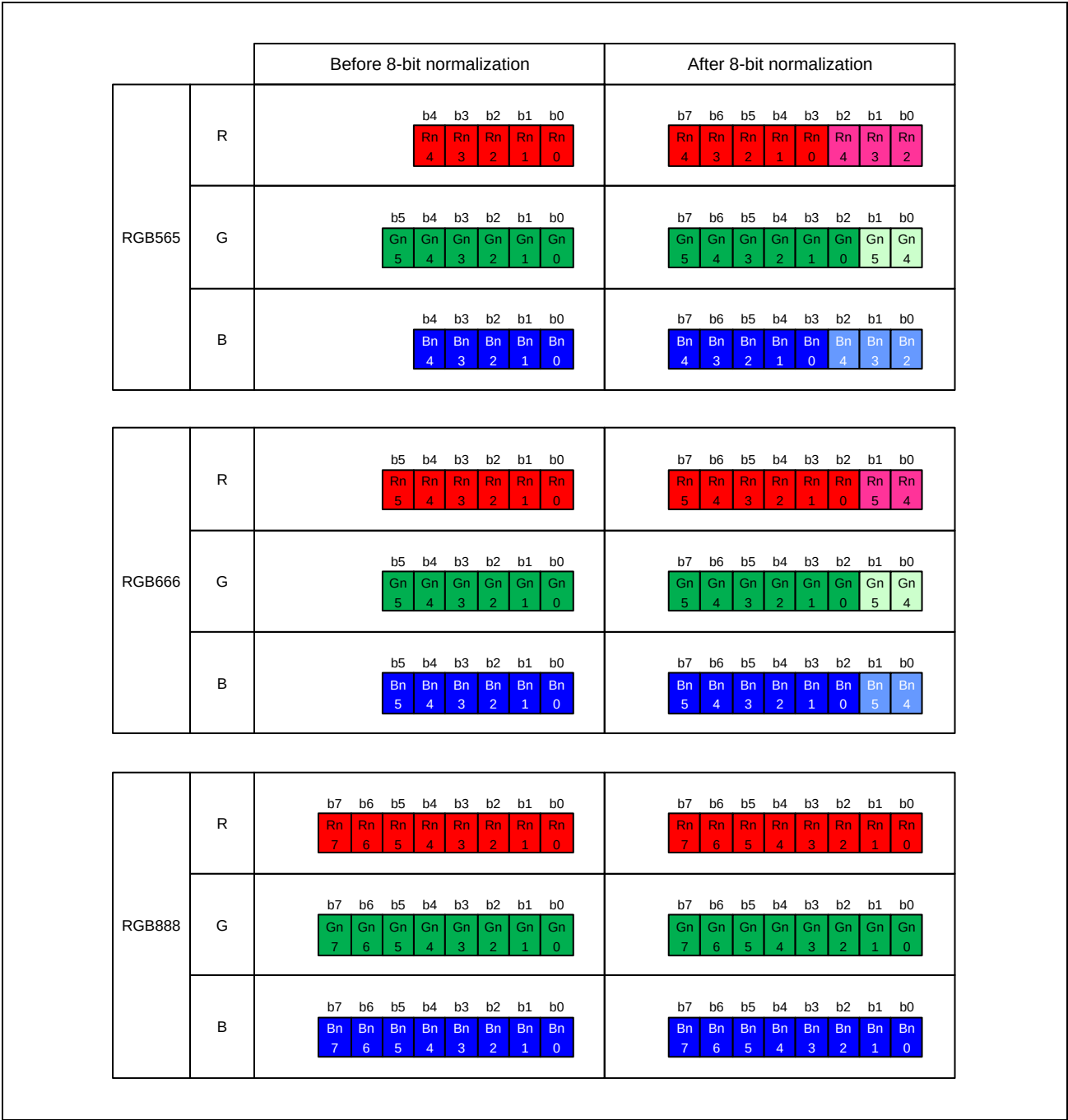


Figure 9.3-14 8-bit Normalization for RGB Format

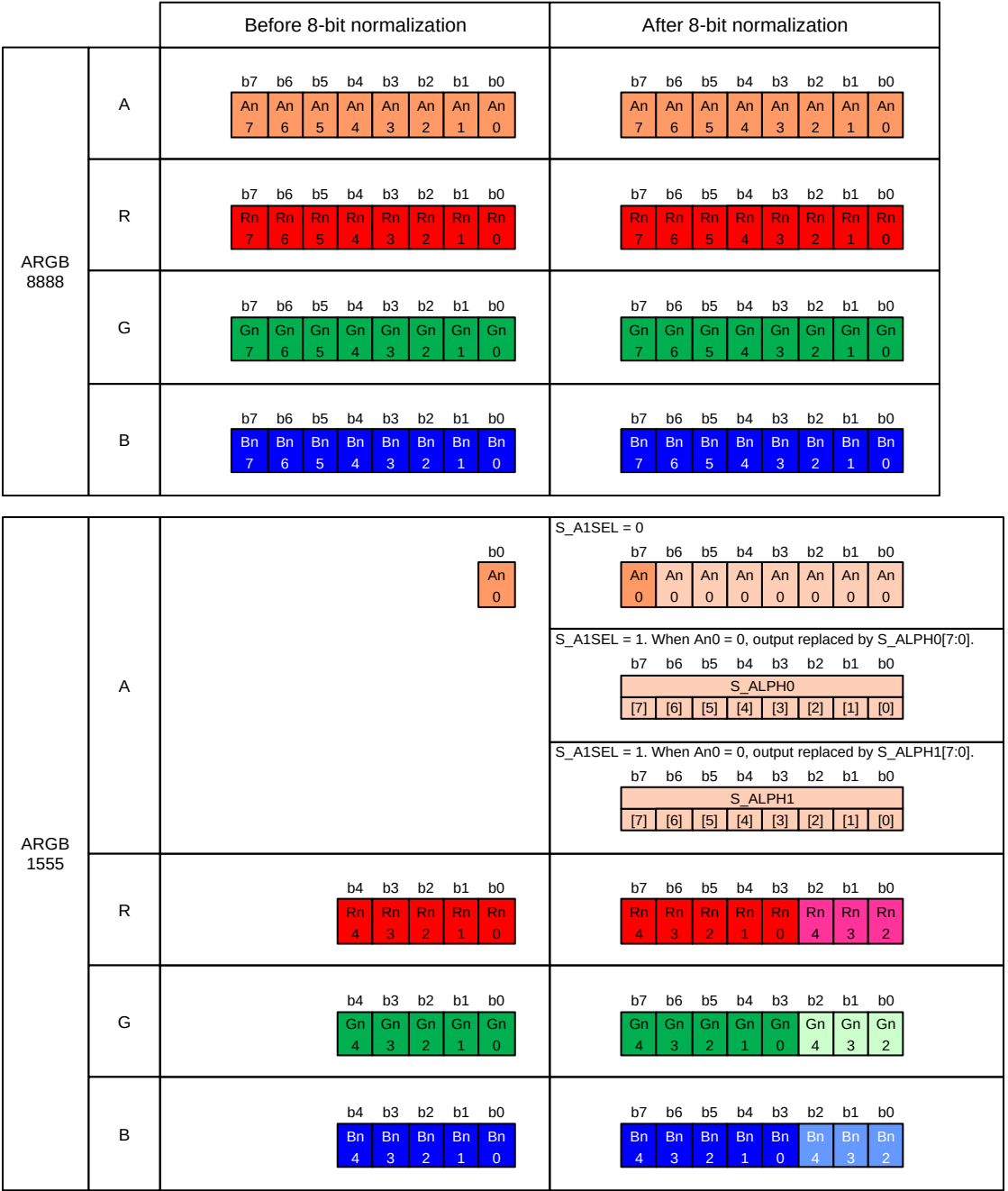


Figure 9.3-15 8-bit Normalization for ARGB Format

■ YCbCr/YUV

For YCbCr/YUV420 Semi-Planar format, even line re-use previous odd line’s chroma information.

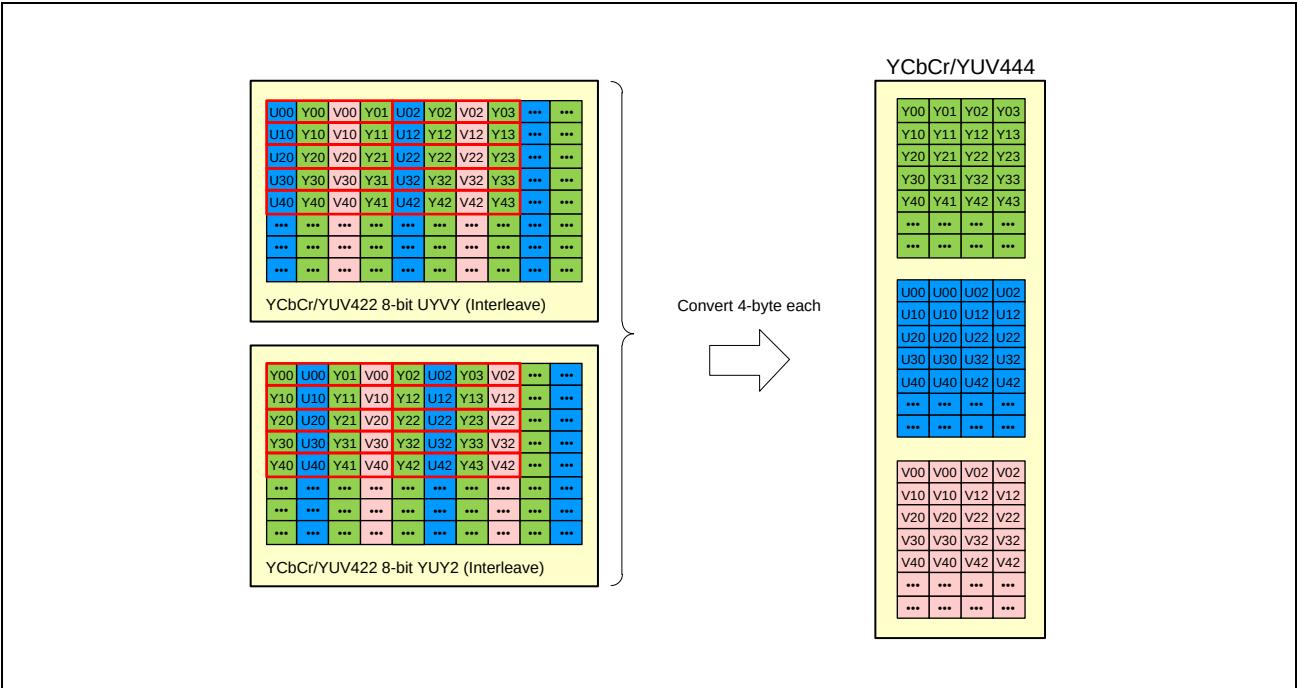


Figure 9.3-16 YCbCr/YUV444 Conversion from YCbCr/YUV422 Format

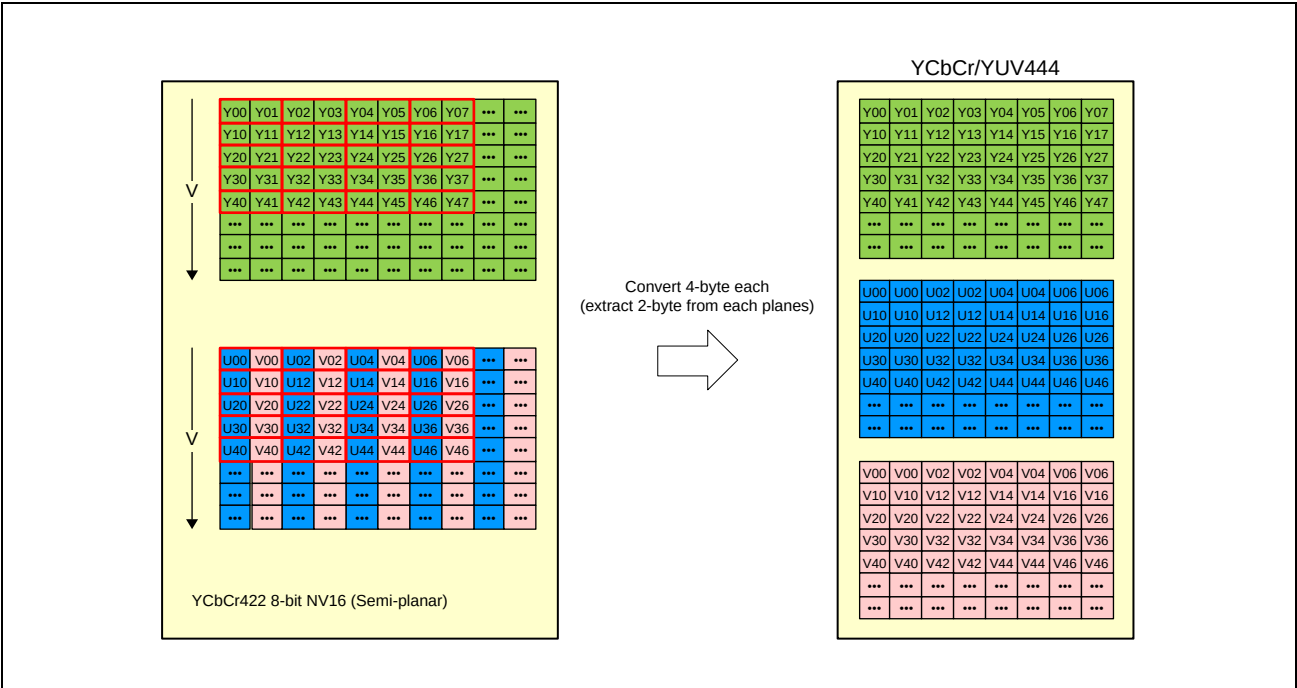


Figure 9.3-17 YCbCr/YUV444 Conversion from YCbCr/YUV422 Semi-Planar Format

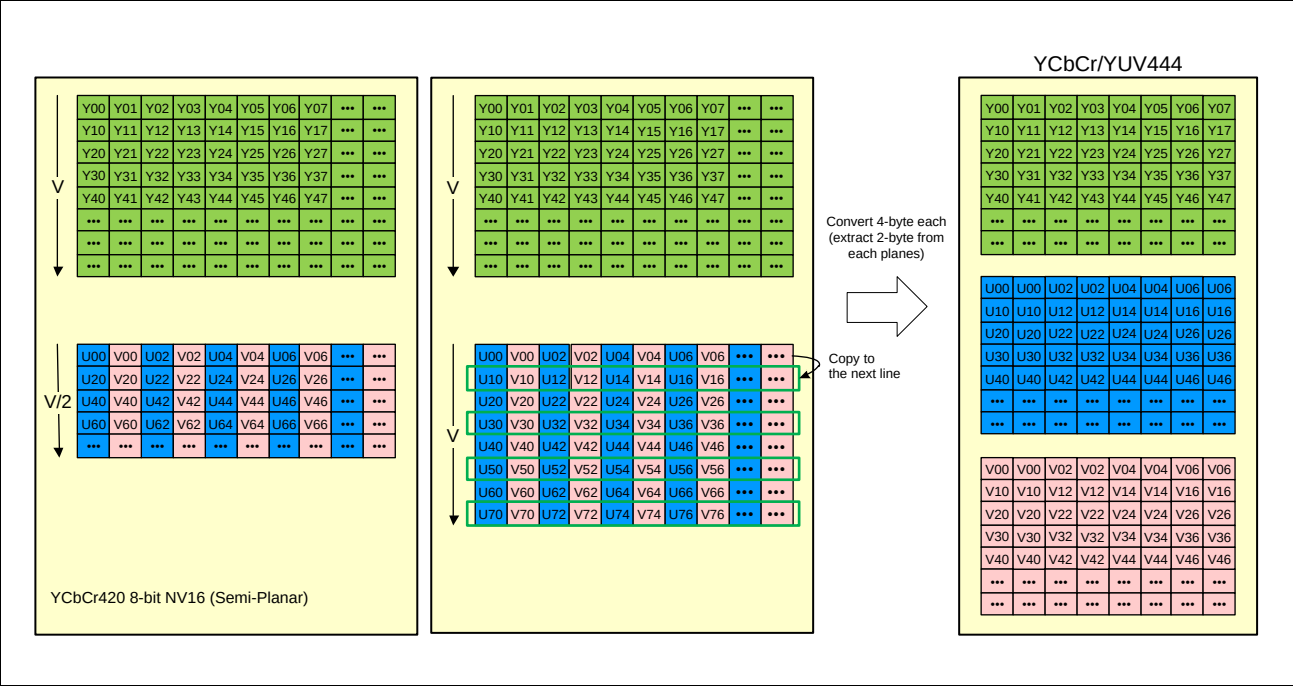


Figure 9.3-18 YCbCr/YUV444 Conversion from YCbCr/YUV420 Semi-Planar Format

■ RAW

RAW is converted to 8 bits by using higher 8 bits of RAW6-/7-/8-/10-/12-/14-/16-/20-bit data.

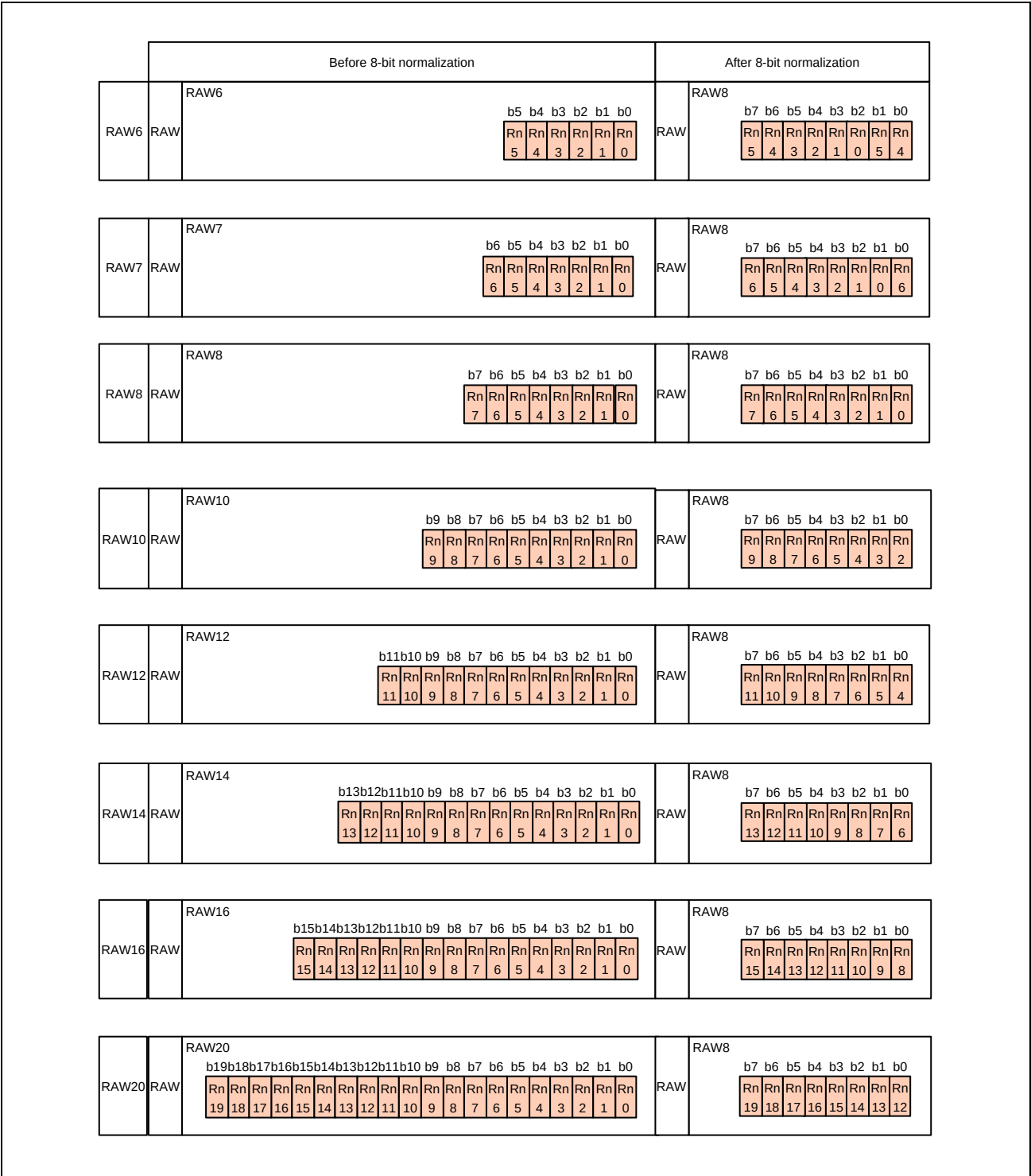


Figure 9.3-19 8-bit Normalization for RAW Format

9.3.1.3.4 Test Pattern Generation

Instead of the input image data from the AXI-Master, RPF can generates a test pattern and outputs it to the Resizer.

The test pattern is an 8-bit gradation image from the upper left to the lower right in the ARGB8888 format according to the register setting.

Please switch between the test pattern and the input image data from the AXI-Master according below procedure.

1. Enable the test pattern. (ISU_RPF_SRC_TD1.TMODE_EN = 1b)
2. Start the resizing process. (ISU_FM_FRCON.START = 1b)

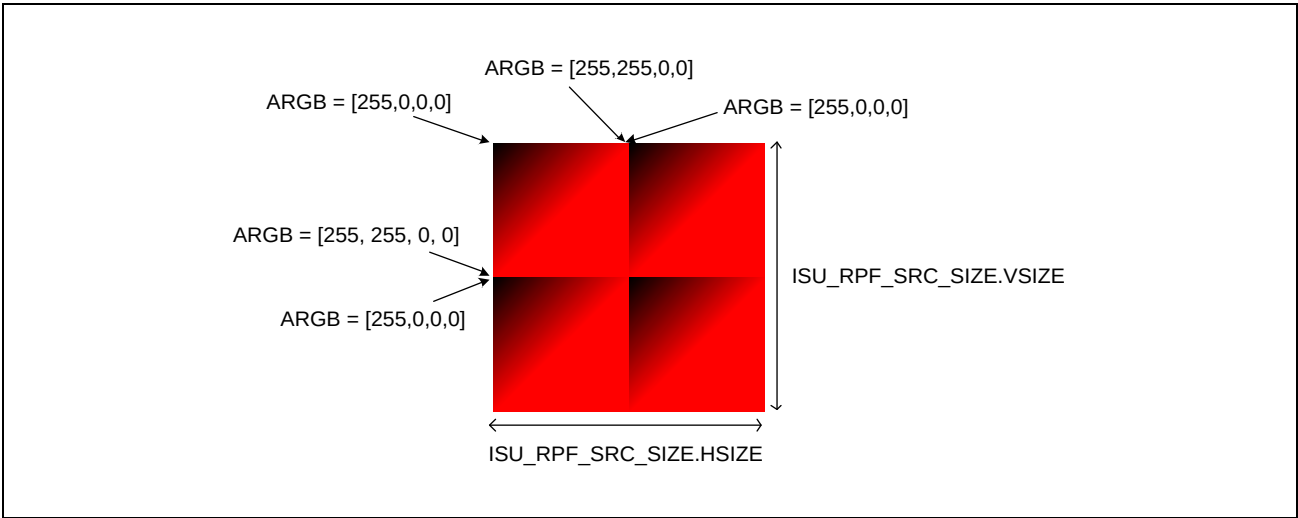


Figure 9.3-20 Test Pattern of R Gradation

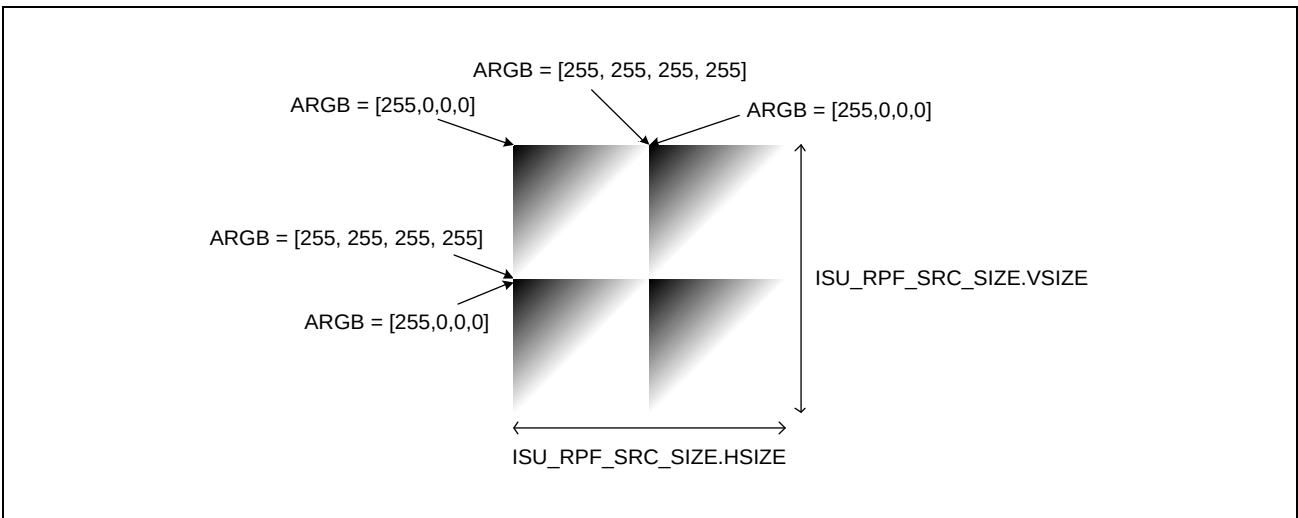


Figure 9.3-21 Pattern of R,G,B Gradation

9.3.1.4 Resizer (RS)

Resizer (RS) reduces the image size by using bi-linear algorithm (4-point linear interpolation) according to register setting of horizontal / vertical resize ratio.

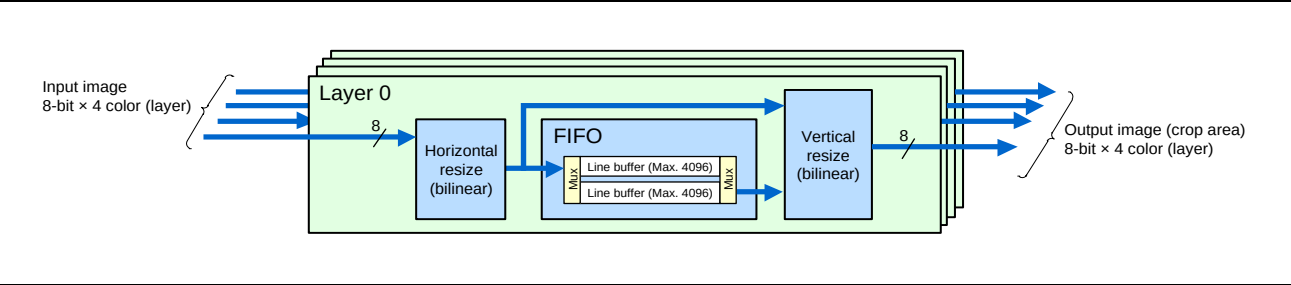


Figure 9.3-22 Resizer Block Diagram

The resize processing is different whether the Resized image size is smaller or larger than the Crop area.

If the Resized image size is smaller than the Crop area, fill in the missing edge pixels according to the register settings below.

When ISU_RS_PADDMODE.PADDSEL = 0b, fill in the padding area by copying the right / bottom edge pixels after the reduction.

When ISU_RS_PADDMODE.PADDSEL = 1b, fill in the padding area with the value specified by the register.

If the Resized image size is larger than the Crop area, it will be cut off by the Crop area.

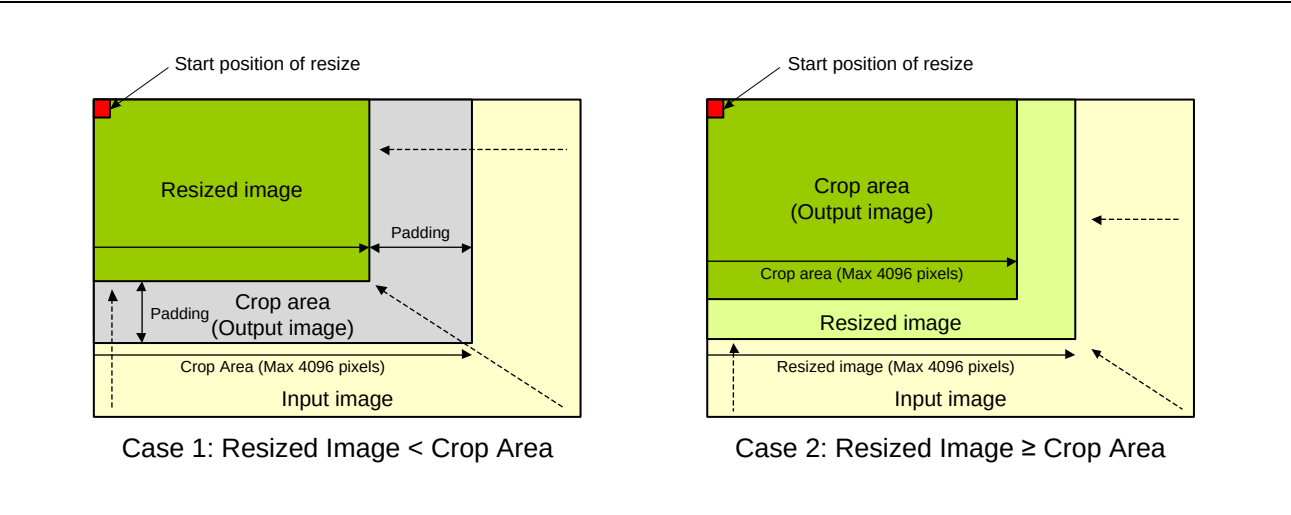


Figure 9.3-23 Relationship of Resized Image and Crop Area

9.3.1.4.1 Resize start position

The resize processing start from Resize Start Position (ISU_RS_STPOS) on the input image in according to the register setting. This position setting is specified in pixel coordinates from the top left of the image. If 0 is specified, the input image will be the Resizer (RS) target image as it is.

Padding area will be filled based on the setting of ISU_RS_PADDMODE.PADDSEL.

- Register ISU_RS_STPOS.HSTART = Resize Start H position
- Register ISU_RS_STPOS.VSTART = Resize Start V position

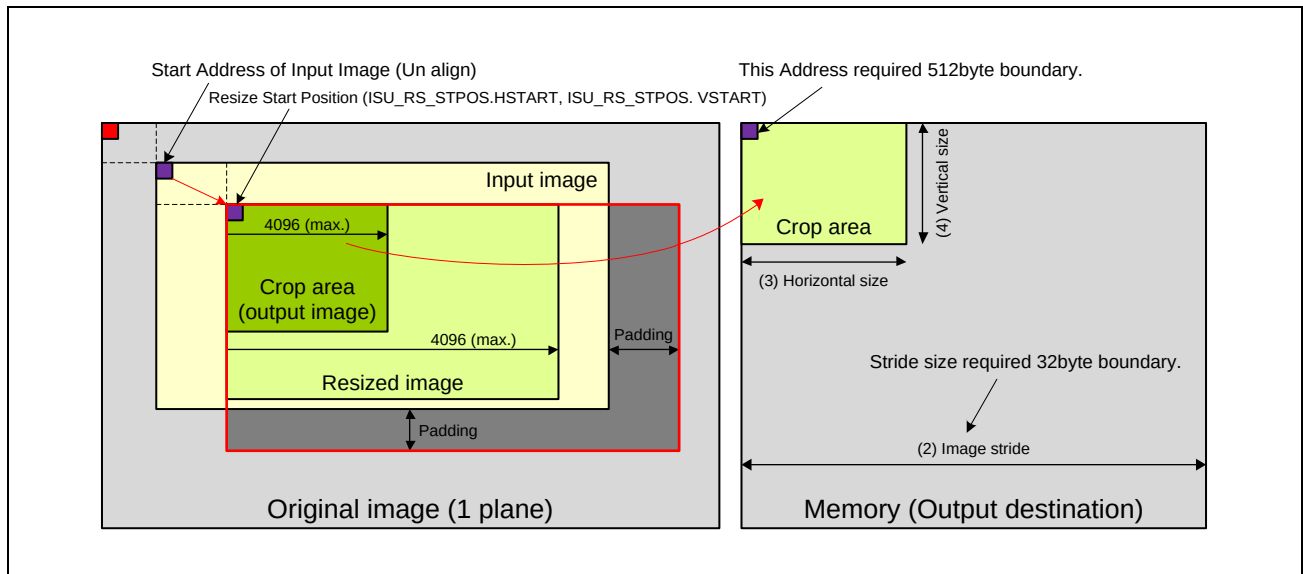


Figure 9.3-24 Resize Start Position

9.3.1.4.2 Sub pixel offset

The Resize Start Position Tuning (ISU_RS_POS_TUNE) can be used to adjust the start position by sub pixel level to reduce moiré.

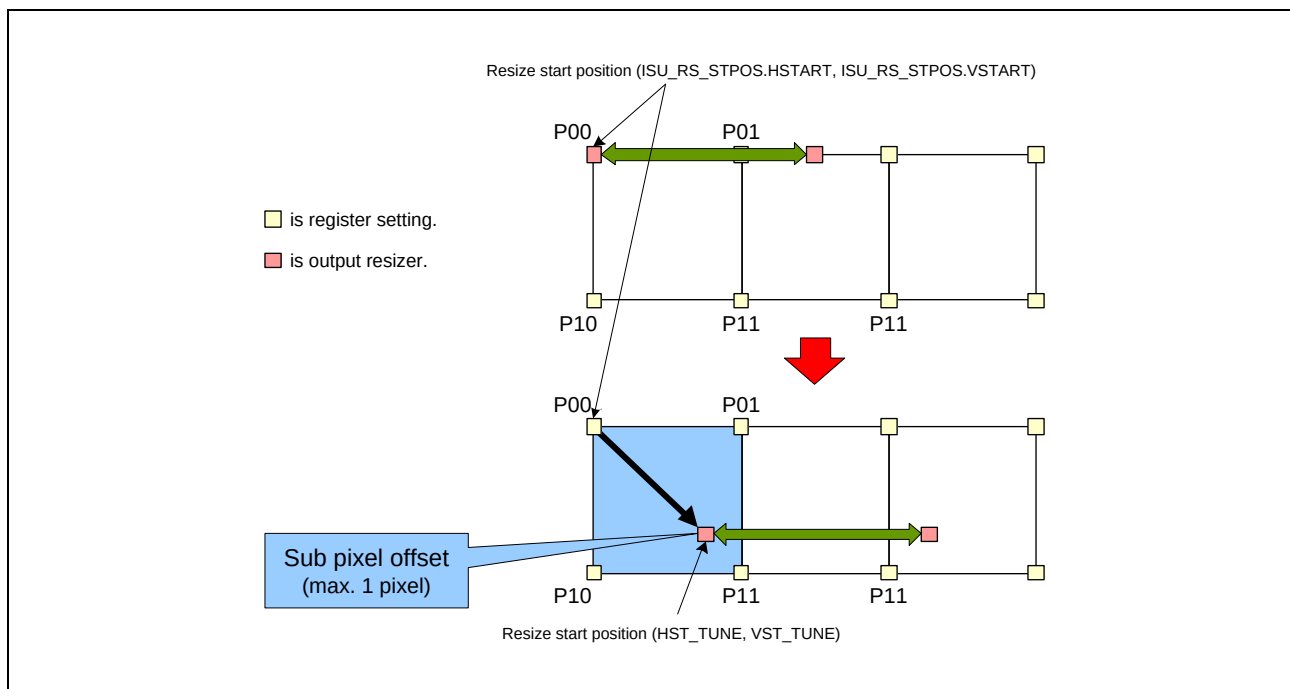


Figure 9.3-25 Sub Pixel Position Adjustment

9.3.1.4.3 **Scaling calculation**

The coefficients are defined in **Table 9.3-8**.

Table 9.3-8 Scaling Coefficient Setting

No.	Category	Register	Variation
(1)	Horizontal Scaling Ratio	RS_HSCALE.HMANT[3:0]	• m _h : Integer part of Horizontal scaling coefficient
		RS_HSCALE.HFRAC[11:0]	• f _h : Fraction part of Horizontal scaling coefficient
(2)	Vertical Scaling Ratio	RS_VSCALE.VMANT[3:0]	• m _v : Integer part of Vertical scaling coefficient
		RS_VSCALE.VFRAC[11:0]	• f _v : Fraction part of Vertical scaling coefficient

Follow the procedure below to set the coefficients.

Step 1: Estimation of scaling factor

The resizer can be handled resizing magnification independently for horizontally and vertically. The image size of this processing result can be calculated approximately by using “Input image horizontal size × hscale” and “Input image vertical size × vscale”.

$$\text{hscale} = \frac{4096}{4096 \times m_h + f_h} \qquad \text{vscale} = \frac{4096}{4096 \times m_v + f_v}$$

Figure 9.3-26 Sub Pixel Position Adjustment

Step 2: Conversion of set values

Convert the set values according to **Table 9.3-9**.

Table 9.3-9 m_h' or m_v' Settings

ISU_RS_HSCALE.HMANTsetting (ISU_RS_VSCALE.VMANTsetting)	m _h ' (m _v ')
1 to 3	1
4 to 7	2
8 to 15	4

Step 3: Fine-tuning of size calculations

• Definitions

hsize_{org}: Input horizontal image size

vsize_{org}: Input vertical image size

hsize_{down_scaled}: Horizontal image size after scale down

vsize_{down_scaled}: Vertical image size after scale down

< >: Rounding down to the decimal point

• Formula

$$\text{hsize}_{\text{down\_scaled}} = \left\langle 1 + \left[\left\langle 1 + \frac{\text{hsize}_{\text{org}} - 1}{m_h} \right\rangle - 1 \right] \times m_h \times 4096 \right\rangle / (4096 \times m_h + f_h)$$

$$\text{vsize}_{\text{down\_scaled}} = \left\langle 1 + \left[\left\langle 1 + \frac{\text{vsize}_{\text{org}} - 1}{m_v} \right\rangle - 1 \right] \times m_v \times 4096 \right\rangle / (4096 \times m_v + f_v)$$

Step 4: Adjustment

If the down scaled image size does not match the intended size, the values of f_h and f_v are fine-tuned and recalculated to find f_h and f_v that result in the desired output down scaled size.

The values thus obtained are set to the corresponding register values RS_HSCALE and RS_VSCALE.

9.3.1.4.4 Resize processing

The following figure shows the processing image of bi-linear algorithm (4-point interpolation).

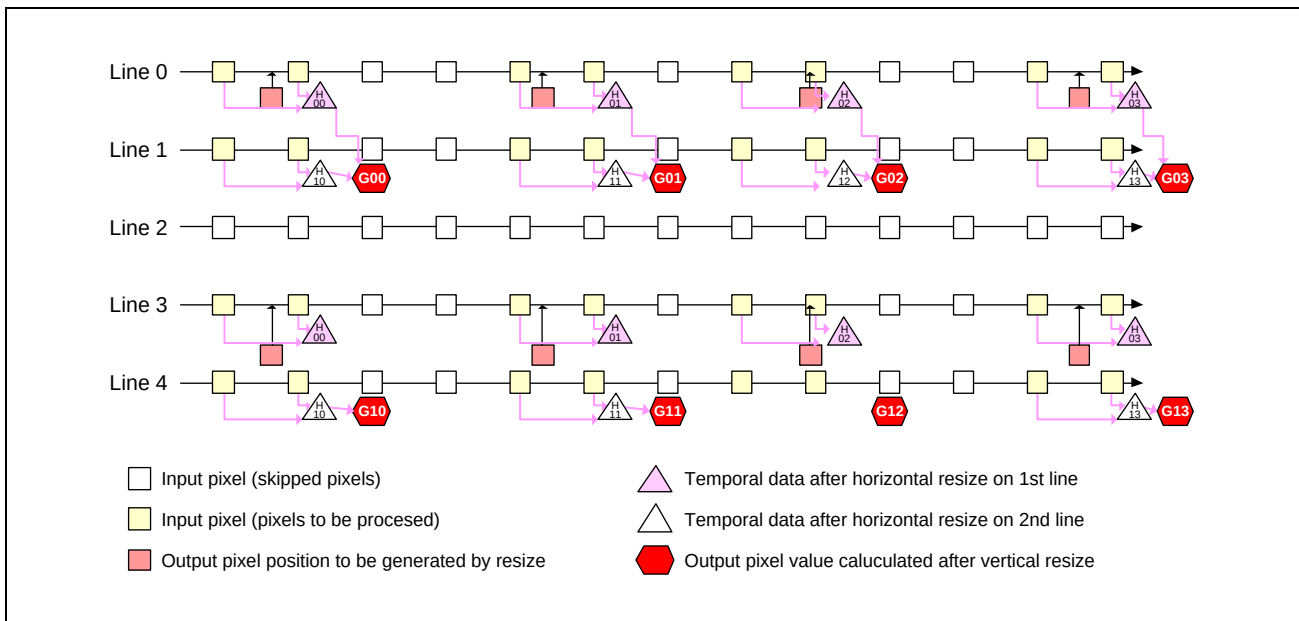


Figure 9.3-27 Points Bi-Linear Resize

The outline of the calculation procedure of 4-point linear interpolation is shown below.

Although dH and dV are not exist actually, it is used to help understanding for this algorithm. The pixel value of the target pixel position $G00$ will be calculated from the nearest 4 input pixels, and dH and dV indicate sub pixel position of $G00$ calculated from Horizontal/Vertical scaling ratio.

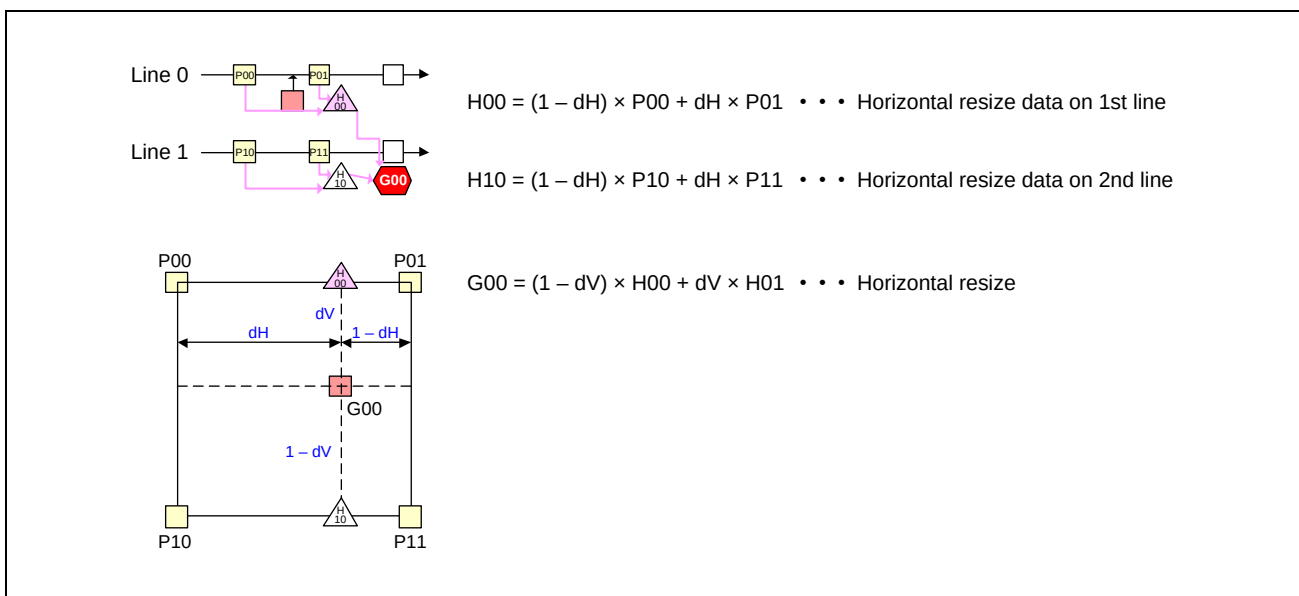


Figure 9.3-28 Calculation for 4 Points Bi-linear Resize

9.3.1.5 Write Pixel Formatting (WPF)

The WPF converts the resized image data from RS and outputs it to the AXI-Master.

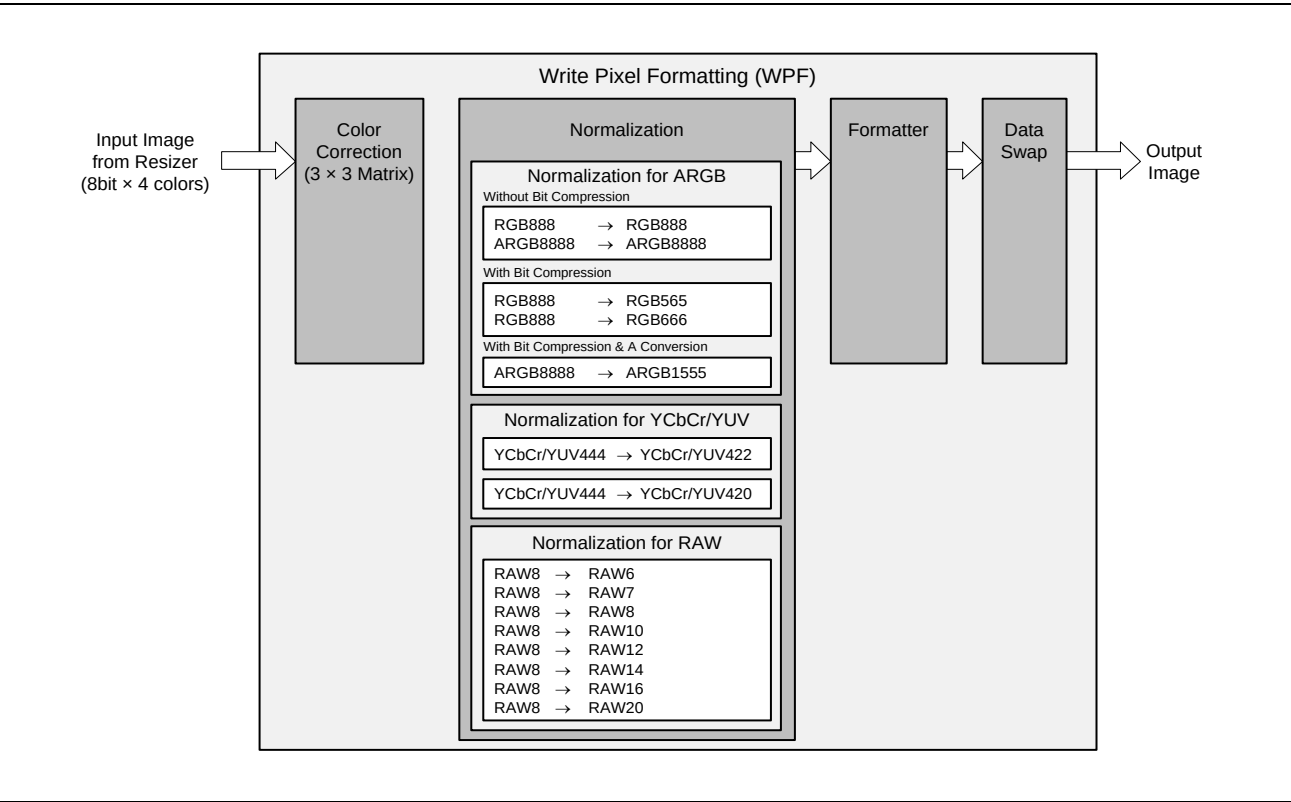


Figure 9.3-29 WPF Block Diagram

9.3.1.5.1 Color correction

This function is for the color space conversion among RGB, YCbCr, and YUV by using 3×3 color matrix. Matrix coefficients can be set by registers. The matrix calculation formula and circuit block diagram are shown below.

3×3 matrix $\begin{bmatrix} C1' \\ C2' \\ C3' \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \cdot \begin{bmatrix} A - \text{OFST\_A1} \\ B - \text{OFST\_B1} \\ C - \text{OFST\_C1} \end{bmatrix} + \begin{bmatrix} \text{OFST\_A2} \\ \text{OFST\_B2} \\ \text{OFST\_C2} \end{bmatrix}$	Level clipping $C1'' = \text{MAX}(\text{CLPMIN\_A}, \text{MIN}(\text{CLPMAX\_A}, C1'))$ $C2'' = \text{MAX}(\text{CLPMIN\_B}, \text{MIN}(\text{CLPMAX\_B}, C2'))$ $C3'' = \text{MAX}(\text{CLPMIN\_C}, \text{MIN}(\text{CLPMAX\_C}, C3'))$

Figure 9.3-30 Color Correction Equation

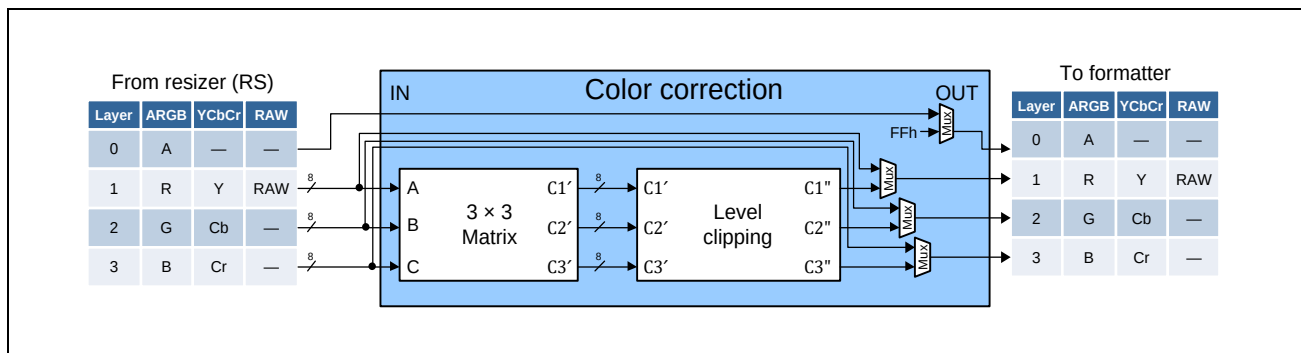


Figure 9.3-31 Color Correction Block Diagram

Sample settings of color matrix are shown below.

Conversion between RGB and YUV (PAL, SECAM)

$$\begin{bmatrix} Y_{OUT} \\ U_{OUT} \\ V_{OUT} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.580 \\ 1.000 & 2.032 & -0.000 \end{bmatrix} \begin{bmatrix} Y_{in} \\ (U_{in} - 128) \\ (V_{in} - 128) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Conversion between RGB and YCbCr ITU-R BT.601 / BT.709 (1250/50/2:1)

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.772 & -0.000 \end{bmatrix} \begin{bmatrix} Y_{in} \\ (Cb_{in} - 128) \\ (Cr_{in} - 128) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Conversion between RGB and YCbCr ITU-R BT.709 (1125/60/2:1)

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \begin{bmatrix} 0.213 & 0.715 & 0.072 \\ -0.115 & -0.385 & 0.500 \\ 0.500 & -0.454 & -0.046 \end{bmatrix} \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.575 \\ 1.000 & -0.187 & -0.468 \\ 1.000 & 1.856 & -0.000 \end{bmatrix} \begin{bmatrix} Y_{in} \\ (Cb_{in} - 128) \\ (Cr_{in} - 128) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Conversion from RAW to RGB or YCbCr/YUV (Grayscale)

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 \end{bmatrix} \begin{bmatrix} RAW_{in} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \begin{bmatrix} RAW_{in} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

9.3.1.5.2 Normalization

This function converts each color data into appropriate bit depth.

		Before normalization	After normalization																																									
RGB565	R	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3	2	1	0	<table><tr><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td></tr></table>	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3		
	b7	b6	b5	b4	b3	b2	b1	b0																																				
	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn																																				
7	6	5	4	3	2	1	0																																					
b4	b3	b2	b1	b0																																								
Rn	Rn	Rn	Rn	Rn																																								
7	6	5	4	3																																								
G	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2	1	0	<table><tr><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr></table>	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2
b7	b6	b5	b4	b3	b2	b1	b0																																					
Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn																																					
7	6	5	4	3	2	1	0																																					
b5	b4	b3	b2	b1	b0																																							
Gn	Gn	Gn	Gn	Gn	Gn																																							
7	6	5	4	3	2																																							
B	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3	2	1	0	<table><tr><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td></tr></table>	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3			
b7	b6	b5	b4	b3	b2	b1	b0																																					
Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn																																					
7	6	5	4	3	2	1	0																																					
b4	b3	b2	b1	b0																																								
Bn	Bn	Bn	Bn	Bn																																								
7	6	5	4	3																																								

RGB666	R	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3	2	1	0	<table><tr><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr></table>	b5	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3	2
	b7	b6	b5	b4	b3	b2	b1	b0																																					
	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn																																					
7	6	5	4	3	2	1	0																																						
b5	b4	b3	b2	b1	b0																																								
Rn	Rn	Rn	Rn	Rn	Rn																																								
7	6	5	4	3	2																																								
G	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2	1	0	<table><tr><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr></table>	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2	
b7	b6	b5	b4	b3	b2	b1	b0																																						
Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn																																						
7	6	5	4	3	2	1	0																																						
b5	b4	b3	b2	b1	b0																																								
Gn	Gn	Gn	Gn	Gn	Gn																																								
7	6	5	4	3	2																																								
B	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3	2	1	0	<table><tr><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td></tr></table>	b5	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3	2	
b7	b6	b5	b4	b3	b2	b1	b0																																						
Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn																																						
7	6	5	4	3	2	1	0																																						
b5	b4	b3	b2	b1	b0																																								
Bn	Bn	Bn	Bn	Bn	Bn																																								
7	6	5	4	3	2																																								

RGB888	R	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3	2	1	0	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td><td>Rn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn	7	6	5	4	3	2	1	0
	b7	b6	b5	b4	b3	b2	b1	b0																																											
	Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn																																											
7	6	5	4	3	2	1	0																																												
b7	b6	b5	b4	b3	b2	b1	b0																																												
Rn	Rn	Rn	Rn	Rn	Rn	Rn	Rn																																												
7	6	5	4	3	2	1	0																																												
G	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2	1	0	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td><td>Gn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn	7	6	5	4	3	2	1	0	
b7	b6	b5	b4	b3	b2	b1	b0																																												
Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn																																												
7	6	5	4	3	2	1	0																																												
b7	b6	b5	b4	b3	b2	b1	b0																																												
Gn	Gn	Gn	Gn	Gn	Gn	Gn	Gn																																												
7	6	5	4	3	2	1	0																																												
B	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3	2	1	0	<table><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td><td>Bn</td></tr><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	b7	b6	b5	b4	b3	b2	b1	b0	Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn	7	6	5	4	3	2	1	0	
b7	b6	b5	b4	b3	b2	b1	b0																																												
Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn																																												
7	6	5	4	3	2	1	0																																												
b7	b6	b5	b4	b3	b2	b1	b0																																												
Bn	Bn	Bn	Bn	Bn	Bn	Bn	Bn																																												
7	6	5	4	3	2	1	0																																												

Figure 9.3-32 Normalization for RGB Format

■ ARGB

		Before normalization	After normalization
ARGB 8888	A	b7 b6 b5 b4 b3 b2 b1 b0 An An An An An An An An 7 6 5 4 3 2 1 0	D_A8SEL = 0 b7 b6 b5 b4 b3 b2 b1 b0 An An An An An An An An 7 6 5 4 3 2 1 0 D_A8SEL = 1. $An[7:0] \leq D\_A8THR0$ D_ALPH0[7:0] b7 b6 b5 b4 b3 b2 b1 b0 D_ALPH0 [7] [6] [5] [4] [3] [2] [1] [0] D_A8SEL = 1. $D\_A8THR0 < An[7:0] < D\_A8THR1$ D_ALPH1[7:0] b7 b6 b5 b4 b3 b2 b1 b0 D_ALPH1 [7] [6] [5] [4] [3] [2] [1] [0] D_A8SEL = 1. $D\_A8THR1 \leq An[7:0]$ D_ALPH2[7:0] b7 b6 b5 b4 b3 b2 b1 b0 D_ALPH2 [7] [6] [5] [4] [3] [2] [1] [0]
	R	b7 b6 b5 b4 b3 b2 b1 b0 Rn Rn Rn Rn Rn Rn Rn Rn 7 6 5 4 3 2 1 0	b7 b6 b5 b4 b3 b2 b1 b0 Rn Rn Rn Rn Rn Rn Rn Rn 7 6 5 4 3 2 1 0
	G	b7 b6 b5 b4 b3 b2 b1 b0 Gn Gn Gn Gn Gn Gn Gn Gn 7 6 5 4 3 2 1 0	b7 b6 b5 b4 b3 b2 b1 b0 Gn Gn Gn Gn Gn Gn Gn Gn 7 6 5 4 3 2 1 0
	B	b7 b6 b5 b4 b3 b2 b1 b0 Bn Bn Bn Bn Bn Bn Bn Bn 7 6 5 4 3 2 1 0	b7 b6 b5 b4 b3 b2 b1 b0 Bn Bn Bn Bn Bn Bn Bn Bn 7 6 5 4 3 2 1 0

Figure 9.3-33 Normalization for ARGB8888 Format

“A(An)” of ARGB8888 is normalized according to the following logic.

If ISU_WPF_ALPH_SEL2.D_A8SEL = 0b, An is output as it is.

If ISU_WPF_ALPH_SEL2.D_A8SEL = 1b, replace it with the following value according to the relationship between ISU_WPF_ALPH_SEL2.D_A8THR0 to 1 and An.

If $An \leq D\_A8THR0$, replace it with ISU_WPF_ALPH_VAL.D_ALPH0.

If $D\_A8THR0 < An < D\_A8THR1$, replace it with ISU_WPF_ALPH_VAL.D_ALPH1.

If $D\_A8THR1 \leq An$, replace it with ISU_WPF_ALPH_VAL.D_ALPH2.

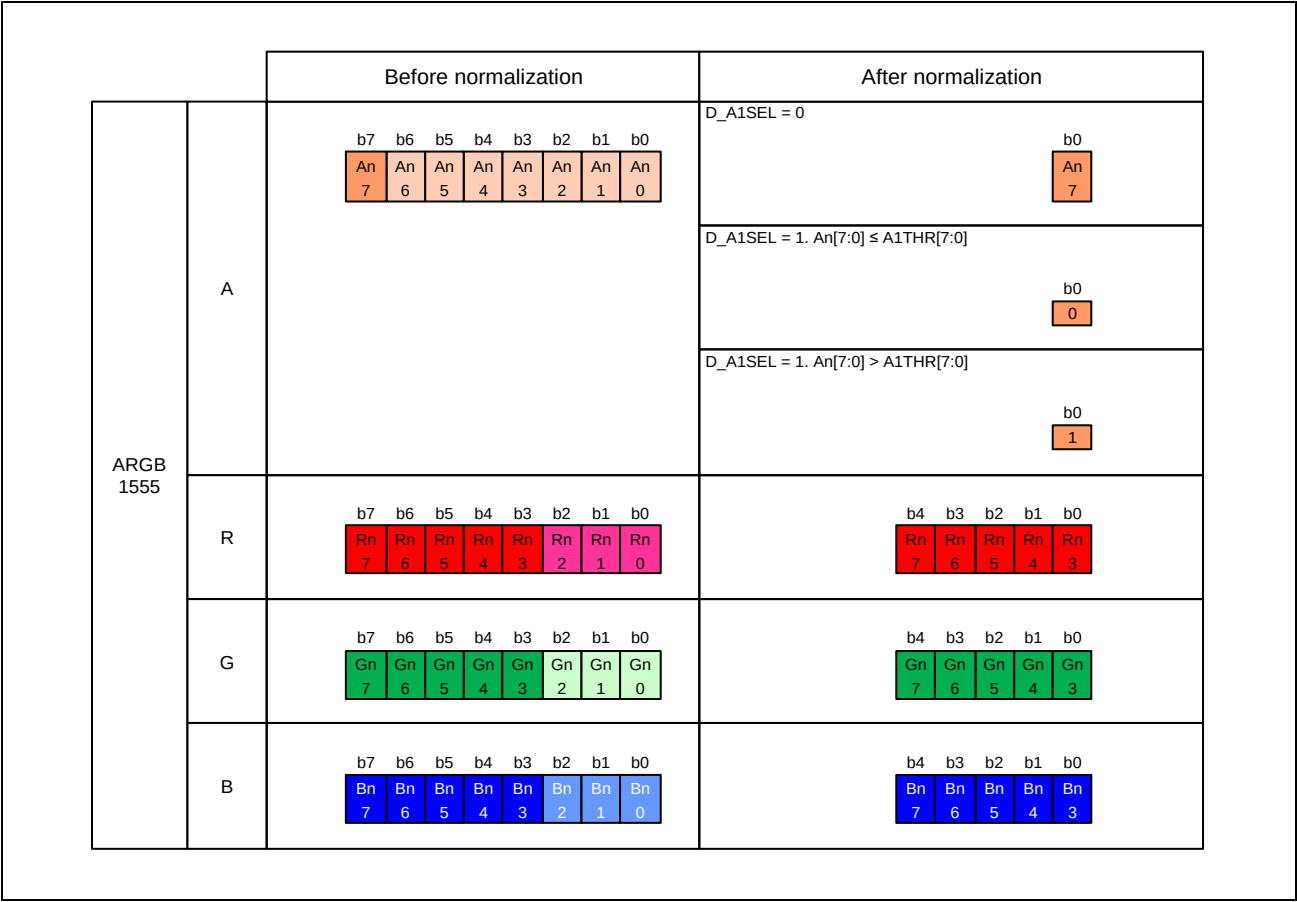


Figure 9.3-34 Normalization for ARGB1555 Format

“A(An)” of ARGB1555 A is compressed according to the following.

- If ISU_WPF_ALPH_SEL1.D_A1SEL = 0b, An7 is output.
- If ISU_WPF_ALPH_SEL1.D_A1SEL = 1b, replace it with the following value according to the relationship between ISU_WPF_ALPH_SEL1.D_A1THR and An.
- If $An \leq D\_A1THR$, 0 is output.
- If $D\_A1THR < An$, 1 is output.

■ YCbCr/YUV

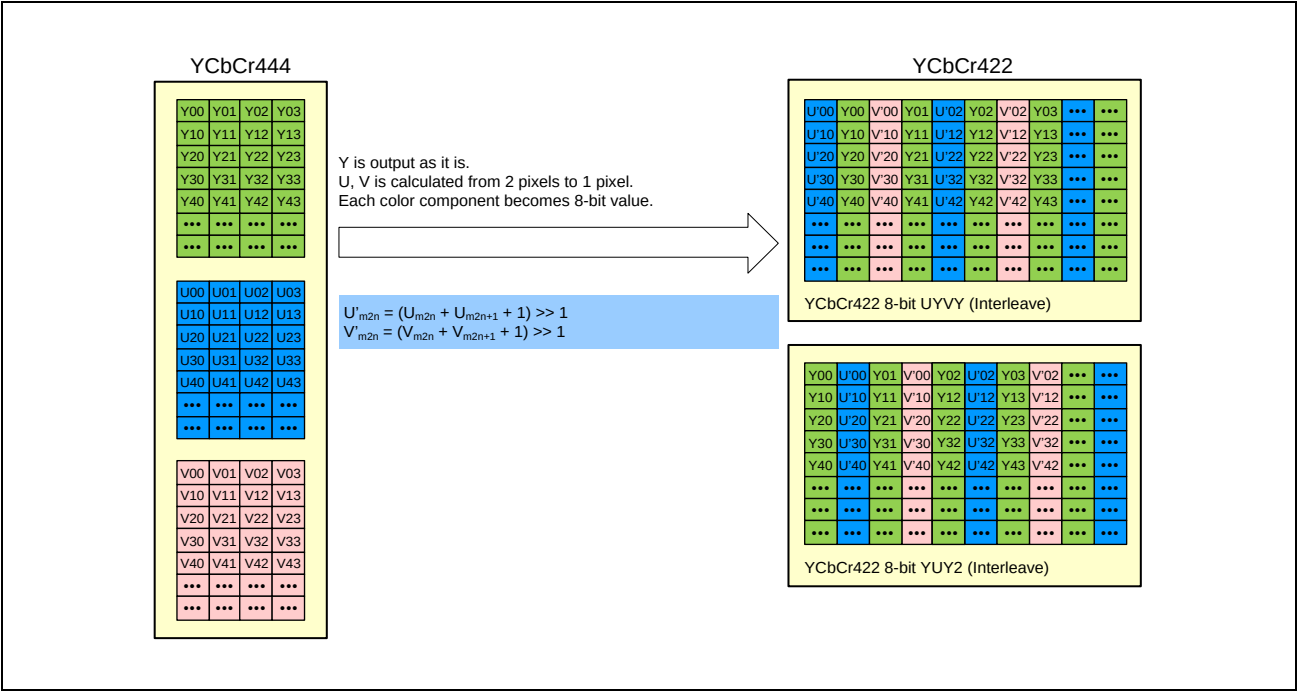


Figure 9.3-35 Normalization for YCbCr/YUV422 (Interleave) Format

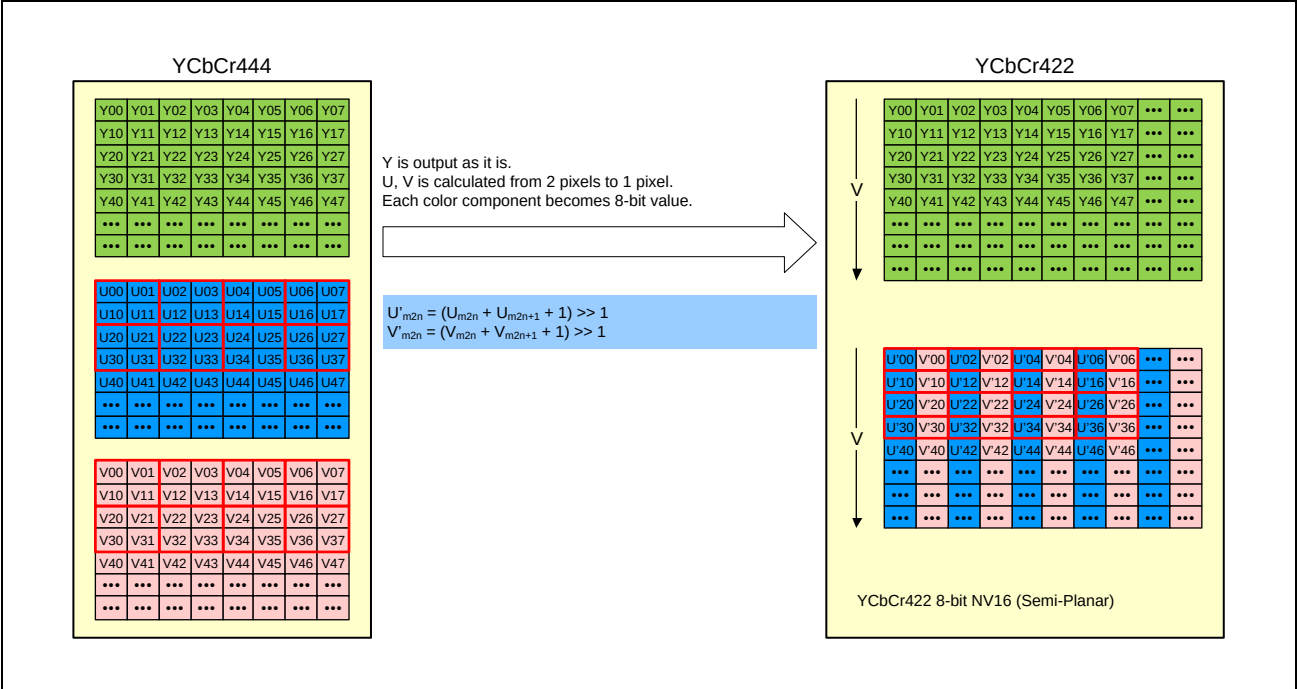


Figure 9.3-36 Normalization for YCbCr/YUV422 (Semi-Planar) Format

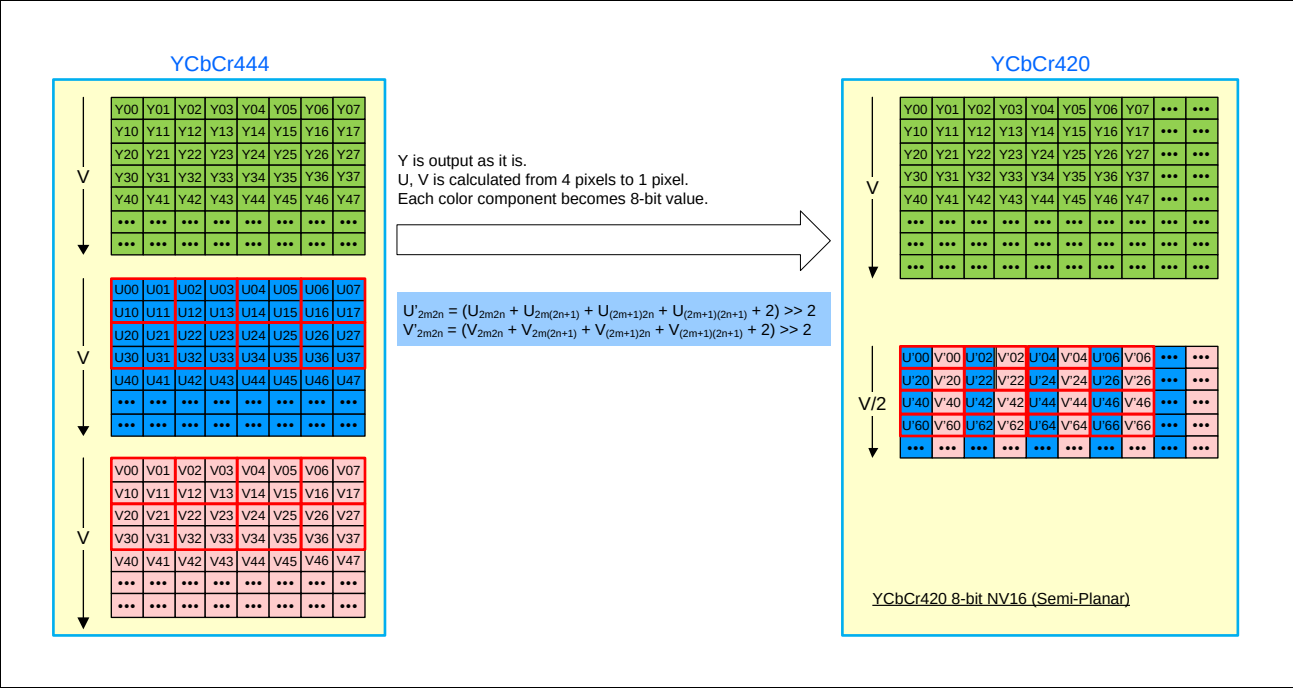


Figure 9.3-37 Normalization for YCbCr/YUV420 Format

■ RAW

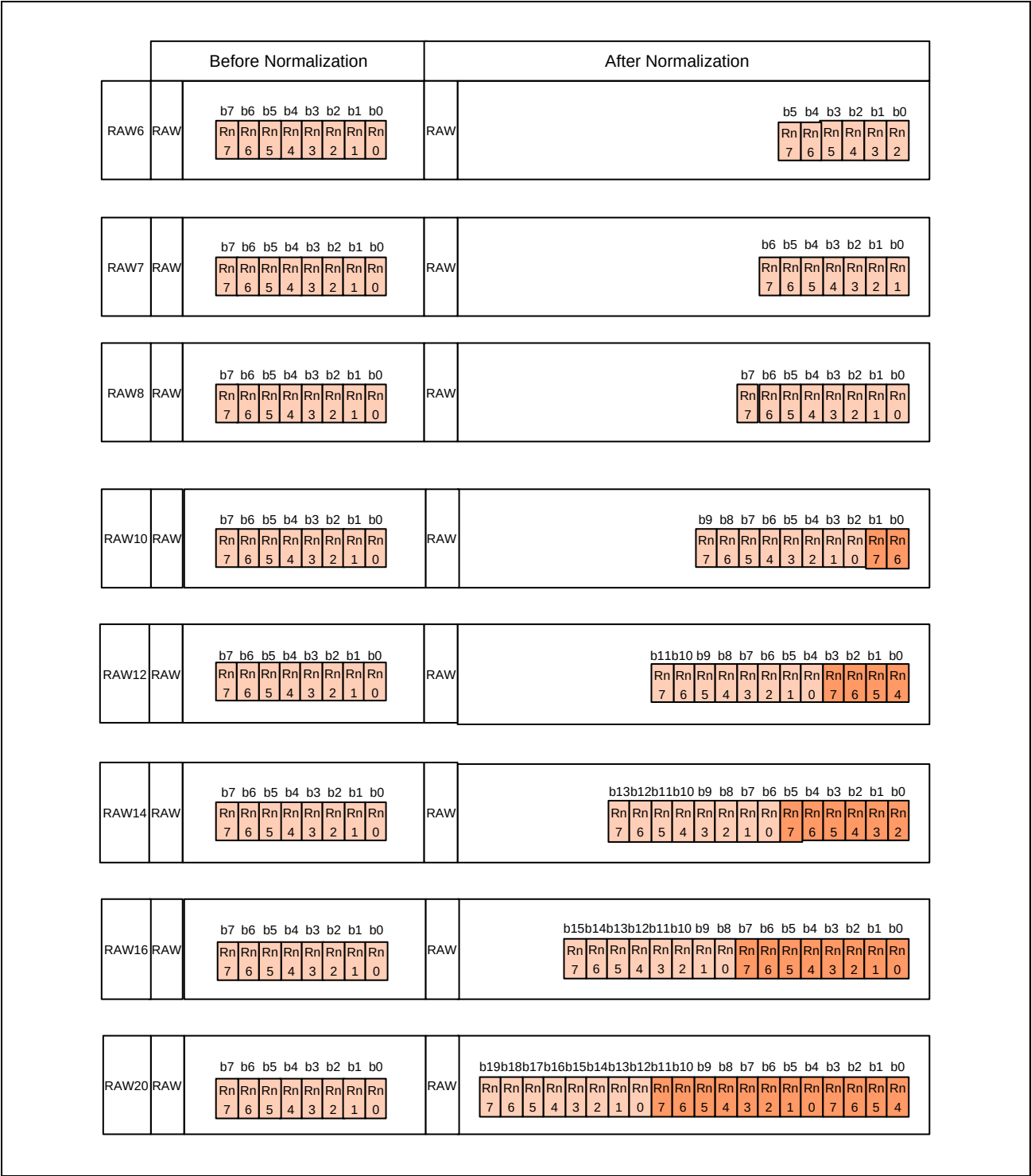


Figure 9.3-38 Normalization for RAW Format

9.3.1.5.3 WPF Formatter

The following data formats are described as 2-/3-/4-/8-byte per unit. Eventually, these data unit are packed in 32-byte (256-bit) each so that it can be handled in next process. Image format setting is indicated by ISU_WPF_FMT.WDFMT[5:0].

■ RGB/ARGB

This function supports 8 ARGB/RGB formats as shown below. All formats are “Number of planes = 1”.

Table 9.3-10 ARGB Support Format

	Format				n								n + 1								n + 2								n + 3							
WDFMT [5:0]	Color Format	Padding	bit/pixel	phase	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
01h	RGB565	—	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1
05h	ARGB8888	—	24	—	A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0
06h	RGBA8888	—		—	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A
03h	RGB888	—		0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1
				1	G1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2
			2	B2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3	B3
04h	BGR888	—	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1
				1	G1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2
				2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3
00h	ARGB1555	—	15	—	A	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	B1	
02h	BGR666	BGR upper	18	0			B0	B0	B0	B0	B0	B0			G0	G0	G0	G0	G0	G0			R0	R0	R0	R0	R0			B1	B1	B1	B1	B1	B1	
				1			G1	G1	G1	G1	G1	G1				R1	R1	R1	R1	R1	R1			B2	B2	B2	B2	B2	B2			G2	G2	G2	G2	G2
				2			R2	R2	R2	R2	R2	R2				B3	B3	B3	B3	B3	B3				G3	G3	G3	G3	G3	G3			R3	R3	R3	R3
07h	ABGR8888	—	24	—	A	A	A	A	A	A	A	A	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	

- **YCbCr/YUV**

This function supports 4 YCrCb/YUV formats as shown below.

Table 9.3-11 YCbCr/YUV Support Format

Data Format	RDFMT[5:0]	Color Format	Alias	Figure Index
Interleave (Number of Planes = 1)	20h	YCbCr422 8-bit	UYVY	(1)
	21h	YCbCr422 8-bit	YUY2	(2)
Semi-Planar (Number of Planes = 2)	22h	YCbCr422 8-bit	NV16	(3)
	23h	YCbCr420 8-bit	NV12	(4)

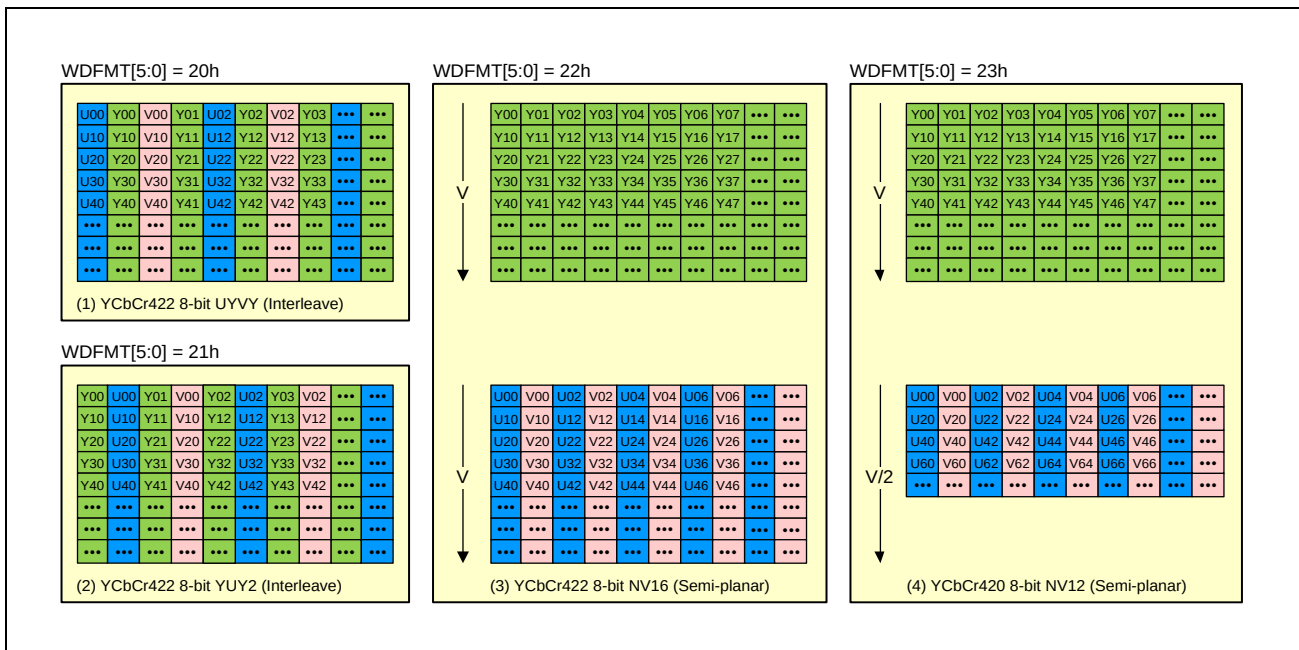


Figure 9.3-39 YCbCr/YUV Support Format

■ **RAW**

This function supports 8 RAW formats as shown below. All formats are “Number of planes = 1”.

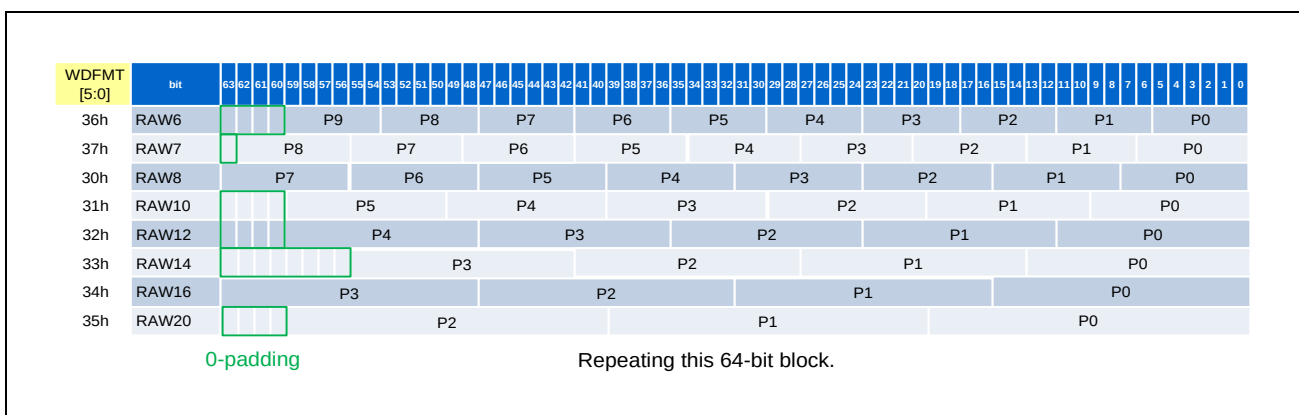


Figure 9.3-40 RAW Support Format

9.3.1.5.4 Data Swap (Endian Correction)

This function corrects the byte order so that modify endian of data on memory. Each bytes can be arranged any order according to the register setting.

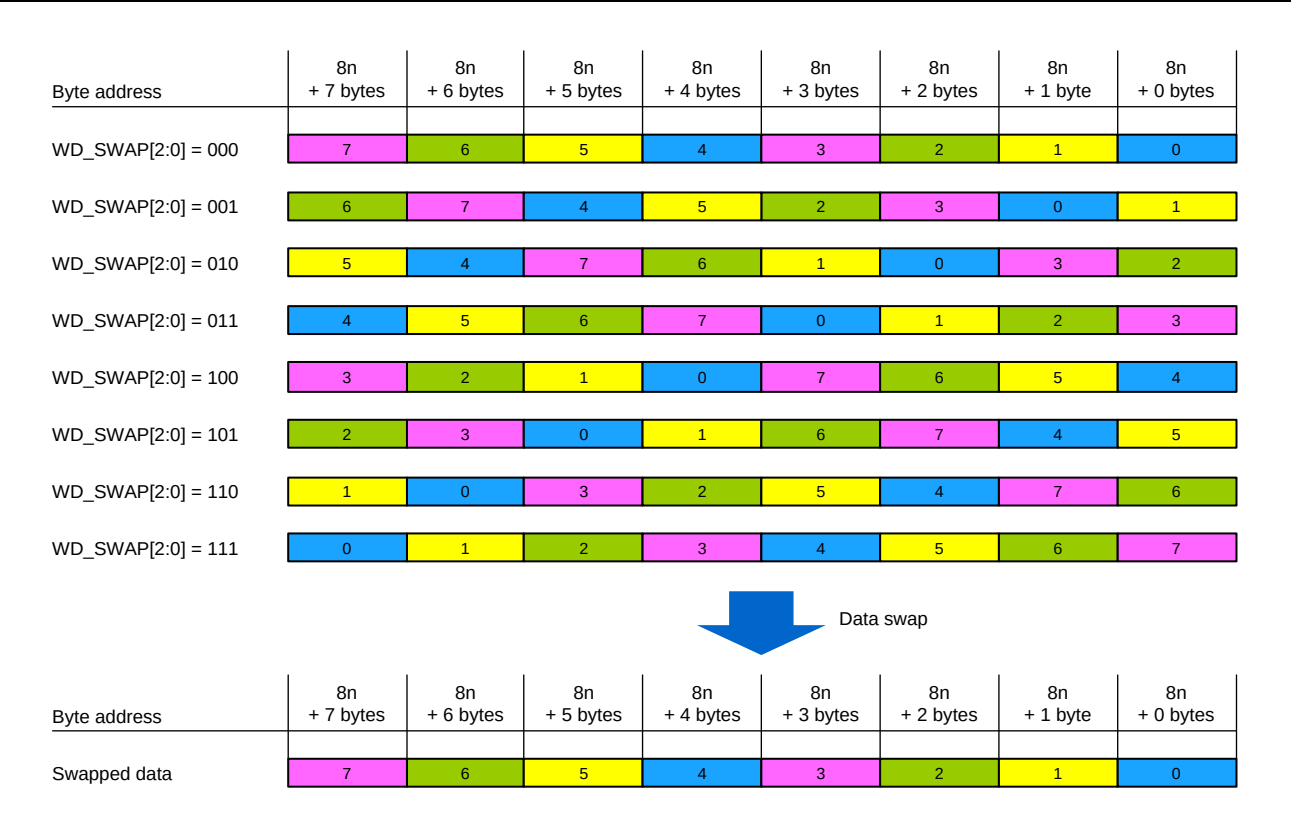


Figure 9.3-41 Data Swap Correction

9.3.2 Registers

The address of the ISU register is represented by the offset address from the base address. ISU base address is as follows:

Table 9.3-12 Register Base Address

Base Register Name	Base Address
<ISU_base>	0_1645_0000h (5645_0000h* ¹ , 4645_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.3.2.1 List of Registers

9.3.2.1.1 System Management Register

“System Management Register” sets descriptor processing and interrupts control.

The descriptor cannot access these registers.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
FM Descriptor List Address Registers 0	ISU_FM_DL_STADDH	0000_0000h	0000h	32
FM Descriptor List Address Registers 1	ISU_FM_DL_STADDL	0000_0000h	0004h	32
FM Frame Control Registers	ISU_FM_FRCON	0000_0000h	0008h	32
FM Module Stop Registers	ISU_FM_STOP	0000_0000h	000Ch	32
FM Interrupt Enable Registers	ISU_FM_INT_EN	0000_0000h	0010h	32
FM Interrupt Status Registers	ISU_FM_INT_STA	0000_0000h	0014h	32
FM Frame End Interrupt Control Registers	ISU_FM_INT_FREQ	0000_0000h	0018h	32
Reserve	-	-	001Ch to 001Fh	-
AXI Error Action Registers	ISU_AXI_ERACT	0000_0000h	0020h	32
Reserve	-	-	0024h to 002Bh	-
AXI FIFO Capability Registers	ISU_AXI_FIFO_CAP	4000_4000h	002Ch	32
Reserve	-	-	0030h to 00FFh	-

9.3.2.1.2 Common Register

“Common Register” sets input/output, reduction ratio and color conversion coefficient.

Prohibits write to these registers during frame processing.

The descriptor can access these registers.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
RPF Source Image Size Registers	ISU_RPF_SRC_SIZE	0000_0000h	0100h	32
RPF Source Stride Registers	ISU_RPF_SRC_STRD	0000_0000h	0104h	32
RPF Source Plane0 Address Registers 0	ISU_RPF_SRC_ADDH_PL0	0000_0000h	0108h	32
RPF Source Plane0 Address Registers 1	ISU_RPF_SRC_ADDL_PL0	0000_0000h	010Ch	32
RPF Source Plane1 Address Registers 0	ISU_RPF_SRC_ADDH_PL1	0000_0000h	0110h	32
RPF Source Plane1 Address Registers 1	ISU_RPF_SRC_ADDL_PL1	0000_0000h	0114h	32
RPF Source Image Format Registers	ISU_RPF_FMT	0000_0000h	0118h	32
RPF Source Image UV Format Registers	ISU_RPF_UVBIN	0000_0000h	011Ch	32
RPF Source Image Data Swap Registers	ISU_RPF_SRC_DSWAP	0000_0000h	0120h	32
RPF Source ALPHA Data Selection Registers	ISU_RPF_ALPH_SEL	0000_0000h	0124h	32
RPF Source TEST Data Registers 1	ISU_RPF_SRC_TD1	0000_0000h	0128h	32
RPF Source TEST Data Registers 2	ISU_RPF_SRC_TD2	0000_0000h	012Ch	32
Reserve	-	-	0130h to 013Fh	-
RS Scaling Factor Registers 0	ISU_RS_HSCALE	0000_0000h	0140h	32
RS Scaling Factor Registers 1	ISU_RS_VSCALE	0000_0000h	0144h	32
RS Output Image Start Position Registers	ISU_RS_STPOS	0000_0000h	0148h	32
RS Output Image Start Position Tuning Registers	ISU_RS_POS_TUNE	0000_0000h	014Ch	32
RS Output Size Crop Registers	ISU_RS_OS_CROP	0000_0000h	0150h	32
RS CROP Padding Mode Registers	ISU_RS_PADDMODE	0000_0000h	0154h	32
RS CROP Padding Value Registers	ISU_RS_PADDVAL	0000_0000h	0158h	32
Reserve	-	-	015Ch to 017Fh	-
WPF Destination Plane0 Address Registers 0	ISU_WPF_DST_ADDH_PL0	0000_0000h	0180h	32
WPF Destination Plane0 Address Registers 1	ISU_WPF_DST_ADDL_PL0	0000_0000h	0184h	32
WPF Destination Plane1 Address Registers 0	ISU_WPF_DST_ADDH_PL1	0000_0000h	0188h	32
WPF Destination Plane1 Address Registers 1	ISU_WPF_DST_ADDL_PL1	0000_0000h	018Ch	32
WPF Destination Stride Registers	ISU_WPF_DST_STRD	0000_0000h	0190h	32
WPF Destination Image Format Registers	ISU_WPF_FMT	0000_0000h	0194h	32
WPF Color Correction Control Registers	ISU_WPF_CCOL	0000_0000h	0198h	32
WPF Color Correction MUL Coefficient Registers1	ISU_WPF_MUL1	0000_0000h	019Ch	32
WPF Color Correction MUL Coefficient Registers2	ISU_WPF_MUL2	0000_0000h	01A0h	32
WPF Color Correction MUL Coefficient Registers3	ISU_WPF_MUL3	0000_0000h	01A4h	32
WPF Color Correction MUL Coefficient Registers4	ISU_WPF_MUL4	0000_0000h	01A8h	32
WPF Color Correction MUL Coefficient Registers5	ISU_WPF_MUL5	0000_0000h	01ACh	32
WPF Color Correction MUL Coefficient Registers6	ISU_WPF_MUL6	0000_0000h	01B0h	32
WPF Color Correction Offset Coefficient Registers1	ISU_WPF_OFST1	0000_0000h	01B4h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
WPF Color Correction Offset Coefficient Registers2	ISU_WPF_OFST2	0000_0000h	01B8h	32
WPF Color Correction Clip Registers1	ISU_WPF_CLP1	0000_0000h	01BCh	32
WPF Color Correction Clip Registers2	ISU_WPF_CLP2	0000_0000h	01C0h	32
WPF Destination Image Data Swap Registers	ISU_WPF_DST_DSWAP	0000_0000h	01C4h	32
WPF Destination ALPHA Selection Registers1	ISU_WPF_ALPH_SEL1	0000_0000h	01C8h	32
WPF Destination ALPHA Selection Registers2	ISU_WPF_ALPH_SEL2	0000_0000h	01CCh	32
WPF Destination ALPHA Value Registers	ISU_WPF_ALPH_VAL	0000_0000h	01D0h	32
Reserve	-	-	01D4h to 01EFh	-
AXI Max Burst Length Registers	ISU_AXI_BLEN	000F_000Fh	01F0h	32

9.3.2.2 Register Descriptions

9.3.2.2.1 FM Descriptor List Address Register 0 (ISU_FM_DL_STADDH)

This register sets the start address [34:32] of the “Descriptor List”.

Access Size :		32 bits														
Address :		<ISU_base> + 0000h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DLADD[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	DLADD[34:32]		0h		RW		Set the start address [34:32] of the “Descriptor-List”.									

9.3.2.2.2 FM Descriptor List Address Register 1 (ISU_FM_DL_STADDL)

This register sets the start address [31:0] of the “Descriptor List”.

Access Size :		32 bits															
Address :		<ISU_base> + 0004h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																	DLADD[31:16]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	DLADD[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	DLADD[31:0]																
31 to 0	DLADD[31:0]																
Initial Value		0h															
R/W		RW															
Description		Set the start address [31:0] of the “Descriptor List”. Note: Set an address that is a multiple of 32 bytes.															

9.3.2.2.3 FM Frame Control Register (ISU_FM_FRCON)

This register controls the frame processing method selection and processing start.

Access Size :		32 bits															
Address :		<ISU_base> + 0008h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DESON	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	START	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													
16	DESON	0h	RW	Processing method selection Select "Register Mode" or "Descriptor Mode". 0b: Register Mode When executing in this mode, set the register before starting frame processing. 1b: Descriptor Mode When executing in this mode, write a "Descriptor List" before starting frame processing.													
15 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													
0	START	0h	RW	Frame processing start instruction [Write] 0b: NOP 1b: Start frame processing [Read] This bit is always read as 0. Note: If 1 is written during frame processing, it will be ignored. Note: When 1 is written, the descriptor footer read completion interrupt (ISU_FM_INT_STA.DESEND) is automatically cleared.													

9.3.2.2.4 FM Module Stop Register (ISU_FM_STOP)

This register controls an emergency stop.

Access Size :		32 bits														
Address :		<ISU_base> + 000Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STOP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	STOP		0h		RW	Emergency stop instruction [Write] 0b: NOP 1b: Start emergency stop [Read] This bit is always read as 0. Note: It will be cleared automatically after the emergency stop is completed. If 1 is written during the emergency stop process, it will be ignored. Note: If you want to stop frame synchronously, you can stop it by setting the footer of the "Descriptor List" not to process the next frame.										

9.3.2.2.5 FM Interrupt Enable Register (ISU_FM_INT_EN)

This register controls the interrupt factor.

Access Size : 32 bits

Address : <ISU_base> + 0010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	AXIRX ERRE	-	-	-	AXITX ERRE	-	-	-	-	-	-	-	LISTER RE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SRSTE NDE	-	-	-	-	-	-	DESEN DE	FRENDE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	AXIRXERRE	0h	RW	Control of AXI bus read error interrupt (ISU_INT_ERR) Controls interrupt by "ISU_FM_INT_STA.RESPERR". 0b: Do not output interrupt factors 1b: Output interrupt factor
27 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	AXITXERRE	0h	RW	Control of AXI bus write error interrupt (ISU_INT_ERR) Controls interrupt by "ISU_FM_INT_STA.BRESPERR". 0b: Do not output interrupt factors 1b: Output interrupt factor
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	LISTERRE	0h	RW	Control of "Descriptor List" format violation interrupt (ISU_INT_ERR) Controls interrupt by the number of bytes set in the header exceeding the limit or not being in 8-byte units. 0b: Do not output interrupt factors 1b: Output interrupt factor
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	SRSTENDE	0h	RW	Control of emergency stop completion interrupt (ISU_INT_STOPE) Controls interrupt by the completion of emergency stop. 0b: Do not output interrupt factors 1b: Output interrupt factor
7 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	DESENDE	0h	RW	Control of descriptor footer read completion interrupt (ISU_INT_DESE) Controls interrupt by the completion of descriptor footer read. 0b: Do not output interrupt factors 1b: Output interrupt factor Note: When an error occurs due to a format violation of "Descriptor List", the ISU_INT_DESE interrupt does not occur even if this bit is set to 1.
0	FRENDE	0h	RW	Control of frame processing end interrupt (ISU_INT_FRE) Controls interrupt by the frame processing end. 0b: Not output interrupt factor 1b: Output interrupt factor Note: This bit is updated according to the setting of the footer, if the descriptor mode is set.

9.3.2.2.6 FM Interrupt Status Register (ISU_FM_INT_STA)

This register displays the status of the interrupt factors.

Access Size :		32 bits														
Address :		<ISU_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	RRESPERR[2:0]			-	BRESPERR[2:0]			-	-	-	-	-	-	LISTERR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW0	RW0	RW0	R	RW0	RW0	RW0	R	R	R	R	R	R	RW0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SRSTEND	-	-	-	-	-	-	DESEN D	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW0	R	R	R	R	R	R	RW0	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
29 to 27	RRESPERR [2:0]	0h	RW0	Displays the status of the AXI read response. Each bit means the following. [0] EXOKAY 0b: No EXOKAY response 1b: EXOKAY response [1] SLVERR 0b: No SLVERR response 1b: SLVERR response [2] DECERR 0b: No DECERR response 1b: DECERR response Note: If you want to clear these bits, clear the 3 bits at the same time.
26	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
25 to 23	BRESPERR [2:0]	0h	RW0	Displays the status of the AXI write response. Each bit means the following. [0] EXOKAY 0b: No EXOKAY response 1b: EXOKAY response [1] SLVERR 0b: No SLVERR response 1b: SLVERR response [2] DECERR 0b: No DECERR response 1b: DECERR response Note: If you want to clear these bits, clear the 3 bits at the same time.
22 to 17	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
16	LISTERR	0h	RW0	Displays the status of “Descriptor List” format violation (ISU_INT_ERR). 0b: Descriptor list does not violate format 1b: Descriptor list does violate format
15 to 9	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
8	SRSTEND	0h	RW0	Displays the status of the emergency stop completion (ISU_INT_STOPE). 0b: Emergency stop not completed 1b: Emergency stop completed
7 to 2	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.

Bit	Bit Name	Initial Value	R/W	Description
1	DESEND	0h	RW0	Displays the status of the descriptor footer read completion (ISU_INT_DESE). 0b: Descriptor footer read incomplete 1b: Descriptor footer read completed
0	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.

9.3.2.2.7 FM Frame End Interrupt Control Register (ISU_FM_INT_FREQ)

This register sets the number of toggles when issuing an ISU_INT_FRE interrupt.

Access Size :		32 bits														
Address :		<ISU_base> + 0018h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FREQ[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 2	-		All 0		R		Reserved Whenever it is read, 0 is read. The written value will be ignored.									
1, 0	FREQ[1:0]		0h		RW		Sets the number of ISU_INT_FRE interrupt toggles. ISU_INT_FRE interrupt toggle counter is reset by writing 1b to ISU_FM_FRCON.START. 0: Always outputs a frame end interrupt from ISU_INT_FRE0. 1: A frame end interrupt is output from ISU_INT_FRE0 to FRE1. ISU_INT_FRE0 to FRE1 are toggled in order each time the frame ends. 2: A frame end interrupt is output from ISU_INT_FRE0 to FRE2. ISU_INT_FRE0 to FRE2 are toggled in order each time the frame ends. 3: A frame end interrupt is output from ISU_INT_FRE0 to FRE3. ISU_INT_FRE0 to FRE3 are toggled in order each time the frame ends.									

9.3.2.2.8 AXI Error Action Register (ISU_AXI_ERACT)

This register sets the operation of the AXI-Master after an AXI error occurs.

For each AXI error condition, choose whether to ignore and continue or stop subsequent transactions.

Transaction continuation settings are used for debugging purposes.

Access Size :		32 bits														
Address :		<ISU_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	RRESP EXO_A CT	RRESP SLV_A CT	RRESP DEC_A CT	-	BRESP EXO_A CT	BRESP SLV_A CT	BRESP DEC_A CT	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	R	RW	RW	RW	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
29	RRESPEXO_AC T	0h	RW	Operation setting after AXI read response EXOKAY occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
28	RRESPSLV_AC T	0h	RW	Operation setting after AXI read response SLVERR occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
27	RRESPDEC_AC T	0h	RW	Operation setting after AXI read response DECERR occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
26	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
25	BRESPEXO_AC T	0h	RW	Operation setting after AXI write response EXOKAY occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
24	BRESPSLV_AC T	0h	RW	Operation setting after AXI write response SLVERR occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
23	BRESPDEC_AC T	0h	RW	Operation setting after AXI write response DECERR occurs 0b: Transaction stop 1b: Transaction continuation (For debugging)
22 to 0	-	0h	R	When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.

9.3.2.2.9 AXI FIFO Capability Register (ISU_AXI_FIFO_CAP)

This register is a register that displays the free space of the FIFO in the AXI-Master.

Access Size : 32 bits

Address : <ISU_base> + 002Ch

Initial Value : 4000_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	SCAP_PL0[6:0]							-	-	SCAP_PL1[5:0]					
Initial Value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	DCAP_PL0[6:0]							-	-	DCAP_PL1[5:0]					
Initial Value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 24	SCAP_PL0[6:0]	40h	R	Free space in FIFO of the input image "Plane 0" 0 to 32 in Semi-Planar mode. 0 to 64 in other modes. The result of multiplying the value of this register by 32 bytes indicates the free space of the FIFO.
23, 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21 to 16	SCAP_PL1[5:0]	0h	R	Free space in FIFO of the input image "Plane 1" 0 to 32 in Semi-Planar mode. 0 in other modes. The result of multiplying the value of this register by 32 bytes indicates the free space of the FIFO.
15	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14 to 8	DCAP_PL0[6:0]	40h	R	Free space in FIFO of the output image "Plane 0" 0 to 32 in Semi-Planar mode. 0 to 64 in other modes. The result of multiplying the value of this register by 32 bytes indicates the free space of the FIFO.
7, 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	DCAP_PL1[5:0]	0h	R	Free space in FIFO of the output image "Plane 1" 0 to 32 in Semi-Planar mode. 0 in other modes. The result of multiplying the value of this register by 32 bytes indicates the free space of the FIFO.

9.3.2.2.10 RPF Source Image Size Register (ISU_RPF_SRC_SIZE)

This register sets the size of the input image.

Access Size :		32 bits														
Address :		<ISU_base> + 0100h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	S_HSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	S_VSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	S_HSIZE[12:0]	0h	RW	Horizontal size of the input image Set the horizontal size of the input image in pixels. Set the size from 1 to 4096. 0: Setting prohibited 4097 or higher: Setting prohibited Note: For YCbCr type, set it to an even number.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	S_VSIZE[12:0]	0h	RW	Vertical size of the input image Set the vertical size of the input image in pixels. Set the size from 1 to 4096. 0: Setting prohibited 4097 or higher: Setting prohibited Note: For YCbCr420, set it to an even number.

9.3.2.2.11 RPF Source Stride Register (ISU_RPF_SRC_STRD)

This register sets the stride of the input image.

Access Size :		32 bits															
Address :		<ISU_base> + 0104h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																	SSTRPL0[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	SSTRPL1[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SSTRPL0[15:0]	0h	RW	Set the "Plane 0 (Y/RGB/RAW)" stride Set the "Plane 0 (Y/RGB/RAW)" stride in bytes. Note: The setting must be a multiple of 32 bytes and at least the number of horizontal pixels (ISU_RPF_SRC_SIZE.S_HSIZE).
15 to 0	SSTRPL1[15:0]	0h	RW	Set the "Plane 1 (C/Cb/Cr)" stride Set the "Plane 1 (C/Cb/Cr)" stride in bytes. Note: The setting must be a multiple of 32 bytes and at least the number of horizontal pixels (ISU_RPF_SRC_SIZE.S_HSIZE).

9.3.2.2.12 RPF Source Plane 0 Address Register 0 (ISU_RPF_SRC_ADDH_PL0)

This register sets the start address [34:32] of the input image of “Plane 0”.

Access Size :		32 bits														
Address :		<ISU_base> + 0108h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SADD_PL0[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3		-		All 0		R		Reserved								
								Whenever it is read, 0b is read. The written value will be ignored.								
2 to 0		SADD_PL0 [34:32]		0h		RW		Set the start address [34:32] for "Plane 0 (Y/RGB/RAW)".								

9.3.2.2.13 RPF Source Plane 0 Address Register 1 (ISU_RPF_SRC_ADDL_PL0)

This register sets the start address [31:0] of the input image of “Plane 0”.

Access Size :		32 bits															
Address :		<ISU_base> + 010Ch															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																	SADD_PL0[31:16]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	SADD_PL0[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit																	
31 to 0	SADD_PL0	Initial Value		R/W	Description												
		[31:0]			Set the start address [31:0] of “Plane 0 (Y/RGB/RAW)”.												
					Note: Set an address that is a multiple of 32 bytes.												

9.3.2.2.14 RPF Source Plane 1 Address Register 0 (ISU_RPF_SRC_ADDH_PL1)

This register sets the start address [34:32] of the input image of “Plane 1”.

Access Size :		32 bits														
Address :		<ISU_base> + 0110h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	SADD_PL1[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3	-		All 0		R/W		Description									
		Reserved														
		Whenever it is read, 0b is read. The written value will be ignored.														
2 to 0	SADD_PL1 [34:32]		0h		RW		Set the start address [34:32] of "Plane 1 (C(Cb/Cr))".									

9.3.2.2.15 RPF Source Plane 1 Address Register 1 (ISU_RPF_SRC_ADDL_PL1)

This register sets the start address [31:0] of the input image of “Plane 1”.

Access Size :		32 bits															
Address :		<ISU_base> + 0114h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																	SADD_PL1[31:16]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	SADD_PL1[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit																	
31 to 0	SADD_PL1	Initial Value		R/W		Description											
		[31:0]				Set the start address [31:0] for “Plane 1 (C(Cb/Cr))”.											
						Note: Set an address that is a multiple of 32 bytes.											

9.3.2.2.16 RPF Source Image Format Register (ISU_RPF_FMT)

This register sets the color format of the input image.

For details, see **9.3.1.3.2 RPF** Formatter.

Access Size :		32 bits														
Address :		<ISU_base> + 0118h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	RDFMT[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	RDFMT[5:0]	0h	RW	Color format of input image The following formats can be configured. 00h: ARGB1555 01h: RGB565 02h BGR666 03h: RGB888 04h: BGR888 05h: ARGB8888 06h: RGBA8888 07h: ABGR8888 20h: YCbCr422 8-bit UYVY 21h: YCbCr422 8-bit YUY2 22h: YCbCr422 8-bit NV16 23h: YCbCr420 8-bit NV12 30h: RAW8 31h: RAW10 32h: RAW12 33h: RAW14 34h: RAW16 35h: RAW20 36h: RAW6 37h: RAW7 Others: Setting prohibited

9.3.2.2.17 RPF Source Image UV Format Register (ISU_RPF_UVBIN)

This register sets whether to convert CbCr/UV to offset binary.

For details, see 9.3.1.3.2 RPF Formatter.

Access Size :		32 bits														
Address :		<ISU_base> + 011Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UVCH G
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name															
31 to 1	-															
	Initial Value		All 0		R/W		Description									
							Reserved									
							Whenever it is read, 0b is read. The written value will be ignored.									
0	UVCHG		0h		RW		Offset binary conversion selection for CbCr / UV data									
							0b: No conversion									
							1b: Conversion									

9.3.2.2.18 RPF Source Image Data Swap Register (ISU_RPF_SRC_DSWAP)

This register sets the data swap of the input image.

For details, see **9.3.1.3.1 Data swap (endian correction)**.

Access Size :		32 bits														
Address :		<ISU_base> + 0120h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	RD_SWAP[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	RD_SWAP[2:0]		0h		RW		Data swap for input images									

9.3.2.2.19 RPF Source ALPHA Data Selection Register (ISU_RPF_ALPH_SEL)

This register sets the extension method from A (Alpha) 1-bit to 8-bit at the time of ARGB1555 input.

For details, see **9.3.1.3.3 Data extraction**.

Access Size :		32 bits														
Address :		<ISU_base> + 0124h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S_A1SEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	S_ALPH1[7:0]								S_ALPH0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name	Initial Value	R/W	Description												
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												
16	S_A1SEL	0h	RW	Setting of how to extend A(Alpha) from 1-bit to 8-bit 0b: Copy input Alpha value 1-bit to all 8-bit 1b: Select the register value according to the input Alpha value 1 bit S_ALPH0 when Alpha = 0b S_ALPH1 when Alpha = 1b												
15 to 8	S_ALPH1[7:0]	0h	RW	Extended Alpha value when S_A1SEL = 1b and input Alpha value is 1b.												
7 to 0	S_ALPH0[7:0]	0h	RW	Extended Alpha value when S_A1SEL = 1b and input Alpha value is 0b.												

9.3.2.2.20 RPF Source TEST Data Register 1 (ISU_RPF_SRC_TD1)

This register controls the test pattern generation.

For details, see **9.3.1.3.4 Test Pattern Generation**.

Access Size :		32 bits														
Address :		<ISU_base> + 0128h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMOD E_EN	-	-	-	-	-	-	-	-	-	-	-	GRAD A_A	GRAD A_R	GRAD A_G	GRAD A_B
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	GRAD A_TYP E	GRADA_STEP[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	TMODE_EN	0h	RW	Control of test pattern generation 0b: Normal operation 1b: Test pattern generation After one frame of the test pattern is generated, it is automatically cleared to 0b. Note: Setting in descriptor mode is prohibited.
30 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	GRADA_A	0h	RW	A color when generating a test pattern 0b: Fixed initial value (ISU_RPF_SRC_TD2.INIT_A) 1b: Gradation
18	GRADA_R	0h	RW	R color when generating a test pattern 0b: Fixed initial value (ISU_RPF_SRC_TD2.INIT_R) 1b: Gradation
17	GRADA_G	0h	RW	G color when generating a test pattern 0b: Fixed initial value (ISU_RPF_SRC_TD2.INIT_G) 1b: Gradation
16	GRADA_B	0h	RW	B color when generating a test pattern 0b: Fixed initial value (ISU_RPF_SRC_TD2.INIT_B) 1b: Gradation
15 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	GRADA_TYPE	0h	RW	Gradient type 0b: Gradation from 0 to 255 1b: Gradation from 255 to 0
3 to 0	GRADA_STEP [3:0]	0h	RW	Gradient step Note: If the setting is 0b, the initial value (ISU_RPF_SRC_TD2.INIT_*) is entered.

9.3.2.2.21 RPF Source TEST Data Register 2 (ISU_RPF_SRC_TD2)

This register sets the initial value of each color of ARGB for test pattern generation.

For details, see **9.3.1.3.4 Test Pattern Generation**.

Access Size :		32 bits															
Address :		<ISU_base> + 012Ch															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
INIT_A[7:0]								INIT_R[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
INIT_G[7:0]								INIT_B[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	INIT_A[7:0]	0h	RW	Initial value of A color
23 to 16	INIT_R[7:0]	0h	RW	Initial value of R color
15 to 8	INIT_G[7:0]	0h	RW	Initial value of G color
7 to 0	INIT_B[7:0]	0h	RW	Initial value of B color

9.3.2.2.22 RS Scaling Factor Register 0 (ISU_RS_HSCALE)

This register sets the reduction factor of the input image in the horizontal direction.

For details, see 9.3.1.4.3 Scaling calculation.

Access Size :		32 bits														
Address :		<ISU_base> + 0140h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	HMANT[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	HFRAC[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 16	HMANT[3:0]	0h	RW	Horizontal reduction factor (integer part) 0: Setting prohibited
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	HFRAC[11:0]	0h	RW	Horizontal multiplication factor (decimal part)

9.3.2.2.23 RS Scaling Factor Register 1 (ISU_RS_VSCALE)

This register sets the reduction factor of the input image in the vertical direction.

For details, see 9.3.1.4.3 Scaling calculation.

Access Size :		32 bits														
Address :		<ISU_base> + 0144h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	VMANT[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	VFRAC[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 20	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
19 to 16	VMANT[3:0]		0h		RW	Vertical reduction factor (integer part) 0: Setting prohibited										
15 to 12	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 0	VFRAC[11:0]		0h		RW	Vertical reduction factor (decimal part)										

9.3.2.2.24 RS Output Image Start Position Register (ISU_RS_STPOS)

This register sets the resize start position.

For details, see **9.3.1.4.1 Resize start position**.

Access Size :		32 bits																	
Address :		<ISU_base> + 0148h																	
Initial Value :		0000_0000h																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
	-	-	-	-	HSTART[11:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	-	-	-	-	VSTART[11:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 16	HSTART[11:0]	0h	RW	Resize start horizontal position Set the resizing start horizontal position by the number of pixels with the upper left of the input image as the origin. Set it with less than the number of input horizontal pixels (ISU_RPF_SRC_SIZES_HSIZE).
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	VSTART[11:0]	0h	RW	Resize start vertical position Set the resizing start vertical position by the number of pixels with the upper left of the input image as the origin. Set it with less than the number of input vertical pixels (ISU_RPF_SRC_SIZES_VSIZE).

9.3.2.2.25 RS Output Image Start Position Tuning Register (ISU_RS_POS_TUNE)

This register sets the offset for fine adjustment of the reduction start position.

For details, see 9.3.1.4.2 Sub pixel offset.

Access Size :		32 bits														
Address :		<ISU_base> + 014Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	HST_TUNE[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	VST_TUNE[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 28	-		All 0		R		Reserved									
							Whenever it is read, 0b is read. The written value will be ignored.									
27 to 16	HST_TUNE [11:0]		0h		RW		Offset from the resize-starting horizontal position (ISU_RS_STPOS.HSTART)									
							Set the offset to be added to the horizontal position at the start of resizing in 4,096 steps (0 to less than 1 pixel of the input image).									
15 to 12	-		All 0		R		Reserved									
							Whenever it is read, 0b is read. The written value will be ignored.									
11 to 0	VST_TUNE [11:0]		0h		RW		Offset from the resize-starting vertical position (ISU_RS_STPOS.VSTART)									
							Set the offset to be added to the vertical position at the start of resizing in 4,096 steps (0 to less than 1 pixel of the input image).									

9.3.2.2.26 RS Output Size Crop Register (ISU_RS_OS_CROP)

This register sets the crop size of the reduced image.

For details, see **9.3.1.4 Resizer (RS)**.

Access Size :		32 bits														
Address :		<ISU_base> + 0150h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	O_HSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	O_VSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	O_HSIZE[12:0]	0h	RW	Horizontal crop size Set the horizontal crop size in pixels. Set it in the range of 1 to 4096. 0: Setting prohibited 4097 or higher: Setting prohibited Note: For YCbCr type, set it to an even number.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	O_VSIZE[12:0]	0h	RW	Vertical crop size Set the vertical crop size in pixels. Set it in the range of 1 to 4096. 0: Setting prohibited 4097 or higher: Setting prohibited Note: For YCbCr420, set it to an even number.

9.3.2.2.27 RS CROP Padding Mode Register (ISU_RS_PADDMODE)

This register sets the padding method when the reduced image is smaller than the set crop size.

For details, see 9.3.1.4 Resizer (RS).

Access Size :		32 bits														
Address :		<ISU_base> + 0154h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PADDSEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	PADDSEL		0h		RW		Selecting a padding method Set the padding method when the reduced image is smaller than the set crop size. 0b: Fill in the padding area by copying the edge pixels after the reduction. 1b: Fill in the padding area with the color specified by the register (ISU_RS_PADDVAL.VAL_*).									

9.3.2.2.28 RS CROP Padding Value Register (ISU_RS_PADDVAL)

This register sets the color that fills the edges with padding.

For details, see **9.3.1.4 Resizer (RS)**.

Access Size :		32 bits														
Address :		<ISU_base> + 0158h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VAL_L3[7:0]									VAL_L2[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL_L1[7:0]									VAL_L0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	VAL_L3[7:0]	0h	RW	The value to fill Layer 3, if the reduced image is smaller than the set crop size.
23 to 16	VAL_L2[7:0]	0h	RW	The value to fill Layer 2, if the reduced image is smaller than the set crop size.
15 to 8	VAL_L1[7:0]	0h	RW	The value to fill Layer 1, if the reduced image is smaller than the set crop size.
7 to 0	VAL_L0[7:0]	0h	RW	The value to fill Layer 0, if the reduced image is smaller than the set crop size.

9.3.2.2.29 WPF Destination Plane 0 Address Register 0 (ISU_WPF_DST_ADDH_PL0)

This register sets the start address [34:32] of the output image of “Plane 0”.

Access Size :		32 bits														
Address :		<ISU_base> + 0180h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DADD_PL0[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3		-	All 0		R/W		Description									
		Reserved														
		Whenever it is read, 0b is read. The written value will be ignored.														
2 to 0		DADD_PL0 [34:32]	0h		RW		Set the write start address [34:32] for "Plane 0 (Y/RGB/RAW)".									

9.3.2.2.30 WPF Destination Plane 0 Address Register 1 (ISU_WPF_DST_ADDL_PL0)

This register sets the start address [31:0] of the output image of “Plane 0”.

Access Size :		32 bits														
Address :		<ISU_base> + 0184h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DADD_PL0[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADD_PL0[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
		Initial Value														
		R/W														
		Description														
31 to 0	DADD_PL0 [31:0]	0h		RW	Set the write start address [31:0] for "Plane 0 (Y/RGB/RAW)". Note: Set an address that is a multiple of 512 bytes.											

9.3.2.2.31 WPF Destination Plane 1 Address Register 0 (ISU_WPF_DST_ADDH_PL1)

This register sets the start address [34:32] of the output image of “Plane 1”.

Access Size :		32 bits														
Address :		<ISU_base> + 0188h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DADD_PL1[34:32]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name															
31 to 3	-															
	All 0															
	R															
	Reserved															
	Whenever it is read, 0b is read. The written value will be ignored.															
2 to 0	DADD_PL1 [34:32]															
	0h															
	RW															
	Set the write start address [34:32] for "Plane 1 (C(Cb/Cr))".															

9.3.2.2.32 WPF Destination Plane 1 Address Register 1 (ISU_WPF_DST_ADDL_PL1)

This register sets the start address [31:0] of the output image of “Plane 1”.

Access Size :		32 bits														
Address :		<ISU_base> + 018Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DADD_PL1[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADD_PL1[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
Initial Value	Initial Value															
R/W	R/W															
Description	Description															
31 to 0	DADD_PL1 [31:0] Set the write start address [31:0] for "Plane 1 (C/Cb/Cr)". Note: Set an address that is a multiple of 512 bytes.															

9.3.2.2.33 WPF Destination Stride Register (ISU_WPF_DST_STRD)

This register sets the stride of the output image.

Access Size :		32 bits														
Address :		<ISU_base> + 0190h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTRPL0[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTRPL1[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	DSTRPL0[15:0]	0h	RW	Set the "Plane 0 (Y/RGB/RAW)" stride Set the "Plane 0 (Y/RGB/RAW)" stride in bytes. Note: The setting must be a multiple of 32 bytes and at least the number of horizontal crop size (ISU_RS_OS_CROP.O_HSIZE).
15 to 0	DSTRPL1[15:0]	0h	RW	Set the "Plane 1 (C/Cb/Cr)" stride Set the "Plane 1 (C/Cb/Cr)" stride in bytes. Note: The setting must be a multiple of 32 bytes and at least the number of horizontal crop size (ISU_RS_OS_CROP.O_HSIZE).

9.3.2.2.34 WPF Destination Image Format Register (ISU_WPF_FMT)

This register sets the color format of the output image.

For details, see 9.3.1.5.3 WPF Formatter.

Access Size :		32 bits														
Address :		<ISU_base> + 0194h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	WDFMT[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	WDFMT[5:0]	0h	RW	Color format of output image The following formats can be configured. 00h: ARGB1555 01h: RGB565 02h: BGR666 03h: RGB888 04h: BGR888 05h: ARGB8888 06h: RGBA8888 07h: ABGR8888 20h: YCbCr422 8-bit UYVY 21h: YCbCr422 8-bit YUY2 22h: YCbCr422 8-bit NV16 23h: YCbCr420 8-bit NV12 30h: RAW8 31h: RAW10 32h: RAW12 33h: RAW14 34h: RAW16 35h: RAW20 36h: RAW6 37h: RAW7 Others: Setting prohibited

9.3.2.2.35 WPF Color Correction Control Register (ISU_WPF_CCOL)

This register sets the color conversion correction control.

For details, see **9.3.1.5.1 Color correction**.

Access Size :		32 bits														
Address :		<ISU_base> + 0198h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMAT_SEL	CMAT_ASEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 2	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
1	CMAT_SEL		0h	RW	Color conversion correction control 0b: No conversion 1b: Conversion											
0	CMAT_ASEL		0h	RW	A (Alpha) color conversion correction control 0b: No conversion 1b: Conversion											

9.3.2.2.36 WPF Color Correction MUL Coefficient Register 1 (ISU_WPF_MUL1)

This register sets the matrix value K11 for the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits														
Address :		<ISU_base> + 019Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K11[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 14	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
13 to 0	K11[13:0]		0h	RW	Color Correction Coefficient K11 Signed fixed point: 4-bit integer part, 10-bit decimal part											

9.3.2.2.37 WPF Color Correction MUL Coefficient Register 2 (ISU_WPF_MUL2)

This register sets the matrix values K12 and K13 for the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits														
Address :		<ISU_base> + 01A0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	K12[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K13[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 16	K12[13:0]	0h	RW	Color Correction Coefficient K12 Signed fixed point: 4-bit integer part, 10-bit decimal part
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 0	K13[13:0]	0h	RW	Color Correction Coefficient K13 Signed fixed point: 4-bit integer part, 10-bit decimal part

9.3.2.2.38 WPF Color Correction MUL Coefficient Register 3 (ISU_WPF_MUL3)

This register sets the matrix value K21 for the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits														
Address :		<ISU_base> + 01A4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K21[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 14	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
13 to 0	K21[13:0]		0h	RW	Color Correction Coefficient K21 Signed fixed point: 4-bit integer part, 10-bit decimal part											

9.3.2.2.39 WPF Color Correction MUL Coefficient Register 4 (ISU_WPF_MUL4)

This register sets the matrix values K22 and K23 for the color conversion correction.

For details, see **9.3.1.5.1 Color correction**.

Access Size :		32 bits														
Address :		<ISU_base> + 01A8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	K22[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K23[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 16	K22[13:0]	0h	RW	Color Correction Coefficient K22 Signed fixed point: 4-bit integer part, 10-bit decimal part
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 0	K23[13:0]	0h	RW	Color Correction Coefficient K23 Signed fixed point: 4-bit integer part, 10-bit decimal part

9.3.2.2.40 WPF Color Correction MUL Coefficient Register 5 (ISU_WPF_MUL5)

This register sets the matrix value K31 for the color conversion correction.

For details, see **9.3.1.5.1 Color correction**.

Access Size :		32 bits														
Address :		<ISU_base> + 01ACh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K31[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 14	-		All 0		R		Reserved									
							Whenever it is read, 0b is read. The written value will be ignored.									
13 to 0	K31[13:0]		0h		RW		Color Correction Coefficient K31									
							Signed fixed point: 4-bit integer part, 10-bit decimal part									

9.3.2.2.41 WPF Color Correction MUL Coefficient Register 6 (ISU_WPF_MUL6)

This register sets the matrix values K32 and K33 for the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits														
Address :		<ISU_base> + 01B0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	K32[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	K33[13:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 16	K32[13:0]	0h	RW	Color Correction Coefficient K32 Signed fixed point: 4-bit integer part, 10-bit decimal part
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 0	K33[13:0]	0h	RW	Color Correction Coefficient K33 Signed fixed point: 4-bit integer part, 10-bit decimal part

9.3.2.2.42 WPF Color Correction Offset Coefficient Register 1 (ISU_WPF_OFST1)

This register sets the offset 1 of the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits																
Address :		<ISU_base> + 01B4h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	OFST_A1[7:0]								OFST_B1[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	OFST_C1[7:0]								-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R		
Bit	Bit Name		Initial Value	R/W	Description													
31 to 24	OFST_A1[7:0]		0h	RW	Color Correction Offset value A1 Unsigned integer 8-bit													
23 to 16	OFST_B1[7:0]		0h	RW	Color Correction Offset value B1 Unsigned integer 8-bit													
15 to 8	OFST_C1[7:0]		0h	RW	Color Correction Offset value C1 Unsigned integer 8-bit													
7 to 0	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

9.3.2.2.43 WPF Color Correction Offset Coefficient Register 2 (ISU_WPF_OFST2)

This register sets the offset 2 of the color conversion correction.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits															
Address :		<ISU_base> + 01B8h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	OFST_A2[7:0]								OFST_B2[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	OFST_C2[7:0]								-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R	
Bit																	
Bit	Bit Name																
		Initial Value															
		R/W															
		Description															
31 to 24	OFST_A2[7:0]	0h	RW	Color Correction Offset value A2 Unsigned integer 8-bit													
23 to 16	OFST_B2[7:0]	0h	RW	Color Correction Offset value B2 Unsigned integer 8-bit													
15 to 8	OFST_C2[7:0]	0h	RW	Color Correction Offset value C2 Unsigned integer 8-bit													
7 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													

9.3.2.2.44 WPF Color Correction Clip Register 1 (ISU_WPF_CLP1)

This register sets the maximum and minimum values of the color conversion correction for Layer 1.

For details, see 9.3.1.5.1 Color correction.

Access Size :		32 bits														
Address :		<ISU_base> + 01BCh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLPMAX_A[7:0]								CLPMIN_A[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 8	CLPMAX_A [7:0]	0h	RW	Color Correction maximum clip level value A Set the value to CLPMAX_A ≥ CLPMIN_A
7 to 0	CLPMIN_A[7:0]	0h	RW	Color Correction minimum clip level value A Set the value to CLPMAX_A ≥ CLPMIN_A

9.3.2.2.45 WPF Color Correction Clip Register 2 (ISU_WPF_CLP2)

This register sets the maximum and minimum values of the color conversion correction for Layer 2 and Layer 3.

For details, see **9.3.1.5.1 Color correction**.

Access Size :		32 bits															
Address :		<ISU_base> + 01C0h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CLPMAX_B[7:0]									CLPMIN_B[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CLPMAX_C[7:0]									CLPMIN_C[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	Bit Name							Initial Value							R/W		Description
31 to 24	CLPMAX_B [7:0]							0h							RW		Color Correction maximum clip level value B Set the value to CLPMAX_B ≥ CLPMIN_B
23 to 16	CLPMIN_B [7:0]							0h							RW		Color Correction minimum clip level value B Set the value to CLPMAX_B ≥ CLPMIN_B
15 to 8	CLPMAX_C [7:0]							0h							RW		Color Correction maximum clip level value C Set the value to CLPMAX_C ≥ CLPMIN_C
7 to 0	CLPMIN_C [7:0]							0h							RW		Color Correction minimum clip level value C Set the value to CLPMAX_C ≥ CLPMIN_C

9.3.2.2.46 WPF Destination Image Data Swap Register (ISU_WPF_DST_DSWAP)

This register sets the data swap of the output image data.

For details, see 9.3.1.5.4 Data Swap (Endian Correction).

Access Size :		32 bits														
Address :		<ISU_base> + 01C4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	WD_SWAP[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 3	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
2 to 0	WD_SWAP [2:0]		0h		RW		Data swap for output images									

9.3.2.2.47 WPF Destination ALPHA Selection Register 1 (ISU_WPF_ALPH_SEL1)

This register sets the method of compressing the A (Alpha) value at the ARGB1555 output from 8-bit to 1-bit.
For details, see **9.3.1.5.2 Normalization**.

Access Size :		32 bits																
Address :		<ISU_base> + 01C8h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	D_A1SEL		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	D_A1THR[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW		

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	D_A1SEL	0h	RW	Setting of how to compress A (Alpha) from 8-bit to 1-bit 0b: Outputs the MSB of alpha value 1b: Selects the output according to the alpha value and threshold level Outputs 0b when Alpha ≤ D_A1THR Outputs 1b when Alpha > D_A1THR
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7 to 0	D_A1THR[7:0]	0h	RW	Threshold level value Value used to determine when D_A1SEL = 1b.

9.3.2.2.48 WPF Destination ALPHA Selection Register 2 (ISU_WPF_ALPH_SEL2)

This register sets the 8-bit A (Alpha) value at the time of ARGB8888 output.

For details, see **9.3.1.5.2 Normalization**.

Access Size :		32 bits														
Address :		<ISU_base> + 01CCh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	D_A8SEL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D_A8THR1[7:0]								D_A8THR0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	D_A8SEL	0h	RW	A (Alpha) 8-bit output selection 0b: Output Alpha value as it is 1b: Outputs the register value according to the alpha value and the threshold level (D_A8THR0 and D_A8THR1) Outputs D_ALPH2 when D_A8THR1 ≤ Alpha Outputs D_ALPH1 when D_A8THR0 < Alpha < D_A8THR1 Outputs D_ALPH0 when Alpha ≤ D_A8THR0
15 to 8	D_A8THR1 [7:0]	0h	RW	Threshold level value 1 Value used to determine which register value is output as Alpha when D_A8SEL = 1b. Set the value to D_A8THR0 < D_A8THR1.
7 to 0	D_A8THR0 [7:0]	0h	RW	Threshold level value 0 Value used to determine which register value is output as Alpha when D_A8SEL = 1b. Set the value to D_A8THR0 < D_A8THR1.

9.3.2.2.49 WPF Destination ALPHA Value Register (ISU_WPF_ALPH_VAL)

This register sets the A (Alpha) value to be output when ARGB8888 and D_A8SEL = 1.

For details, see 9.3.1.5.2 Normalization.

Access Size :		32 bits														
Address :		<ISU_base> + 01D0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	D_ALPH2[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D_ALPH1[7:0]								D_ALPH0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	D_ALPH2[7:0]	0h	RW	Conversion Alpha value 2 The value to be output when $D\_A8THR1 \leq \text{Alpha}$.
15 to 8	D_ALPH1[7:0]	0h	RW	Conversion Alpha value 1 The value to be output when $D\_A8THR0 < \text{Alpha} < D\_A8THR1$.
7 to 0	D_ALPH0[7:0]	0h	RW	Conversion Alpha value 0 The value to be output when $\text{Alpha} \leq D\_A8THR0$.

9.3.2.2.50 AXI Max Burst Length Register (ISU_AXI_BLEN)

This register sets the upper limit to the burst length of the AXI-Master.

Access Size :		32 bits														
Address :		<ISU_base> + 01F0h														
Initial Value :		000F_000Fh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	ARLEN_MAX[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	AWLEN_MAX[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19 to 16	ARLEN_MAX [3:0]	Fh	RW	Upper limit of read data burst length 3h: Up to 4-beat burst 7h: Up to 8-beat burst Fh: Up to 16-beat burst Others: Setting prohibited
15 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3 to 0	AWLEN_MAX [3:0]	Fh	RW	Upper limit of write data burst length 3h: Up to 4-beat burst 7h: Up to 8-beat burst Fh: Up to 16-beat burst Others: Setting prohibited

9.3.3 Operation

9.3.3.1 Start Processing

After setting various registers, write 1b to the start instruction bit to start.

At that time, start it together with the setting of whether to execute by the descriptor method.

Table 9.3-13 Explanation for ISU FM Frame Control Registers (ISU_FM_FRCON)

ICU_FM_FRCON		Description
Bit 16	DESON	0b: Execute with Descriptor read OFF. 1b: Execute with Descriptor read ON.
Bit 0	START	When the frame processing start is executed, the frame completion interrupt and descriptor capture completion interrupt are automatically cleared. [Write] 0b: NOP 1b: Start Frame Processing [Read] Always Zero is read.

9.3.3.1.1 Descriptor List Address

In the case of the descriptor method (DESON = 1 and START is set to 1), the access to acquire the descriptor list is performed first. The access destination is set in advance in the following register.

- FM Descriptor List Address Register 0 (ISU_FM_DL_STADDH [2: 0])
- FM Descriptor List Address Register 1 (ISU_FM_DL_STADDL [31: 0])

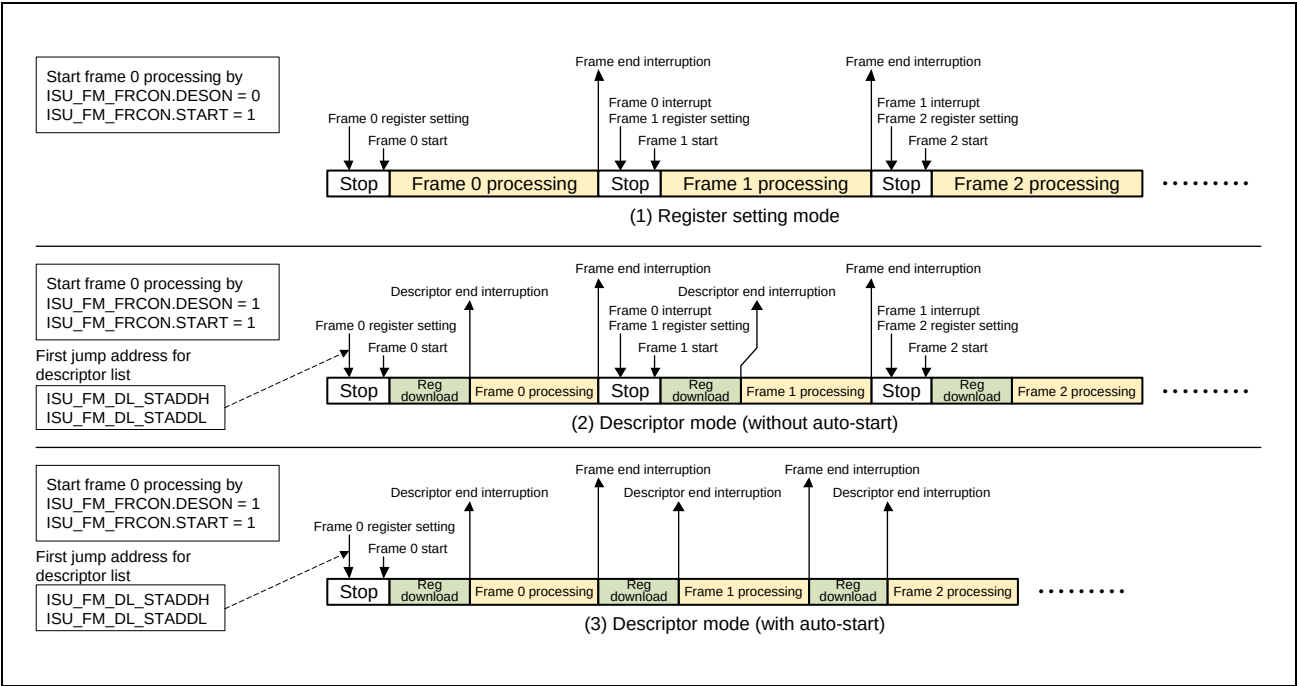


Figure 9.3-42 Relationship Between Register ISU_FM_FRCON Setting and Frame Processing Mode

9.3.3.1.2 Frame End Interruption

- When started with ISU_FM_FRCON.DESON = 0
 - An interrupt is generated when one frame processing is completed inside the ISU.
 - Completion of 1-frame processing shall meet the following conditions.
 - 1) Write the reduced data from the AXI bus and receive the response of the response channel.
 - 2) The FIFO pointer of the input image is in the initial state (empty state), and the FIFO pointer of the output image is also in the initial state (empty state).
 - 3) You have not accidentally started capturing the image of the next frame.
 - Always generate an interrupt from ISU_INT_FRE0, regardless of ISU_FM_INT_FREQ.
- When started with ISU_FM_FRCON.DESON = 1
 - One frame processing by the descriptor is completed, and an interrupt is issued in the descriptor list. An interrupt is generated when it is enabled (see the corresponding bit in the footer).
 - In the descriptor method, it advances to the next frame without doing anything after the interrupt occurs. In this case, the interrupt remains on.
 - ISU_INT_FRE0 to 3 (maximum) will be issued in sequence each time the frame ends.
 - The number of toggles for ISU_INT_FRE0 to 3 can be set in the register ISU_FM_INT_FREQ.
 - ISU_INT_FRE interrupt toggle counter is reset by writing 1 to ISU_FM_FRCON.START.

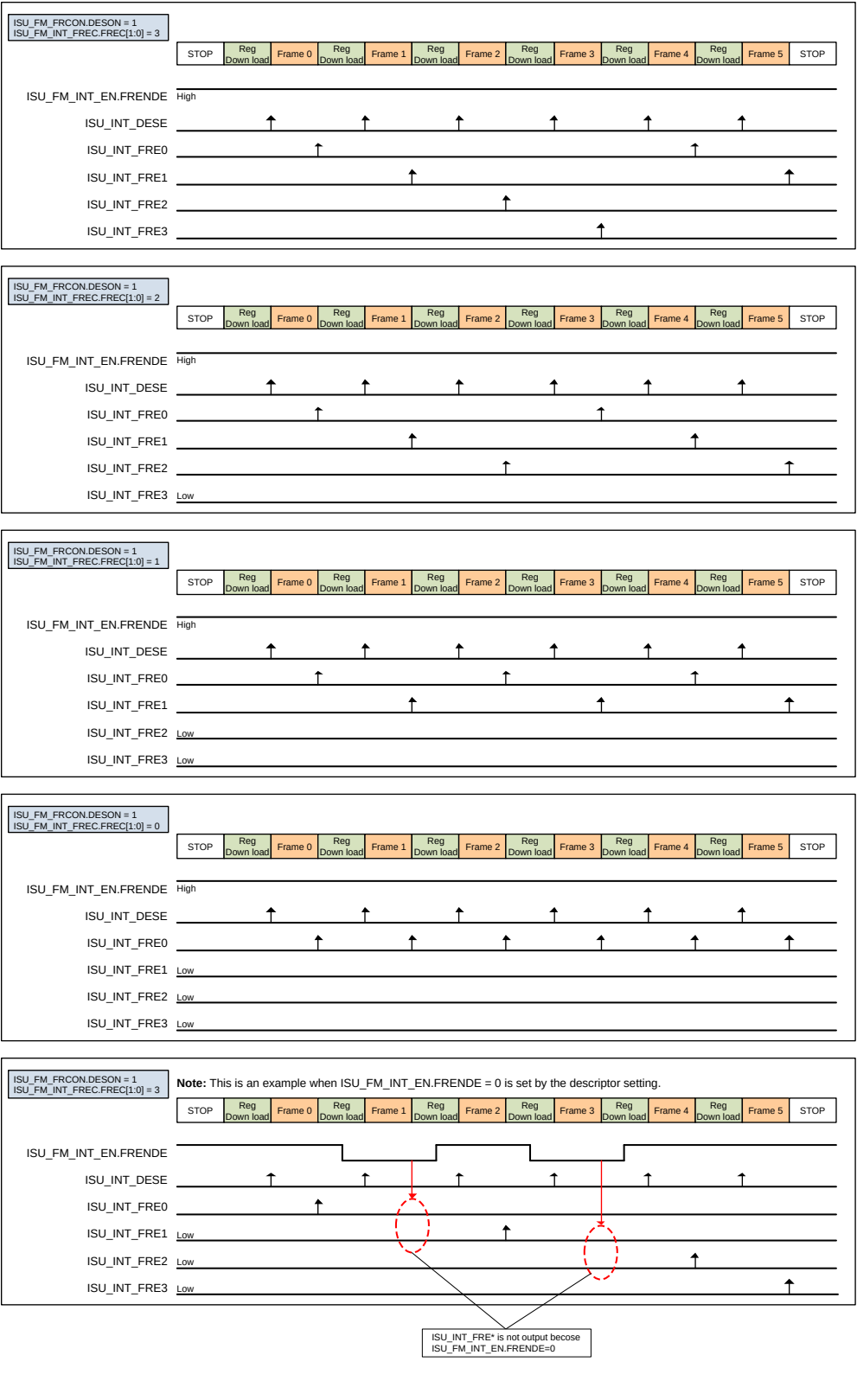


Figure 9.3-43 Relationship Between Register ISU_FREC Setting and ISU_INT_FRE0 to FRE3

9.3.3.2 Stop Processing

There are two ways to stop it.

- Normal stop
 - When started with ISU_FM_FRCON.DESON = 0b
It will end automatically when the processing of one frame is completed. After completion, the read and write FIFOs on the AXI-Master side will also be empty.
 - When started with ISU_FM_FRCON.DESON = 1b
If there is no instruction to execute the next frame in the footer of the descriptor, it will automatically end after executing one frame. After completion, the read and write FIFOs on the AXI-Master side will also be empty.

- Emergency stop

Emergency stop will be stopped by register setting from APB.

In response to the instruction from APB, the transfer to the bus in process of the AXI-Master is executed until a correct response is made, and after it is completed, an interrupt signal is issued. After that, when restarting from the system side, the ISU system reset is always applied from the outside and executed from initialization.

If the APB instructs the emergency stop again during the emergency stop process, only the first emergency stop instruction will be valid.

9.3.3.3 Interrupt

Table 9.3-14 lists the interrupt sources, registers, and interrupt signals to the ICU.

Table 9.3-14 List of Interrupts

Interrupt Signal	Function	Interrupt Source Status/clear	Interrupt Enable register	Pulse/Level	Min Pulse	Active Level	Clock
ISU_INT_FRE0	ISU Frame end interrupt 0	—	ISU_FM_INT_EN. FRENDE	Pulse	1	High	ACLK
ISU_INT_FRE1	ISU Frame end interrupt 1	—		Pulse	1	High	ACLK
ISU_INT_FRE2	ISU Frame end interrupt 2	—		Pulse	1	High	ACLK
ISU_INT_FRE3	ISU Frame end interrupt 3	—		Pulse	1	High	ACLK
ISU_INT_ERR	ISU AXI bus read error interrupt	ISU_FM_INT_STA. RRESPERR[2:0]	ISU_FM_INT_EN. AXIRXERRE	Level	—	High	PCLK
	ISU AXI bus write error interrupt	ISU_FM_INT_STA. BRESPERR[2:0]	ISU_FM_INT_EN. AXITXERRE	Level	—	High	PCLK
	ISU Descriptor List format violation interrupt	ISU_FM_INT_STA. LISTERR	ISU_FM_INT_EN. LISTERRE	Level	—	High	PCLK
ISU_INT_STOPE	ISU Stop completion interrupt	ISU_FM_INT_STA. SRSTEND	ISU_FM_INT_EN. SRSTEND	Level	—	High	PCLK
ISU_INT_DESE	ISU Descriptor footer receive interrupt	ISU_FM_INT_STA. DESEND	ISU_FM_INT_EN. DESENDE	Level	—	High	PCLK

9.3.4 Usage Notes

This section shows examples of converting image size and color format using the image scaling unit (ISU).

9.3.4.1 List of Setting Examples

Table 9.3-15 List of Examples

Item	Exec. Mode	Description
Image Size Conversion 1	Descriptor Mode	Input size: 3840 × 2160 (YUV422 UYVY)
	Next frame auto-start on	Output size: 1920 × 1080 (ARGB8888) <i>Note: ALPHA = FFh</i> Scaling Ratio 1st frame: H = 1/2, V = 1/2 2nd frame: H = 1/3, V = 1/3
Image Size Conversion 2	Descriptor Mode	Input size: 4096 × 4096 (YUV422 UYVY)
	Next frame auto-start off	Output size: 4096 × 4096 (ARGB8888) <i>Note: ALPHA = 5Ah</i> Scaling Ratio 1st frame: H = 1/1, V = 1/1 2nd frame: H = 1/2, V = 1/2
Image Size Conversion 3	Descriptor Mode	Input size: 4096 × 2160 (YUV422 UYVY)
	Next frame auto-start on	Output size: 4096 × 2160 (ARGB8888) <i>Note: ALPHA = A5h</i> Scaling Ratio 1st frame: H = 1/1, V = 1/1 2nd frame: H = 1/2, V = 1/2
Image Size Conversion 4	Register Mode	5M (2800 × 2047) → FullHD (1920 × 1080)
		Input size: 5M (2800 × 2047) (ARGB8888) Output size: FullHD (1920 × 1080) (ARGB8888) Scaling Ratio 1st frame: H = 1920/2800, V = 1080/2047
Color format Conversion 1	Register Mode	YCbCr422 → YCbCr420
		Input size: 320 × 240 (YCbCr422 UYVY) Output size: 320 × 240 (YCbCr420 NV12) Scaling Ratio 1st frame: H = 1/1, V = 1/1
Color format Conversion 2	Register Mode	YCbCr422 → ARGB8888
		Input size: 320 × 240 (YCbCr422 UYVY) Output size: 320 × 240 (ARGB8888) Scaling Ratio 1st frame: H = 1/1, V = 1/1
Color format Conversion 3	Register Mode	BGR888 → YCbCr422
		Input size: 320 × 240 (BGR888) Output size: 320 × 240 (YCbCr422 UYVY) Scaling Ratio 1st frame: H = 1/1, V = 1/1

For details on the settings, refer to each setting example.

9.3.4.2 Image Size Conversion

9.3.4.2.1 Example 1: 3840 × 2160 (YUV422 UYVY) → 1920 × 1080 (ARGB8888)

Set as follows to handle image size conversion.

Table 9.3-16 Image Size Conversion Example 1

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	3840 × 2160
Input format	YCbCr422 8-bit UYVY (Interleave)
Input start address	1st frame: Plane 0 = 0_5000_0000h 2nd frame: Plane 0 = 0_7000_0000h
Input stride	Plane 0 = 7680 (1E00h) Note: 3840 × 2 bytes/pixel
ISU setting	
Output (CROP) size	1920 × 1080
Output format	ARGB8888
Output start address	1st frame: Plane 0 = 0_6000_0000h 2nd frame: Plane 0 = 0_8000_0000h
Output stride	Plane 0 = 7680 (1E00h) Note: 1920 × 4 bytes/pixel
Scaling ratio	1st frame: H = 1/2, V = 1/2 2nd frame: H = 1/3, V = 1/3
Processing method	Descriptor method with automatic start of the next frame
Color format conversion	YCbCr422 → RGB (BT.601 SDTV)
Descriptor address	Table 1: 0_4000_0000h Table 2: 0_4000_0100h
Padding mode	PADDSEL = 1 (Register value)

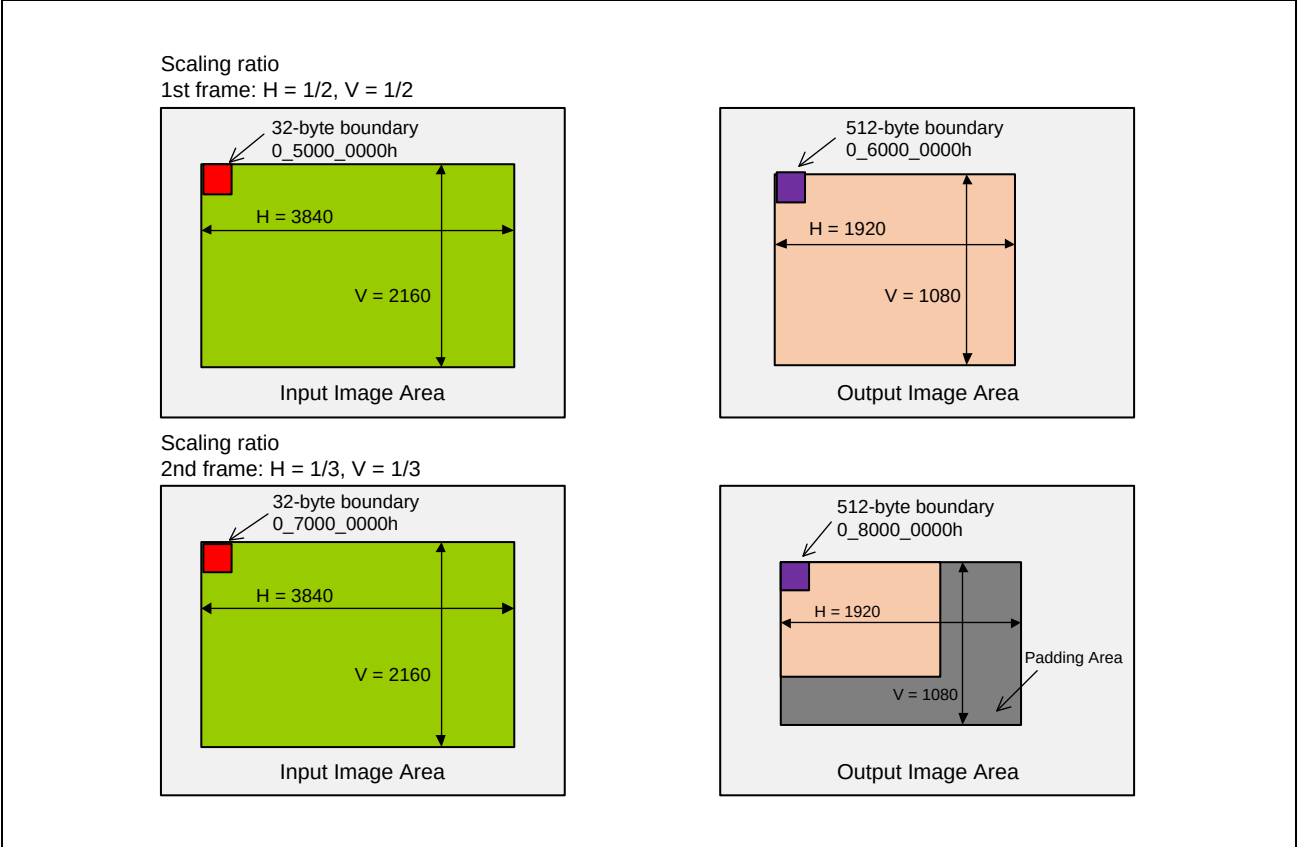


Figure 9.3-44 Image Size Conversion Example 1

The following shows the matrix operation formula for conversion and color correction.

$$\begin{bmatrix} C1' \\ C2' \\ C3' \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \cdot \begin{bmatrix} A - \text{OFS\_A1} \\ B - \text{OFS\_B1} \\ C - \text{OFS\_C1} \end{bmatrix} + \begin{bmatrix} \text{OFST\_A2} \\ \text{OFST\_B2} \\ \text{OFST\_C2} \end{bmatrix}$$

$$\begin{aligned} C1'' &= \text{MAX}(\text{CLPMIN\_A}, \text{Min}(\text{CLPMAX\_A}, C1')) \\ C2'' &= \text{MAX}(\text{CLPMIN\_B}, \text{Min}(\text{CLPMAX\_B}, C2')) \\ C3'' &= \text{MAX}(\text{CLPMIN\_C}, \text{Min}(\text{CLPMAX\_C}, C3')) \end{aligned}$$

Figure 9.3-45 Matrix Operation Formula for Conversion and Color Correction

The following shows the YCbCr422-to-RGB conversion formula (BT.601 SDTV).

$$\begin{aligned} R &= 1.164 (Y - 16) + 1.596 (Cr - 128) \\ G &= 1.164 (Y - 16) - 0.391 (Cb - 128) - 0.813 (Cr - 128) \\ B &= 1.164 (Y - 16) + 2.018 (Cb - 128) \end{aligned}$$

Note: R/G/B: 0 to 255

Perform YCbCr-to-RGB conversion with the following allocation.

$$\begin{aligned} R &= K11 (Y - \text{OFST\_A1}) + K12 (Cb - \text{OFST\_B1}) + K13 (Cr - \text{OFST\_C1}) + \text{OFST\_A2} \\ G &= K21 (Y - \text{OFST\_A1}) + K22 (Cb - \text{OFST\_B1}) + K23 (Cr - \text{OFST\_C1}) + \text{OFST\_B2} \\ B &= K31 (Y - \text{OFST\_A1}) + K32 (Cb - \text{OFST\_B1}) + K33 (Cr - \text{OFST\_C1}) + \text{OFST\_C2} \end{aligned}$$

Note: K** is set by signed fixed point Q10.

Example: $K11 = 1.164 \times 1024 = 1191.936 = 4A8h$

Rounded to the nearest whole number.

Table 9.3-17 Image Size Conversion Example 1-1 (Descriptor Table 1, 1st Frame) (1/2)

Concept	Address	Set Value	Comment
Header	4000_0000h	0000_00D8h	Total number of bytes (register address + data) = D8h
Register setting main body	4000_0004h	0000_0100h	ISU_RPF_SRC_SIZE address
	4000_0008h	0F00_0870h	Input size H = 3840, V = 2160
	4000_000Ch	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0010h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_0014h	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0018h	5000_0000h	Lower 32 bits of Plane 0 input start address 5000_0000h
	4000_001Ch	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0020h	1E00_0000h	Input stride 1E00h
	4000_0024h	0000_0118h	ISU_RPF_FMT address
	4000_0028h	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
	4000_002Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0030h	0002_0000h	Multiplication coefficient in the horizontal direction HMANT = 2h, HFRAC = 000h
	4000_0034h	0000_0144h	ISU_RS_VSCALE address
	4000_0038h	0002_0000h	Multiplication coefficient in the vertical direction VMANT = 2h, VFRAC = 000h
	4000_003Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0040h	0780_0438h	CROP size H = 1920, V = 1080
	4000_0044h	0000_0154h	ISU_RS_PADDMODE address
	4000_0048h	0000_0001h	PADDSEL = 1
	4000_004Ch	0000_0158h	ISU_RS_PADDVAL address
	4000_0050h	12EF_CDABh	VAL_L3 = 12h, VAL_L2 = EFh, VAL_L1 = CDh, VAL_L0 = ABh
	4000_0054h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0058h	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
	4000_005Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0060h	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
	4000_0064h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0068h	1E00_0000h	Output stride 1E00h
	4000_006Ch	0000_0194h	ISU_WPF_FMT address
	4000_0070h	0000_0005h	Output format ARGB8888
	4000_0074h	0000_0198h	ISU_WPF_CCOL address
	4000_0078h	0000_0003h	CMAT_SEL = 1h, CMAT_ASEL = 1h
	4000_007Ch	0000_019Ch	ISU_WPF_MUL1 address
	4000_0080h	0000_04A8h	K11 = 04A8h
	4000_0084h	0000_01A0h	ISU_WPF_MUL2 address
	4000_0088h	0000_0662h	K12 = 0000h, K13 = 0662h
	4000_008Ch	0000_01A4h	ISU_WPF_MUL3 address
	4000_0090h	0000_04A8h	K21 = 04A8h
	4000_0094h	0000_01A8h	ISU_WPF_MUL4 address
	4000_0098h	3E70_3CBFh	K22 = 3E70h, K23 = 3CBFh

Table 9.3-17 Image Size Conversion Example 1-1 (Descriptor Table 1, 1st Frame) (2/2)

Concept	Address	Set Value	Comment
Register setting main body	4000_009Ch	0000_01ACh	ISU_WPF_MUL5 address
	4000_00A0h	0000_04A8h	K31 = 04A8h
	4000_00A4h	0000_01B0h	ISU_WPF_MUL6 address
	4000_00A8h	0812_0000h	K32 = 0812h, K33 = 0000h
	4000_00ACh	0000_01B4h	ISU_WPF_OFST1 address
	4000_00B0h	1080_8000h	OFST_A1 = 10h, OFST_B1 = 80h, OFST_C1 = 80h
	4000_00B4h	0000_01B8h	ISU_WPF_OFST2 address
	4000_00B8h	0000_0000h	OFST_A2 = 00h, OFST_B2 = 00h, OFST_C2 = 00h
	4000_00BCh	0000_01BCh	ISU_WPF_CLP1 address
	4000_00C0h	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
	4000_00C4h	0000_01C0h	ISU_WPF_CLP2 address
	4000_00C8h	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h
	4000_00CCh	0000_01CCh	ISU_WPF_ALPH_SEL2 address
	4000_00D0h	0001_FE00h	D_A8SEL = 1h, D_A8THR1 = FEh, D_A8THR0 = 00h
	4000_00D4h	0000_01D0h	ISU_WPF_ALPH_VAL address
	4000_00D8h	00FF_0000h	D_ALPH2 = FFh, D_ALPH1 = 00h, D_ALPH0 = 00h
Footer	4000_00DCh	4000_0100h	Lower 32 bits of the start address in the next descriptor table 4000_0100h
	4000_00E0h	0000_0000h	Higher 3 bits of the start address in the next descriptor table 0h
	4000_00E4h	0000_0003h	Automatic start of the next frame with a frame end interrupt

Table 9.3-18 Image Size Conversion Example 1-2 (Descriptor Table 2, 2nd Frame)

Concept	Address	Set Value	Comment
Header	4000_0100h	0000_0048h	Total number of bytes (register address + data) = 48h
Register setting main body	0000_0104h	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0108h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_010Ch	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0110h	7000_0000h	Lower 32 bits of Plane 0 input start address 7000_0000h
	4000_0114h	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0118h	1E00_0000h	Plane 0 input stride 1E00h
	4000_011Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0120h	0003_0000h	Multiplication coefficient in the horizontal direction HMANT = 3h, HFRAC = 000h
	4000_0124h	0000_0144h	ISU_RS_VSCALE address
	4000_0128h	0003_0000h	Multiplication coefficient in the vertical direction VMANT = 3h, VFRAC = 000h
	4000_012Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0130h	0780_0438h	CROP size H = 1920, V = 1080
	4000_0134h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0138h	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
	4000_013Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0140h	8000_0000h	Lower 32 bits of Plane 0 output start address 8000_0000h
	4000_0144h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0148h	1E00_0000h	Output stride 1E00h
Footer	4000_014Ch	0000_0000h	Since "No automatic start of the next frame" is selected in end of the Footer, the set value is ignored.
	4000_0150h	0000_0000h	Since "No automatic start of the next frame" is selected in end of the Footer, the set value is ignored.
	4000_0154h	0000_0002h	No automatic start of the next frame with a frame end interrupt

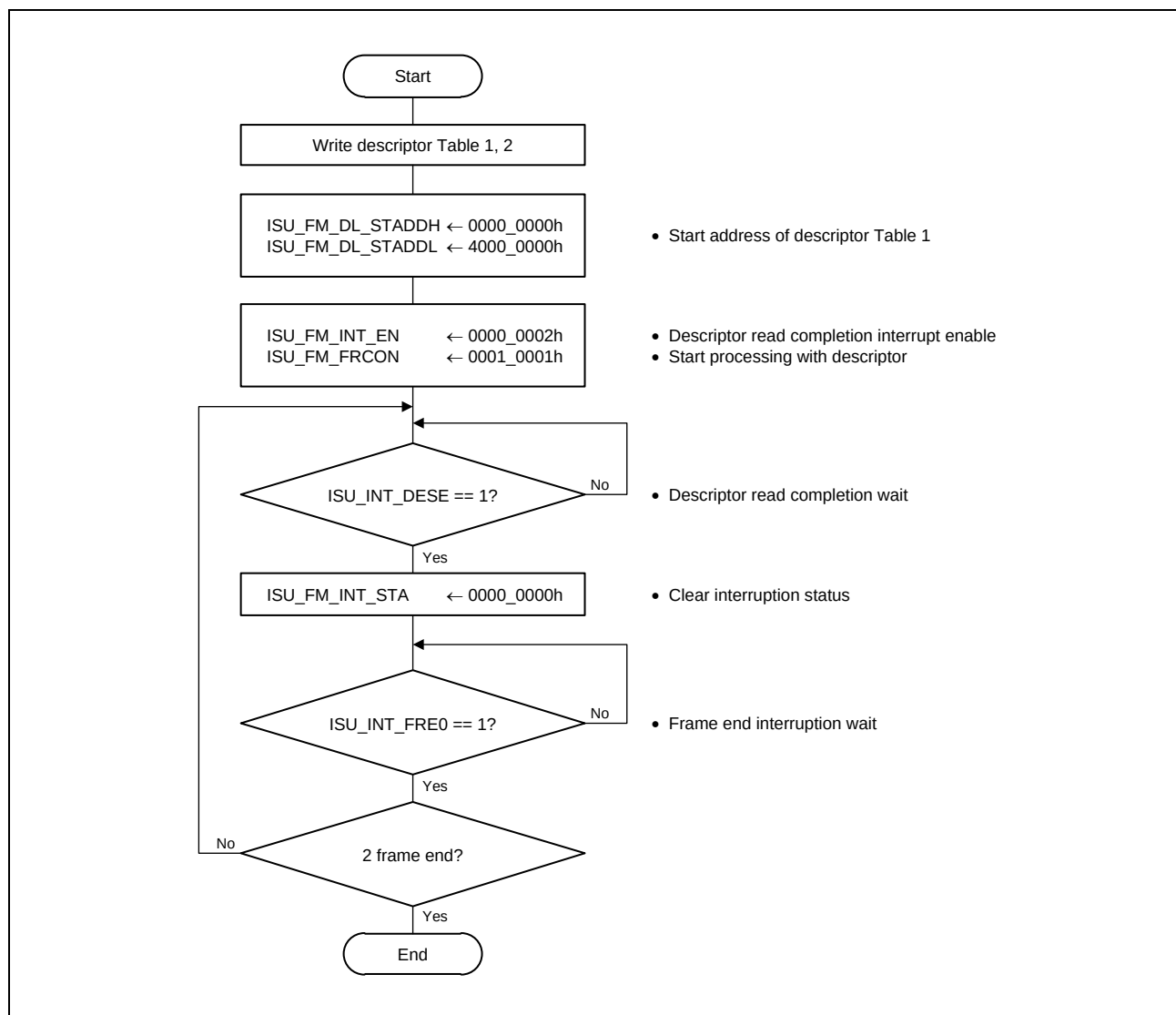


Figure 9.3-46 Flowchart of Image Size Conversion Example 1

9.3.4.2.2 Example 2: 4096 × 4096 (YUV422 UYVY) → 4096 × 4096 (ARGB8888)

Set as follows to handle image size conversion.

Table 9.3-19 Image Size Conversion Example 2

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	4096 × 4096
Input format	YCbCr422 8-bit UYVY (Interleave)
Input start address	1st frame: Plane 0 = 0_5000_0000h 2nd frame: Plane 0 = 0_7000_0000h
Input stride	Plane 0 = 8192 (2000h) <i>Note: 4096 × 2 bytes/pixel</i>
ISU setting	
Output (CROP) size	4096 × 4096
Output format	ARGB8888
Output start address	1st frame: Plane 0 = 0_6000_0000h 2nd frame: Plane 0 = 0_8000_0000h
Output stride	Plane 0 = 16384 (4000h) <i>Note: 4096 × 4 bytes /pixel</i>
Scaling ratio	1st frame: H = 1/1, V = 1/1 2nd frame: H = 1/2, V = 1/2
Processing method	Descriptor method with nonautomatic start of the next frame
Color format conversion	YCbCr422 → RGB (BT.601 SDTV)
Descriptor address	Table 1: 0_4000_0000h Table 2: 0_4000_0100h
Padding mode	PADDSEL = 1b (Register value)

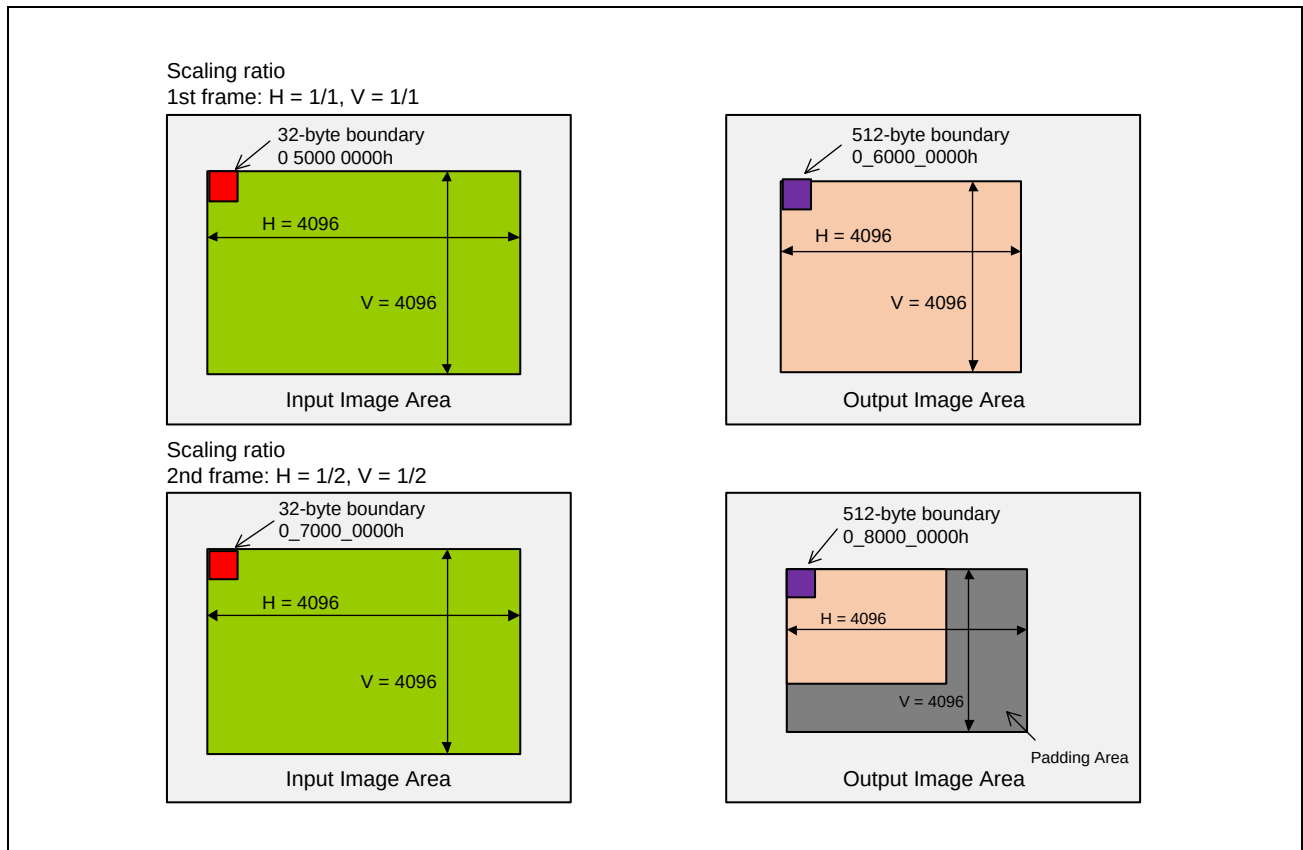


Figure 9.3-47 Image Size Conversion Example 2

Set the color conversion coefficient in the same way as in Image Size Conversion Example 1.

Table 9.3-20 Image Size Conversion Example 2-1 (Descriptor Table 1, 1st frame) (1/2)

Concept	Address	Set Value	Comment
Header	4000_0000h	0000_00D8h	Total number of bytes (register address + data) = D8h
Register setting main body	4000_0004h	0000_0100h	ISU_RPF_SRC_SIZE address
	4000_0008h	1000_1000h	Input size H = 4096, V = 4096
	4000_000Ch	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0010h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_0014h	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0018h	5000_0000h	Lower 32bits of Plane 0 input start address 5000_0000h
	4000_001Ch	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0020h	2000_0000h	Input stride 2000h
	4000_0024h	0000_0118h	ISU_RPF_FMT address
	4000_0028h	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
	4000_002Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0030h	0001_0000h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 000h
	4000_0034h	0000_0144h	ISU_RS_VSCALE address
	4000_0038h	0001_0000h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = 000h
	4000_003Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0040h	1000_1000h	CROP size H = 4096, V = 4096

Table 9.3-20 Image Size Conversion Example 2-1 (Descriptor Table 1, 1st frame) (2/2)

Concept	Address	Set Value	Comment
Register setting main body	4000_0044h	0000_0154h	ISU_RS_PADDMODE address
	4000_0048h	0000_0001h	PADDSEL = 1
	4000_004Ch	0000_0158h	ISU_RS_PADDVAL address
	4000_0050h	12EF_CDABh	VAL_L3 = 12h, VAL_L2 = EFh, VAL_L1 = CDh, VAL_L0 = ABh
	4000_0054h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0058h	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
	4000_005Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0060h	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
	4000_0064h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0068h	4000_0000h	Output stride 4000h
	4000_006Ch	0000_0194h	ISU_WPF_FMT address
	4000_0070h	0000_0005h	Output format ARGB8888
	4000_0074h	0000_0198h	ISU_WPF_CCOL address
	4000_0078h	0000_0003h	CMAT_SEL = 1h, CMAT_ASEL = 1h
	4000_007Ch	0000_019Ch	ISU_WPF_MUL1 address
	4000_0080h	0000_04A8h	K11 = 04A8h
	4000_0084h	0000_01A0h	ISU_WPF_MUL2 address
	4000_0088h	0000_0662h	K12 = 0000h, K13 = 0662h
	4000_008Ch	0000_01A4h	ISU_WPF_MUL3 address
	4000_0090h	0000_04A8h	K21 = 04A8h
	4000_0094h	0000_01A8h	ISU_WPF_MUL4 address
	4000_0098h	3E70_3CBFh	K22 = 3E70h, K23 = 3CBFh
	4000_009Ch	0000_01ACh	ISU_WPF_MUL5 address
	4000_00A0h	0000_04A8h	K31 = 04A8h
	4000_00A4h	0000_01B0h	ISU_WPF_MUL6 address
	4000_00A8h	0812_0000h	K32 = 0812h, K33 = 0000h
	4000_00ACh	0000_01B4h	ISU_WPF_OFST1 address
	4000_00B0h	1080_8000h	OFST_A1 = 10h, OFST_B1 = 80h, OFST_C1 = 80h
	4000_00B4h	0000_01B8h	ISU_WPF_OFST2 address
	4000_00B8h	0000_0000h	OFST_A2 = 00h, OFST_B2 = 00h, OFST_C2 = 00h
	4000_00BCh	0000_01BCh	ISU_WPF_CLP1 address
	4000_00C0h	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
	4000_00C4h	0000_01C0h	ISU_WPF_CLP2 address
	4000_00C8h	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h
	4000_00CCh	0000_01CCh	ISU_WPF_ALPH_SEL2 address
	4000_00D0h	0001_FE00h	D_A8SEL = 1h, D_A8THR1 = FEh, D_A8THR0 = 00h
	4000_00D4h	0000_01D0h	ISU_WPF_ALPH_VAL address
	4000_00D8h	005A_0000h	D_ALPH2 = 5Ah, D_ALPH1 = 00h, D_ALPH0 = 00h
Footer	4000_00DCh	4000_0100h	Lower 32 bits of the start address in the next descriptor table 4000_0100h
	4000_00E0h	0000_0000h	Higher 3 bits of the start address in the next descriptor table 0h
	4000_00E4h	0000_0002h	No automatic start of the next frame with a frame end interrupt

Table 9.3-21 Image Size Conversion Example 2-2 (Descriptor Table 2, 2nd frame) (1/2)

Concept	Address	Set Value	Comment
Header	4000_0100h	0000_00D8h	Total number of bytes (register address + data) = D8h
Register setting main body	4000_0104h	0000_0100h	ISU_RPF_SRC_SIZE address
	4000_0108h	1000_1000h	Input size H = 4096, V = 4096
	4000_010Ch	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0110h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_0114h	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0118h	7000_0000h	Lower 32 bits of Plane 0 input start address 7000_0000h
	4000_011Ch	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0120h	2000_0000h	Input stride 2000h
	4000_0124h	0000_0118h	ISU_RPF_FMT address
	4000_0128h	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
	4000_012Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0130h	0002_0000h	Multiplication coefficient in the horizontal direction HMANT = 2h, HFRAC = 000h
	4000_0134h	0000_0144h	ISU_RS_VSCALE address
	4000_0138h	0002_0000h	Multiplication coefficient in the vertical direction VMANT = 2h, VFRAC = 000h
	4000_013Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0140h	1000_1000h	CROP size H = 4096, V = 4096
	4000_0144h	0000_0154h	ISU_RS_PADDMODE address
	4000_0148h	0000_0001h	PADDSEL = 1
	4000_014Ch	0000_0158h	ISU_RS_PADDVAL address
	4000_0150h	12EF_CDABh	VAL_L3 = 12h, VAL_L2 = EFh, VAL_L1 = CDh, VAL_L0 = ABh
	4000_0154h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0158h	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
	4000_015Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0160h	8000_0000h	Lower 32 bits of Plane 0 output start address 8000_0000h
	4000_0164h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0168h	4000_0000h	Output stride 4000h
	4000_016Ch	0000_0194h	ISU_WPF_FMT address
	4000_0170h	0000_0005h	Output format ARGB8888
	4000_0174h	0000_0198h	ISU_WPF_CCOL address
	4000_0178h	0000_0003h	CMAT_SEL = 1h, CMAT_ASEL = 1h
	4000_017Ch	0000_019Ch	ISU_WPF_MUL1 address
	4000_0180h	0000_04A8h	K11 = 04A8h
	4000_0184h	0000_01A0h	ISU_WPF_MUL2 address
	4000_0188h	0000_0662h	K12 = 0000h, K13 = 0662h
	4000_018Ch	0000_01A4h	ISU_WPF_MUL3 address
	4000_0190h	0000_04A8h	K21 = 04A8h
	4000_0194h	0000_01A8h	ISU_WPF_MUL4 address
	4000_0198h	3E70_3CBFh	K22 = 3E70h, K23 = 3CBFh
	4000_019Ch	0000_01ACh	ISU_WPF_MUL5 address
	4000_01A0h	0000_04A8h	K31 = 04A8h
	4000_01A4h	0000_01B0h	ISU_WPF_MUL6 address
	4000_01A8h	0812_0000h	K32 = 0812h, K33 = 0000h
	4000_01ACh	0000_01B4h	ISU_WPF_OFST1 address
	4000_01B0h	1080_8000h	OFST_A1 = 10h, OFST_B1 = 80h, OFST_C1 = 80h

Table 9.3-21 Image Size Conversion Example 2-2 (Descriptor Table 2, 2nd frame) (2/2)

Concept	Address	Set Value	Comment
Register setting main body	4000_01B4h	0000_01B8h	ISU_WPF_OFST2 address
	4000_01B8h	0000_0000h	OFST_A2 = 00h, OFST_B2 = 00h, OFST_C2 = 00h
	4000_01BCh	0000_01BCh	ISU_WPF_CLP1 address
	4000_01C0h	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
	4000_01C4h	0000_01C0h	ISU_WPF_CLP2 address
	4000_01C8h	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h
	4000_01CCh	0000_01CCh	ISU_WPF_ALPH_SEL2 address
	4000_01D0h	0001_FE00h	D_A8SEL = 1h, D_A8THR1 = FEh, D_A8THR0 = 00h
	4000_01D4h	0000_01D0h	ISU_WPF_ALPH_VAL address
	4000_01D8h	005A_0000h	D_ALPH2 = 5Ah, D_ALPH1 = 00h, D_ALPH0 = 00h
Footer	4000_01DCh	0000_0000h	Lower 32 bits of the start address in the next descriptor table 0000_0000h
	4000_01E0h	0000_0000h	Higher 3 bits of the start address in the next descriptor table 0h
	4000_01E4h	0000_0002h	No automatic start of the next frame with a frame end interrupt

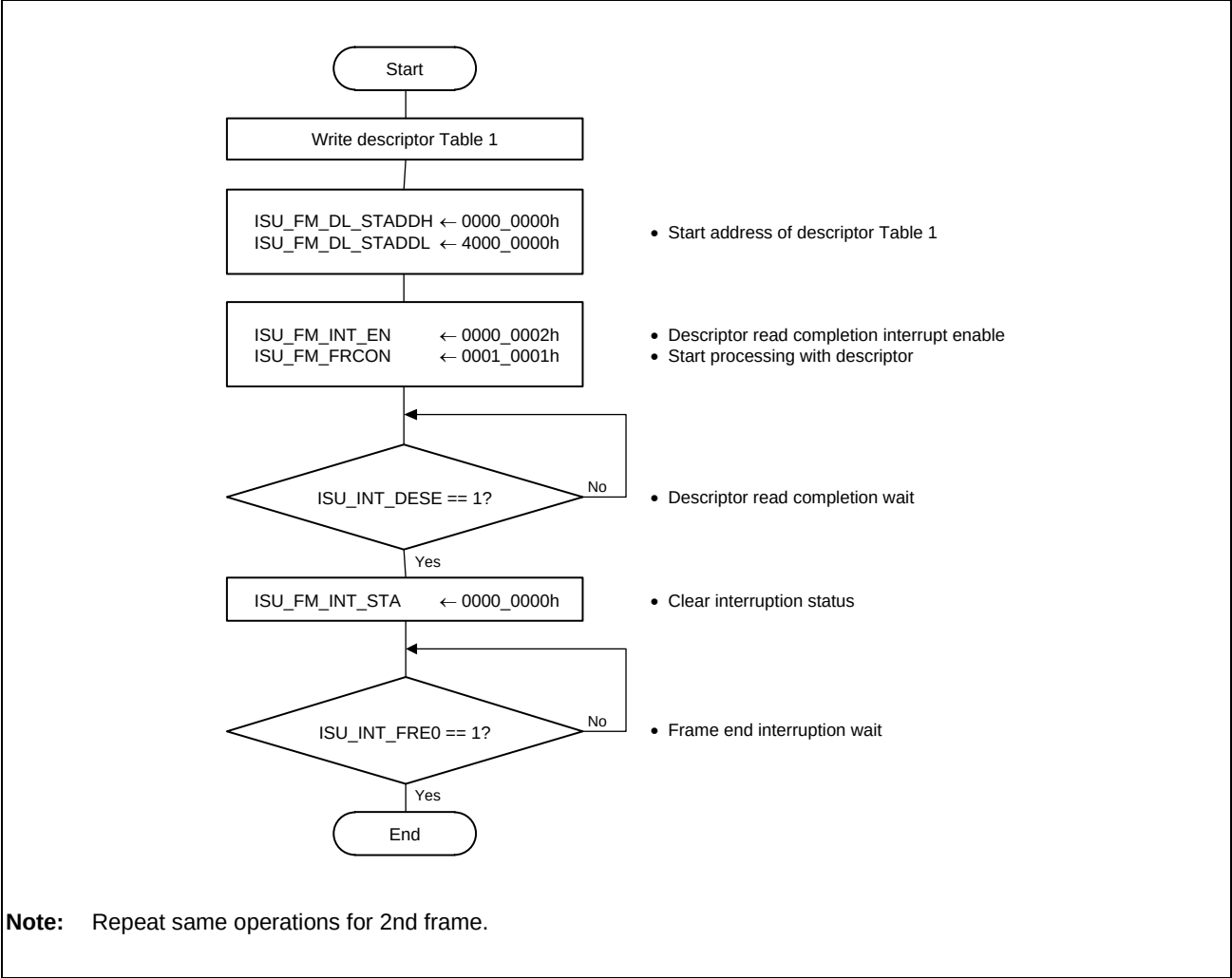


Figure 9.3-48 Flowchart of Image Size Conversion Example 2

9.3.4.2.3 Example 3: 4096 × 2160 (YUV422 UYVY) → 4096 × 2160 (ARGB8888)

Set as follows to handle image size conversion.

Table 9.3-22 Image Size Conversion Example 3

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	4096 × 2160
Input format	YCbCr422 8-bit UYVY (Interleave)
Input start address	1st frame: Plane 0 = 0_4100_0000h 2nd frame: Plane 0 = 0_4100_0000h
Input stride	Plane 0 = 8192 (2000h) <i>Note: 4096 × 2 byte/pixel</i>
ISU setting	
Output (CROP) size	4096 × 2160
Output format	ARGB8888
Output start address	1st frame: Plane 0 = 0_4300_0000h 2nd frame: Plane 0 = 2_4000_0000h
Output stride	Plane 0 = 16384 (4000h) <i>Note: 4096 × 4 byte/pixel</i>
Scaling ratio	1st frame: H = 1/1, V = 1/1 2nd frame: H = 1/2, V = 1/2
Processing method	Descriptor method with automatic start of the next frame
Color format conversion	YCbCr422 → RGB (BT.601 SDTV)
Descriptor address	Table 1: 0_4000_0000h Table 2: 0_4000_0100h
Padding mode	PADDSEL = 1b (Register value)

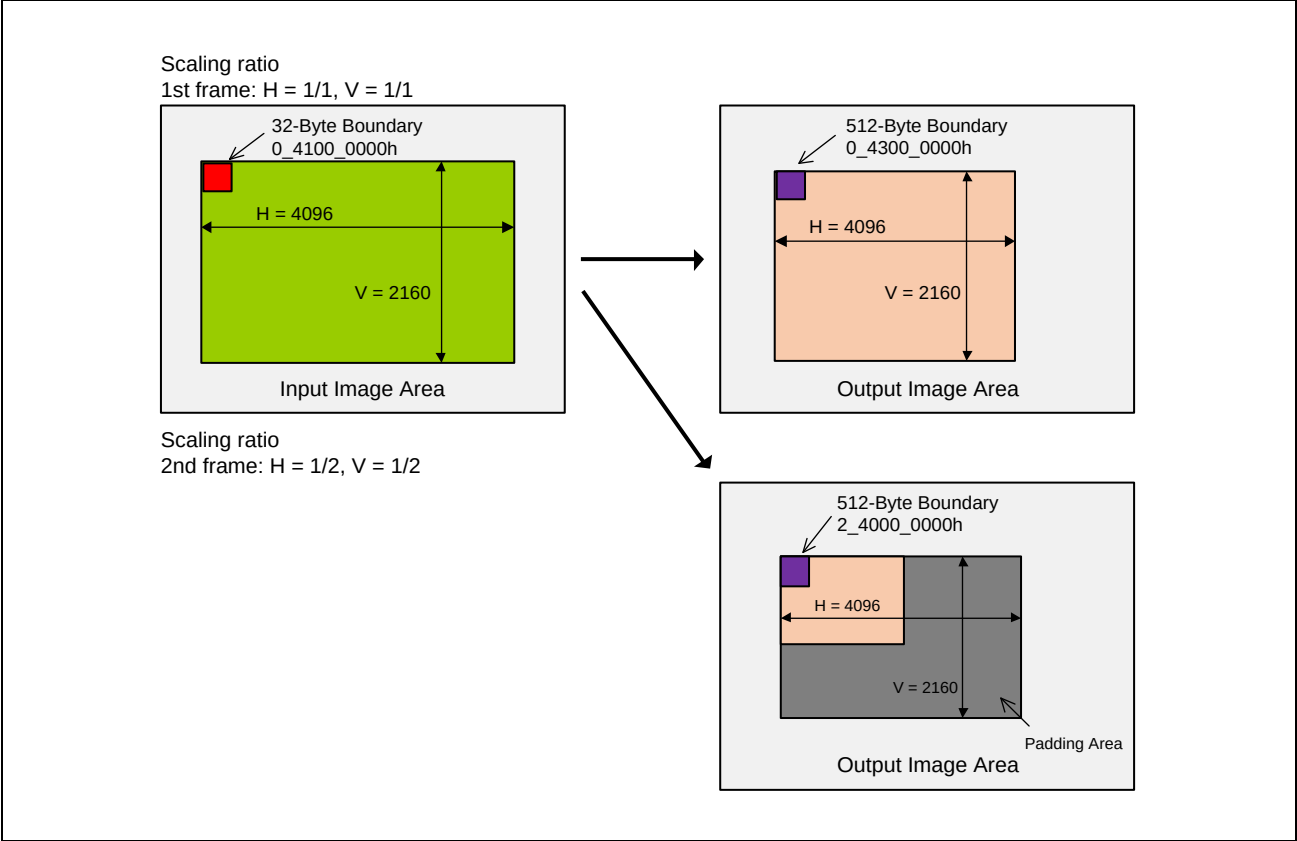


Figure 9.3-49 Image Size Conversion Example 3

Set the color conversion coefficient in the same way as in Image Size Conversion Example 1.

Table 9.3-23 Image Size Conversion Example 3-1 (Descriptor Table 1, 1st frame) (1/2)

Concept	Address	Set Value	Comment
Header	4000_0000h	0000_00D8h	Total number of bytes (register address + data) = D8h
Register setting main body	4000_0004h	0000_0100h	ISU_RPF_SRC_SIZE address
	4000_0008h	1000_0870h	Input size H = 4096, V = 2160
	4000_000Ch	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0010h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_0014h	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0018h	4100_0000h	Lower 32bits of Plane 0 input start address 4100_0000h
	4000_001Ch	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0020h	2000_0000h	Input stride 2000h
	4000_0024h	0000_0118h	ISU_RPF_FMT address
	4000_0028h	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
	4000_002Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0030h	0001_0000h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 000h
	4000_0034h	0000_0144h	ISU_RS_VSCALE address
	4000_0038h	0001_0000h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = 000h
	4000_003Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0040h	1000_0870h	CROP size H = 4096, V = 2160
	4000_0044h	0000_0154h	ISU_RS_PADDMODE address
	4000_0048h	0000_0001h	PADDSEL = 1
	4000_004Ch	0000_0158h	ISU_RS_PADDVAL address
	4000_0050h	12EF_CDABh	VAL_L3 = 12h, VAL_L2 = EFh, VAL_L1 = CDh, VAL_L0 = ABh
	4000_0054h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0058h	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
	4000_005Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0060h	4300_0000h	Lower 32 bits of Plane 0 output start address 4300_0000h
	4000_0064h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0068h	4000_0000h	Output stride 4000h
	4000_006Ch	0000_0194h	ISU_WPF_FMT address
	4000_0070h	0000_0005h	Output format ARGB8888
	4000_0074h	0000_0198h	ISU_WPF_CCOL address
	4000_0078h	0000_0003h	CMAT_SEL = 1h, CMAT_ASEL = 1h
	4000_007Ch	0000_019Ch	ISU_WPF_MUL1 address
	4000_0080h	0000_04A8h	K11 = 04A8h
	4000_0084h	0000_01A0h	ISU_WPF_MUL2 address
	4000_0088h	0000_0662h	K12 = 0000h, K13 = 0662h
	4000_008Ch	0000_01A4h	ISU_WPF_MUL3 address
	4000_0090h	0000_04A8h	K21 = 04A8h
	4000_0094h	0000_01A8h	ISU_WPF_MUL4 address
	4000_0098h	3E70_3CBFh	K22 = 3E70h, K23 = 3CBFh
	4000_009Ch	0000_01ACh	ISU_WPF_MUL5 address
	4000_00A0h	0000_04A8h	K31 = 04A8h
	4000_00A4h	0000_01B0h	ISU_WPF_MUL6 address

Table 9.3-23 Image Size Conversion Example 3-1 (Descriptor Table 1, 1st frame) (2/2)

Concept	Address	Set Value	Comment
Register setting main body	4000_00A8h	0812_0000h	K32 = 0812h, K33 = 0000h
	4000_00ACh	0000_01B4h	ISU_WPF_OFST1 address
	4000_00B0h	1080_8000h	OFST_A1 = 10h, OFST_B1 = 80h, OFST_C1 = 80h
	4000_00B4h	0000_01B8h	ISU_WPF_OFST2 address
	4000_00B8h	0000_0000h	OFST_A2 = 00h, OFST_B2 = 00h, OFST_C2 = 00h
	4000_00BCh	0000_01BCh	ISU_WPF_CLP1 address
	4000_00C0h	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
	4000_00C4h	0000_01C0h	ISU_WPF_CLP2 address
	4000_00C8h	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h
	4000_00CCh	0000_01CCh	ISU_WPF_ALPH_SEL2 address
	4000_00D0h	0001_FE00h	D_A8SEL = 1h, D_A8THR1 = FEh, D_A8THR0 = 00h
	4000_00D4h	0000_01D0h	ISU_WPF_ALPH_VAL address
	4000_00D8h	00A5_0000h	D_ALPH2 = A5h, D_ALPH1 = 00h, D_ALPH0 = 00h
Footer	4000_00DCh	4000_0100h	Lower 32 bits of the start address in the next descriptor table 4000_0100h
	4000_00E0h	0000_0000h	Higher 3 bits of the start address in the next descriptor table 0h
	4000_00E4h	0000_0003h	Automatic start of the next frame with a frame end interrupt

Table 9.3-24 Image Size Conversion Example 3-2 (Descriptor Table 2, 2nd frame)

Concept	Address	Set Value	Comment
Header	4000_0100h	0000_0048h	Total number of bytes (register address + data) = 48h
Register setting main body	4000_0104h	0000_0108h	ISU_RPF_SRC_ADDH_PL0 address
	4000_0108h	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
	4000_010Ch	0000_010Ch	ISU_RPF_SRC_ADDL_PL0 address
	4000_0110h	4100_0000h	Lower 32 bits of Plane 0 input start address 4100_0000h
	4000_0114h	0000_0104h	ISU_RPF_SRC_STRD address
	4000_0118h	2000_0000h	Plane 0 input stride 2000h
	4000_011Ch	0000_0140h	ISU_RS_HSCALE address
	4000_0120h	0002_0000h	Multiplication coefficient in the horizontal direction HMANT = 2h, HFRAC = 000h
	4000_0124h	0000_0144h	ISU_RS_VSCALE address
	4000_0128h	0002_0000h	Multiplication coefficient in the vertical direction VMANT = 2h, VFRAC = 000h
	4000_012Ch	0000_0150h	ISU_RS_OS_CROP address
	4000_0130h	1000_0870h	CROP size H = 4096, V = 2160
	4000_0134h	0000_0180h	ISU_WPF_DST_ADDH_PL0 address
	4000_0138h	0000_0002h	Higher 3 bits of Plane 0 output start address 2h
	4000_013Ch	0000_0184h	ISU_WPF_DST_ADDL_PL0 address
	4000_0140h	4000_0000h	Lower 32 bits of Plane 0 output start address 4000_0000h
	4000_0144h	0000_0190h	ISU_WPF_DST_STRD address
	4000_0148h	4000_0000h	Output stride 4000h
Footer	4000_014Ch	0000_0000h	Since "No automatic start of the next frame" is selected in end of the Footer, the set value is ignored.
	4000_0150h	0000_0000h	Since "No automatic start of the next frame" is selected in end of the Footer, the set value is ignored.
	4000_0154h	0000_0002h	No automatic start of the next frame with a frame end interrupt

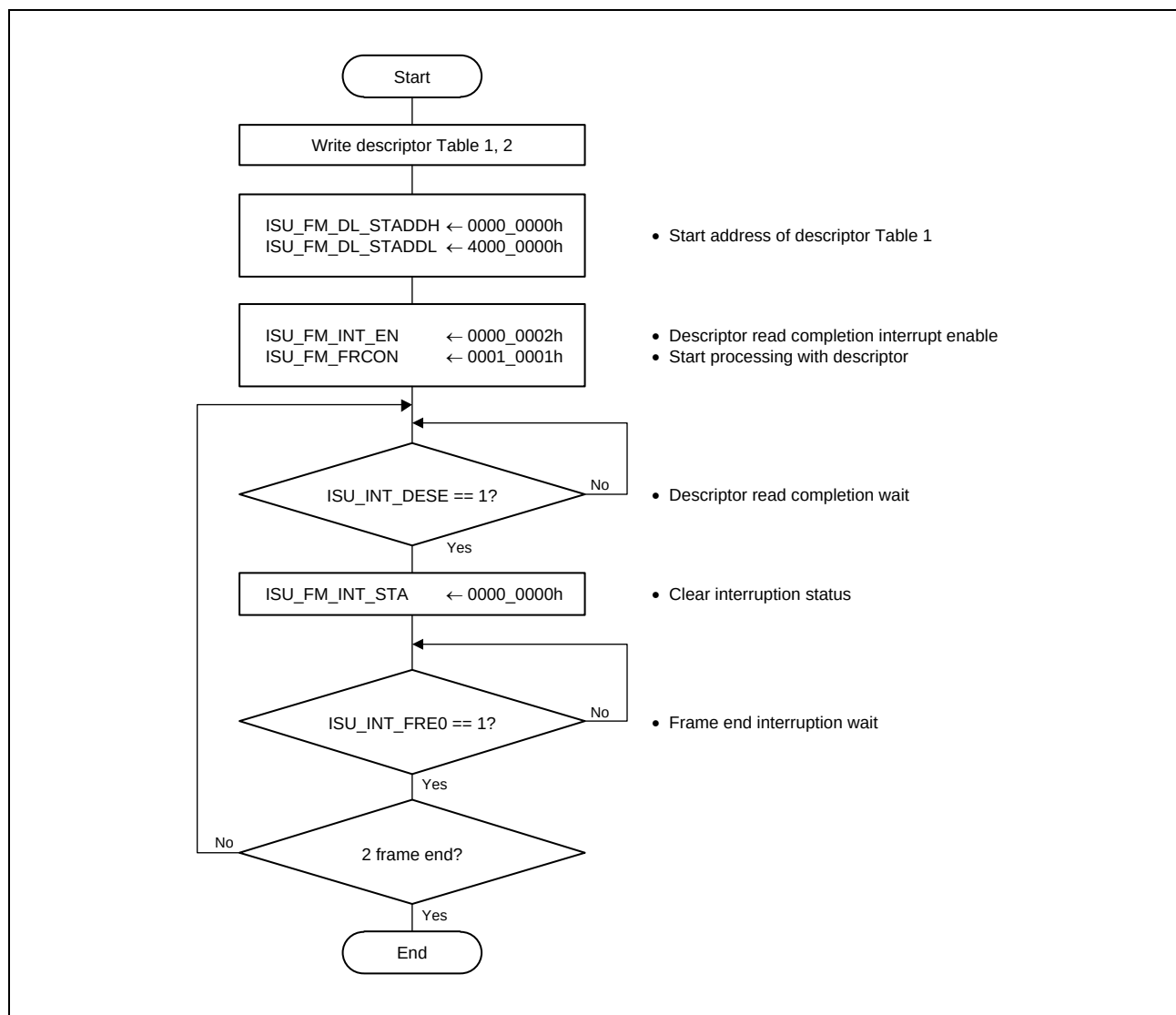


Figure 9.3-50 Flowchart of Image Size Conversion Example 3

9.3.4.2.4 5M (2800 × 2047) → Full HD (1920 × 1080)

Set as follows to handle image size conversion.

Table 9.3-25 Image Size Conversion Example 4

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	2800 × 2047
Input format	ARGB8888 * (The ISU swaps data because of the BGRA format.)
Input start address	Plane 0 = 0_5000_0000h
Input stride	Plane 0 = 2BC0h
ISU setting	
Output (CROP) size	1920 × 1080
Output format	ARGB8888
Output start address	Plane 0 = 0_6000_000h
Output stride	Plane 0 = 1E00h
Processing method	Register
Color format conversion	None
Padding mode	PADDSEL = 0 (Last pixel copy)

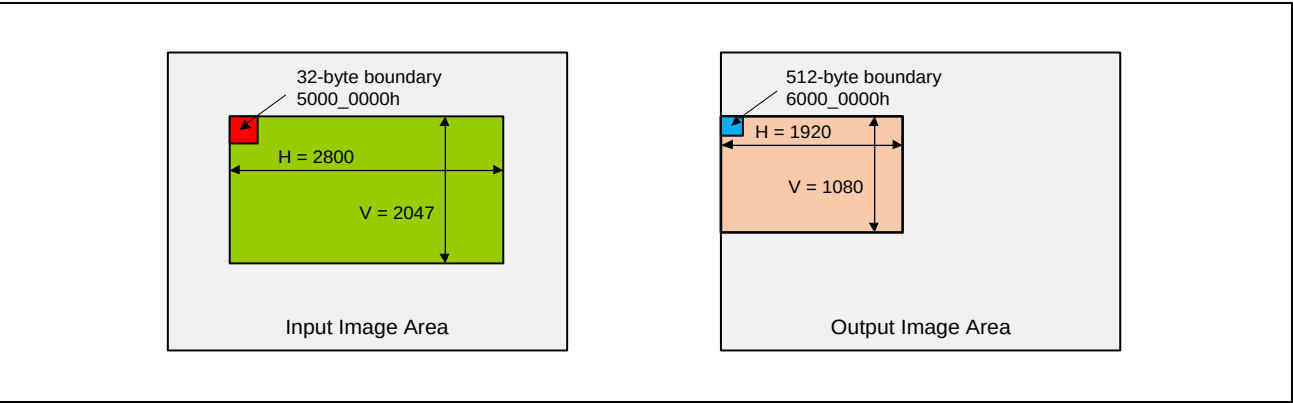


Figure 9.3-51 Image Size Conversion Example 4

Obtain the multiplication coefficient in the horizontal direction/

$2800 / 1920 = 1.458333...$

HMANT (integer part) = 1 = 1h

HFRAC (decimal part) = 0.458333×4096 (Q12) = 1877.3320 ... = 755h

Obtain the multiplication coefficient in the vertical direction.

$2047 / 1080 = 1.895370...$

VMANT (integer part) = 1 = 1h

VFRAC (decimal part) = 0.895370×4096 (Q12) = 3667.43552... = E53h

Note: HFRAC and VFRAC are set by unsigned fixed point Q12 and are truncated after the decimal point.

Table 9.3-26 Image Size Conversion Example 4 (Register Settings)

Register	Set Value	Comment
ISU_RPF_SRC_SIZE	0AF0_07FFh	Input size H = 2800, V = 2047
ISU_RPF_SRC_STRD	2BC0_0000h	Input stride PL0 = 2BC0h
ISU_RPF_SRC_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
ISU_RPF_SRC_ADDL_PL0	5000_0000h	Lower 32 bits of Plane 0 input start address 5000_0000h
ISU_RPF_FMT	0000_0005h	Input format ARGB8888
ISU_RPF_SRC_DSWAP	0000_0003h	Data swap 3
ISU_RS_HSCALE	0001_0755h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 755h
ISU_RS_VSCALE	0001_0E53h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = E35h
ISU_RS_OS_CROP	0780_0438h	CROP size H = 1920, V = 1080
ISU_WPF_DST_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
ISU_WPF_DST_ADDL_PL0	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
ISU_WPF_DST_STRD	1E00_0000h	Output stride PL0 = 1E00h
ISU_WPF_FMT	0000_0005h	Output format ARGB8888
ISU_WPF_MUL1	0000_0400h	K11 = 400h
ISU_WPF_MUL4	0400_0000h	K22 = 400h
ISU_WPF_MUL6	0000_0400h	K33 = 400h
ISU_WPF_CLP1	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
ISU_WPF_CLP2	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h

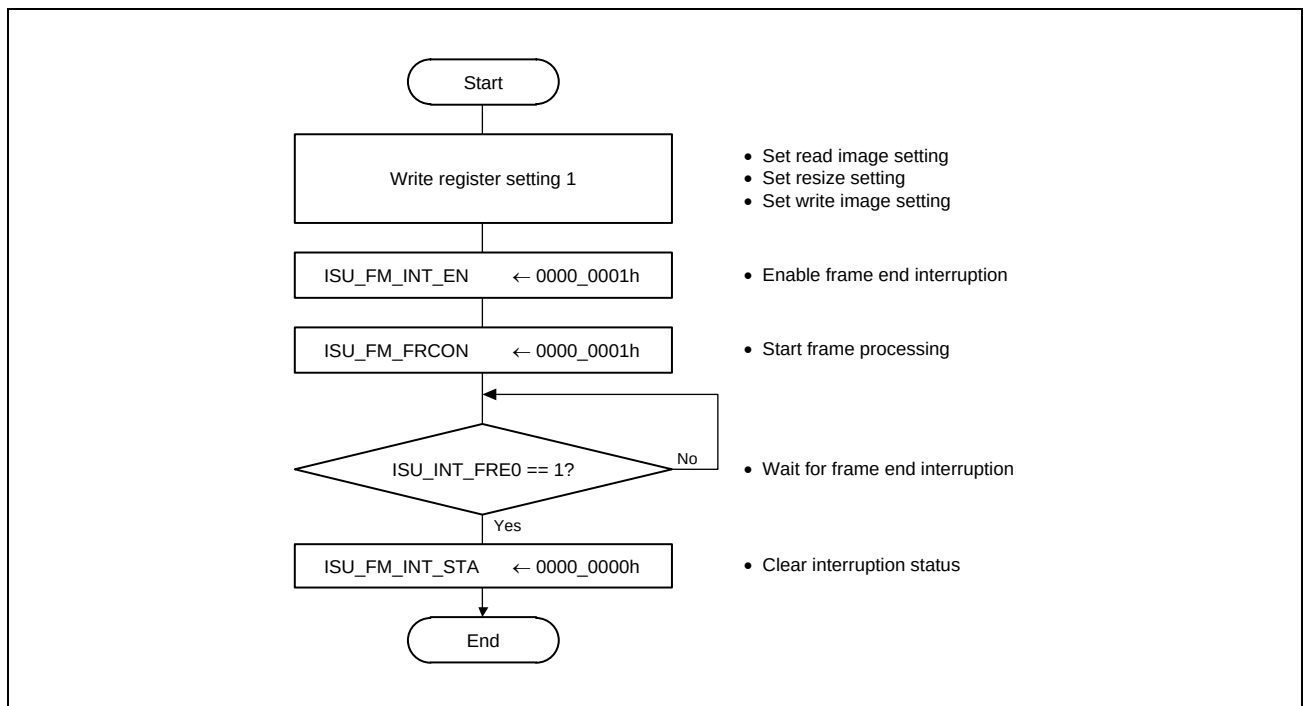


Figure 9.3-52 Flowchart of Image Size Conversion Example 4

9.3.4.3 Color Format Conversion

9.3.4.3.1 YCbCr422 → YCbCr420

Set as follows to handle color format conversion.

Table 9.3-27 Color Format Conversion Example 1

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	320 × 240
Input format	YCbCr422 8-bit UYVY (Interleave)
Input start address	Plane 0 = 0_5000_0000h
Input stride	Plane 0 = 280h
ISU setting	
Output (CROP) size	320 × 240
Output format	YCbCr420 8-bit NV12 (Semi-Planar)
Output start address	Plane 0 = 0_6000_000h Plane 1 = 0_6002_000h
Output stride	Plane 0 = 140h Plane 1 = 140h
Processing method	Register
Color format conversion	None

Table 9.3-28 Color Format Conversion Example 1 (Register Settings)

Register	Set Value	Comment
ISU_RPF_SRC_SIZE	0140_00F0h	Input size H = 320, V = 240
ISU_RPF_SRC_STRD	0280_0000h	Input stride PL0 = 0280h
ISU_RPF_SRC_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
ISU_RPF_SRC_ADDL_PL0	5000_0000h	Lower 32 bits of Plane 0 input start address 5000_0000h
ISU_RPF_FMT	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
ISU_RS_HSCALE	0001_0000h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 000h
ISU_RS_VSCALE	0001_0000h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = 000h
ISU_RS_OS_CROP	0140_00F0h	CROP size H = 320, V = 240
ISU_WPF_DST_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
ISU_WPF_DST_ADDL_PL0	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
ISU_WPF_DST_ADDH_PL1	0000_0000h	Higher 3 bits of Plane 1 output start address 0h
ISU_WPF_DST_ADDL_PL1	6002_0000h	Lower 32 bits of Plane 1 output start address 6002_0000h
ISU_WPF_DST_STRD	0140_0140h	Output stride PL0 = 0140h, PL1 = 0140h
ISU_WPF_FMT	0000_0023h	Output format YCbCr420 8-bit NV12 (Semi-Planar)
ISU_WPF_MUL1	0000_0400h	K11 = 400h
ISU_WPF_MUL4	0400_0000h	K22 = 400h
ISU_WPF_MUL6	0000_0400h	K33 = 400h
ISU_WPF_CLP1	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
ISU_WPF_CLP2	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h

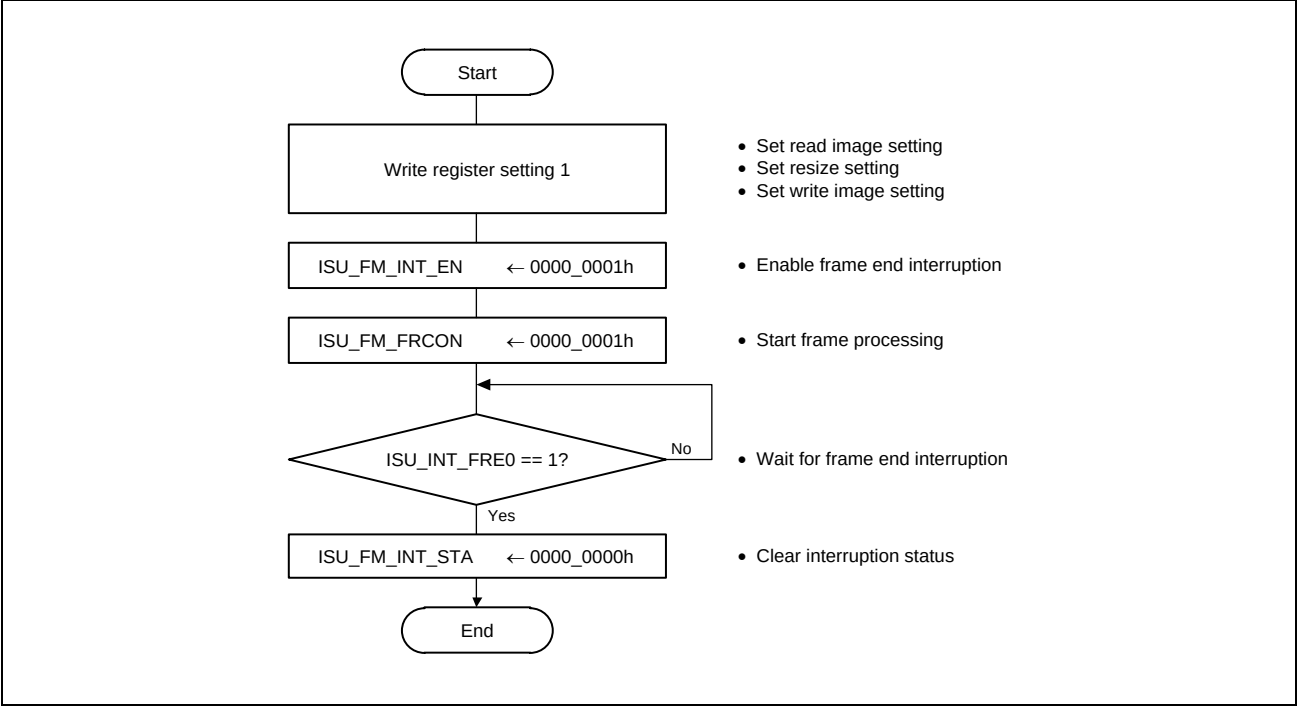


Figure 9.3-53 Flowchart of Color Format Conversion 1

9.3.4.3.2 YCbCr422 → ARGB8888

Set as follows to handle color format conversion.

Table 9.3-29 Color Format Conversion Example 2

Item	Description
Assuming that the following image is stored in DDR memory	
Input size	320 × 240
Input format	YCbCr422 8-bit UYVY (Interleave)
Input start address	Plane 0 = 0_5000_0000h
Input stride	Plane 0 = 280h
ISU setting	
Output (CROP) size	320 × 240
Output format	ARGB8888
Output start address	Plane 0 = 0_6000_000h
Output stride	Plane 0 = 500h
Processing method	Register
Color format conversion	YCbCr → RGB (BT.601 SDTV)

The following shows the matrix operation formula for conversion and color correction.

$$\begin{bmatrix} C1' \\ C2' \\ C3' \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \cdot \begin{bmatrix} A - OFS\_A1 \\ B - OFS\_B1 \\ C - OFS\_C1 \end{bmatrix} + \begin{bmatrix} OFST\_A2 \\ OFST\_B2 \\ OFST\_C2 \end{bmatrix}$$

$C1'' = \text{MAX} (CLPMIN\_A, \text{Min} (CLPMAX\_A, C1'))$ $C2'' = \text{MAX} (CLPMIN\_B, \text{Min} (CLPMAX\_B, C2'))$ $C3'' = \text{MAX} (CLPMIN\_C, \text{Min} (CLPMAX\_C, C3'))$

Figure 9.3-54 Matrix Operation Formula for Conversion and Color Correction

The following shows the YCbCr422-to-RGB conversion formula (BT.601 SDTV).

$$R = 1.164 (Y - 16) + 1.596 (Cr - 128)$$
$$G = 1.164 (Y - 16) - 0.391 (Cb - 128) - 0.813 (Cr - 128)$$
$$B = 1.164 (Y - 16) + 2.018 (Cb - 128)$$

Note: R/G/B: 0 to 255

Perform YCbCr-to-RGB conversion with the following allocation.

$$R = K11 (Y - OFST\_A1) + K12 (Cb - OFST\_B1) + K13 (Cr - OFST\_C1) + OFST\_A2$$
$$G = K21 (Y - OFST\_A1) + K22 (Cb - OFST\_B1) + K23 (Cr - OFST\_C1) + OFST\_B2$$
$$B = K31 (Y - OFST\_A1) + K32 (Cb - OFST\_B1) + K33 (Cr - OFST\_C1) + OFST\_C2$$

Note: K** is set by signed fixed point Q10.
Example: K11 = 1.164 × 1024 = 1191.936 = 4A8h
Rounded to the nearest whole number.

Table 9.3-30 Color Format Conversion Example 2 (Register Settings)

Register	Set Value	Comment
ISU_RPF_SRC_SIZE	0140_00F0h	Input size H = 320, V = 240
ISU_RPF_SRC_STRD	0280_0000h	Input stride PL0 = 0280h
ISU_RPF_SRC_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
ISU_RPF_SRC_ADDL_PL0	5000_0000h	Lower 32 bits of Plane 0 input start address 5000_0000h
ISU_RPF_FMT	0000_0020h	Input format YCbCr422 8-bit UYVY (Interleave)
ISU_RS_HSCALE	0001_0000h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 000h
ISU_RS_VSCALE	0001_0000h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = 000h
ISU_RS_OS_CROP	0140_00F0h	CROP size H = 320, V = 240
ISU_WPF_DST_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
ISU_WPF_DST_ADDL_PL0	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
ISU_WPF_DST_STRD	0500_0000h	Output stride PL0 = 0500h
ISU_WPF_FMT	0000_0005h	Output format ARGB8888
ISU_WPF_CCOL	0000_0002h	With color conversion correction
ISU_WPF_MUL1	0000_04A8h	K11 = 04A8h
ISU_WPF_MUL2	0000_0662h	K12 = 0000h, K13 = 0662h
ISU_WPF_MUL3	0000_04A8h	K21 = 04A8h
ISU_WPF_MUL4	3E70_3CBFh	K22 = 3E70h, K23 = 3CBFh
ISU_WPF_MUL5	0000_04A8h	K31 = 04A8h
ISU_WPF_MUL6	0812_0000h	K32 = 0812h, K33 = 0000h
ISU_WPF_OFST1	1080_8000h	OFST_A1 = 10h, B1 = 80h, C1 = 80h
ISU_WPF_OFST2	0000_0000h	OFST_A2 = 00h, B2 = 00h, C2 = 00h
ISU_WPF_CLP1	0000_FF00h	CLP (MAX/MIN)_A = FFh, 00h
ISU_WPF_CLP2	FF00_FF00h	CLP (MAX/MIN)_B = FFh, 00h, C = FFh, 00h

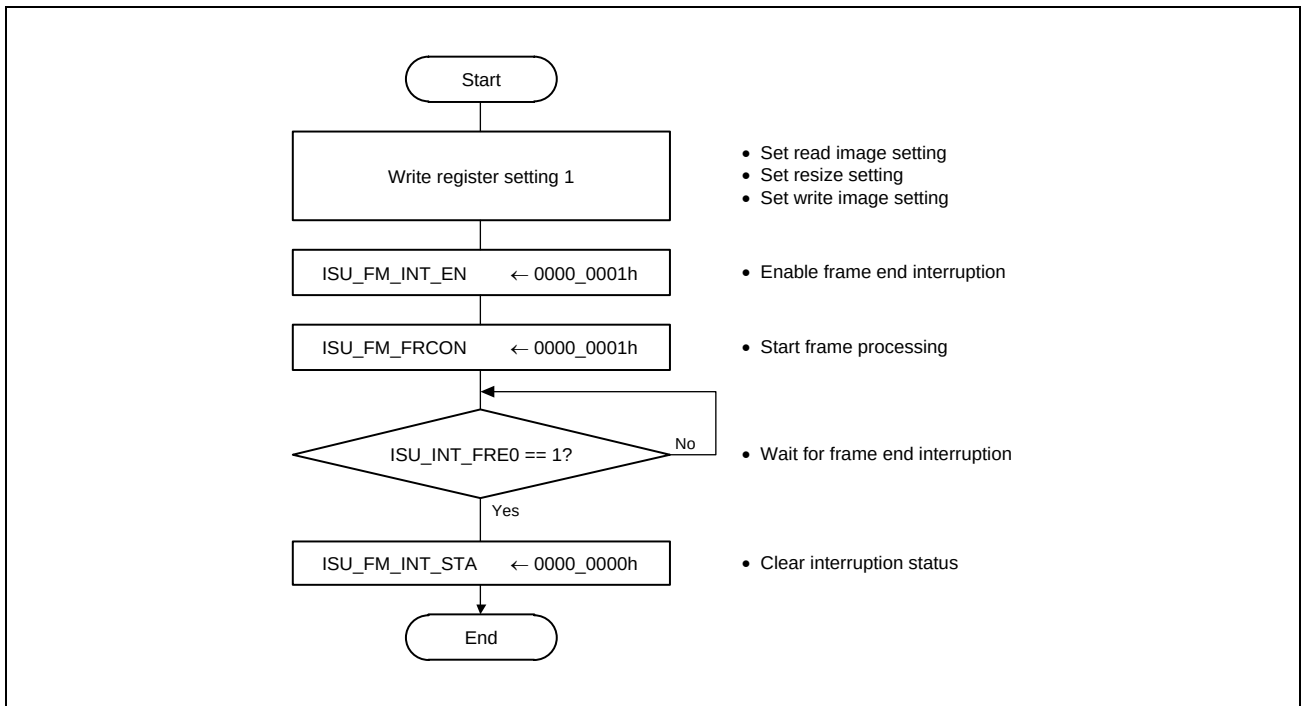


Figure 9.3-55 Flowchart of Color Format Conversion 2

9.3.4.3.3 BGR888 → YCbCr422

Set as follows to handle color format conversion.

Table 9.3-31 Color Format Conversion Example 3

Item	Description
Assuming that the following image is input from the CRU	
Input size	320 × 240
Input format	BGR888
Input start address	Plane 0 = 0_5000_0000h
Input stride	Plane 0 = 03C0h
ISU setting	
Output (CROP) size	320 × 240
Output format	YCbCr422 8-bit UYVY (Interleave)
Output start address	Plane 0 = 0_6000_000h
Output stride	Plane 0 = 0280h
Processing method	Register
Color format conversion	RGB → YCbCr (BT.601 SDTV)

The following shows the matrix operation formula for conversion and color correction.

$$\begin{bmatrix} C1' \\ C2' \\ C3' \end{bmatrix} = \begin{bmatrix} K11 & K12 & K13 \\ K21 & K22 & K23 \\ K31 & K32 & K33 \end{bmatrix} \cdot \begin{bmatrix} A - OFS\_A1 \\ B - OFS\_B1 \\ C - OFS\_C1 \end{bmatrix} + \begin{bmatrix} OFST\_A2 \\ OFST\_B2 \\ OFST\_C2 \end{bmatrix}$$

$C1'' = \text{MAX} (CLPMIN\_A, \text{Min} (CLPMAX\_A, C1'))$ $C2'' = \text{MAX} (CLPMIN\_B, \text{Min} (CLPMAX\_B, C2'))$ $C3'' = \text{MAX} (CLPMIN\_C, \text{Min} (CLPMAX\_C, C3'))$

Figure 9.3-56 Matrix Operation Formula for Conversion and Color Correction

The following shows the RGB-to-YCbCr422 conversion formula (BT.601 SDTV).

$Y = 0.257R + 0.504G + 0.098B + 16$ $Cb = -0.148R - 0.291G + 0.439B + 128$ $Cr = 0.439R - 0.368G - 0.071B + 128$

Note: Brightness: 16 to 235
Color difference: 16 to 240

Perform RGB-to-YCbCr conversion with the following allocation.

$Y = K11 (R - OFST\_A1) + K12 (G - OFST\_B1) + K13 (B - OFST\_C1) + OFST\_A2$ $Cb = K21 (R - OFST\_A1) + K22 (G - OFST\_B1) + K23 (B - OFST\_C1) + OFST\_B2$ $Cr = K31 (R - OFST\_A1) + K32 (G - OFST\_B1) + K33 (B - OFST\_C1) + OFST\_C2$

Note: K** is set by signed fixed point Q10.
Example: K11 = 0.257 × 1024 = 263.168 = 107h
Rounded to the nearest whole number.

Table 9.3-32 Color Format Conversion Example 3 (Register Settings)

Register	Set Value	Comment
ISU_RPF_SRC_SIZE	0140_00F0h	Input size H = 320, V = 240
ISU_RPF_SRC_STRD	03C0_0000h	Input stride PL0 = 03C0h
ISU_RPF_SRC_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 input start address 0h
ISU_RPF_SRC_ADDL_PL0	5000_0000h	Lower 32 bits of Plane 0 input start address 5000_0000h
ISU_RPF_FMT	0000_0004h	Input format BGR888
ISU_RS_HSCALE	0001_0000h	Multiplication coefficient in the horizontal direction HMANT = 1h, HFRAC = 000h
ISU_RS_VSCALE	0001_0000h	Multiplication coefficient in the vertical direction VMANT = 1h, VFRAC = 000h
ISU_RS_OS_CROP	0140_00F0h	CROP size H = 320, V = 240
ISU_WPF_DST_ADDH_PL0	0000_0000h	Higher 3 bits of Plane 0 output start address 0h
ISU_WPF_DST_ADDL_PL0	6000_0000h	Lower 32 bits of Plane 0 output start address 6000_0000h
ISU_WPF_DST_STRD	0280_0000h	Output stride PL0 = 0280h
ISU_WPF_FMT	0000_0020h	Output format YCbCr422 8-bit UYVY (Interleave)
ISU_WPF_CCOL	0000_0002h	With color conversion correction
ISU_WPF_MUL1	0000_0107h	K11 = 0107h
ISU_WPF_MUL2	0204_0064h	K12 = 0204h, K13 = 0064h
ISU_WPF_MUL3	0000_3F68h	K21 = 3F68h
ISU_WPF_MUL4	3ED6_01C2h	K22 = 3ED6h, K23 = 01C2h
ISU_WPF_MUL5	0000_01C2h	K31 = 01C2h
ISU_WPF_MUL6	3E87_3FB7h	K32 = 3E87h, K33 = 3FB7h
ISU_WPF_OFST1	0000_0000h	OFST_A1 = 00h, B1 = 00h, C1 = 00h
ISU_WPF_OFST2	1080_8000h	OFST_A2 = 10h, B2 = 80h, C2 = 80h
ISU_WPF_CLP1	0000_EB10h	CLP (MAX/MIN)_A = EBh, 10h
ISU_WPF_CLP2	F010_F010h	CLP (MAX/MIN)_B = F0h, 10h, C = F0h, 10h

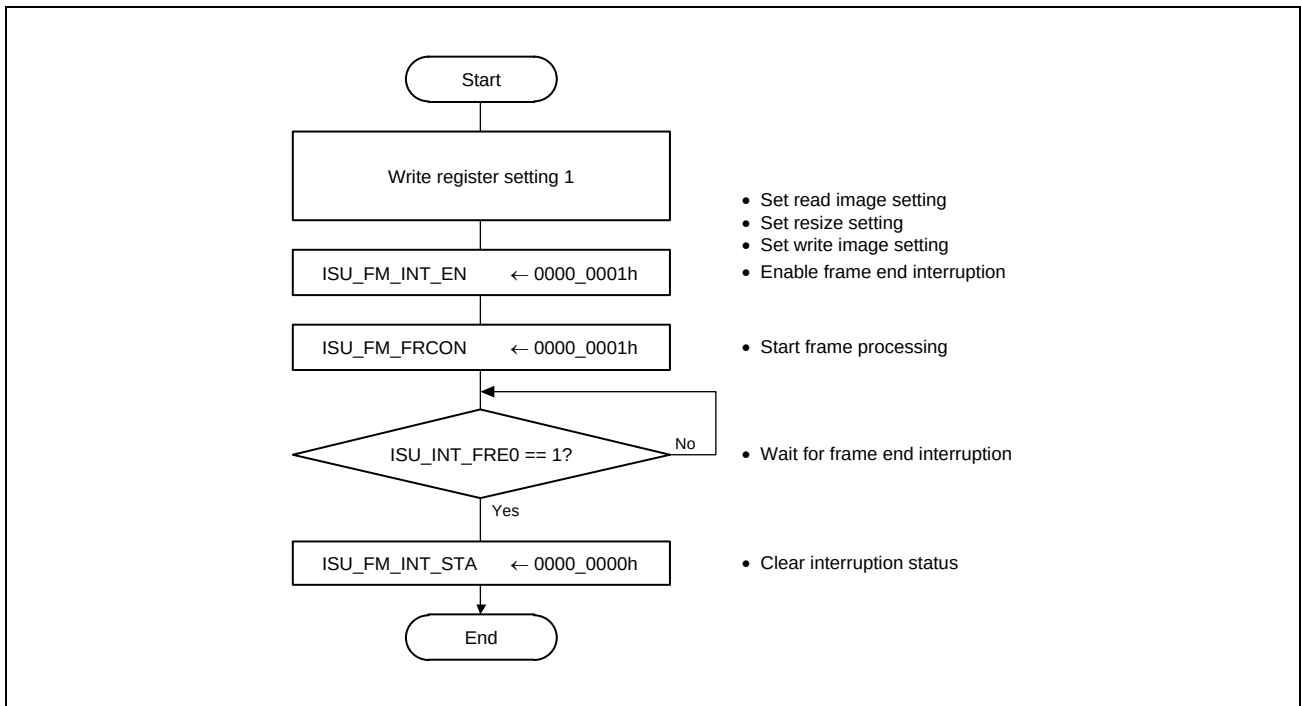


Figure 9.3-57 Flowchart of Color Format Conversion Example 3

SECTION 9 IMAGE

9.4 LCD Controller (LCDC)

9.4.1 Overview

This chapter describes the features of the LCDC unit of this LSI.

This unit is an LCD controller and it is composed of Frame Compression Processor (FCPVD), Video Signal Processor (VSPD), and Display Unit (DU).

This LSI has one this unit and it is connected to DSI. The video data, which is output by this unit, is input to DSI, and DSI converts to MIPI DSI (Display Serial Interface) and output.

9.4.1.1 Features

The following is key features of this unit.

- FCPVD
 - Supports out-of-order for the whole outstanding transactions
 - Read linear addressing image data
 - Read display list data
 - Write image data
- VSPD
 - Supports various data formats and conversion
 - ◆ Supports YCbCr444/422/420, RGB, α RGB, α plane
 - ◆ Color space conversion and changes to the number of colors by dithering
 - ◆ Color keying
 - ◆ Supports combination between pixel alpha and global alpha
 - ◆ Supports generating pre multiplied alpha
 - Video processing
 - ◆ Blending of two picture layers and raster operations (ROPs)
 - ◆ Clipping
 - ◆ 1D look up table
 - ◆ Vertical flipping in case of output to memory
 - Direct connection to display module
 - ◆ Supports up to 1920 pixels in horizontal direction
 - ◆ Writing back image data which is transferred to Display Unit (DU) to memory

- DU

- Display timing master
 - ◆ Generates video timings (Front porch, Back porch, Sync active, Active video area)
 - ◆ Selects the polarity of output DCLK, HSYNC, VSYNC, and DE
- Supports Progressive (Non-interlace)
- Not supports Interlace
- Input data format (from VSPD): RGB888, RGB666 (not supports dithering of RGB565)
- Output data format: same as Input data format
- Supports up to 187.5 MHz of the pixel clock frequency
- Examples of the supported resolutions is below:
 - ◆ RGB888, 1920 pixels × 1200 lines, 60 fps
 - ◆ RGB888, 1920 pixels × 1080 lines, 60 fps (FHD)
 - ◆ RGB888, 1280 pixels × 1024 lines, 120 fps

9.4.1.2 Block Diagram

Figure 9.4-1 shows a block diagram.

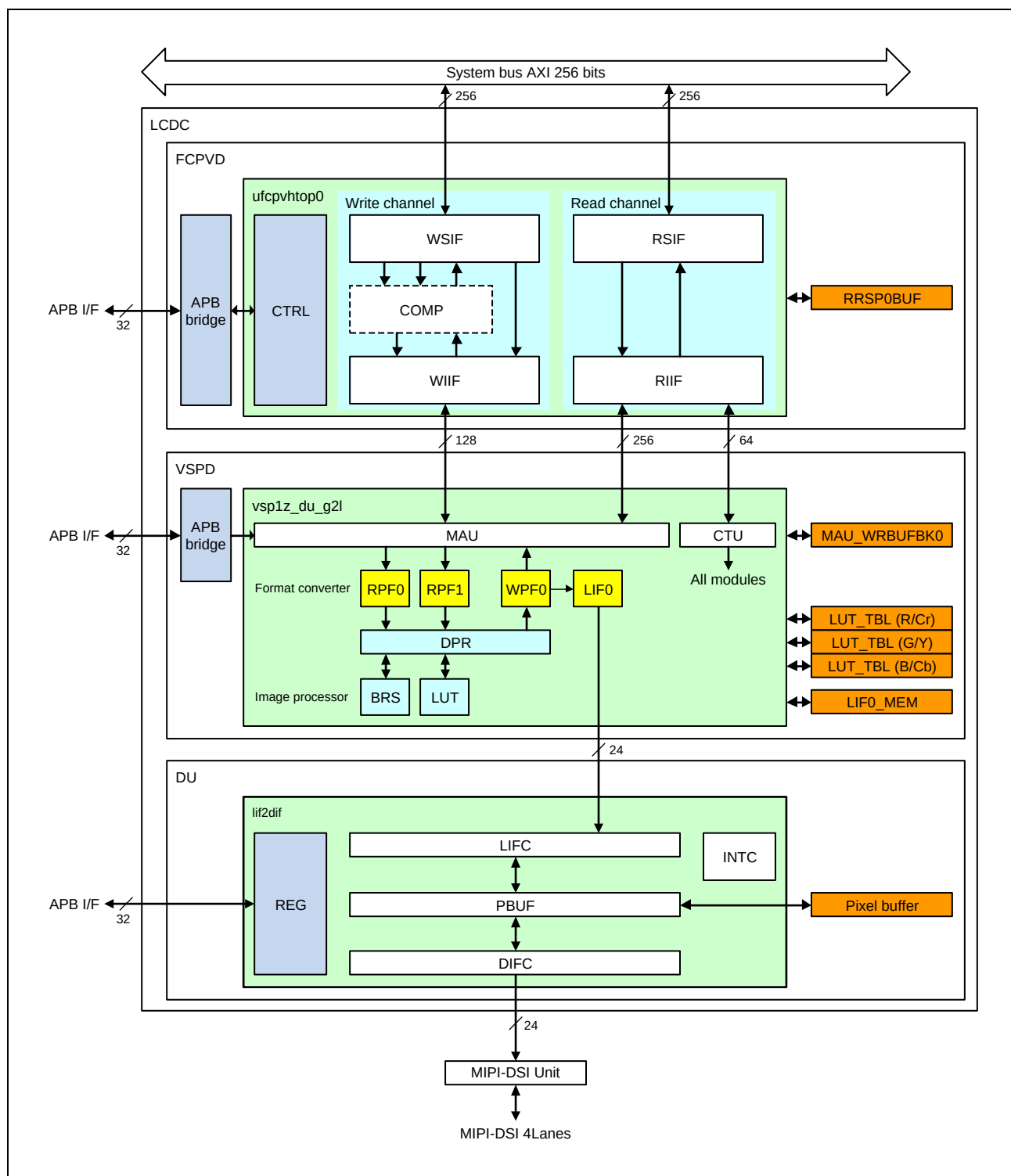


Figure 9.4-1 Block Diagram

9.4.1.2.1 FCPVD

FCPVD has the following sub modules.

Table 9.4-1 Sub Modules in FCPVD

Abbreviation	Description
CTRL	FCPVD controller
WIIF	VSPD write channel interface
WSIF	System AXI write channel interface
RIIF	VSPD read channel interface
RSIF	System AXI read channel interface
COMP	Compression module (This is a dummy module)

9.4.1.2.2 VSPD

VSPD has the following sub modules.

(1) Memory Access Unit (MAU)

The VSPD applies processing to the image data stored in the external memory and writes the resultant data back to the external memory. The data transfer between the external memory and VSPD necessary for this operation is done by the MAU, which works as the bus master, according to the register settings. The MAU executes this data transfer between the external memory and VSPD.

(2) Command Transfer Unit (CTU)

The VSPD can directly read register parameters for image processing by display lists stored in external memory. The CTU module is a bus interface and controls display lists when the CTU reads display lists as a bus master.

(3) Read Pixel Formatter (RPF)

The RPF reads image data from the external memory through the MAU, unpacks data according to the specified format, converts the color space, converts the number of colors, executes color keying, ROP operation, Multiply-alpha and OSD processing, and outputs the resultant data to the DPR. The RPF has an input format unpacking unit, a 1-bit mask generator, a raster operation unit (ROP unit), a color keying unit, a color space converter and multiply-alpha. **Figure 9.4-2** show the processing flow in the RPF.

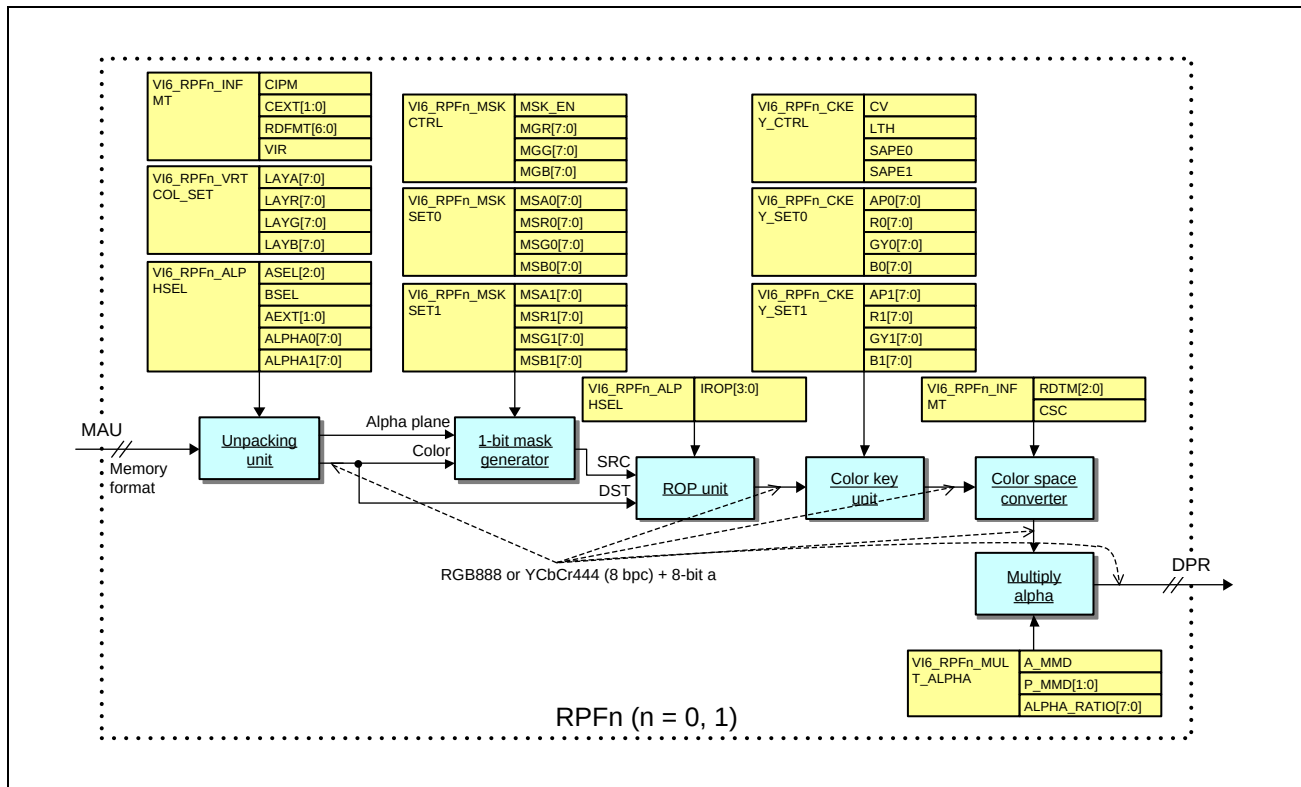


Figure 9.4-2 RPF Processing Flow

The input format unpacking unit expands the image data input from the MAU into the image format for internal processing (YCbCr444 8bpc or RGB888 8bpc), and the 1-bit mask generator generates a 1-bit image mask from the image data expanded through the unpacking unit. Alternately, a 1-bit image mask can be generated from the alpha plane that is different from the picture plane read through the MAU.

The raster operation unit (ROP unit) executes raster operation between the data from the 1-bit mask generator and the image data expanded from the input format, and the color keying unit applies color replacement and specifies the transparent color for the image data input from the ROP unit. The color space converter converts the color space (RGB-YCbCr) of the image data input from the color keying unit as necessary. The multiply-alpha unit multiplies pixel-alpha by specified alpha or/and multiplies image data by pixel-alpha or/and specified alpha.

The VSPD provides maximum two RPF modules (RPF0 to RPF1).

(4) Data Path Router (DPR)

The DPR controls the data paths among RPFs, function modules, and WPFs. The DPR selects one of the images input from RPFs, outputs it to a function module (BRS or LUT), and selects one of WPFs as the destination where the image data processed in the function module will be output. Before output to the WPF, the output from each function module can be input to another function module, which enables multiple image processing functions to be executed continuously without involving the external memory.

(5) Look Up Table (LUT)

This is a 1D-LUT that converts each of three color components by using a lookup table. The LUT is connected to the DPR and can be used for gamma correction, negative-positive conversion, posterization, and binarization through desired tone curve settings.

The LUT supports local area processing for Y component. The LUT can apply 1D look up table for Y component per area where a image is divided into multiple area specified by registers and command list.

(6) Blend ROP Sub Unit (BRS)

The BRS is a module connected to the DPR, which executes the image blending processing and ROP operation. The BRS has two blend/ROP operation units (blend/ROP unit m, m = A to B), a blend/ROP input switch (SEL) for selecting the input to these operation units, and a divider for normalization (div unit).

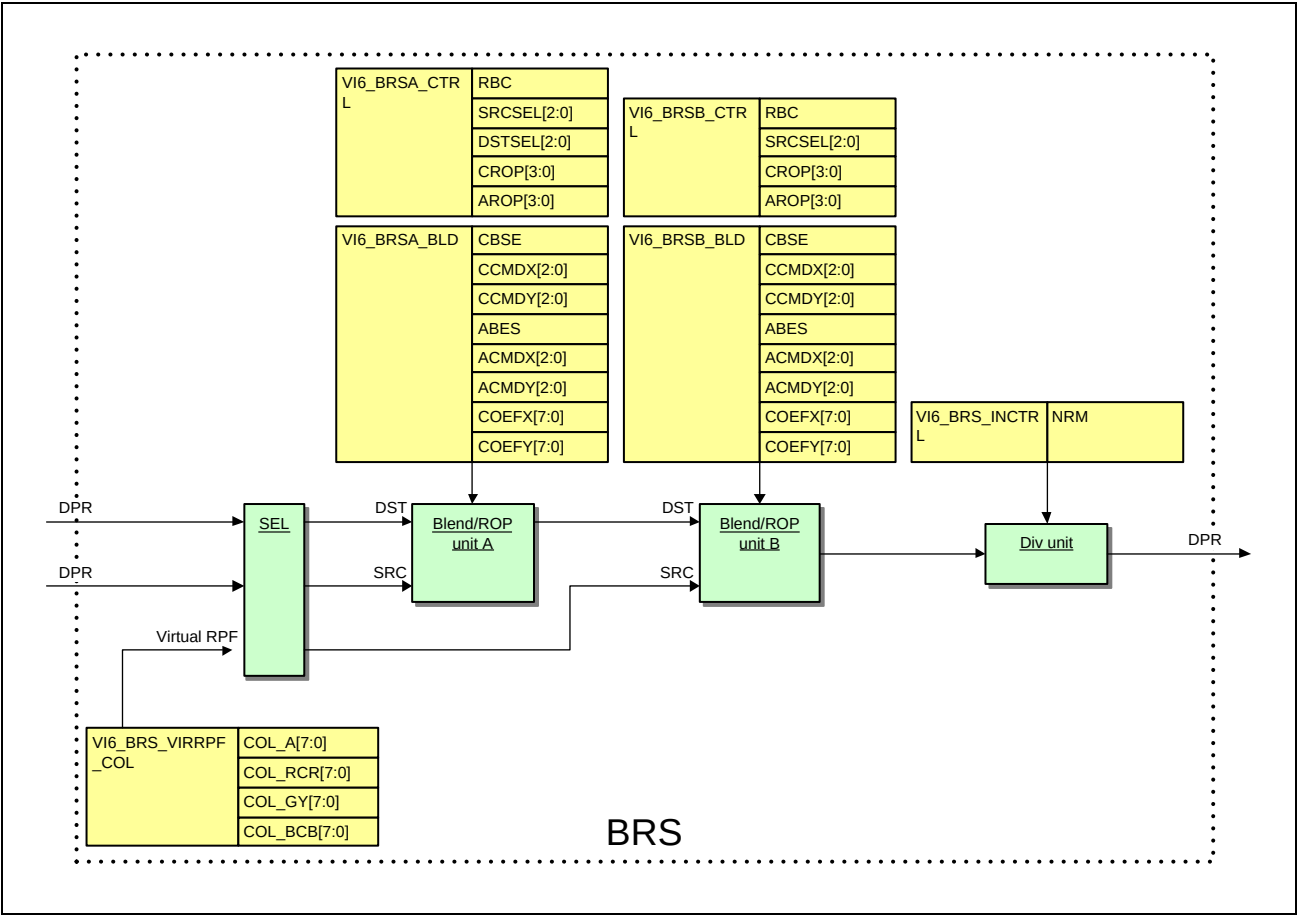


Figure 9.4-3 BRS Processing Flow

Each of the two blend-ROP operation units (blend/ROP unit m) receives the output from the SEL or blend/ROP Unit m, and executes blending or raster operation (ROP) of images.

The divider for normalization (div unit) divides the pixel value by the α value.

(7) Write Pixel Formatter (WPF)

The WPF is an output module that receives 32bits image data (YCbCr444 or RGB888 + 8-bit α) from the DPR, converts the color space, number of colors, and format of the data, and outputs the results of VSPD image processing to external memory through the MAU. The WPF is mainly configured from a color space converter and an output format converter (the packing unit).

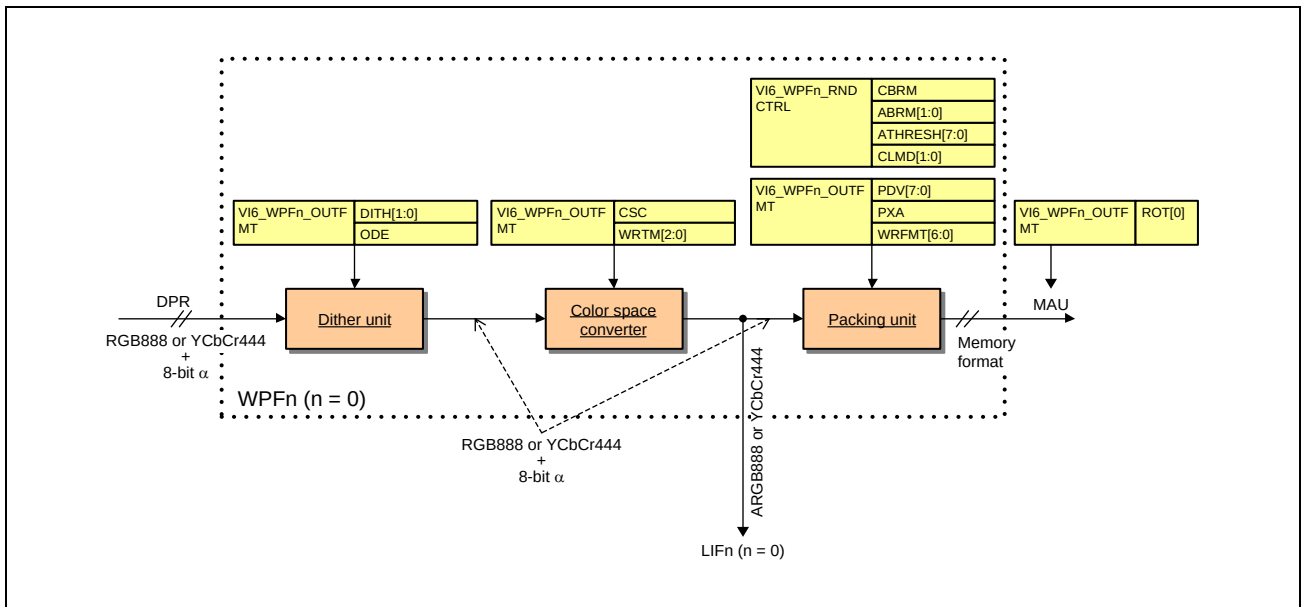


Figure 9.4-4 WPF Processing Flow

The color space converter converts the color space between RGB and YCbCr, and the packing unit converts the format into the picture plane storing format. The VSPD provides one WPF (WPF0). The WPF has the output to LIF module after the color space conversion for transferring the video data to the display module. Image data to LIF is 24bits (YCbCr444 or RGB888) format, and its' color space is after color space conversion.

(8) Display Unit Interface (LIF)

The LIF module is used for transferring image data to the display module. The input port of the LIFn (n = 0) module is connected to WPFn (n = 0), and the output port of the LIF module is connected to DU (Display Unit). Data flow in LIF is shown in **Figure 9.4-30**.

(9) Detail Function

Table 9.4-2 Detail Function of VSPD (1/4)

• Image Data Transfer Function			
Bus interface	Protocol		AXI 256 bits through FCPVD
	Data alignment	Conversion method	Byte, Word, LW, or LLW data swapping
		Channel	Data alignment can be specified separately for each input/output channel
Image memory	Input	Address setting	1-byte units
	Output	Address setting	1-byte units
		Memory area	Images can be written to the same memory area where the master layer is stored. Note the restrictions shown in Table 9.4-38
Tile transfer mode			Supported by RPF0 to RPF1
YCbCr memory storage format			Interleaved, planar, or semi-planar
• Display list/Extended Display List Transfer Function			
Bus interface	Protocol		AXI 256 bits through FCPVD
	Data alignment	Conversion method	Byte, Word, LW data swapping
			Data alignment can be specified separately for display list and each entry of extended display list
Data memory	Load	Address setting	8-byte units
	Store	—	Not available

Table 9.4-2 Detail Function of VSPD (2/4)

• Read Pixel Formatter (RPF)			
Number of channels		Two channels (RPF0 to RPF1)	
Input color bit-depth		Color	2- to 8-bit
		Alpha	1-/2-/4-/8-bit
Operation bit-depth		Color	8-bit
		Alpha	8-bit
Image format	Input	RGB	RGB888, RGB565, RGB666, αRGB8888, αRGB4444, αRGB1555 α plane (8 bpp, 1 bpp)
		YCbCr	YCbCr4:4:4 (8 bpc) Planar/Semi planar/Interleaved YCbCr4:2:2 (8 bpc) Planar/Semi planar/Interleaved YCbCr4:2:0 (8 bpc) Planar/Semi planar/Interleaved α plane (8 bpp, 1 bpp)
		Maximum size	1920 × 1200 pixels The internal data path modules have separate restrictions on the maximum image size. For details, refer to 9.4.6.3 Input Image Size .
		Minimum size	1 × 1 pixel The internal data path modules have separate restrictions on the minimum image size. For details, refer to 9.4.6.3 Input Image Size .
		Size setting unit	YCbCr420: 2-pixel units both horizontally and vertically. YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. In other formats, the size can be set in 1-pixel units.
Color keying	Color replacement	Compared data	RGB or Y (8 bpc)
		Replaced data	αRGB or αYCbCr (8 bpc for R, G, B, Y, Cb, Cr, α)
		Comparison Mode	Matched color mode
		Input source	RPF0 to RPF1
	Transparent color	Compared data	RGB or Y (8 bpc)
		Replaced data	α 8 bpp
		Comparison Mode	Matched color mode, Luma threshold mode
		Input source	RPF0 to RPF1
Raster operation	ROP2 (within input channels)	Operator	16 types (OpenGL2.0 is supported)
		Sources of operation	ROP2 operation between the 1-bpp α plane and RGB/YCbCr data in RPF0 to RPF1. Note that 1-bpp α is converted to αRGB or αYCbCr4:4:4. (8 bpc)
Color space conversion	RGB to YCbCr	Conversion expression	RGB (0, $2^n - 1$) to BT601 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) RGB (0, $2^n - 1$) to BT709 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) RGB (0, $2^n - 1$) to BT601 (0, $2^n - 1$) RGB (2^{n-4} , $235 \times 2^{n-8}$) to BT709 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) (n = 8)
		Target of conversion	RPF0 to RPF1
	YCbCr to RGB	Conversion expression	BT.601 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) to RGB (0, $2^n - 1$) BT.709 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) to RGB (0, $2^n - 1$) BT.601 (0, $2^n - 1$) to RGB (0, $2^n - 1$) BT.709 (2^{n-4} , $235 \times 2^{n-8}/240 \times 2^{n-8}$) to RGB (2^{n-4} , $235 \times 2^{n-8}$) (n = 8)
		Target of conversion	RPF0 to RPF1

Table 9.4-2 Detail Function of VSPD (3/4)

● Read Pixel Formatter (RPF)			
Changing number of colors	Bit extension	Extended bits	Input bitdepth (2bpc to 6bpc) to operation bitdepth 8bpc
		Target format	RGB666, RGB565, RGB555, RGB444, or RGB332
		Bit reduction method	Padded with 0. Copied from the most significant bits.
		Target of conversion	RPF0 to RPF1
	YCbCr444 generation	Vertical	CbCr copying
		Horizontal	copying or interpolation.
α bit count conversion	Bit extension	Extended bits	Input bitdepth 1bpc to operation bitdepth 8bpc
		Method	Padded with 0. Copied from the most significant bits.
Multiply-alpha function	Fade-alpha		RPF0 to RPF1 Available for straight pixel and pre-multiplied pixel.
	Generate pre-multiplied alpha		RPF0 to RPF1
Virtual display	RPF0 to RPF1	Color format	αRGB8888 or αYCbCr4:4:4 single-color
		Display size	Same as the size of the input channel
	Virtual RPF	Color format	αRGB8888 or αYCbCr4:4:4 single-color
		Display size	Maximum: 1920 × 1200 pixels Minimum: 4 × 4 pixels
● Write Pixel Formatter (WPF)			
Number of channels			One channel (WPF0)
Image format	Output	RGB	RGB332, RGB444, RGB565, RGB666, RGB888, αRGB8666, αRGB8888, αRGB4444, αRGB1555
		YCbCr	YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0
		Maximum size	1920 × 1200 pixels The internal data path modules have separate restrictions on the maximum image size. For details, refer to 9.4.6.4 Output Image Size .
		Minimum size	1 × 1 pixel The internal data path modules have separate restrictions on the minimum image size. For details, refer to 9.4.6.4 Output Image Size .
		Size setting unit	YCbCr420: 2-pixel units both horizontally and vertically. YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. In other formats, the size can be set in 1-pixel units.
Color space conversion	RGB to YCbCr	Conversion expression	RGB (0, 255) to BT.601 (16, 235/240) RGB (0, 255) to BT.709 (16, 235/240) RGB (0, 255) to BT.601 (0, 255) RGB (16, 235) to BT.709 (16, 235/240)
		Target of conversion	WPF0
	YCbCr to RGB	Conversion expression	BT.601 (16, 235/240) to RGB (0, 255) BT.709 (16, 235/240) to RGB (0, 255) BT.601 (0, 255) to RGB (0, 255) BT.709 (16, 235/240) to RGB (16, 235)
		Target of conversion	WPF0
Changing number of colors	Output	Reducing RGB color depth	Dithering, lower-order bit truncation, or rounding
		YCbCr422/420	CbCr skipping or CbCr vertical skipping and horizontal skipping
α bit count conversion	Output	Bit reduction	Truncation, rounding, or comparison with threshold (for 1 bpp)

Table 9.4-2 Detail Function of VSPD (4/4)

● Image Compositing			
α blending	Input α value selection	RGB	Pixel α, fixed α value, α plane, or 1-bit α converted from the color specified for pixels
		YCbCr	α plane, fixed α value, or 1-bit α converted from the color specified for pixels
	α blending expression	Layer A: Upper layer Layer B: Lower layer	xA + yB, xA – yB Coefficients x and y should be selected from the following. Fixed α value, (α for layer A), (1 – α for layer A), (α for layer B), (1 – α for layer B)
	Output α value selection	RGB	Fixed α value, x (α for layer A) + y (α for layer B), x (α for layer A) – y (α for layer B) Coefficients x and y should be selected from the following. Fixed α value, (α for layer A), (1 – α for layer A), (α for layer B), (1 – α for layer B)
	Blending layers	Number of layers	Two layers selected from RPF0 to RPF1 and video processing function output, and virtual RPF; three layers in total
		Order of layers	The order of three layers selected from RPF0 to RPF1, virtual RPF, and video processing function output can be changed as desired.
	α plane	Format	8 bpp or 1 bpp (α value can be specified through register)
		Input source	RPF0 to RPF1
	Fixed α value	Format	8 bpp
		Input source	RPF0 to RPF1, virtual RPF, or video processing function output
Raster operation	ROP2 (between input channels)	Operator	16 types (OpenGL2.0 is supported)
		Sources of operation	RPF0 to RPF1, virtual RPF, and video processing function output
		Operation control	RGB/YCbCr and α are operated separately.
	ROP3 (between input channels)	Operator	256-type ROP3 is available by combining ROP2 operations
		Sources of operation	RPF0 to RPF1, virtual RPF, and video processing function output
		Operation control	RGB/YCbCr and α are operated separately.
● Color Adjustment Function			
1D-LUT	LUT configuration	Independent R/Y, G/Cb, and B/Cr. 256 entries each	
● Direct connection to Display module			
Display I/F	Destination	DU	
	Direct connection	Can transfer data without going through external memory to DU	
Line padding	Padding cycles	1 to 32 cycles	
	Padding pattern	Arbitrarily setting up to 32 cycles	

9.4.1.2.3 DU

DU has the following sub modules.

(1) REG

The REG controls registers for DU. The DU has APB4 interface, its transaction is transfer with no wait states.

(2) LIFC

The LIFC is the interface block from/to VSPD. The LIFC receives image data from VSPD, sends to PBUF in a pixel unit. The LIFC continues to receive the image data from VSPD unless FIFO will be full.

(3) PBUF

The PBUF is the asynchronous FIFO which convert clock domain of image data from System Bus Clock to Video Clock. The PBUF monitors the FIFO underflow.

(4) DIFC

The DIFC is the display timing master. The DIFC controls the polarity and the timing of the Video Output signals. Controllable timings are the followings.

- Hfront: Horizontal front porch (pixels)
- Hback: Horizontal back porch (pixels)
- Hsync: Horizontal sync active (pixels)
- Hactive: Horizontal active area (pixels)
- Vfront: Vertical front porch (lines)
- Vback: Vertical back porch (lines)
- Vsync: Vertical sync active (lines)
- Vactive: Vertical active area (lines)

(5) INTC

The INTC generates the interrupt when PBUF will detect a FIFO underflow. The interrupt is level signal.

9.4.1.3 Clock and Interrupt signal

The following clocks are supplied to the LCDC.

Table 9.4-3 Clock List

Clock Name	Clock Symbol	Description	Frequency (MHz)
LCDC_0_clk_a	M0 ϕ	AXI clock.	400/200/100/50/13
LCDC_0_clk_p	ZT ϕ	APB clock.	100
LCDC_0_clk_d	M3 ϕ	Video clock.	5.44 to 187.5

The LCDC transmits the following interrupt signals to the interrupt controller.

Table 9.4-4 Interrupt List

Signal Name	Interrupt Condition	Interrupt Source Register
VSPD_INT	Frame End	VI6_WPF0_IRQ_STA.FRE
	Display List Frame End	VI6_WPF0_IRQ_STA.DFE
	DU Connection UnderRun Error	VI6_WPF0_IRQ_STA.UND
DU_INT	RBUF FIFO Underflow	DU_MSR0.ST_PB_RUF

Note: Refer to (13) **WPF0 Interrupt Status Register (LCDC_VSPD_VI6_WPF0_IRQ_STA)** and **9.4.4.2.1 DU Module Control Register 0 (LCDC_DU_MCR0)** for details.

9.4.2 FCPVD Registers

The FCPVD contains the registers which can directly be read and written to by the host CPU. The host can access any long word (32 bits) in the registers. Therefore, read-modify-write is needed to change partial bits in the 32-bit register.

Never access the registers when the FCPVD is in the module standby mode (with the operation clock placed in hold status).

Table 9.4-5 FCPVD Register Base Address

Base Register Name	Base Address
<LCDC_fcprd_base>	0_1647_0000h (5647_0000h* ¹ , 4647_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.4.2.1 List of FCPVD Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
FCPV version control register	LCDC_FCPVD_FCP_VCR	0000_0109h	0000h	32

9.4.2.2 FCPVD Register Description

The prefix (LCDC_FCPVD_) of the register names is omitted in this and subsequent sections.

9.4.2.2.1 FCPVD Version Control Register (LCDC_FCPVD_FCP_VCR)

FCP_VCR is the version control register of the FCPVD. The value read is fixed.

Access Size :		32 bits														
Address :		<LCDC_fcpvd_base> + 0000h														
Initial Value :		0000_0109h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CATEGORY[7:0]								REVISION[7:0]							
Initial Value	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 8	CATEGORY [7:0]	1h	R	FCP use case category.
7 to 0	REVISION[7:0]	9h	R	Version of LSI product 09h: FCP revision is the Simple LCDC

9.4.3 VSPD Registers

In the following, “n” in a register name or a offset address indicates an integer, and the range of value n is defined when necessary. For RPF_n and WPF_n, when the range of value n is not defined, RPF_n (n = 0, 1) and WPF_n (n = 0) are assumed.

Table 9.4-6 VSPD Register Base Address

Base Register Name	Base Address
<LCDC_vspd_base>	0_1648_0000h (5648_0000h* ¹ , 4648_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.4.3.1 List of VSPD Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
VSPD Start Register 0	LCDC_VSPD_VI6_CMD0	0000_0000h	0000h	32
Reserve	-	-	0004h to 000Fh	-
Clock Control Register 0	LCDC_VSPD_VI6_CLK_CTRL0	0000_0000h	0010h	32
Clock Control Register 1	LCDC_VSPD_VI6_CLK_CTRL1	0000_0000h	0014h	32
Dynamic Clock Stop Control Register	LCDC_VSPD_VI6_CLK_DCSWT	0000_0000h	0018h	32
Dynamic Clock Stop Disable Register 0	LCDC_VSPD_VI6_CLK_DCSM0	0000_0000h	001Ch	32
Dynamic Clock Stop Disable Register 1	LCDC_VSPD_VI6_CLK_DCSM1	0000_0000h	0020h	32
Reserve	-	-	0024h to 0027h	-
Software Reset Register	LCDC_VSPD_VI6_SRESET	0000_0000h	0028h	32
Module Reset Enable Register 0	LCDC_VSPD_VI6_MRESET_ENB 0	0000_0000h	002Ch	32
Module Reset Enable Register 1	LCDC_VSPD_VI6_MRESET_ENB 1	0000_0000h	0030h	32
Module Reset Issuing Register	LCDC_VSPD_VI6_MRESET	0000_0000h	0034h	32
Operating Status Register	LCDC_VSPD_VI6_STATUS	0000_0000h	0038h	32
Reserve	-	-	003Ch to 0047h	-
WPF0 Interrupt Enable Register	LCDC_VSPD_VI6_WPF0_IRQ_EN B	0000_0000h	0048h	32
WPF0 Interrupt Status Register	LCDC_VSPD_VI6_WPF0_IRQ_ST A	0000_0000h	004Ch	32
Reserve	-	-	0050h to 0077h	-
Display-0 Interrupt Enable Register	LCDC_VSPD_VI6_DISP0_IRQ_EN B	0000_0000h	0078h	32
Display-0 Interrupt Status Register	LCDC_VSPD_VI6_DISP0_IRQ_ST A	0000_0000h	007Ch	32
Reserve	-	-	0080h to 00FFh	-
Display List Control Register	LCDC_VSPD_VI6_DL_CTRL	0000_0000h	0100h	32
Reserve	-	-	0108h to 0113h	-
Display List-0 Data Swapping Register	LCDC_VSPD_VI6_DL_SWAP0	0000_0000h	0114h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Reserve	-	-	0118h to 011Bh	-
Extended Display List-0 Control Register	LCDC_VSPD_VI6_DL_EXT_CTRL0	0000_0000h	011Ch	32
Display List Body Size Register 0	LCDC_VSPD_VI6_DL_BODY_SIZE0	0000_0000h	0120h	32
Display List-0 Header Address Register	LCDC_VSPD_VI6_DL_HDR_ADDR0	0000_0000h	0104h	32
Reserve	-	-	0124h to 012Fh	-
Display List-0 Header Reference Address Register	LCDC_VSPD_VI6_DL_HDR_REF_ADDR0	0000_0000h	0130h	32
Reserve	-	-	0134h to 0157h	-
Display List-0 Wake Up Counter Register	LCDC_VSPD_VI6_DL_WUPCNT0	0000_0000h	0158h	32
Reserve	-	-	015Ch to 02FFh	-
RPF0 Basic Read Size Register	LCDC_VSPD_VI6_RPF0_SRC_SIZE	0000_0000h	0300h	32
RPF0 Extended Read Size Register	LCDC_VSPD_VI6_RPF0_SRC_SIZE	0000_0000h	0304h	32
RPF0 Input Format Register	LCDC_VSPD_VI6_RPF0_INFMT	0000_0000h	0308h	32
RPF0 Data Swapping Register	LCDC_VSPD_VI6_RPF0_DSWAP	0000_0000h	030Ch	32
RPF0 Display Location Register	LCDC_VSPD_VI6_RPF0_LOC	0000_0000h	0310h	32
RPF0 a Plane Selection Control Register	LCDC_VSPD_VI6_RPF0_ALPH_SEL	0000_0000h	0314h	32
RPF0 Virtual Plane Color Information Register	LCDC_VSPD_VI6_RPF0_VRTCOL_SET	0000_0000h	0318h	32
RPF0 Mask Control Register	LCDC_VSPD_VI6_RPF0_MSKCTRL	0000_0000h	031Ch	32
RPF0 IROP-SRC Input Value Register 0	LCDC_VSPD_VI6_RPF0_MSKSET0	0000_0000h	0320h	32
RPF0 IROP-SRC Input Value Register 1	LCDC_VSPD_VI6_RPF0_MSKSET1	0000_0000h	0324h	32
RPF0 Color Keying Control Register	LCDC_VSPD_VI6_RPF0_CKEY_CTRL	0000_0000h	0328h	32
RPF0 Color Keying Color Setting Register 0	LCDC_VSPD_VI6_RPF0_CKEY_SET0	0000_0000h	032Ch	32
RPF0 Color Keying Color Setting Register 1	LCDC_VSPD_VI6_RPF0_CKEY_SET1	0000_0000h	0330h	32
RPF0 Source Picture Memory Stride Setting Register	LCDC_VSPD_VI6_RPF0_SRCM_PSTRIDE	0000_0000h	0334h	32
RPF0 Source α Memory Stride Setting Register	LCDC_VSPD_VI6_RPF0_SRCM_ASTRIDE	0000_0000h	0338h	32
RPF0 Source Y/RGB Address Register	LCDC_VSPD_VI6_RPF0_SRCM_ADDR_Y	0000_0000h	033Ch	32
RPF0 Source Chroma Address Register 0	LCDC_VSPD_VI6_RPF0_SRCM_ADDR_C0	0000_0000h	0340h	32
RPF0 Source Chroma Address Register 1	LCDC_VSPD_VI6_RPF0_SRCM_ADDR_C1	0000_0000h	0344h	32
RPF0 Source α Address Register	LCDC_VSPD_VI6_RPF0_SRCM_ADDR_AI	0000_0000h	0348h	32
Reserve	-	-	034Ch to 034Fh	-
RPF0 Bus Access Control Register	LCDC_VSPD_VI6_RPF0_BAC	0000_0000h	0350h	32
Reserve	-	-	0354h to 036Bh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
RPF0 Multiple Alpha Control Register	LCDC_VSPD_VI6_RPF0_MULT_ALPH	0000_0000h	036Ch	32
Reserve	-	-	0370h to 03FFh	-
RPF1 Basic Read Size Register	LCDC_VSPD_VI6_RPF1_SRC_SIZE	0000_0000h	0400h	32
RPF1 Extended Read Size Register	LCDC_VSPD_VI6_RPF1_SRC_SIZE	0000_0000h	0404h	32
RPF1 Input Format Register	LCDC_VSPD_VI6_RPF1_INFMT	0000_0000h	0408h	32
RPF1 Data Swapping Register	LCDC_VSPD_VI6_RPF1_DSWAP	0000_0000h	040Ch	32
RPF1 Display Location Register	LCDC_VSPD_VI6_RPF1_LOC	0000_0000h	0410h	32
RPF1 a Plane Selection Control Register	LCDC_VSPD_VI6_RPF1_ALPH_SEL	0000_0000h	0414h	32
RPF1 Virtual Plane Color Information Register	LCDC_VSPD_VI6_RPF1_VRTCOL_SET	0000_0000h	0418h	32
RPF1 Mask Control Register	LCDC_VSPD_VI6_RPF1_MSKCTRL	0000_0000h	041Ch	32
RPF1 IROP-SRC Input Value Register 0	LCDC_VSPD_VI6_RPF1_MSKSET0	0000_0000h	0420h	32
RPF1 IROP-SRC Input Value Register 1	LCDC_VSPD_VI6_RPF1_MSKSET1	0000_0000h	0424h	32
RPF1 Color Keying Control Register	LCDC_VSPD_VI6_RPF1_CKEY_CTRL	0000_0000h	0428h	32
RPF1 Color Keying Color Setting Register 0	LCDC_VSPD_VI6_RPF1_CKEY_SET0	0000_0000h	042Ch	32
RPF1 Color Keying Color Setting Register 1	LCDC_VSPD_VI6_RPF1_CKEY_SET1	0000_0000h	0430h	32
RPF1 Source Picture Memory Stride Setting Register	LCDC_VSPD_VI6_RPF1_SRCM_PSTRIDE	0000_0000h	0434h	32
RPF1 Source α Memory Stride Setting Register	LCDC_VSPD_VI6_RPF1_SRCM_ASTRIDE	0000_0000h	0438h	32
RPF1 Source Y/RGB Address Register	LCDC_VSPD_VI6_RPF1_SRCM_ADDR_Y	0000_0000h	043Ch	32
RPF1 Source Chroma Address Register 0	LCDC_VSPD_VI6_RPF1_SRCM_ADDR_C0	0000_0000h	0440h	32
RPF1 Source Chroma Address Register 1	LCDC_VSPD_VI6_RPF1_SRCM_ADDR_C1	0000_0000h	0444h	32
RPF1 Source α Address Register	LCDC_VSPD_VI6_RPF1_SRCM_ADDR_AI	0000_0000h	0448h	32
Reserve	-	-	044Ch to 044Fh	-
RPF1 Bus Access Control Register	LCDC_VSPD_VI6_RPF1_BAC	0000_0000h	0450h	32
Reserve	-	-	0454h to 046Bh	-
RPF1 Multiple Alpha Control Register	LCDC_VSPD_VI6_RPF1_MULT_ALPH	0000_0000h	046Ch	32
Reserve	-	-	0470h to 0FFFh	-
WPF0-Source-RPF Register	LCDC_VSPD_VI6_WPF0_SRCRPF	0000_0000h	1000h	32
WPF0 Horizontal Input Size Clipping Register	LCDC_VSPD_VI6_WPF0_HSZCLIP	0000_0000h	1004h	32
WPF0 Vertical Input Size Clipping Register	LCDC_VSPD_VI6_WPF0_VSZCLIP	0000_0000h	1008h	32
WPF0 Output Format Register	LCDC_VSPD_VI6_WPF0_OUTFMT	0000_0000h	100Ch	32
WPF0 Data Swapping Register	LCDC_VSPD_VI6_WPF0_DSWAP	0000_0000h	1010h	32
WPF0 Rounding Control Register	LCDC_VSPD_VI6_WPF0_RNDCTRL	0000_0000h	1014h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Reserve	-	-	1018h to 101Bh	-
WPF0 Destination Y Plane Memory Stride Register	LCDC_VSPD_VI6_WPF0_DSTM_STRIDE_Y	0000_0000h	101Ch	32
WPF0 Destination C Plane Memory Stride Register	LCDC_VSPD_VI6_WPF0_DSTM_STRIDE_C	0000_0000h	1020h	32
WPF0 Destination Y/RGB Address Register	LCDC_VSPD_VI6_WPF0_DSTM_ADDR_Y	0000_0000h	1024h	32
WPF0 Destination Chroma Address Register 0	LCDC_VSPD_VI6_WPF0_DSTM_ADDR_C0	0000_0000h	1028h	32
WPF0 Destination Chroma Address Register 1	LCDC_VSPD_VI6_WPF0_DSTM_ADDR_C1	0000_0000h	102Ch	32
Reserve	-	-	1030h to 1033h	-
WPF0 LIF Write Back Control Register	LCDC_VSPD_VI6_WPF0_WRBCK_CTRL	0000_0000h	1034h	32
Reserve	-	-	1038h to 1FFFh	-
RPF0 Routing Register	LCDC_VSPD_VI6_DPR_RPF0_ROUTE	0000_0000h	2000h	32
RPF1 Routing Register	LCDC_VSPD_VI6_DPR_RPF1_ROUTE	0000_0000h	2004h	32
Reserve	-	-	2008h to 2013h	-
WPF0 Timing Control Register	LCDC_VSPD_VI6_DPR_WPF0_FPORCH	0000_0000h	2014h	32
Reserve	-	-	2018h to 203Bh	-
LUT Routing Register	LCDC_VSPD_VI6_DPR_LUT_ROUTE	0000_0000h	203Ch	32
Reserve	-	-	2040h to 204Fh	-
BRS Routing Register	LCDC_VSPD_VI6_DPR_BRS_ROUTE	0000_0000h	2050h	32
Reserve	-	-	2054h to 27FFh	-
LUT Control Register	LCDC_VSPD_VI6_LUT_CTRL	0000_0000h	2800h	32
Reserve	-	-	2804h to 38FFh	-
BRS Input Control Register	LCDC_VSPD_VI6_BRS_INCTRL	0000_0000h	3900h	32
Size Register of BRS Input Virtual RPF	LCDC_VSPD_VI6_BRS_VIRRRPF_SIZE	0000_0000h	3904h	32
Display Location Register of BRS Input Virtual RPF	LCDC_VSPD_VI6_BRS_VIRRRPF_LOC	0000_0000h	3908h	32
Color Information Register of BRS Input Virtual RPF	LCDC_VSPD_VI6_BRS_VIRRRPF_COL	0000_0000h	390Ch	32
BRS Control Register A	LCDC_VSPD_VI6_BRSA_CTRL	0000_0000h	3910h	32
BRS Blend Control Register A	LCDC_VSPD_VI6_BRSA_BLD	0000_0000h	3914h	32
BRS Control Register B	LCDC_VSPD_VI6_BRSB_CTRL	0000_0000h	3918h	32
BRS Blend Control Register B	LCDC_VSPD_VI6_BRSB_BLD	0000_0000h	391Ch	32
Reserve	-	-	3920h to 3AFFh	-

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
LIFO Control Register	LCDC_VSPD_VI6_LIF0_CTRL	0000_0000h	3B00h	32
LIFO Clock Stop Buffer Control Register	LCDC_VSPD_VI6_LIF0_CSBTH	0000_0000h	3B04h	32
Reserve	-	-	3B08h to 3B0Bh	-
LIFO Buffer Attribute Register	LCDC_VSPD_VI6_LIF0_LBA	0000_0000h	3B0Ch	32
Reserve	-	-	3B10h to 3B2Fh	-
LIFO Padding Line Cycle Register	LCDC_VSPD_VI6_LIF0_PADLN_ CYC	0000_0000h	3B30h	32
LIFO Padding Line Pattern Register	LCDC_VSPD_VI6_LIF0_PADLN_ PT	0000_0000h	3B34h	32
LIFO Padding Line Value Register	LCDC_VSPD_VI6_LIF0_PADLN_ VAL	0000_0000h	3B38h	32
LIFO Padding Line Size Register	LCDC_VSPD_VI6_LIF0_PADLN_ SIZE	0000_0000h	3B3Ch	32

Note: Read only bits or write only bits or read/write bits are mixed in the registers.

9.4.3.2 VSPD Register Description

The prefix (LCDC_VSPD_) of the register names is omitted in this and subsequent sections.

9.4.3.2.1 Notational Conventions for Registers and Bit Fields

This section uses the following notational conventions for the VSPD registers and bit fields.

1. The names of registers and bits are written in uppercase.
2. A bit or bit field in a register is indicated as [register name.bit name]. For example, the STRCMD bit in the VI6_CMD0 register is indicated as VI6_CMD0.STRCMD.
3. Lowercase “n” in a register name or a bit name indicates an integer and the range of value n is defined when necessary. For RPFn, and WPFn, when the range of value n is not defined, RPFn (n = 0, 1), WPFn (n = 0) are assumed.
4. In each subsection for register description in **9.4.3.2 VSPD Register Description**, when only a bit name is written without showing its register name, the bit is in the register described in that subsection.
5. A wildcard (*) indicates any characters in a name and represents all registers or bits that match the specified first part of a name. For example, when there are two registers VI6_RPF_SRC_BSIZE and VI6_RPF_SRC_ESIZE, VI6_RPF_SRC_* indicates both registers.

9.4.3.2.2 Register Classification

The VSPD registers are arranged in the following order; the general control registers to display list control registers control operation of the entire VSPD, and the other registers control each image processing and specify parameters for the processing. The functions of the registers are described in this order starting from **9.4.3.2.4 General Control Registers**.

1. General control registers (VI6_CMDn (n = 0, 1), VI6_SRESET, VI6_STATUS, VI6_WPFn_IRQ_* (n = 0))
2. Display list control registers (VI6_DL_*)
3. RPF control registers (VI6_RPFn_* (n = 0, 1))
4. WPF control registers (VI6_WPFn_* (n = 0))
5. DPR control registers (VI6_DPR_*)
6. LUT control register (VI6_LUT_CTRL)
7. BRS control registers (VI6_BRS_*)
8. LIFn control registers (VI6_LIFn_* (n = 0))

The VSPD has two RPF channels and the register configuration is the same for all of RPFn (n = 0, 1). However, some bit fields have restrictions in certain RPFs. These restrictions are included in the description of the corresponding bit fields and registers. Likewise, the register configuration is the same for all WPFn (n = 0), but some bit fields have restrictions in certain WPFs; the restrictions are included in the description of the corresponding bit fields and registers.

For each register address, refer to **9.4.3.1 List of VSPD Registers**.

9.4.3.2.3 Restrictions on Access to Registers and Lookup Tables

The VSPD has control registers and lookup tables. All VSPD registers are writable and readable by only 32 bits unit. To write partial bits in each register, read-modify-write is needed. When accessing the addresses where these registers and lookup tables are allocated, the following restrictions should be satisfied. If any restriction is violated, the VSPD will not operate correctly.

1. For the read-only bits and reserved bits in all VSPD registers, writing 1 is prohibited unless otherwise specified.
2. Addresses undefined in **9.4.3.1 List of VSPD Registers** are reserved areas and write access is prohibited in these areas.
3. For all registers and lookup tables, except VI6_CMDn, VI6_SRESET, VI6_*IRQ* and two plane registers such as VI6_DL_HDR_ADDRn, VI6_DL_BODY_SIZE0, modifying register values during operation of the module is prohibited. Modify registers while the corresponding module is stopped. For the operating status of the target module, refer to **9.4.5.4 Concept of VSPD Operation Starting and Stopping**.
4. There are three types about General control registers (**9.4.3.2.4 General Control Registers**) and Display List Control Registers (**9.4.3.2.5 Display List Control Registers**) as below.
 - Controlling WPF0 (ex. VI6_CMD0, VI6_SRESET.SRST0)
 - Common setting of WPF0 (ex. VI6_DL_CTRL.AR_WAIT [15:0])

Table 9.4-7 Correspondence between Modules and Register Names

Module Name	Register Name
RPFn (n = 0, 1)	VI6_RPFn_*
WPFn (n = 0)	VI6_WPFn_*
DPR	VI6_DPR_*
LUT	VI6_LUT_CTRL
BRS	VI6_BRS_*
LIFn (n = 0)	VI6_LIFn_*

9.4.3.2.4 General Control Registers

(1) VSPD Start Register 0 (LCDC_VSPD_VI6_CMD0)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0000h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	UPDHDR	-	-	-	STRCMD
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	UPDHDR	0h	RW	Reserved state of updating Display List Header (DLH) address. UPDHDR can be negated by software reset (VI6_SRESET.SRST0). [Write] 0b: NOP 1b: Do not set 1b to this bit [Read] 0b: Not reserved to update DLH address. 1b: Reserved to update DLH address. Set 0b to this bit when STRCMD bit is set to 1b.
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	STRCMD	0h	RW	Start reservation of WPF VI6_CMD0.STRCMD controls WPF0. Writing 1b to this bit starts WPF0 in VSPD. Set this bit for activation only after all register settings in each output channel have been completed. If WPF0 is idle, WPF0 starts right after this bit is set to 1b. If WPF0 is active, writing 1b to this bit reserves starting WPF0 operation. Wait to set 1b to VI6_CMD0.STRCMD until this bit is read as 0b. VI6_CMD0.STRCMD can be negated by software reset (VI6_SRESET.SRST0). [Write] 0b: Start reservation of WPF0 is canceled. 1b: Start reservation of WPF0 is set. [Read] 0b: Starting VSPD is not reserved. 1b: Starting VSPD is reserved.

The basic concept of image processing operation started by activating the VSPD is shown in **Figure 9.4-5**. The actual data input/output is executed by the MAU, which is the bus interface module, as described in **9.4.1.2 Block Diagram**, but conceptually the RPF works as the data entry point to the VSPD and the WPF works as the data exit point. To process images through the VSPD, the RPF (entrance) and WPF (exit) should be connected and a data path from the RPF to the WPF should be formed.

To connect RPF_n to WPF_n, specify RPF_n as the source RPF for WPF_n in VI6_WPF_n_SRCRPF, which is a register for WPF_n (refer to **(1) WPF0-Source-RPF Register (LCDC_VSPD_VI6_WPF0_SRCRPF)**). This setting determines that RPF_n will be started when WPF_n is started through VI6_CMD_n.

A data path to execute desired image processing should then be formed between the RPF (entrance) and WPF (exit). To form a data path, connect the necessary function modules in the VSPD between RPF and WPF. This function is

provided by the DPR; specify the information for each module connection in data path routing registers VI6_DPR_*_ROUTE (refer to **9.4.3.2.8 DPR Control Registers**).

After a data path is formed (RPF_n → WPF_n) as described above, starting output module WPF_n in the VSPD through VI6_CMD_n starts all function modules connected to WPF_n and the desired image processing is executed. According to this design concept, starting a WPF module means starting the VSPD.

There are two types of data path configuration in the VSPD; one is “a single input module to a single output module” as shown in **Figure 9.4-5 (A)**, and the other is “multiple input modules to a single output module” as shown in **Figure 9.4-5 (B)**.

Figure 9.4-5 (A) shows an example of a configuration where modules with single input and single output are implemented through the DPR.

Figure 9.4-5 (B) shows another configuration example where the module with multiple input and single output is implemented through the DPR.

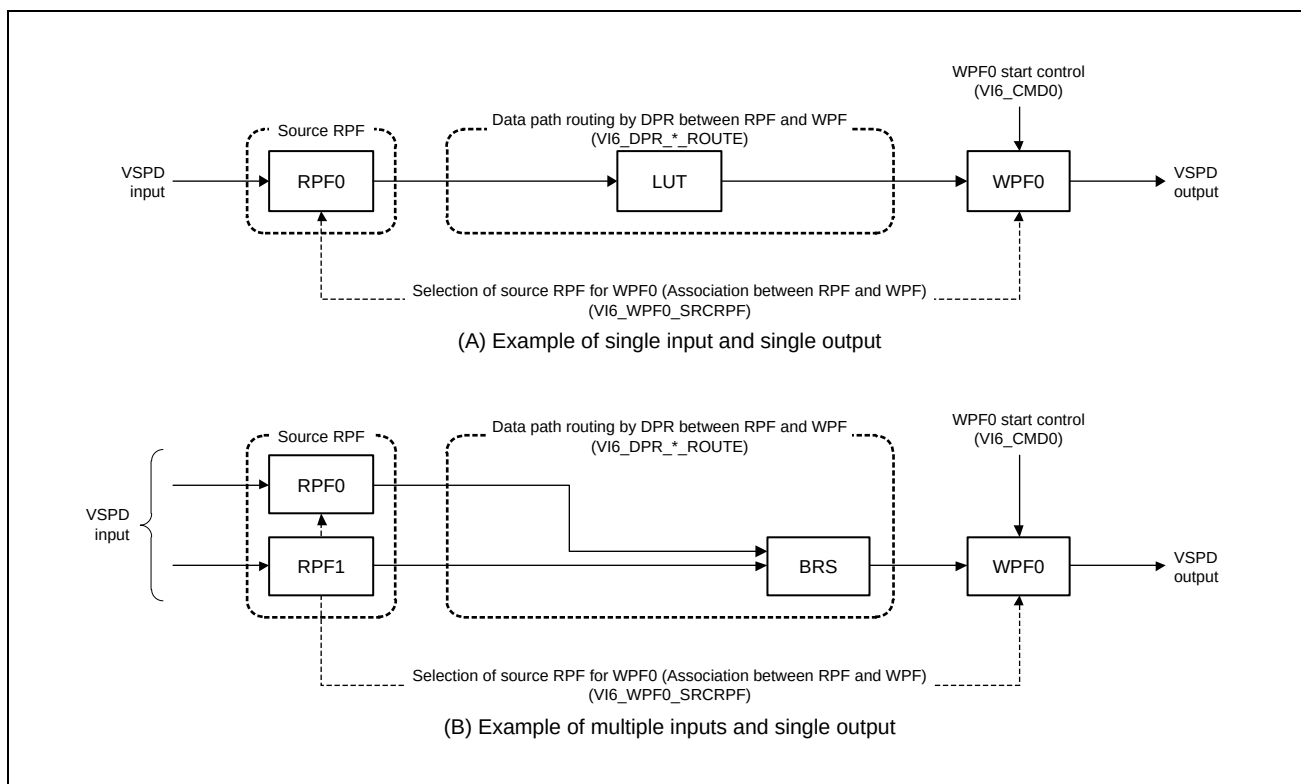


Figure 9.4-5 Basic Concept of VSPD Startup

(2) Clock Control Register 0 (LCDC_VSPD_VI6_CLK_CTRL0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0010h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	GCS0	-	-	-	-	-	-	-	-	-	-	-	GCS1
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	GCS2[3:0]				-	-	-	GCS3[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	GCS0	0h	RW	Clock Control Setting0
27 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	GCS1	0h	RW	Clock Control Setting1
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 8	GCS2[3:0]	0h	RW	Clock Control Setting2
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4 to 0	GCS3[4:0]	00h	RW	Clock Control Setting3

VSPD can stop its operating clock for reducing power consumption.

To enable clock stop function, set following registers:

- VI6_CLK_CTRL0 = H'0000 0000
- VI6_CLK_CTRL1 = H'0000 0000
- VI6_CLK_DCSWT = H'0000 0808
- VI6_CLK_DCSM0 = H'0000 0000
- VI6_CLK_DCSM1 = H'0000 0000

To disable clock stop function, set following registers:

- VI6_CLK_CTRL0 = H'1001 0F1F
- VI6_CLK_CTRL1 = H'FF10 FFFF
- VI6_CLK_DCSWT = H'0033 0808
- VI6_CLK_DCSM0 = H'1FFF 0F1F
- VI6_CLK_DCSM1 = H'FF10 FFFF

When the clock gating is enabled, the supply of the clock signal is stopped for the module which is not operated. If the bit field of this register is changed from 0 to 1, the supply of the clock signal for the module is immediately started. If

the bit field of the register is changed from 1 to 0, the clock supply is immediately stopped. Therefore, please do not change this register field from 1 to 0 in operation because the VSPD may be stalled.

(3) Clock Control Register 1 (LCDC_VSPD_VI6_CLK_CTRL1)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 0014h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
GCS4[7:0]									-	-	-	GCS6	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
GCS5[15:0]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	Bit Name																
31 to 24	GCS4[7:0]																
Initial Value	00h																
R/W	RW																
23 to 21	-																
Initial Value	All 0																
R	Reserved Whenever it is read, 0b is read. The written value will be ignored.																
20	GCS6																
Initial Value	0h																
R/W	RW																
19 to 16	-																
Initial Value	All 0																
R	Reserved Whenever it is read, 0b is read. The written value will be ignored.																
15 to 0	GCS5[15:0]																
Initial Value	0000h																
R/W	RW																

See (2) Clock Control Register 0 (LCDC_VSPD_VI6_CLK_CTRL0) for detail.

(4) Dynamic Clock Stop Control Register (LCDC_VSPD_VI6_CLK_DCSWT)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0018h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	DCC0[1:0]		-	-	DCC1	DCC2
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSTPW[7:0]								CSTRW[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
21, 20	DCC0[1:0]	0h	RW	Dynamic Clock Control Setting0
19, 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	DCC1	0h	RW	Dynamic Clock Control Setting1
16	DCC2	0h	RW	Dynamic Clock Control Setting2
15 to 8	CSTPW[7:0]	00h	RW	Dynamic Clock Stop Control 1 Always specify 8.
7 to 0	CSTRW[7:0]	00h	RW	Dynamic Clock Stop Control 2 Always specify 8.

See (2) Clock Control Register 0 (LCDC_VSPD_VI6_CLK_CTRL0) for detail.

(5) Dynamic Clock Stop Disable Register 0 (LCDC_VSPD_VI6_CLK_DCSM0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 001Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DCD0[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	DCD1[3:0]				-	-	-	DCD2[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW
Bit	Bit Name			Initial Value	R/W	Description										
31 to 29	-			All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
28 to 16	DCD0[12:0]			0000h	RW	Dynamic Clock Stop Disable Setting0										
15 to 12	-			All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 8	DCD1[3:0]			0h	RW	Dynamic Clock Stop Disable Setting1										
7 to 5	-			All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
4 to 0	DCD2[4:0]			00h	RW	Dynamic Clock Stop Disable Setting2										

See (2) Clock Control Register 0 (LCDC_VSPD_VI6_CLK_CTRL0) for detail.

(6) Dynamic Clock Stop Disable Register 1 (LCDC_VSPD_VI6_CLK_DCSM1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCD3[7:0]								-	-	-	DCD5	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCD4[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name															
31 to 24		DCD3[7:0]		Initial Value		00h		R/W		Description						
23 to 21		-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.								
20		DCD5		0h		RW		Dynamic Clock Stop Disable Setting5								
19 to 16		-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.								
15 to 0		DCD4[15:0]		0000h		RW		Dynamic Clock Stop Disable Setting4								

See (2) Clock Control Register 0 (LCDC_VSPD_VI6_CLK_CTRL0) for detail.

(7) Software Reset Register (LCDC_VSPD_VI6_SRESET)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0028h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SRST0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	SRST0	0h	RW	WPF0 Software Reset Writing 1b to this bit aborts the current processing in WPF0 (the partially-completed image undergoing processing is output). The period until this software reset processing is completed depends on the bus state. When this reset processing is completed, the VI6_WPF0_IRQ_STA.FRE interrupt source bit is set to 1b; when the FRE interrupt is enabled, the FRE end interrupt is output to notify the end of the reset processing. This bit is always read as 0b. 0b: NOP 1b: WPF0 software reset* ¹

Note 1. Applying a software reset to each WPF has the following restrictions.

1. A software reset can be applied to only one of WPF0 through single write access to VI6_SRESET.
2. After a software reset is issued, no more software reset can be issued to another WPF until the issued software reset processing is completed
3. The end of software reset processing is notified through the FRE bit in VI6_WPFn_IRQ_STA, but the software reset issued while WPF is stopped is ignored as NOP. As it takes a while until the reset is actually issued after the reset bit is set, the VSPD may complete operation before the reset is actually issued. In this case, no interrupt is output for the software reset that is issued after the VSPD completes operation.
4. If a software reset is issued during downloading of a display list, the downloading processing is not aborted. After the end of downloading that is in progress when a software reset is issued, a frame end interrupt is output.

(8) Module Reset Enable Register 0 (LCDC_VSPD_VI6_MRESET_ENB0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 002Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	MRSTEO[1:0]		-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	MRSTE1[3:0]				-	-	-	MRSTE2[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31, 30	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
29, 28	MRSTEO[1:0]		0h	RW	Module Reset Enable 0											
27 to 12	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
11 to 8	MRSTE1[3:0]		0h	RW	Module Reset Enable 1											
7 to 5	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
4 to 0	MRSTE2[4:0]		00h	RW	Module Reset Enable 2											

This register is for purpose of h/w debugging.

(9) Module Reset Enable Register 1 (LCDC_VSPD_VI6_MRESET_ENB1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0030h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRSTE3[7:0]									-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRSTE4[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 24	MRSTE3[7:0]		00h		RW		Module Reset Enable 3									
23 to 16	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
15 to 0	MRSTE4[15:0]		0000h		RW		Module Reset Enable 4									

This register is for purpose of h/w debugging.

(10) Module Reset Issuing Register (LCDC_VSPD_VI6_MRESET)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0034h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MRST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	MRST		0h		RW	Module Reset Assertion										

This register is for purpose of h/w debugging.

(11) Operating Status Register (LCDC_VSPD_VI6_STATUS)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0038h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	FLDST 0	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SYS0_ ACT	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	FLDST0	0h	R	Field status of previous frame of WPF0. 0b: Previous finished frame is TOP field. 1b: Previous finished frame is BOT field. This bit can be referred in case of AUTO-FLD or AUTO-DISP. This bit is changed at the timing of VI6_WPF0_IRQ_STA.FRE.
27 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	SYS0_ACT	0h	R	WPF0 Operating Status Each bit indicates the operating or stopped state of control channel n (WPF0). 0b: WPF0 is stopped. 1b: WPF0 is operating.
7 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(12) WPF0 Interrupt Enable Register (LCDC_VSPD_VI6_WPF0_IRQ_ENB)

Each bit controls the interrupt enable of the corresponding interrupt source.

0: Interrupt disabled

1: Interrupt enabled

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0048h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UNDE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DFEE	FREE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	UNDE	0h	RW	Interrupt Enable for WPF0 Underrun in case of DU connection.
15 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	DFEE	0h	RW	Interrupt Enable for WPF0 Display List Frame End
0	FREE	0h	RW	Interrupt Enable for WPF0 Frame End

Each bit in VI6_WPFn_IRQ_STA is set to 1 when the corresponding interrupt source is generated.

VI6_WPFn_IRQ_ENB specifies whether to output an interrupt signal for the generated source. When an interrupt is disabled in this register, no interrupt signal is generated even when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1. When an interrupt is enabled in this register, an interrupt signal is output when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1.

(13) WPF0 Interrupt Status Register (LCDC_VSPD_VI6_WPF0_IRQ_STA)

The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.

[Read Access] Interrupt Status

- 0: No interrupt
- 1: Interrupt activated

[Write Access] Interrupt Clear

- 0: The interrupt status is cleared to 0
- 1: Hold the interrupt status value

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 004Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UND
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DFE	FRE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	UND	0h	RW	Interrupt Status and Clear for WPF0 Underrun in case of DU connection. This interrupt source bit is set to 1b when data underrun occurs in case of DU connection. Timing that this bit is set to 1b is at end of 1 frame.
15 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	DFE	0h	RW	Interrupt Status and Clear for WPF0 Display List Frame End This interrupt source bit is set to 1b when VSPD completes one-frame processing while the current frame in enable value stored in the display list header is 1 (refer to 9.4.5.4 Concept of VSPD Operation Starting and Stopping). When display lists are not used, this bit is not used. In this case, clear VI6_WPF0_IRQ_ENB.DFEE to 0b to mask the interrupt generation by this interrupt source. This bit can hold the most recent of two times of interrupt status at the maximum. When writing 0b to this bit, the oldest interrupt status is cleared. And the status bit becomes 0b after all interrupt statuses are cleared.
0	FRE	0h	RW	Interrupt Status and Clear for WPF0 Frame End This interrupt source bit is set to 1b when VSPD completes one-frame processing. This bit is also set to 1b when one-frame processing using a display list is completed. The interrupt status is set to 1b by any of the following conditions. a) Processing one frame is finished normally b) Software reset is issued during VSPD is processing. c) One frame's data from DU is displayed while VSPD couldn't transfer one frame data to DU. (in case of Linked with DU) This bit can hold the most recent of two times of interrupt status at the maximum. When writing 0b to this bit, the oldest interrupt status is cleared. And the status bit becomes 0b after all interrupt statuses are cleared.

VI6_WPFn_IRQ_STA indicates the state of the interrupt sources generated in the VSPD. Whether to output a VSPD interrupt when an interrupt source is generated and the corresponding bit is set to 1 is determined by the corresponding bit setting in VI6_WPFn_IRQ_ENB. While an interrupt is disabled in VI6_WPFn_IRQ_ENB, the VSPD does not output an interrupt signal even when an interrupt source is generated, but the source flag in this register is set to 1.

Note that the interrupt source bits in this register cannot be cleared by write access using a display list.

(14) Display-0 Interrupt Enable Register (LCDC_VSPD_VI6_DISP0_IRQ_ENB)

Each bit controls the interrupt enable of the corresponding interrupt source.

- 0: Interrupt disabled
- 1: Interrupt enabled

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0078h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DSTE	-	-	MAEE	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	RW	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	DSTE	0h	RW	Interrupt Enable for Display Start
7 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5	MAEE	0h	RW	Interrupt Enable for Display Read Data End
4 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(15) Display-0 Interrupt Status Register (LCDC_VSPD_VI6_DISP0_IRQ_STA)

The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.

[Read Access] Interrupt Status

0: No interrupt

1: Interrupt activated

[Write Access] Interrupt Clear

0: The interrupt status is cleared to 0

1: Hold the interrupt status value

		Access Size :		32 bits												
		Address :		<LCDC_vspd_base> + 007Ch												
		Initial Value :		0000_0000h												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DST	-	-	MAE	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	RW	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	DST	0h	RW	Interrupt Status and Clear for Display Start This bit is set to 1b when LIF module transfers the first data to the display module at the beginning of each frame. The timing depends on the output buffer status of LIF module.
7 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5	MAE	0h	RW	Interrupt Status and Clear for Display Read Data End This bit is set to 1b when all RPFs transfer the last data of the frame to LIF module. The RPF module which is not used by LIF module does not affect this bit.
4 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

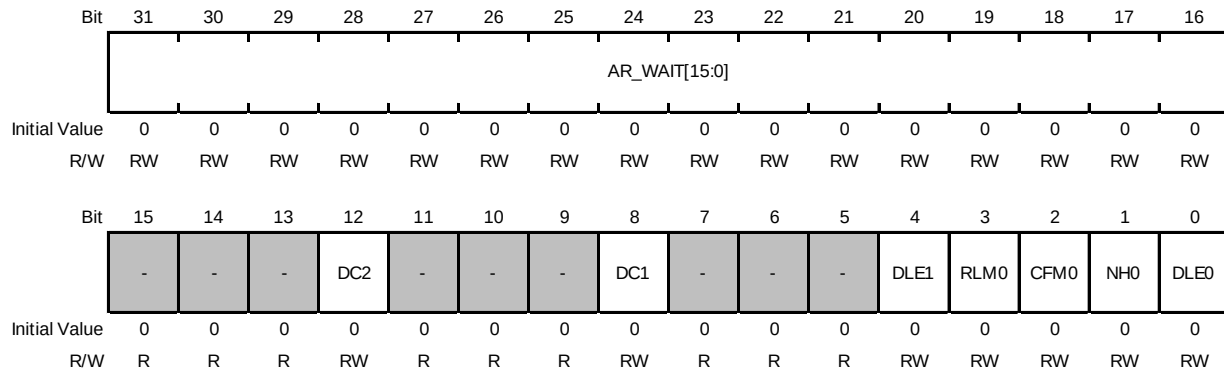
9.4.3.2.5 Display List Control Registers

(1) Display List Control Register (LCDC_VSPD_VI6_DL_CTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0100h

Initial Value : 0000_0000h



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	AR_WAIT[15:0]	0000h	RW	Display List Control Setting Always specify 256.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	DC2	0h	RW	This bit doesn't affect anything to VSPD (NOP)
11 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	DC1	0h	RW	This bit doesn't affect anything to VSPD (NOP)
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	DLE1	0h	RW	This bit doesn't affect anything to VSPD (NOP)
3	RLM0	0h	RW	Loading two plane Registers Mode for WPF0. 0b: Reserved status (VI6_CMD0.UPDHDR) is accepted by next frame auto start. Two plane registers are downloaded by next_frame_start also. 1b: Reserved status (VI6_CMD0.UPDHDR) is not accepted by next frame auto start. pNextHeader in previous frame's DLH is used for loading DisplayList. Two plane registers are not downloaded by next_frame_start also.
2	CFM0	0h	RW	Continuous Frame Mode for Header-less Display List for WPF0 This bit determines whether the next frame is automatically started or not. When the updated flag of the display list, VI6_DL_BODY_SIZE0.UPD0, is not updated, the display list of the next frame is not transferred and the same register values are used for the next frame. When the value of VI6_DL_BODY_SIZE0.UPD0 is updated, the new display list is transferred. 0b: Stopped at the end of every frame 1b: The next frame is automatically started
1	NH0	0h	RW	Header-less Display List Mode This bit is used for specifying the header-less display list mode. In case of header-less mode, the number of the display lists is 1. The address of the display body is set in VI6_DL_HDR_ADDR0 register, and the body size is set in VI6_DL_BODY_SIZE0 register. When this bit is changed, make sure that VSPD is stopped. And also make sure the following before starting VSPD. - Header Address (VI6_DL_HDR_ADDR0) - Body Size (VI6_DL_BODY_SIZE0) in case of header-less mode 0b: Use Display List Header (Normal DL Mode) 1b: Do not use Display List Header (Header-less Mode) Note1: Only WPF0 supports header-less display list. WPF1 work as the normal display list mode even if the WPF0 is set to header-less display list mode. Note2: When DLE0 bit is 0b, set 0b to NH0 bit.

Bit	Bit Name	Initial Value	R/W	Description
0	DLE0	0h	RW	Display List Enable/Disable for WPF0 Enables or disables the WPF0 display list function. When the display list function is enabled through this bit, all WPF processing channels work in display list mode. When using display lists, note the restrictions in 9.4.3.2.3 Restrictions on Access to Registers and Lookup Tables . 0b: The display list function is disabled 1b: The display list function is enabled

(2) Display List-0 Header Address Register (LCDC_VSPD_VI6_DL_HDR_ADDR0)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 0104h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
																	DL_HDR_PTR-0[31:16]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																	DL_HDR_PTR-0[15:0]
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DL_HDR_PTR-0[31:0]	0000_0000h	RW	Display List-0 Header Address These bits specify the address of the display list header to be read for display list-n in 16-byte units (the lower-order four bits are read-only). When WPF0 is first started in display list mode, the display list header is loaded from the address specified in this register. After loading of the header is completed, the register value of the display list address is updated to the next header address stored in the loaded header to prepare for loading of the next display list header. After that, this header address updating is repeated. A value from 0000 0000h to FFFF FFF0h can be specified.

(3) Display List-0 Data Swapping Register (LCDC_VSPD_VI6_DL_SWAP0)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 0114h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	IND	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	-	LWS	WDS	BTS	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31	IND	0h	RW	Enabling independent swap setting per WPF. This bit is available only for VI6_DL_SWAP1. This bit is reserved for VI6_DL_SWAP0. 0b: Display list swap for WPF1 is specified by LWS, WDS and BTS in VI6_DL_SWAP0. 1b: Display list swap for WPF1 is specified by LWS, WDS and BTS in VI6_DL_SWAP1
30 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	LWS	0h	RW	Display List Data Swapping in long word Units The effect of this bit setting is defined in Table 9.4-8 . 0b: Data swapping in long word (32-bit) units is disabled 1b: Data swapping in long word (32-bit) units is enabled
1	WDS	0h	RW	Display List Data Swapping in Word Units The effect of this bit setting is defined in Table 9.4-8 . 0b: Data swapping in word (16-bit) units is disabled 1b: Data swapping in word (16-bit) units is enabled
0	BTS	0h	RW	Display List Data Swapping in Byte Units The effect of this bit setting is defined in Table 9.4-8 . 0b: Data swapping in byte (8-bit) units is disabled 1b: Data swapping in byte (8-bit) units is enabled

Table 9.4-8 shows the data order before and after swapping according to the long word, word, and byte swapping settings.

When data order in memory for each format is the same as **Table 9.4-34**, set 111b to {LWS, WDS, BTS}. If data order is not the same as the definition, change data order within 8byte unit by these bits as shown in **Table 9.4-8**.

Table 9.4-8 Changing Data Order According to Display List Swap Register

Data order in memory									*_LWS	*_WDS	*_BTS
Byte address	8n+0	8n+1	8n+2	8n+3	8n+4	8n+5	8n+6	8n+7			
Data	0	1	2	3	4	5	6	7	1	1	1
	1	0	3	2	5	4	7	6	1	1	0
	2	3	0	1	6	7	4	5	1	0	1
	3	2	1	0	7	6	5	4	1	0	0
	4	5	6	7	0	1	2	3	0	1	1
	5	4	7	6	1	0	3	2	0	1	0
	6	7	4	5	2	3	0	1	0	0	1
	7	6	5	4	3	2	1	0	0	0	0



Data order defined in Table 9.4-34								
Byte address	8n+0	8n+1	8n+2	8n+3	8n+4	8n+5	8n+6	8n+7
Data	0	1	2	3	4	5	6	7

(4) Extended Display List-0 Control Register (LCDC_VSPD_VI6_DL_EXT_CTRL0)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 011Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	NWE
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	POLINT[5:0]					-	-	DLPRI	EXPRI	-	-	-	-	EXT
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	RW	RW	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	NWE	0h	RW	No Wait for Polling When this bit is set to 1b, the polling condition for extended display lists is always assumed to be true.
15 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 8	POLINT[5:0]	00h	RW	Extended Display List Command Control Always specify 2h.
7 to 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5	DLPRI	0h	RW	Display List Control 0 Always specify 1b.
4	EXPRI	0h	RW	Display List Control 1 Always specify 0b.
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	EXT	0h	RW	Extended Display List for WPF0 Enables or disables the extended display list function. When extended display lists are used, the display list header size is 96 bytes; when they are not used, the header size is 80 bytes. 0b: No extended display lists are used 1b: Extended display lists are used [Note] When using extended display lists, be sure to also use normal display list mode (VI6_DL_CTRL.DLE0); executing only extended display lists is not possible. When the header-less display list mode is activated, this bit should be set to 0. The extended display list cannot be used with the header-less display list mode. Extended Display List is available only for VSPD/LIFO to realized AUTO-FLD or AUTO-DISP.

(5) Display List Body Size Register-0 (LCDC_VSPD_VI6_DL_BODY_SIZE0)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0120h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	UPD0	-	-	-	-	-	-	-	BS0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BS0[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	UPD0	0h	RW	Update Flag This bit controls the download of the display list at the next downloading timing in case that the header-less display list mode is used. Set 1b to this bit when display list need to be downloaded for next frame. This bit is automatically cleared to 0b after VI6_DL_HDR_ADDRn and VI6_DL_BODY_SIZE0.BS0 is downloaded in H/W side. When this bit is set to 1b, the value of VI6_DL_HDR_ADDR0 and VI6_DL_BODY_SIZE0.BS0 should not be changed. 0b: Updating display list for the next frame is not reserved 1b: Updating display list for the next frame is reserved
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16 to 0	BS0[16:0]	00000h	RW	Header-less Display List Body Size (WPF0) These bits are used for specifying the body size of the display list in case of header-less display list mode. The unit of the size is byte. The value should be set in multiples of 8.

See **9.4.5.8.2 Controlling two register planes using display lists** for detail.

(6) Display List-0 Header Reference Address Register (LCDC_VSPD_VI6_DL_HDR_REF_ADDR0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0130h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DL_HDR_REF_PTR-0[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL_HDR_REF_PTR-0[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DL_HDR_REF_PTR-0[31:0]	00000000h	R	Display List-n reference Header Address Following value is read out from the VI6_DL_HDR_REF_ADDR0 for each period. (1) When H/W is reading display list from external memory, header address of the display list referred by VSPD-H/W is read out. (2) When H/W is not reading display list from external memory, the value of VI6_DL_HDR_ADDR0 is read out. For details, refer to 9.4.5.8.1 Operation flow of VSPD and DU .

(7) Display List-0 Wake Up Counter Register (VI6_DL_WUPCNT0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0158h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	Bit Name															
31 to 0	-															
		Initial Value		R/W		Description										
		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.										

9.4.3.2.6 RPF Control Registers

(1) RPF_n Basic Read Size Registers (LCDC_VSPD_VI6_RPF_n_SRC_BSIZE) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0300h (n = 0) <LCDC_vspd_base> + 0400h (n = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BHSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BVSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	BHSIZE[12:0]	0000h	RW	Horizontal Size of RPF Basic Read Area These bits specify the horizontal size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units. A value from 1 to 8190 can be specified. Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.BHSIZE).
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	BVSIZ[12:0]	0000h	RW	Vertical Size of RPF Basic Read Area These bits specify the vertical size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:0, specify the size in 2-pixel units. A value from 1 to 8190 can be specified. Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.EVSIZE).

Figure 9.4-6 shows the relationship between the basic read size and extended read size. The RPF reads data from the source memory area specified by the basic read size. The RPF repeats reading the basic read area in the horizontal and vertical directions up to the extended read size and sends the read data to the processing modules in the VSPD.

For basic read size reading, the reading start address, called the RPF_n source image storing address, should be specified in VI6_RPF_n_SRCM_ADDR_*. In the memory area where the basic read area image is stored, the distance (number of bytes) between addresses for lines n and n + 1 of two-dimensional image data, called the memory stride, should be specified in VI6_RPF_n_SRCM_PSTRIDE.

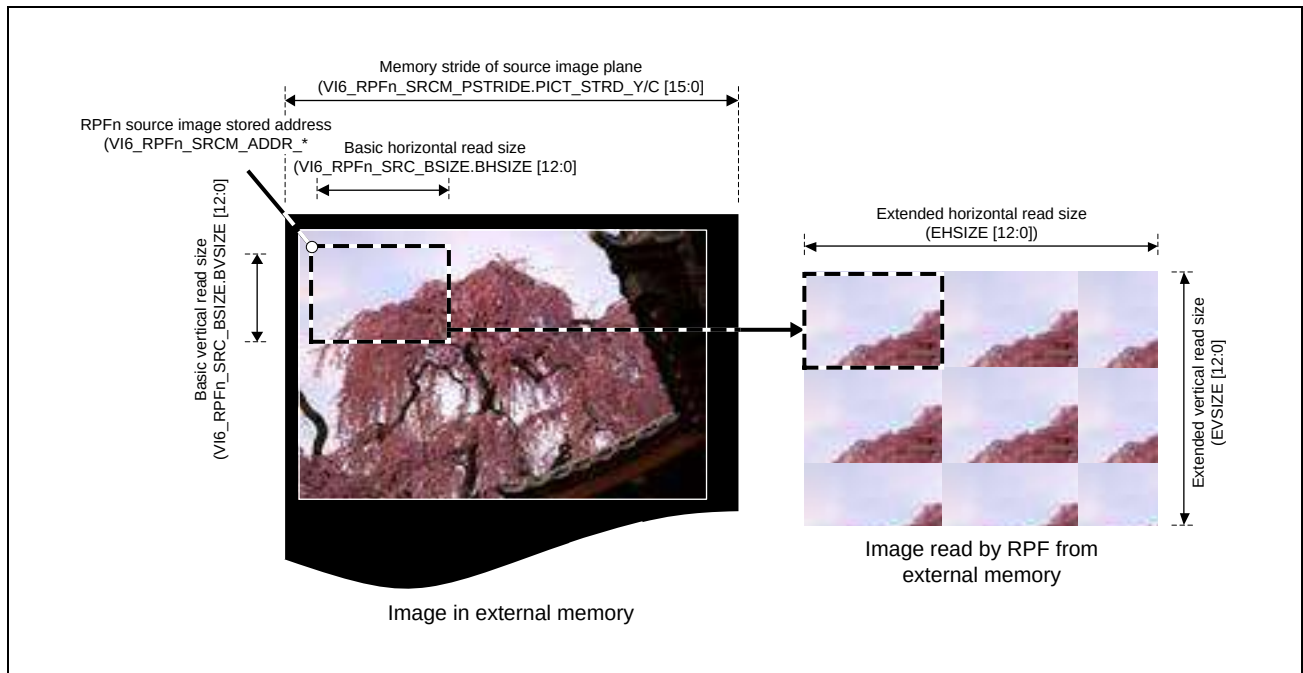


Figure 9.4-6 Relationship between Basic Read Size and Extended Read Size

Refer also to the following sections.

- (2) RPFn Extended Read Size Registers (LCDC_VSPD_VI6_RPFn_SRC_ESIZE)
- (13) RPFn Source Picture Memory Stride Setting Registers (LCDC_VSPD_VI6_RPFn_SRCM_PSTRIDE)
- (15) RPFn Source Y/RGB Address Registers (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_Y)
- (16) RPFn Source Chroma Address Registers 0 (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_C0)
- (17) RPFn Source Chroma Address Registers 1 (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_C1)

(2) RPFn Extended Read Size Registers (LCDC_VSPD_VI6_RPFn_SRC_ESIZE)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0304h (n = 0)
<LCDC_vspd_base> + 0404h (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	EHSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	EVSIZ[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	EHSIZE[12:0]	0000h	RW	RPF Extended Horizontal Read Size These bits specify the horizontal size of the extended read area to which the RPFn reads data from the external RAM. As shown in Figure 9.4-6 , the basic read area image is repeatedly placed in the extended read area; in the EHSIZE bits, specify a value not smaller than the horizontal size of the basic read area. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units (an even value). A value from 1 to 1920 (8190) can be specified.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	EVSIZ[12:0]	0000h	RW	RPF Extended Vertical Read Size These bits specify the vertical size of the extended read area to which the RPFn reads data from the external RAM. As shown in Figure 9.4-6 , the basic read area image is repeatedly placed in the extended read area; in the EVSIZE bits, specify a value not smaller than the vertical size of the basic read area. When the input format is YCbCr4:2:0, specify the size in 2-pixel units (an even value). A value from 1 to 1920 (8190) can be specified.

VI6_RPFn_SRC_ESIZE specifies the extended size for RPFn. The extended horizontal and vertical sizes should be equal to or greater than the basic sizes specified in VI6_RPFn_SRC_BSIZE. The RPF internal data processing described later and image processing described in **9.4.3.2.8 DPR Control Registers** and later sections are all applied to the image in the extended read size shown on the right side in **Figure 9.4-6**.

(3) RPFn Input Format Registers (LCDC_VSPD_VI6_RPFn_INFMT) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0308h (n = 0)
<LCDC_vspd_base> + 0408h (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VIR	-	-	-	-	-	-	-	-	-	-	-	CIPM
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPYCS	SPUVS	CEXT[1:0]		RDTM[2:0]			CSC	-	RDFMT[6:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	VIR	0h	RW	Virtual Input Enable Enables or disables the virtual input function of the RPF0. The image to be processed by the RPF0 is usually read from the external memory by the MAU. Instead of this input, the virtual input function generates a single-color image within the RPF0 and sends it to the modules in VSPD. When the virtual input function is enabled, the fixed value specified in VI6_RPF0_VRTCOL_SET is used as the input to the RPF0. While the virtual input function is enabled, data is not read from the external memory; that is, the α plane is not read and the IROP calculation thus cannot be executed. In this case, set VI6_RPF0_ALPH_SEL.ASEL to 4. Neither the color space conversion through CSC nor the color keying described in 9.4.3.2.6 (11) RPFn Color Keying Control Registers (VI6_RPFn_CKEY_CTRL) can be used. 0b: RPF0 uses general input. 1b: RPF0 uses virtual input.
27 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	CIPM	0h	RW	Horizontal Chrominance Interpolation Method Setting Image data is processed in the YCbCr444 format inside VSPD in case of YCbCr color space. When the chrominance format of the input image is YCbCr422 or YCbCr420, data is up-sampled as shown in Figure 9.4-7 for internal processing. This bit specifies the method of up-sampling for this purpose. 0b: The nearest-neighbor method is used for horizontal chrominance interpolation. 1b: The bilinear method is used for horizontal chrominance interpolation.
15	SPYCS	0h	RW	RPF Input Mode Setting 1 When the input format is YUY2, set this bit to 1b and set the RDFMT bits to 71 (47h). When the input format is YVYU, set this bit and the SPUVS bit to 1b and set the RDFMT bits to 71 (47h). In other cases, set this bit to 0b.
14	SPUVS	0h	RW	RPF Input Mode Setting 2 When the input format is NV61, set this bit to 1b and set the RDFMT bits to 65 (41h). When the input format is NV21, set this bit to 1b and set the RDFMT bits to 66 (42h). When the input format is YVYU, set this bit and the SPYCS bit to 1b and set the RDFMT bits to 71 (47h). In other cases, set this bit to 0b.
13, 12	CEXT[1:0]	0h	RW	Lower-Bit Color Data Extension Method Setting When an RGB input format where each color component is expressed in less than 8 bits are selected from Table 9.4-10 through the RDFMT bits, VSPD internally extends each color component to 8 bits before using the data. These bits select this extension method. 00b: Lower-order bits of color data are extended with 0. 01b: Upper-order bits of color data are copied to the lower -order bits. 10b: Lower-order bits of color data are extended with 0. The maximum value is limited to 0xFF. 11b: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	RDTM[2:0]	0h	RW	CSC Conversion Expression Setting These bits select the expression used for color space conversion. The conversion direction is RGB → YCbCr when RGB is selected through the RDFMT bits; the direction is YCbCr → RGB when YCbCr is selected. 0h: BT.601 YCbCr [16,235/240] ↔ RGB [0,255] 1h: BT.601 YCbCr [0,255] ↔ RGB [0,255] 2h: BT.709 YCbCr [16,235/240] ↔ RGB [0,255] 3h: BT.709 YCbCr [16,235/240] ↔ RGB [16,235] 4h to 7h: Setting prohibited
8	CSC	0h	RW	Color Space Conversion Enable Enables or disables color space conversion between YCbCr and RGB to be executed in RPF0. The characteristics of color space conversion are determined by the RDTM bit setting.* ¹ When using the virtual input (VIR = 1b), specify 0b. 0b: Color space conversion is disabled. 1b: Color space conversion is enabled.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 0	RDFMT[6:0]	00h	RW	RPF Input Image Format Setting These bits select the format of the image input from the external RAM to the RPF0. Select a value corresponding to the desired format from those shown in Table 9.4-10 , Table 9.4-11 , and Table 9.4-12 . When the virtual input function is used (VIR = 1b), the color information for the virtual input should be specified in VI6_RPF0_VRTCOL_SET. If this color information is in the RGB format, set the RDFMT bits to 19. If the color information is in the YCbCr format, set these bits to 64. [Note] Number of input pixels When YCbCr4:2:2 is selected through the RDFMT bits, the horizontal size of the input image should be specified in 2-pixel units. When YCbCr4:2:0 is selected, the vertical and horizontal sizes should be specified in 2-pixel units. Observe these restrictions when specifying the image size in VI6_RPF0_SRC_BSIZE and VI6_RPF0_SRC_ESIZE.

Note 1. Note on color space settings

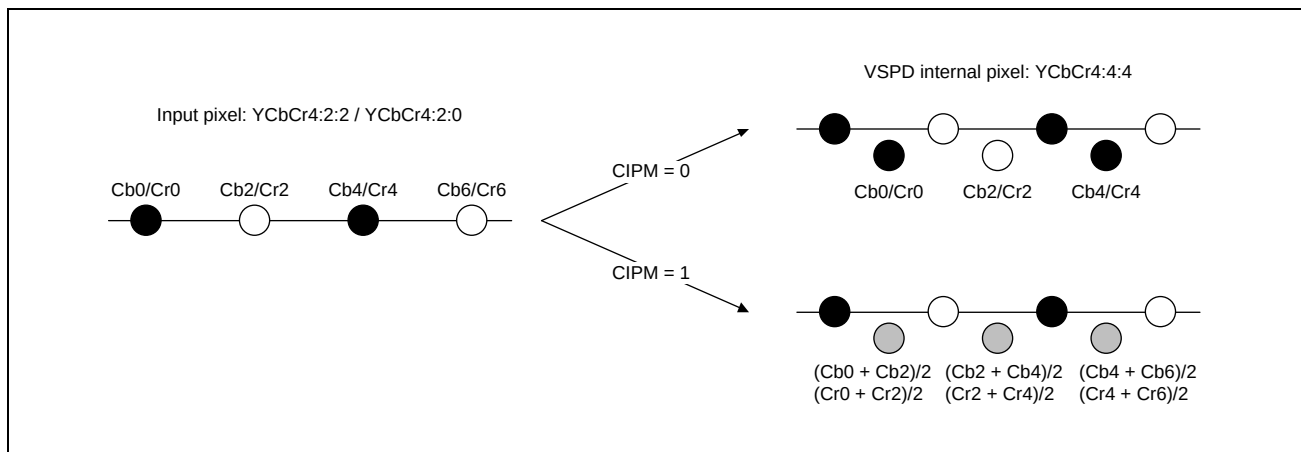


Figure 9.4-7 Chrominance Interpolation Methods Selectable through CIPM Setting

The color space for the image output from the RPF to VSPD internal modules is determined by the combination of the color space for the image input to the RPF, which is selected through the VI6_RPFn_INFMT.RDFMT setting, and the enabled or disabled state of the color space conversion function, which is selected through the VI6_RPFn_INFMT.CSC setting (**Table 9.4-9**). For example, when the image input to the RPF is in the YCbCr format, the RPF outputs data to VSPD internal modules in the YCbCr format if color space conversion is disabled through the CSC bit, and the RPF outputs data in the RGB format if color space conversion is enabled. When the image input to the RPF is in the RGB format, the relationship between the output format and the color space conversion setting is the opposite of the YCbCr case. For some VSPD internal modules, the YCbCr format is recommended for image processing because of the characteristics of the processing, or the same color space needs to be specified between multiple RPF outputs. In these

cases, set VI6_RPFn_INFMT.RDFMT and VI6_RPFn_INFMT.CSC appropriately so that the RPFs can output the required color space according to the color space conditions described above.

Table 9.4-9 RPFn Input Color Space and Output Color Space

RPFn Input Color Space (VI6_RPFn_INFMT.RDFMT)		Color Space Conversion Setting (VI6_RPFn_INFMT.CSC)		RPFn Output Color Space
RGB	(00h to 3Fh)* ¹	Disabled	(0)* ¹	RGB
		Enabled	(1)* ¹	YCbCr
YCbCr	(40h to 7Fh)* ¹	Disabled	(0)* ¹	YCbCr
		Enabled	(1)* ¹	RGB

Note 1. Value specified in the register

A color space conversion function equivalent to that in the RPFn is also provided by the WPF. As shown in **Table 9.4-9**, the color space (YCbCr or RGB) output from the RPF becomes the input format for the WPF. Here, the color space of the output image obtained by the color space conversion function of the WPF must match the color space of the format specified through VI6_WPFn_OUTFMT.WRFMT.

Figure 9.4-8 shows the relationship between the input/output format and color space. The input color space for the RPF is determined when the input image format for the RPF is specified through the RDFMT bits. The color space for the image output from the RPF to subsequent VSPD internal modules depends on the combination of the RPF input format and CSC (color space conversion function) enabled or disabled state in the RPF as shown in **Table 9.4-9** and **Figure 9.4-8**. The user should first determine whether the image processing in the VSPD is done in YCbCr or RGB, and then specify the RPF input format and CSC enabled or disabled state to obtain the desired color space. The color space of the RPF output image is also that of the WPF input image; the color space of the data output from the WPF to the outside of VSPD depends on the combination of the WPF input color space and the enabled or disabled state of the CSC implemented in the WPF as shown in **Figure 9.4-8**. The color space of the WPF output image must match that of the WPF output format (determined by VI6_WPFn_OUTFMT.WRFMT). For example, in the flow shown in **Figure 9.4-8**, YCbCr should not be specified as the WPF output format regardless of the fact that the color space of the WPF output image is in RGB format.

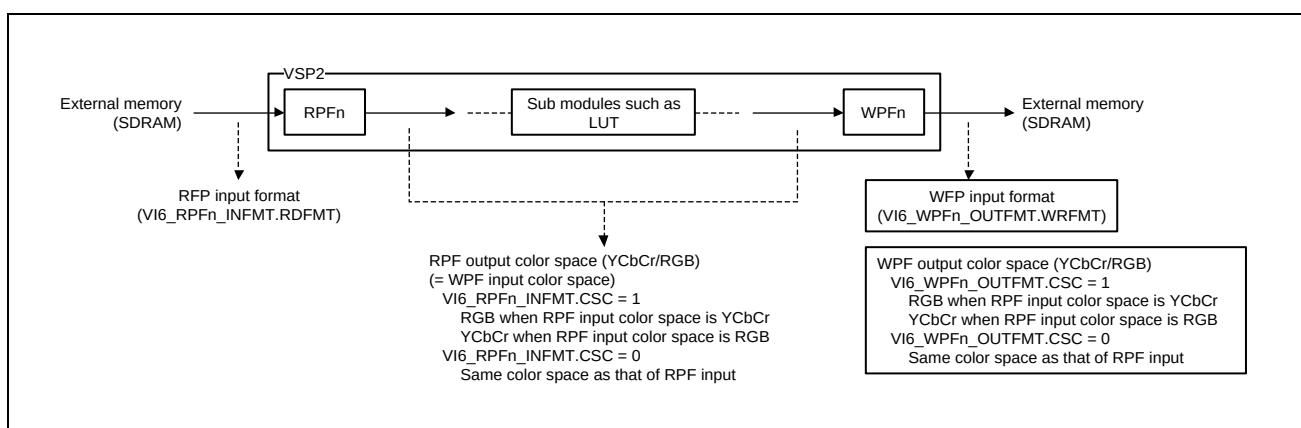


Figure 9.4-8 Relationship between Input/output Format and Color Space

Table 9.4-10 Packed Formats for RPF Input (1/2)

RDFMT[6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																																		
			n								n+1								n+2								n+3										
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
00h	8	—	R0	R0	R0	G0	G0	G0	B0	B0	R1	R1	R1	G1	G1	G1	B1	B1	R2	R2	R2	G2	G2	G2	B2	B2	R3	R3	R3	G3	G3	G3	B3	B3			
01h	12	—					R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0					R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1			
02h							R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0					R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1			
03h	—	—	Reserved								Reserved								Reserved								Reserved										
04h	15	—		R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0		R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1		
05h				R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0		R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	
06h	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B3		
07h	18	—	A	A	A	A	A	A	A								R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0		
08h				R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0									A	A	A	A	A	A	A	A	
09h										R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A	
0Ah										R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	B0							
0Bh					A	A	A	A	A	A	A			R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0	B0	
0Ch							R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A	A	
0Dh																																					
0Eh																																					
0Fh																																					
				0							R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0									R1	R1
		1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1								R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2		
		2	G2	G2	B2	B2	B2	B2	B2	B2							R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3		
10h		0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0									R1	R1	R1	R1	R1	R1	G1	G1			
		1	G1	G1	G1	G1	B1	B1	B1	B1	B1							R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2	B2	B2	B2	B2		
		2	B2	B2														R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3		
11h		0			R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0			R1	R1	R1	R1	R1	R1			
		1			G1	G1	G1	G1	G1	G1			B1	B1	B1	B1	B1	B1			R2	R2	R2	R2	R2	R2			G2	G2	G2	G2	G2	G2			
		2			B2	B2	B2	B2	B2	B2								R3	R3	R3	R3	R3	R3			G3	G3	G3	G3	G3	G3			B3	B3		
12h		0	R0	R0	R0	R0	R0	R0			G0	G0	G0	G0	G0	G0			B0	B0	B0	B0	B0	B0			R1	R1	R1	R1	R1	R1					
		1	G1	G1	G1	G1	G1	G1			B1	B1	B1	B1	B1	B1			R2	R2	R2	R2	R2	R2			G2	G2	G2	G2	G2	G2					
		2	B2	B2	B2	B2	B2	B2			R3	R3	R3	R3	R3	R3			G3	G3	G3	G3	G3	G3			B3	B3	B3	B3	B3	B3					
13h	24	—	A	A	A	A	A	A	A	A	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0			
14h				R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	A	A	A	A	A	A	A	A		
15h				0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1	R1		
		1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2	G2			
		2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3	B3			
16h	18	—								R0	R0	R0	R0	R0	R0	G0	G0	G0								G0	G0	B0	B0	B0	B0	B0	B0	B0			
17h																																					
18h	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1	B1			
			1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2	G2	G2		
			2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3	R3	
19h	12	—	A	A	A	A	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	A	A	A	A	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	B1		
1Ah				R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	A	A	A	A	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	A	A	A	A	A	
1Bh				R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	A	
1Ch	15	—	A	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1		
1Dh				R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	A	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	A		
1Eh				B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	A	A	A	A	B1	B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1	A	A	A	A	
1Fh	15	—	A	B0	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	R0	A	B1	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1		
20h				B0	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	R0	A	B1	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	A		
21h				B0	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	R0	A	B1	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	A		
22h	18	0			B0	B0	B0	B0	B0	B0			G0	G0	G0	G0	G0	G0			R0	R0	R0	R0	R0	R0			B1	B1	B1	B1	B1	B1			
			1			G1	G1	G1	G1	G1	G1			R1	R1	R1	R1	R1	R1			B2	B2	B2	B2	B2	B2			G2	G2	G2	G2	G2	G2		
			2			R2	R2	R2	R2	R2	R2			B3	B3	B3	B3	B3	B3			G3	G3	G3	G3	G3	G3			R3	R3	R3	R3	R3	R3		
23h	24	—	A	A	A	A	A	A	A	A	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	R0				
24h to 2Fh	16	—																	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0			
24h to 2Fh	—	—	Reserved								Reserved								Reserved								Reserved										

Table 9.4-10 Packed Formats for RPF Input (2/2)

RDFMT[6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																							
			n								n+1								n+2							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
30h	*1	*1	*1								*1								*1							
31h to 3Fh	—	—	Reserved								Reserved								Reserved							

Note 1. When the RDFMT[6:0] bits are set to 30h, refer to **Table 9.4-11** for the additional settings and the packed formats

Table 9.4-11 Packed formats of RGB10, RGB10A2, A2RGB10

RDFMT[6:0]	Format name	BYPP_M1[2:0] CPOS[31:0] CLEN[31:0]	upper row - address / bottom row - bit field																							
			n								n+1								n+2							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
30h	RGB10	3h 000A1400h 0A0A0A00h	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0
30h	RGB10A2	3h 000A141Eh 0A0A0A02h	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0
30h	A2RGB10	3h 020C1600h 0A0A0A02h	A0	A0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0

Table 9.4-12 Packed YCbCr Formats for RPF Input

RDFMT[6:0]	Packed YCbCr Input Format	Reference
40h	YCbCr4:4:4 semi-planar	Figure 9.4-9 ^{*4}
41h	YCbCr4:2:2 semi-planar (NV16, NV61 ^{*1})	
42h	YCbCr4:2:0 semi-planar (NV12, NV21 ^{*1}) ^{*5}	
43h to 45h	Reserved	—
46h	YCbCr4:4:4 interleaved	Figure 9.4-10 ^{*4}
47h	YCbCr4:2:2 interleaved type 0 (UYVY, YUY2 ^{*2} , YVYU ^{*3})	
48h	YCbCr4:2:2 interleaved type 1	
49h	YCbCr4:2:0 interleaved ^{*6}	Figure 9.4-11 ^{*4}
4Ah	YCbCr4:4:4 planar	
4Bh	YCbCr4:2:2 planar (YV16)	
4Ch	YCbCr4:2:0 planar (YV12, I420) ^{*5}	—
4Dh to 7Eh	Reserved	
7Fh	Reserved	

Note 1. When the input format is NV61 or NV21, set the SPUVS bit to 1.

Note 2. When the input format is YUY2, set the SPYCS bit to 1.

Note 3. When the input format is YVYU, set the SPUVS bit to 1 and SPYCS bit to 1.

Note 4. **Figure 9.4-12** shows the definition of memory address for each pixel in **Figure 9.4-9** to **Figure 9.4-11**.

Note 5. Each line of C plane is read twice, so byte/pixel of YCbCr420 is 2 byte/pixel (same as YCbCr422).

Note 6. Each line of plane is read twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).

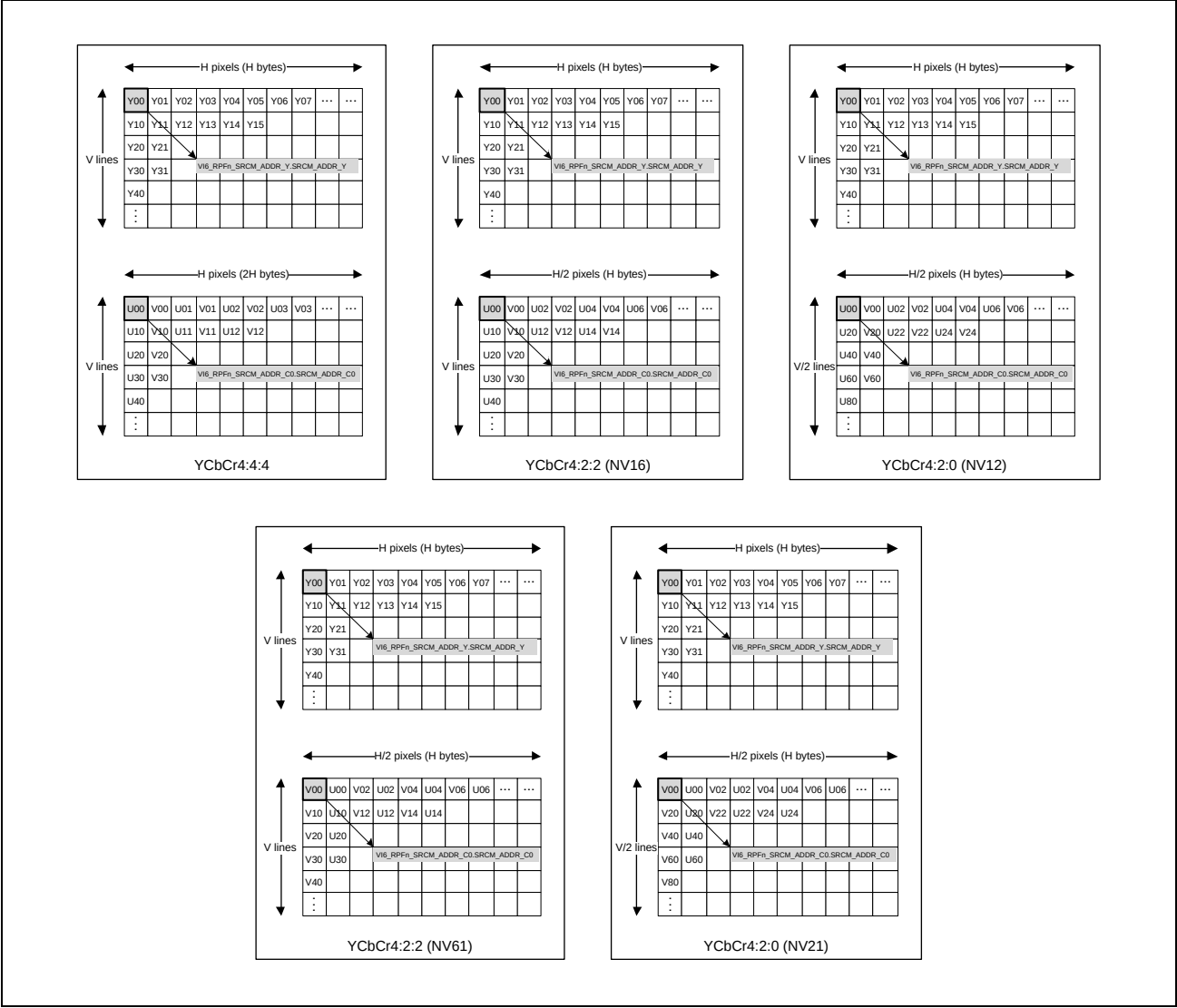


Figure 9.4-9 YCbCr Semi-Planar Formats*1

Note 1. This figure is for 8 bpc.

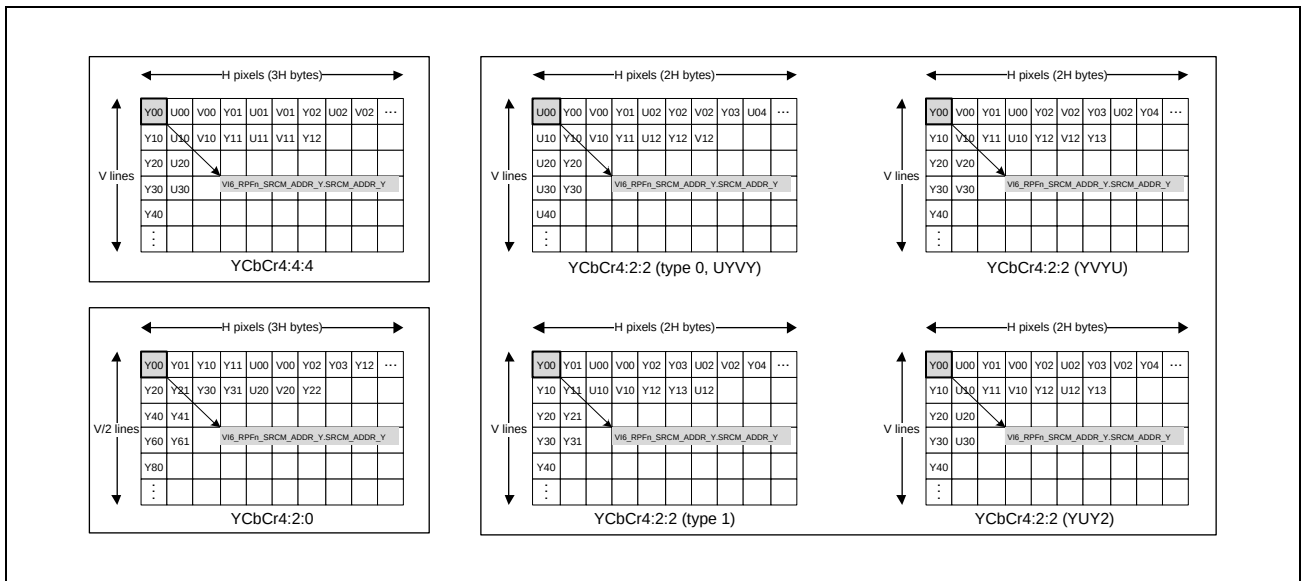


Figure 9.4-10 YCbCr Interleaved Formats*1

Note 1. This figure is for 8 bpc.

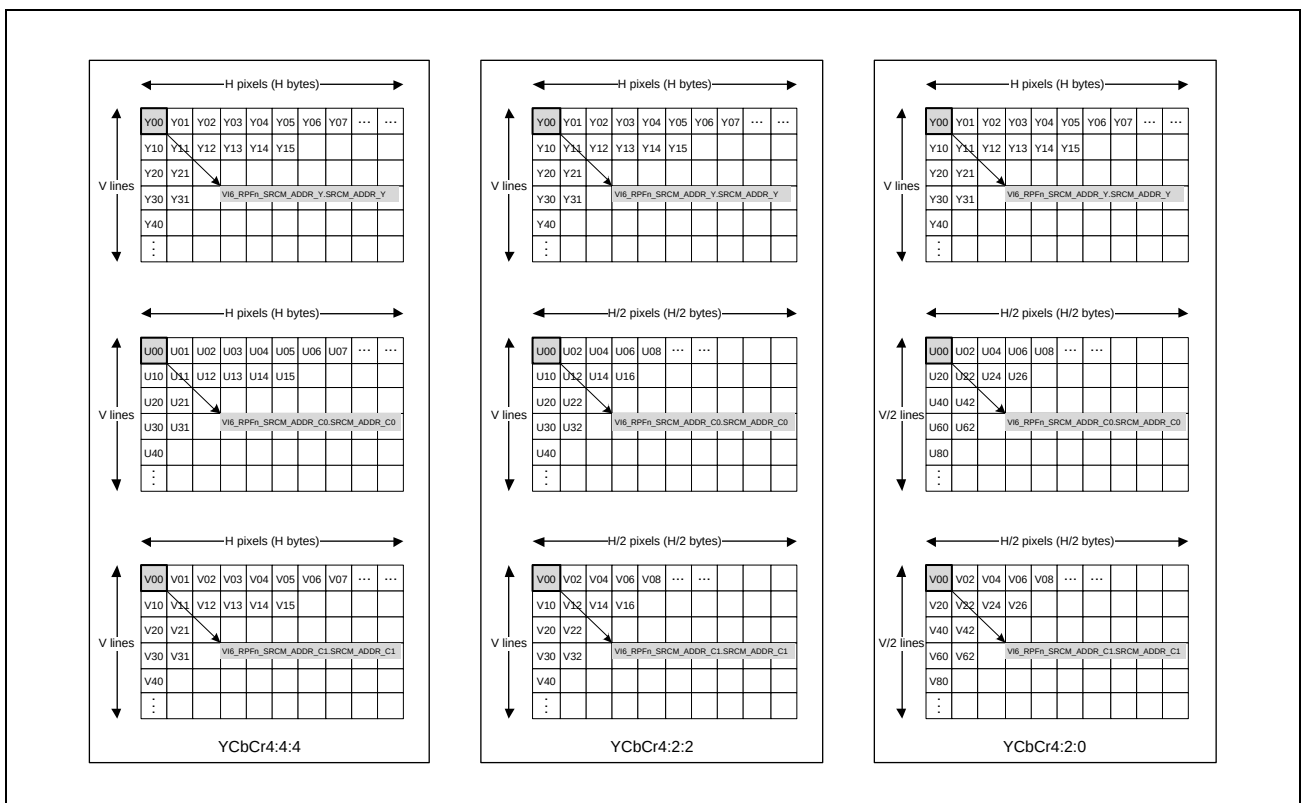


Figure 9.4-11 YCbCr Planar Formats*1

Note 1. This figure is for 8 bpc.

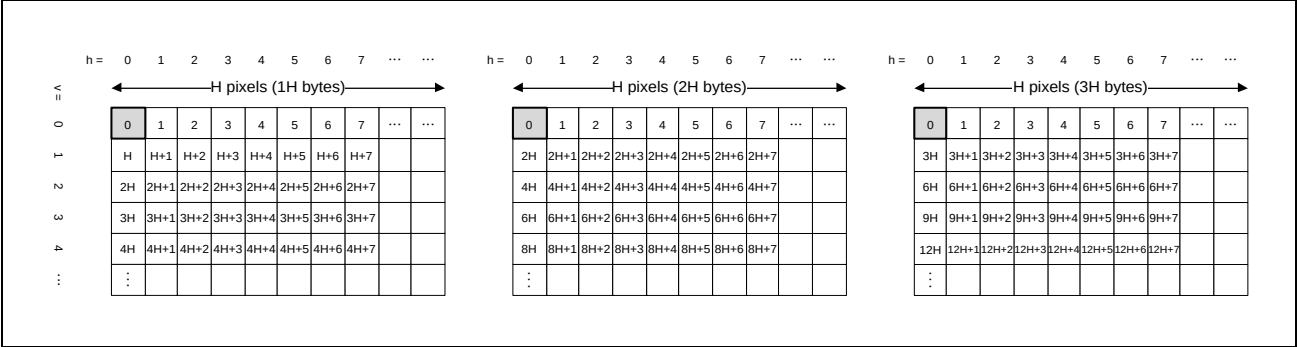


Figure 9.4-12 Memory Address for the Corresponding Pixel's Position

Legend in **Figure 9.4-12**

- H is horizontal image size in pixel unit.
- 1H, 2H, and 3H correspond to the case where 1 pixel has 1 byte, 2 bytes, and 3 bytes of data, respectively.

(4) RPFn Data Swapping Registers (LCDC_VSPD_VI6_RPFn_DSWAP) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 030Ch (n = 0)
<LCDC_vspd_base> + 040Ch (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	A_LLS	A_LWS	A_WDS	A_BTS	-	-	-	-	P_LLS	P_LWS	P_WDS	P_BTS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11	A_LLS	0h	RW	α Plane Data Swapping in LONG LW+H23:R28ORD Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in LONG LWORD (64-bit) units is disabled 1b: Data swapping in LONG LWORD (64-bit) units is enabled
10	A_LWS	0h	RW	α Plane Data Swapping in long word Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in long word (32-bit) units is disabled 1b: Data swapping in long word (32-bit) units is enabled
9	A_WDS	0h	RW	α Plane Data Swapping in Word Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in Word (16-bit) units is disabled 1b: Data swapping in Word (16-bit) units is enabled
8	A_BTS	0h	RW	α Plane Data Swapping in Byte Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in Byte (8-bit) units is disabled 1b: Data swapping in Byte (8-bit) units is enabled
7 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	P_LLS	0h	RW	Picture Plane Data Swapping in LONG LWORD Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in LONG LWORD (64-bit) units is disabled 1b: Data swapping in LONG LWORD (64-bit) units is enabled This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases.
2	P_LWS	0h	RW	Picture Plane Data Swapping in long word Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in long word (32-bit) units is disabled 1b: Data swapping in long word (32-bit) units is enabled This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases.
1	P_WDS	0h	RW	Picture Plane Data Swapping in Word Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in Word (16-bit) units is disabled 1b: Data swapping in Word (16-bit) units is enabled This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases.

Bit	Bit Name	Initial Value	R/W	Description
0	P_BTS	0h	RW	Picture Plane Data Swapping in Byte Units The effect of this bit setting is defined in Table 9.4-14 . 0b: Data swapping in Byte (8-bit) units is disabled 1b: Data swapping in Byte (8-bit) units is enabled This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases.

When the virtual input function of the RPFn is used (VI6_RPFn_INFMT.VIR = 1), this register setting is ignored. Swapping of RPF input data can be specified separately for the α plane and picture plane.

Table 9.4-13 shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

When data order in memory for each format is the same as **Table 9.4-10** for RGB format and **Figure 9.4-9** to **Figure 9.4-12** for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is not the same as the definition, change data order within 16byte unit by these bits as shown in **Table 9.4-13**.

Table 9.4-13 Changing Data Order According to Swap Register

Data order in memory																*_LLS	*_LWS	*_WDS	*_BTS	
Byte address	16n+0	16n+1	16n+2	16n+3	16n+4	16n+5	16n+6	16n+7	16n+8	16n+9	16n+10	16n+11	16n+12	16n+13	16n+14					16n+15
Data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	1	1	1
	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14	1	1	1	0
	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13	1	1	0	1
	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12	1	1	0	0
	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11	1	0	1	1
	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10	1	0	1	0
	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9	1	0	0	1
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	1	0	0	0
	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	0	1	1	1
	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6	0	1	1	0
	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5	0	1	0	1
	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4	0	1	0	0
	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3	0	0	1	1
	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2	0	0	1	0
	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1	0	0	0	1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0



Table 9.4-10 for RGB format and Figure 9.4-9 to Figure 9.4-12 for YCbCr format																
Byte address	16n+0	16n+1	16n+2	16n+3	16n+4	16n+5	16n+6	16n+7	16n+8	16n+9	16n+10	16n+11	16n+12	16n+13	16n+14	16n+15
Data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(5) RPFn Display Location Registers (LCDC_VSPD_VI6_RPFn_LOC) (n = 0, 1)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 0310h (n = 0) <LCDC_vspd_base> + 0410h (n = 1)															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	HCOORD[12:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	VCOORD[12:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	HCOORD[12:0]	0000h	RW	Horizontal Coordinate of Sublayer Display Location on Master Layer These bits specify the left-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the horizontal coordinate of the location in pixel units with the left-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0b. If the sublayer extends beyond the master layer according to the HCOORD setting, the extended section is cut off at the right end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the right end of the master layer. A value from 0 to 8189 can be specified.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	VCOORD[12:0]	0000h	RW	Vertical Coordinate of Sublayer Display Location on Master Layer These bits specify the top-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the vertical coordinate of the location in pixel units with the top-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0b. If the sublayer extends beyond the master layer according to the VCOORD setting, the extended section is cut off at the bottom end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the bottom end of the master layer. A value from 0 to 8189 can be specified.

Figure 9.4-13 shows an example of RPF1 offsets with respect to master layer RPF0. Although this figure only shows sublayers RPF1, specify offsets for all RPFs other than the master layer in the same way as shown in this example.

Whether an RPFn is the master layer or a sublayer is determined through the selection of the source RPF for WPFn (the VI6_WPFn_SRCRPF setting). For details, refer to **(1) WPF0-Source-RPF Register (LCDC_VSPD_VI6_WPF0_SRCRPF)**.

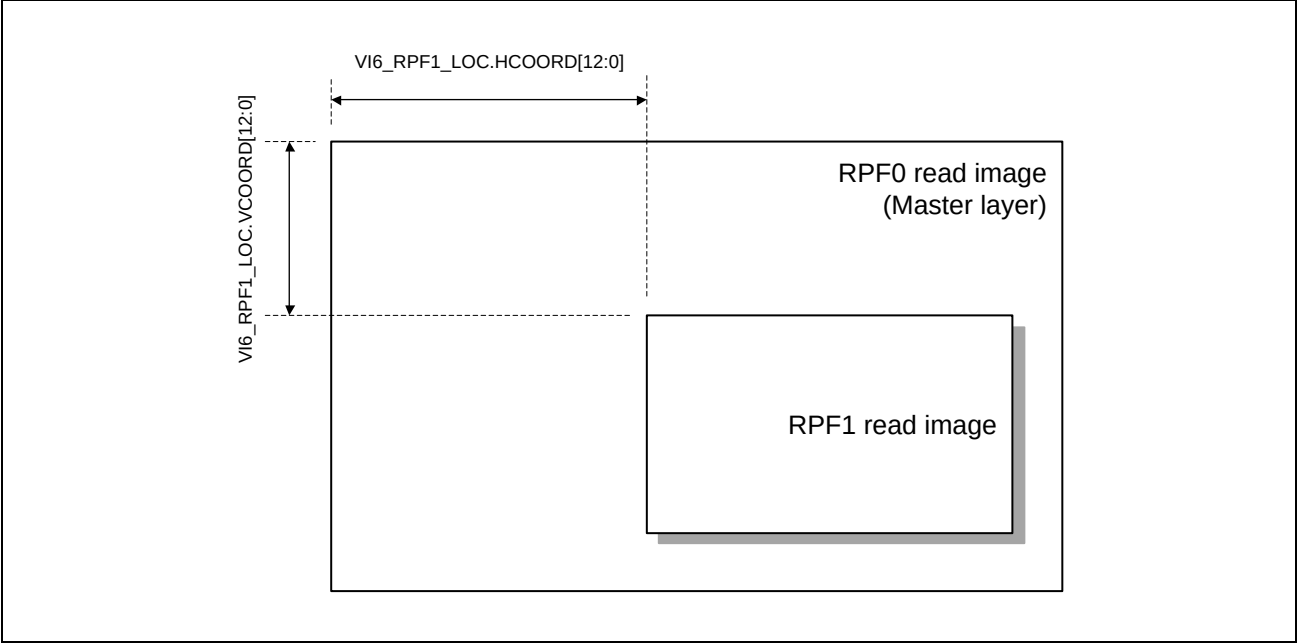


Figure 9.4-13 RPF1 Offsets from Master Layer

(6) RPFn α Plane Selection Control Registers (LCDC_VSPD_VI6_RPFn_ALPH_SEL) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0314h (n = 0)
<LCDC_vspd_base> + 0414h (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	ASEL[2:0]			IROP[3:0]				BSEL	-	-	-	AEXT[1:0]		-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	RW	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALPHA1[7:0]								ALPHA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 28	ASEL[2:0]	0h	RW	α Format and Processing Method Select These bits select how to handle the α value to be used. The RPF handles two types of α value; 8-bit and 1-bit values. When a 1-bit α value is used, VSPD assumes that the 1-bpp α value for each pixel is stored in the order from MSB to LSB in each byte (big endian). The α value is used as either transparency information or mask information. Transparency information is included in the α plane read from external memory when the ASEL bits are set to 1 or 3 and in the α value stored in the packed RGB bit field when these bits are set to 0 or 2. The α value as transparency information is sent as the destination value to the IROP as shown in Figure 9.4-14 and then output to the subsequent modules. The output α value is used, for example, for blending in the BRS. The α value as mask information is used for IROP operation in the RPF. The mask information is included in the α plane read from external RAM when the ASEL bits are set to 0 or 2 and the source value is used in IROP operation (IROP setting other than 0, 5, 10, or 15). This α value is sent as the source value to the IROP as shown in Figure 9.4-14. Note that the α value selected through the ASEL bits has a lower priority than the VI6_RPFn_CKEY_SET*.AP* value replaced through the color keying function. When the color keying function is used, the α value may be replaced with the VI6_RPFn_CKEY_SET*.AP* value regardless of the ASEL bit setting. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1b), specify 4. 0h: 1, 4, or 8-bit packed α + plane α (IROP != 0, 5, 10, 15) The α bit field in 1, 4, or 8-bit packed α is handled as transparency information. Be sure to specify the packed format that includes α through VI6_RPFn_INFMT.RDFMT. When VI6_RPFn_MSKCTRL.MSK_EN is 0b and the IROP bit value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read. 1h: 8-bit plane α The 8-bit α plane is read from external RAM as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified. The α value goes through the 8-bit transparent α generator shown in Figure 9.4-14 without change. When VI6_RPFn_MSKCTRL.MSK_EN is 0b, IROP operation cannot be executed; set the IROP bits to 0 in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1b, IROP operation can be executed.

Bit	Bit Name	Initial Value	R/W	Description
				2h: 1-bit packed α + plane α (IROP != 0, 5, 10, 15) The 1-bit packed α input is converted by the 8-bit transparent α generator shown in Figure 9.4-14 according to the ALPHA0/1 setting into the 8-bit α value as transparency information. Select the packed input format that includes a 1-bit α field. When VI6_RPFn_MSKCTRL.MSK_EN is 0b and the IROP value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read. 3h: 1-bit plane α The 1-bit α plane is read from external RAM and converted by the 8-bit transparent α generator shown in Figure 9.4-14 according to the ALPHA0/1 setting into the 8-bit α value as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified. When VI6_RPFn_MSKCTRL.MSK_EN is 0b, IROP operation cannot be executed; set the IROP bits to 0b in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1b, IROP operation can be executed. 4h: Fixed α The fixed α value (VI6_RPFn_VRTCOL_SET.LAYA value) is output from the RPF. IROP operation cannot be executed; set the IROP bits to 0 in this case. 5 to 7: Setting prohibited.
27 to 24	IROP[3:0]	0h	RW	IROP Operation Setting These bits specify the operator to be executed in the IROP operation unit shown in Figure 9.4-14. The source (S) for the IROP operation is the pixel data and α data specified in the VI6_RPFn_MSKSET0 or VI6_RPFn_MSKSET1 IROP input value register, which is selected according to the value (0 or 1) generated by the 1-bit mask generator. The destination (D) is the image data (RGB/YCbCr) and 8-bit α data output from the unpack/OSD processor. IROP operation is applied both for the image data and α data between the source and destination data. If these bits are set to the operation that involves the source (S) (IROP setting other than 0, 5, 10, or 15) while VI6_RPFn_MSKCTRL.MSK_EN is 0b, the α plane is read from the external RAM to be used for the α value for IROP operation; specify the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the virtual input function is used (VI6_RPFn_INFMT.VIR = 1b), IROP operation is not available; set these bits to 0000b. 0000b: NOP(D) 0001b: AND(S & D) 0010b: AND_REVERSE(S & ~D) 0011b: COPY(S) 0100b: AND_INVERTED(~S & D) 0101b: CLEAR(0) 0110b: XOR(S ^ D) 0111b: OR(S D) 1000b: NOR(~(S D)) 1001b: EQUIV(~(S ^ D)) 1010b: INVERT(~D) 1011b: OR_REVERSE(S ~D) 1100b: COPY_INVERTED(~S) 1101b: OR_INVERTED(~S D) 1110b: NAND(~(S & D)) 1111b: SET(all 1)
23	BSEL	0h	RW	α Bit Count Conversion Selection for 1-Bit Mask Generator Specifies the number of bits in the α plane to be read as mask information from the external RAM. The α value in mask information is used for the source (S) in IROP. When α plane data is 8 bits, it is converted to one bit through the 1-bit mask generator shown in Figure 9.4-14. Note that this bit setting is valid when the ASEL bits are set to 0 or 2 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0b. In other cases, this bit setting has no effect. 0b: 8-bit α is converted to 1-bit α through the 1-bit mask generator. When the 8-bit α value input to the RPF is not 0, it is converted to 1b; when the value is 0, it is converted to 0b. 1b: α value goes through the 1-bit mask generator. The 1-bit α value input to the RPF is output through the 1-bit mask generator without change.
22 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
19, 18	AEXT[1:0]	0h	RW	Lower-Bit α Value Extension Method Set These bits specify the method for extending the input α data to 8 bits through the unpack processing. 00b: The lower-order bits of α value are extended with 0. 01b: The upper-order bits of α value are copied to the lower-order bits. 10b: The lower-order bits of α value are extended with 0. The maximum value is limited to FFh. 11b: Setting prohibited
17, 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 8	ALPHA1[7:0]	00h	RW	8-Bit α Value Output when 1-Bit α Value is 1 These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in Figure 9.4-14 is 1b. This setting is valid when the ASEL bits are set to 010b or 011b. A value from 0 to 255 can be specified.
7 to 0	ALPHA0[7:0]	00h	RW	8-Bit α Value Output when 1-Bit α Value is 0 These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in Figure 9.4-14 is 0b. This setting is valid when the ASEL bits are set to 010b or 011b. A value from 0 to 255 can be specified.

Figure 9.4-14 shows the relationship between the α selector, IROP operation unit, color keying unit, and related registers. The IROP operation unit receives two inputs, source and destination. The image data input from the external memory is processed through the unpack processor and 8-bit transparent α generator and then input to the IROP operation unit as destination data. The α plane data input from the external memory is sent to the 8-bit transparent α generator when the ASEL bits are set to 1 or 3, or sent to the 1-bit mask α generator when the ASEL bits are set to 0 or 2. For the pixel data and 8-bit α value on the source side of the IROP operation unit, either the VI6_RPFn_MSKSET0 value or VI6_RPFn_MSKSET1 values will be selected according to the 1-bit α value output by the 1-bit mask α generator.

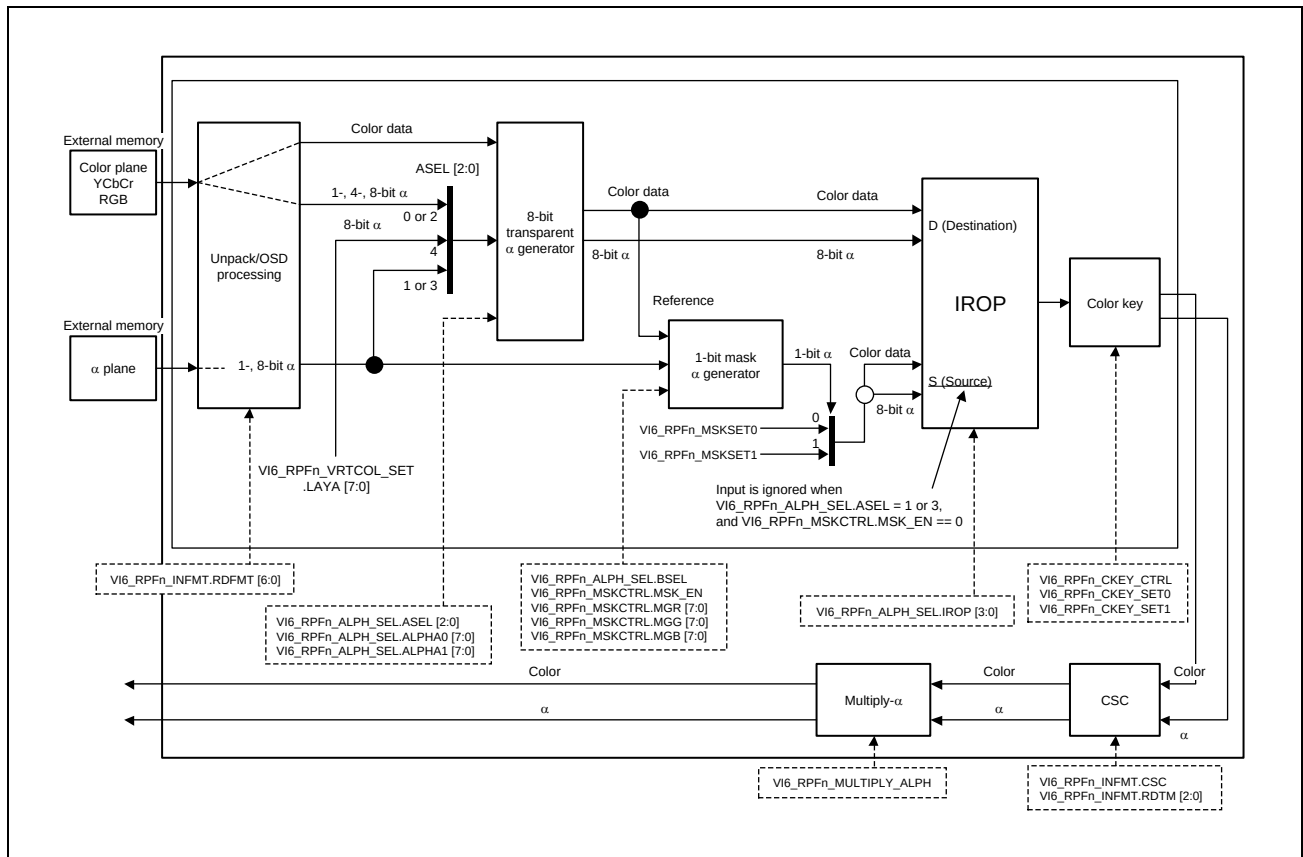


Figure 9.4-14 Configuration of alpha Selector and IROP Operation Unit in RPF

The following describes the function of each block shown in **Figure 9.4-14**. Read the following while referring to the figure as necessary.

Unpack/OSD processor:

Unpacks each component and α value of the image data according to the packed format specified in VI6_RPFn_INFMT.RDFMT.

8-bit transparent α generator:

Converts the input α value into 8-bit α when the input α is four bits or one bit.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α), this generator outputs the input α value without change if the α bit field in the packed α data is 8 bits; if the α bit field is less than 8 bits, it is converted to an 8-bit α value by extending the LSB side according to the VI6_RPFn_ALPH_SEL.AEXT setting.

When VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 4 (fixed α), this generator outputs the input 8-bit plane α without change. If a packed α value is included in RGB data, it is discarded.

When VI6_RPFn_ALPH_SEL.ASEL are set to 2 (1-bit packed α) or 3 (8-bit α generated from 1-bit plane α), an 8-bit α value is generated by using the VI6_RPFn_ALPH_SEL.ALPHA0 [7:0] value when the input 1-bit α value is 0 or by using the VI6_RPFn_ALPH_SEL.ALPHA1 [7:0] value when the input 1-bit α value is 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 3, a packed α value that is included in RGB data is discarded.

1-bit mask α generator:

Generates 1-bit α data from the input 8-bit α data or pixel data. When the input α data is one bit, this generator outputs it without change.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α) or 2 (plane α) and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the α plane read from the external memory to be used in IROP is converted to 1-bit α data when necessary. When the α plane data read from the external RAM is 8 bits (BSEL = 0), if the value is 0, a 1-bit α value of 0b is generated; if the value is not 0, a 1-bit α value of 1b is generated. When the α plane data is one bit (BSEL = 1), this generator outputs it without change.

When the value of the 1-bit α generated by the 1-bit mask α generator is 0b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET0 are output as the source. When the generated 1-bit α value is 1b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET1 are output as the source.

As shown in **Figure 9.4-14**, when VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 3 (1-bit plane α), the α plane read from the external RAM is sent to the 8-bit transparent α generator as transparency information. When VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the 1-bit α for masking is generated according to the input α plane (refer to **(8) RPFn Mask Control Registers (LCDC_VSPD_VI6_RPFn_MSKCTRL)**), but the 1-bit mask α generator does not refer to the input α plane because it is input to the 8-bit transparent α generator as transparency information. Accordingly, the 1-bit mask α generator does not generate a 1-bit α for masking and the data on the source side becomes invalid; that is, IROP operation cannot be executed. Set the IROP bits to 0 in this case. In contrast, when VI6_RPFn_MSKCTRL.MSK_EN is set to 1, the 1-bit mask α generator creates α data for masking according to the pixel data instead of the input α plane data, and IROP operation can be executed in this case.

IROP operation unit:

Executes ROP operation according to the opcode specified in VI6_RPFn_ALPH_SEL.IROP. For ROP operation (other than NOP), valid values should be input both for the source and destination. As described in the above (description of the 1-bit mask α generator), when VI6_RPFn_ALPH_SEL.ASEL are set to 1 or 3 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the source data for the IROP operation unit is treated as invalid; set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP). When VI6_RPFn_ALPH_SEL.ASEL are set to 4, a fixed α value is output from the RPF and IROP operation is not available. In the same way as the above case, set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP).

To specify a valid source value for the IROP operation unit and execute IROP operation (specify an opcode other than NOP in the IROP bits), specify register values as shown in **Table 9.4-14**. Where the source input state is indicated as “Valid” in the table, IROP operation can be executed. In the cases where IROP operation is not available, set the IROP bits to 0 (NOP).

Table 9.4-14 Source Input State in IROP Operation Unit

VI6_RPFn_ALPH_SEL.ASEL[2:0]		VI6_RPFn_MSKCTRL.MSK_EN		
		0 (Source data is generated according to input α plane)	1 (Source data is generated according to the destination-side pixel data)	
000b	(1-, 4-, or 8-bit packed α + plane α)	Valid	(α plane input)	Valid
001b	(8-bit α plane)	Invalid	(IROP operation is not available; α plane is output to the subsequent modules behind RPF)	Valid
010b	(8-bit α generated from 1-bit packed α + plane α)	Valid	(α plane input)	Valid
011b	(8-bit α generated from 1-bit plane α)	Invalid	(IROP operation is not available; α plane is output to the subsequent modules behind RPF)	Valid
100b	(Fixed α)	Invalid (IROP operation is not available; fixed α is output to the subsequent modules behind RPF)		

For the handling of the α values shown in **Figure 9.4-14** and **Table 9.4-14**, the relationship between the RPF input format and RPF output α value is shown in **Table 9.4-15**. Where only bit names are shown in the table, the bits are in VI6_RPFn_ALPH_SEL described in this section.

Table 9.4-15 α Value Selected and Output according to ASEL Bits in Each Input Format

ASEL Setting		α Value Output for Each Input Format	
		RGB	YCbCr
000b	(8-, 4-, or 1-bit packed α is input)	1-, 4-, or 8-bit pixel α	FFh* ¹
001b	(8-bit plane α is input)	8-bit α plane	8-bit α plane
010b	(8-bit α is generated from the 1-bit packed α input)	ALPHA0 or ALPHA1 setting	FFh* ¹
011b	(8-bit α is generated from the 1-bit plane α input)	ALPHA0 or ALPHA1 setting	ALPHA0 or ALPHA1 setting
100b	(Fixed α is output)	VI6_RPFn_VRTCOL_SET.LAYA setting	

Note 1. Fixed value FFh is output because packed α is not included in YCbCr.

(7) RPFn Virtual Plane Color Information Registers (LCDC_VSPD_VI6_RPFn_VRTCOL_SET) (n = 0, 1)**Access Size :** 32 bits**Address :** <LCDC_vspd_base> + 0318h (n = 0)
<LCDC_vspd_base> + 0418h (n = 1)**Initial Value :** 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LAYA[7:0]								LAYR[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAYG[7:0]								LAYB[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	LAYA[7:0]	00h	RW	Virtual-Input Fixed α Value These bits specify the fixed α value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When the virtual input function is disabled (VI6_RPFn_INFMT.VIR = 0b), these bits are used to specify the fixed α value to be output from the RPF while VI6_RPFn_ALPH_SEL.ASEL are set to 4. A value from 0 to 255 can be specified.
23 to 16	LAYR[7:0]	00h	RW	Virtual-Input Fixed R/Cr Component Value These bits specify the fixed R or Cr value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the R value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cr value. A value from 0 to 255 can be specified.
15 to 8	LAYG[7:0]	00h	RW	Virtual-Input Fixed G/Y Component Value These bits specify the fixed G or Y value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the G value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Y value. A value from 0 to 255 can be specified.
7 to 0	LAYB[7:0]	00h	RW	Virtual-Input Fixed B/Cb Component Value These bits specify the fixed B or Cb value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the B value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cb value. A value from 0 to 255 can be specified.

(8) RPFn Mask Control Registers (LCDC_VSPD_VI6_RPFn_MSKCTRL) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 031Ch (n = 0)
<LCDC_vspd_base> + 041Ch (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	MSK_EN	MGR[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MGG[7:0]								MGB[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

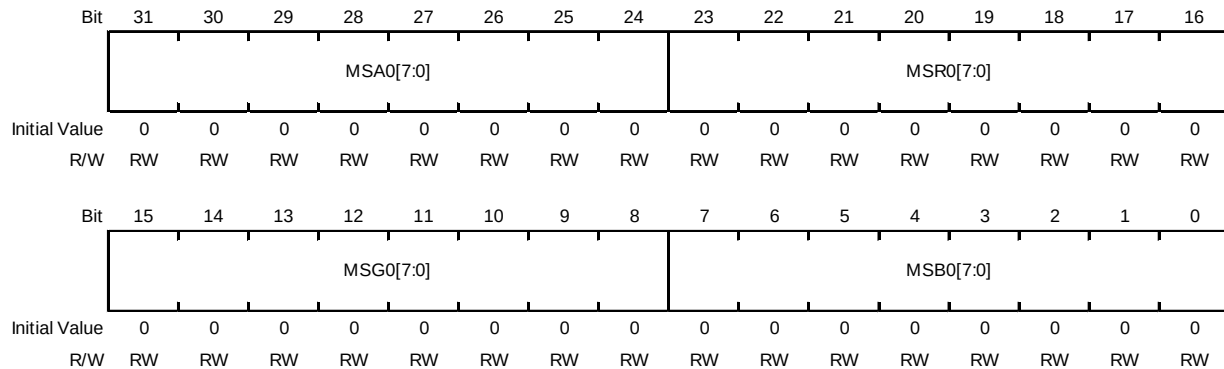
Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
24	MSK_EN	0h	RW	Mask Generation Specification Specifies the method of a value generation in the 1-bit mask α generator shown in Figure 9.4-14 . 0b: A 1-bit mask value is generated according to the input α plane value. When the input α is in the 1-bit format (VI6_RPFn_ALPH_SEL.BSEL = 1b), the 1-bit mask value is output without change. When the input α is in the 8-bit format (VI6_RPFn_ALPH_SEL.BSEL = 0b), the 1-bit mask value is 0 if the α value is 0x00; otherwise, the 1-bit mask value is 1. 1b: The R/Cr, G/Y, and B/Cb components of the image input to the destination side of the IROP operation unit are compared with the values specified in the MGR, MGG, and MGB bits, respectively. When all values match, 1 is output as the 1-bit mask value, and in other cases, 0 is output. When the generated 1-bit mask data is not used, set VI6_RPFn_ALPH_SEL.IROP to 0b.
23 to 16	MGR[7:0]	00h	RW	R/Cr Comparison Value for 1-Bit α Generation These bits specify the R/Cr value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R value for comparison. When YCbCr is specified, specify a Cr value for comparison. This setting is ignored when the MSK_EN bit is set 0b. A value from 0 to 255 can be specified.
15 to 8	MGG[7:0]	00h	RW	G/Y Comparison Value for 1-Bit α Generation These bits specify the G/Y value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G value for comparison. When YCbCr is specified, specify a Y value for comparison. This setting is ignored when MSK_EN is set to 0b. A value from 0 to 255 can be specified.
7 to 0	MGB[7:0]	00h	RW	B/Cb Comparison Value for 1-Bit α Generation These bits specify the B/Cb value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B value for comparison. When YCbCr is specified, specify a Cb value for comparison. This setting is ignored when MSK_EN is set to 0b. A value from 0 to 255 can be specified.

(9) RPFn IROP-SRC Input Value Registers 0 (LCDC_VSPD_VI6_RPFn_MSKSET0) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0320h (n = 0)
<LCDC_vspd_base> + 0420h (n = 1)

Initial Value : 0000_0000h



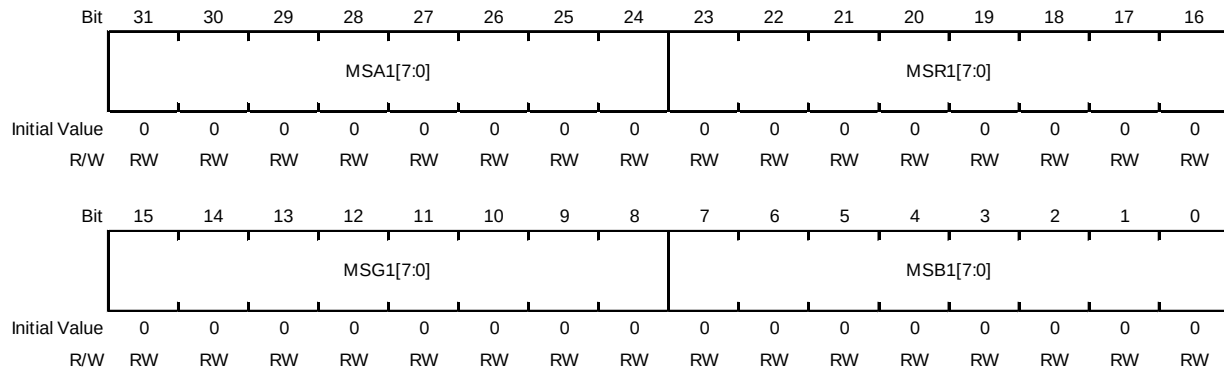
Bit	Bit Name	Initial Value	R/W	Description
31 to 24	MSA0[7:0]	00h	RW	IROP-Source Input α Value when 1-Bit α is 0 These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 9.4-14). A value from 0 to 255 can be specified.
23 to 16	MSR0[7:0]	00h	RW	IROP-Source Input R/Cr Value when 1-Bit α is 0 These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value. A value from 0 to 255 can be specified.
15 to 8	MSG0[7:0]	00h	RW	IROP-Source Input G/Y Value when 1-Bit α is 0 These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value. A value from 0 to 255 can be specified.
7 to 0	MSB0[7:0]	00h	RW	IROP-Source Input B/Cb Value when 1-Bit α is 0 These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value. A value from 0 to 255 can be specified.

(10) RPFn IROP-SRC Input Value Registers 1 (LCDC_VSPD_VI6_RPFn_MSKSET1) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0324h (n = 0)
<LCDC_vspd_base> + 0424h (n = 1)

Initial Value : 0000_0000h



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	MSA1[7:0]	00h	RW	IROP-Source Input α Value when 1-Bit α is 1 These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1. A value from 0 to 255 can be specified (Figure 9.4-14).
23 to 16	MSR1[7:0]	00h	RW	IROP-Source Input R/Cr Value when 1-Bit α is 1 These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value. A value from 0 to 255 can be specified.
15 to 8	MSG1[7:0]	00h	RW	IROP-Source Input G/Y Value when 1-Bit α is 1 These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value. A value from 0 to 255 can be specified.
7 to 0	MSB1[7:0]	00h	RW	IROP-Source Input B/Cb Value when 1-Bit α is 1 These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (Figure 9.4-14). When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value. A value from 0 to 255 can be specified.

(11) RPFn Color Keying Control Registers (LCDC_VSPD_VI6_RPFn_CKEY_CTRL) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0328h (n = 0)
<LCDC_vspd_base> + 0428h (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	LTH	-	-	-	CV	-	-	SAPE1	SAPE0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	RW	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	LTH	0h	RW	Transparent color-luma Threshold Mode Enable/Disable This bit enables or disables transparent color-luma threshold mode for the color keying module. In transparent color-luma threshold mode, color information 0 (GY0) specified in VI6_RPFn_CKEY_SET0 register is compared with the input luma values. When the input data is in YCbCr format, and if input Y value is equal to or smaller than specified color information (VI6_RPFn_CKEY_SET0.GY0), the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP0. This bit is available only when the input data is in YCbCr format. When the input data is in an RGB format, set this bit to 0b. This bit setting is valid only when the CV bit is set to 0b; it is ignored when the CV bit is set to 1b. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1b), set this bit to 0b. 0b: Luma threshold mode is disabled (Matched color mode) 1b: Luma threshold mode is enabled
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	CV	0h	RW	Color Replacement Control This bit controls the color replacement function in the color keying module shown in Figure 9.4-2. When an RGB format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, and if all components of an input pixel match the color components specified in VI6_RPFn_CKEY_SET0, the color replacement function replaces the values of the input α and all RGB components with the α and color components specified in VI6_RPFn_CKEY_SET1. When a YCbCr format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, only the Y data is compared; if the luminance component of an input pixel matches the value specified in VI6_RPFn_CKEY_SET0.GY0, the color replacement function replaces the values of the input α and all YCbCr components with the α and color components specified in VI6_RPFn_CKEY_SET1. When the CV bit is set to 1, the color replacement function is enabled. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0. 0b: Color replacement function is disabled (transparent color mode). 1b: Color replacement function is enabled.
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
1	SAPE1	0h	RW	Comparison Color Data Setting 1 Enable/Disable This bit enables or disables comparison color data setting 1 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1. In transparent color mode, color information 1 (VI6_RPFn_CKEY_SET1.R1/GY1/B1) specified in VI6_RPFn_CKEY_SET1 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET1.AP. When the input data is in YCbCr format, only the Y data is compared. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0. 0b: Comparison color data setting 1 is disabled. 1b: Comparison color data setting 1 is enabled.
0	SAPE0	0h	RW	Comparison Color Data Setting 0 Enable/Disable This bit enables or disables comparison color data setting 0 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0b; it is ignored when the CV bit is set to 1b. In transparent color mode, color information 0 (VI6_RPFn_CKEY_SET0.R0/GY0/B0) specified in VI6_RPFn_CKEY_SET0 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP. When the input data is in YCbCr format, only the Y data is compared. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1b), clear this bit to 0b. 0b: Comparison color data setting 0 is disabled. 1b: Comparison color data setting 0 is enabled.

(12) RPFn Color Keying Color Setting Registers-m (LCDC_VSPD_VI6_RPFn_CKEY_SETm) (n = 0, 1) (m = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 032Ch + m x 0004h (n = 0)
<LCDC_vspd_base> + 042Ch + m x 0004h (n = 1)

Initial Value : 0000_0000h

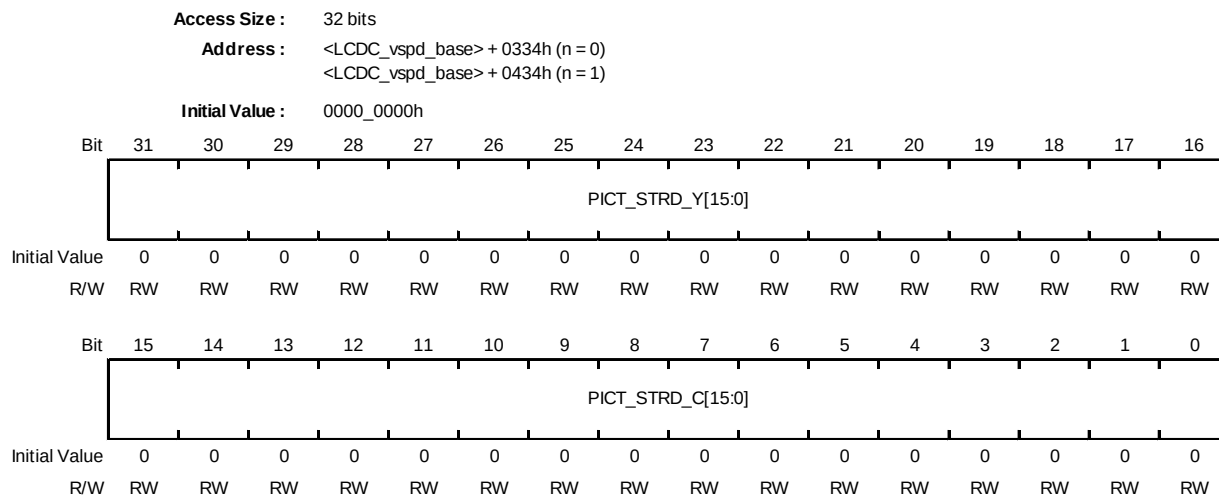
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AP0[7:0]								R0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GY0[7:0]								B0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	AP0[7:0]	00h	RW	α Data in Color Keying Color Information-1 - In transparent color-matched color mode for color keying When the input data matches color setting-0 (R0, GY0, and B0) for transparent color comparison in the color keying module, the input α value is replaced with the value specified in these bits. Specify the α value to replace the input value. - In transparent color-luma threshold mode for color keying When the input data is in YCbCr format, and if input Y value is equal to or smaller than GY0, the input α value is replaced with the value specified in this bit field (AP0). Specify the α value in this bit field (AP0) to replace the input value. - In color replacement mode for color keying These bits are not used in this mode. Clear them to 0. α value replacement through these bits in transparent color mode for color keying takes priority over the α value selected through the VI6_RPFn_ALPH_SEL.ASEL setting. A value from 0 to 255 can be specified.
23 to 16	R0[7:0]	00h	RW	R^{*1}/Cr Component Data in Color Keying Color Information-1 - In transparent color-matched color mode for color keying Specify the R component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cr component, and the setting of these bits is ignored. - In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. - In color replacement mode for color keying Specify the R component value to be compared with the input data in the color replacement function of the color keying module. A value from 0 to 255 can be specified.
15 to 8	GY0[7:0]	00h	RW	G^{*1}/Y Component Data in Color Keying Color Information-1 - In transparent color-matched color mode for color keying Specify the G/Y component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. - In transparent color-luma threshold mode for color keying Specify the Y component value in the GY0 to be compared. - In color replacement mode for color keying Specify the G/Y component value to be compared with the input data in the color replacement function of the color keying module. A value from 0 to 255 can be specified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	B0[7:0]	00h	RW	B*¹/Cb Component Data in Color Keying Color Information-1 - In transparent color-matched color mode for color keying Specify the B component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPE0 setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cb component, and the setting of these bits is ignored. - In transparent color-luma threshold mode for color keying The color keying module does not refer this bit field, and the setting of these bits is ignored. - In color replacement mode for color keying Specify the B component value to be compared with the input data in the color replacement function of the color keying module. A value from 0 to 255 can be specified.

Note 1. When comparison data is specified in an RGB format, if a packed format is selected for RGB input and each of the RGB components is not 8 bits, the lower-order bits of input data are extended as specified through VI6_RPFn_INFMT.CEXT before comparison. The RGB components to be compared with the input should also be extended in the same way and the extended values should be specified in this register.

(13) RPFn Source Picture Memory Stride Setting Registers (LCDC_VSPD_VI6_RPFn_SRCM_PSTRIDE) (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	PICT_STRD_Y [15:0]	0000h	RW	Memory Stride of Source Picture Y/RGB Plane These bits specify in 1-byte units the memory stride of the source picture Y/RGB plane read by the RPFn. A value from 0 to 65535 can be specified. Refer to Figure 9.4-15 for settings.
15 to 0	PICT_STRD_C [15:0]	0000h	RW	Memory Stride of Source Picture C Plane These bits specify in 1-byte units the memory stride of the source picture C plane read by the RPFn. When an RGB-format picture is read, these bits do not need to be set. A value from 0 to 65535 can be specified. Refer to Figure 9.4-15 for settings. In the YCbCr planar format, this setting is used as the memory stride of the Cb and Cr planes.

This register specifies the memory stride of the picture planes in the source area as shown in **Figure 9.4-15**. The memory stride of the α plane should be specified through VI6_RPFn_SRCM_ASTRIDE.ALPH_STRD. When the RPF input is in an RGB format, only the RGB plane is read; when the input is in YCbCr format, the Y and C planes are read as shown in **Figure 9.4-15**. When the α plane is used, the α plane is also read. According to the image format and the necessity of the α plane, specify the necessary addresses where the source image is stored (VI6_RPFn_SRCM_ADDR_Y, VI6_RPFn_SRCM_ADDR_C, and VI6_RPFn_SRCM_ADDR_AI).

Whether the α plane needs to be read is determined according to the α plane selection method and IROP operation type. For details, refer to **(6) RPFn α Plane Selection Control Registers (LCDC_VSPD_VI6_RPFn_ALPH_SEL)**.

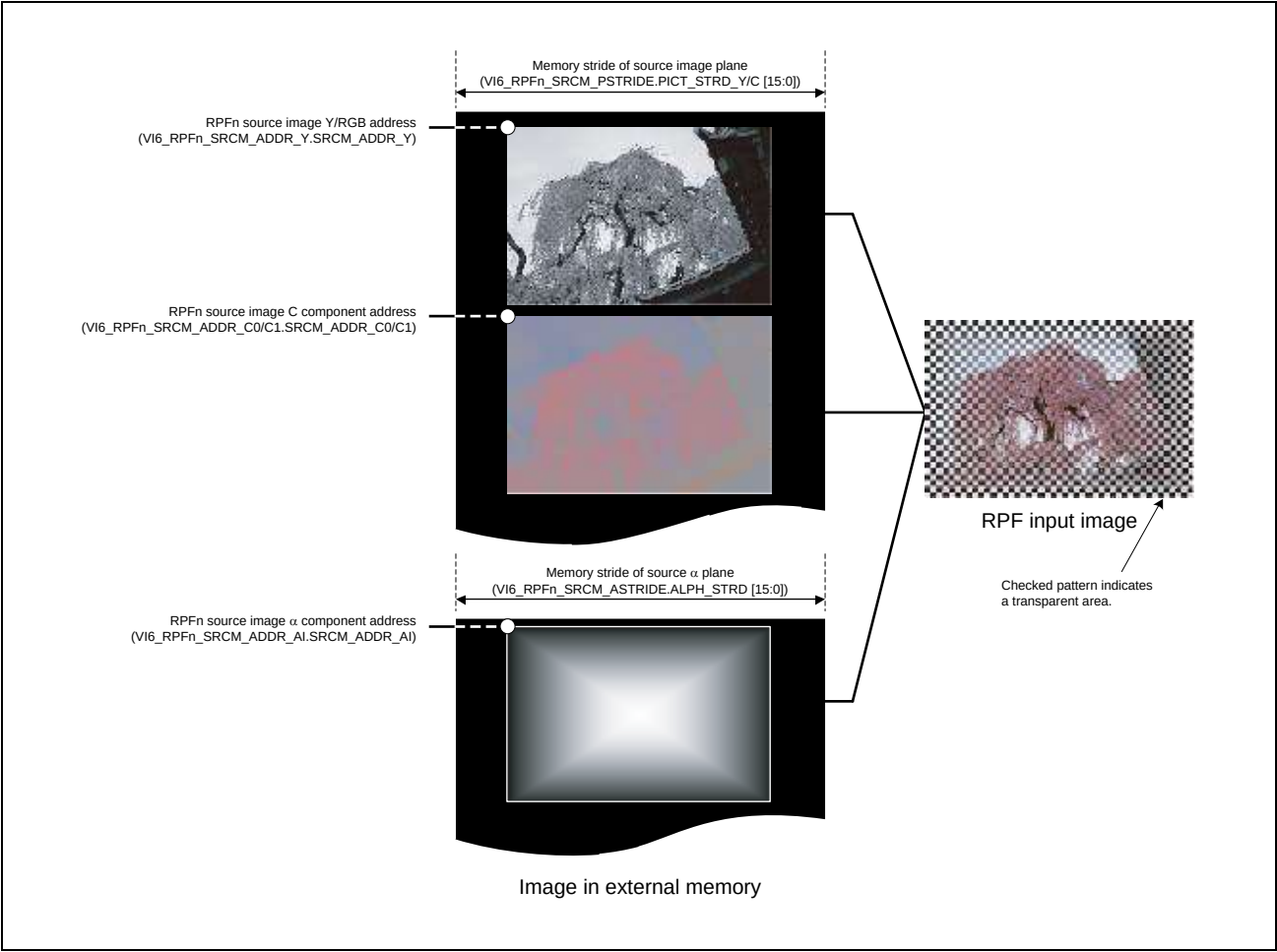
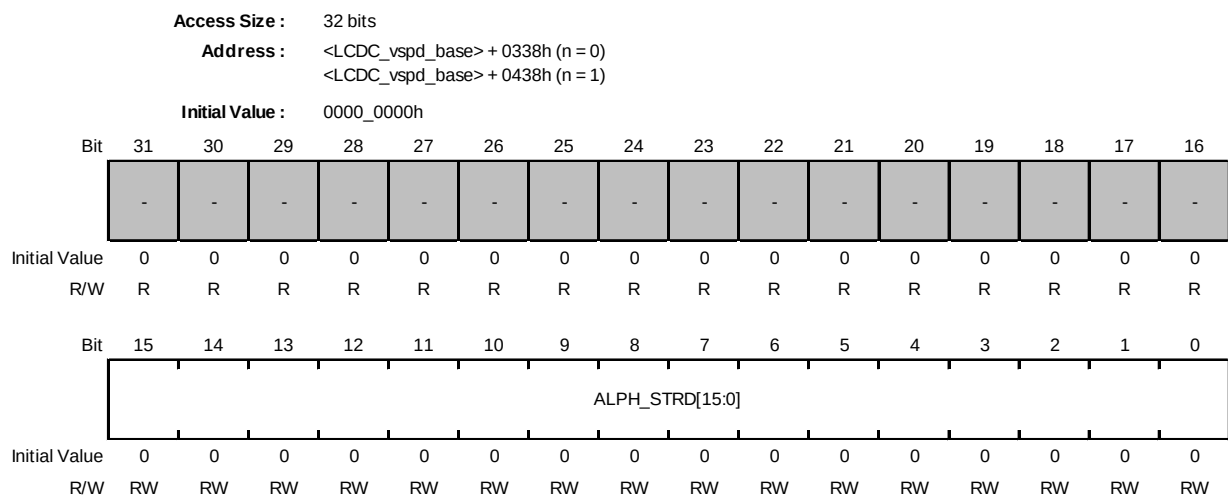


Figure 9.4-15 Reading an Image from RPFn Source Area

(14) RPFn Source α Memory Stride Setting Registers (LCDC_VSPD_VI6_RPFn_SRCM_ASTRIDE) (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	ALPH_STRD [15:0]	0000h	RW	Memory Stride of Source α Plane These bits specify in 1-byte units the memory stride of the source α plane read by the RPFn. A value from 0 to 65535 can be specified. Refer to Figure 9.4-15 for settings.

(15) RPFn Source Y/RGB Address Registers (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_Y) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 033Ch (n = 0) <LCDC_vspd_base> + 043Ch (n = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCM_ADDR_Y[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCM_ADDR_Y[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name	Initial Value	R/W	Description												
31 to 0	SRCM_ADDR_Y[31:0]	0000_0000h	RW	Source Image Y/RGB Plane Storing Address These bits specify in 1-byte units the start address of the source image Y plane and packed RGB plane read by the RPFn. A value from 0000_0000h to FFFF_FFFFh can be specified. Refer to Figure 9.4-15 in (13) RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE) for settings.												

(16) RPFn Source Chroma Address Registers 0 (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_C0) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0340h (n = 0) <LCDC_vspd_base> + 0440h (n = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCM_ADDR_C0[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCM_ADDR_C0[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SRCM_ADDR_C0[31:0]	0000_0000h	RW	Source Image C Plane Storing Address 0 These bits specify in 1-byte units the start address of the source image C plane read by the RPFn. Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in Figure 9.4-15 or the Cb plane when a planar format is selected. When an interleaved format is selected or the RPF input is in an RGB format, this setting is not used. A value from 0000_0000h to FFFF_FFFFh can be specified. Refer to Figure 9.4-15 in (13) RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE) for settings.

(17) RPFn Source Chroma Address Registers 1 (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_C1) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0344h (n = 0) <LCDC_vspd_base> + 0444h (n = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCM_ADDR_C1[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCM_ADDR_C1[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SRCM_ADDR_C1[31:0]	0000_0000h	RW	Source Image C Plane Storing Address 1 These bits specify in 1-byte units the start address of the Cr plane when a planar YCbCr format shown in Figure 9.4-15 is read by the RPFn. This setting is not used when the RPF input is in YCbCr format that is not a planar format or in an RGB format. A value from 0000_0000h to FFFF_FFFFh can be specified. Refer to Figure 9.4-15 in (13) RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE) for settings.

(18) RPFn Source α Address Registers (LCDC_VSPD_VI6_RPFn_SRCM_ADDR_AI) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 0348h (n = 0) <LCDC_vspd_base> + 0448h (n = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRCM_ADDR_AI[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRCM_ADDR_AI[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SRCM_ADDR_AI[31:0]	0000_0000h	RW	Source Image α Plane Storing Address These bits specify in 1-byte units the start address of the α plane of the source image read by the RPFn. Specify in the same way as the start address of the picture plane. When the α plane is not read from the source area, these bits do not need to be set. A value from 0000_0000h to FFFF_FFFFh can be specified. Refer to Figure 9.4-15 in (13) RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE) for settings.

(19) RPFn Bus Access Control Registers (LCDC_VSPD_VI6_RPFn_BAC) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 0350h (n = 0)
<LCDC_vspd_base> + 0450h (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B512
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	B512	0h	RW	Burst access in 512 pixels' enable 0b: burst access in 256 pixels 1b: burst access in 512 pixels Always set 0b to this bit
15 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(20) RPFn Multiple Alpha Control Register (LCDC_VSPD_VI6_RPFn_MULT_ALPH) (n = 0, 1)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 036Ch (n = 0)
<LCDC_vspd_base> + 046Ch (n = 1)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	A_MMD	-	-	P_MMD[1:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	A_MMD	0h	RW	0b: Alpha data go through multiply-alpha unit. 1b: Multiply-alpha unit multiplies alpha data by specified alpha (ALPHA_RATIO[7:0]). When output format from CSC unit is YCbCr, set 0b to this bit.
11, 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9, 8	P_MMD[1:0]	0h	RW	0h: Image data go through multiply-alpha unit. 1h: Multiply-alpha unit multiplies image data by specified alpha (ALPHA_RATIO[7:0]). 2h: Multiply-alpha unit multiplies image data by alpha data. 3h: Multiply-alpha unit multiplies image data by alpha data and specified alpha (ALPHA_RATIO[7:0]). When output format from CSC unit is YCbCr, set 0 to this bit.
7 to 0	ALPHA_RATIO [7:0]	00h	RW	Multiply alpha value

ALPIn: Input Alpha data to Multiply-alpha unit

PIXIn(R): Input R data to Multiply-alpha unit

PIXIn(G): Input G data to Multiply-alpha unit

PIXIn(B): Input B data to Multiply-alpha unit

ALPout: Output Alpha data from Multiply-alpha unit

PIXout(R): Output R data from Multiply-alpha

PIXout(G): Output G data from Multiply-alpha

PIXout(B): Output B data from Multiply-alpha

Table 9.4-16 Expression of Output Alpha Data from Multiply-Alpha Unit

A_MMD	ALPHA_RATIO[7:0]	Expression
0	Don't care	ALPout = ALPIn
1	Not 255	ALPout = ALPIn × ALPHA_RATIO / 256
	255	ALPout = ALPIn

Table 9.4-17 Expression of Output Pixel Data from Multiply-Alpha Unit

P_MMD[1:0]	ALPHA_RATIO[7:0]	ALPin	Expression
0	Don't care	Don't care	$\text{PIXout(R)} = \text{PIXin(R)}$ $\text{PIXout(G)} = \text{PIXin(G)}$ $\text{PIXout(B)} = \text{PIXin(B)}$
1	Not 255	Don't care	$\text{PIXout(R)} = \text{PIXin(R)} \times \text{ALPHA\_RATIO} / 256$ $\text{PIXout(G)} = \text{PIXin(G)} \times \text{ALPHA\_RATIO} / 256$ $\text{PIXout(B)} = \text{PIXin(B)} \times \text{ALPHA\_RATIO} / 256$
	255	Don't care	$\text{PIXout(R)} = \text{PIXin(R)}$ $\text{PIXout(G)} = \text{PIXin(G)}$ $\text{PIXout(B)} = \text{PIXin(B)}$
2	Don't care	Not 255	$\text{PIXout(R)} = \text{PIXin(R)} \times \text{ALPin} / 256$ $\text{PIXout(G)} = \text{PIXin(G)} \times \text{ALPin} / 256$ $\text{PIXout(B)} = \text{PIXin(B)} \times \text{ALPin} / 256$
	Don't care	255	$\text{PIXout(R)} = \text{PIXin(R)}$ $\text{PIXout(G)} = \text{PIXin(G)}$ $\text{PIXout(B)} = \text{PIXin(B)}$
3	Not 255	Not 255	$\text{PIXout(R)} = \text{PIXin(R)} \times \text{ALPin} \times \text{ALPHA\_RATIO} / 256 / 256$ $\text{PIXout(G)} = \text{PIXin(G)} \times \text{ALPin} \times \text{ALPHA\_RATIO} / 256 / 256$ $\text{PIXout(B)} = \text{PIXin(B)} \times \text{ALPin} \times \text{ALPHA\_RATIO} / 256 / 256$
	255	Not 255	$\text{PIXout(R)} = \text{PIXin(R)} \times \text{ALPin} / 256$ $\text{PIXout(G)} = \text{PIXin(G)} \times \text{ALPin} / 256$ $\text{PIXout(B)} = \text{PIXin(B)} \times \text{ALPin} / 256$
	Not 255	255	$\text{PIXout(R)} = \text{PIXin(R)} \times \text{ALPHA\_RATIO} / 256$ $\text{PIXout(G)} = \text{PIXin(G)} \times \text{ALPHA\_RATIO} / 256$ $\text{PIXout(B)} = \text{PIXin(B)} \times \text{ALPHA\_RATIO} / 256$
	255	255	$\text{PIXout(R)} = \text{PIXin(R)}$ $\text{PIXout(G)} = \text{PIXin(G)}$ $\text{PIXout(B)} = \text{PIXin(B)}$

9.4.3.2.7 WPF Control Registers

(1) WPF0-Source-RPF Register (LCDC_VSPD_VI6_WPF0_SRCRPF)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 1000h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	VIR_ACT2[1:0]		-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	RPF1_ACT [1:0]		RPF0_ACT [1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25, 24	VIR_ACT2[1:0]	0h	RW	Virtual RPF Start Enable in BRS These bits enable start of the virtual RPF in the BRS as the source RPF for the WPF0 when the WPF0 is started. For details of the virtual RPF, refer to the following. 9.4.3.2.10 (1) BRS Input Control Register (VI6_BRS_INCTRL) 9.4.3.2.10 (2) Size Register of BRS Input Virtual RPF (VI6_BRS_VIRRRPF_SIZE) 9.4.3.2.10 (3) Display Location Register of BRS Input Virtual RPF (VI6_BRS_VIRRRPF_LOC) 9.4.3.2.10 (4) Color Information Register of BRS Input Virtual RPF (VI6_BRS_VIRRRPF_COL) Note that the virtual RPF is in the BRS as shown in Figure 9.4-28 and there are no register bits for DPR setting related to the virtual RPF. 0h: The virtual RPF in the BRS is not started. 1h: The virtual RPF in the BRS is started as a sublayer source RPF for the WPF0. 2h: The virtual RPF in the BRS is started as the master-layer source RPF for the WPF0. 3h: Setting prohibited
23 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3, 2	RPF1_ACT[1:0]	0h	RW	RPF1 Start Enable These bits enable start of RPF1 as the source RPF for the WPF0 when the WPF0 is started. When RPF1 is not started by the WPF0, set the VI6_DPR_RPF1_ROUTE_RT_RPF1 bits to 63. 0h: RPF1 is not started. 1h: RPF1 is started as a sublayer source RPF for the WPF0. 2h: RPF1 is started as the master-layer source RPF for the WPF0. 3h: Setting prohibited
1, 0	RPF0_ACT[1:0]	0h	RW	RPF0 Start Enable These bits enable start of RPFn as the source RPF for the WPF0 when the WPF0 is started. When RPF0 is not started by the WPF0, set the VI6_DPR_RPF0_ROUTE_RT_RPF0 bits to 63. 0h: RPF0 is not started. 1h: RPF0 is started as a sublayer source RPF for the WPF0. 2h: RPF0 is started as the master-layer source RPF for the WPF0. 3h: Setting prohibited

When the WPFn is started through the VSPD start register n (VI6_CMDn: n = 0), the RPF and virtual RPF in the BRS specified as the source RPF in this register are also started to supply data to the VSPD internal modules.

Note the following when specifying the source RPF.

- (1) When blending or ROP operation is applied to multiple images through the BRS, multiple source RPFs are necessary for one WPF. When multiple source RPFs are used, images should be classified into a master layer and sublayers; assign one of the source RPFs as the master-layer source RPF and other RPFs as sublayer source RPFs. Do not assign all RPFs as sublayer source RPFs (VI6_WPF1_SRCRPF = 00000015h) or two or more RPFs as the master-layer source RPF (VI6_WPF0_SRCRPF = 0000025Ah) (such settings are prohibited).
- (2) When the BRS is not used, there should be only one source RPF for one WPF. In this case, the source RPF should be assigned as the master-layer source RPF.

(2) WPF0 Horizontal Input Size Clipping Register (LCDC_VSPD_VI6_WPF0_HSZCLIP)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 1004h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	HCEN	-	-	-	-	HCL_OFST[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	HCL_SIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	HCEN	0h	RW	Horizontal Size Clipping Enable/Disable Enables or disables clipping of the horizontal size of the WPF0 input image. 0b: Horizontal size clipping is disabled 1b: Horizontal size clipping is enabled
27 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	HCL_OFST[7:0]	00h	RW	Horizontal Size Clipping Offset Value Setting These bits specify the offset size (pixels) from the left end of the image in horizontal size clipping when the HCEN bit is 1b (Figure 9.4-16). The left side of the image input to the WPF is cut off for the size specified in these bits. When the HCEN bit is 0b, this setting is ignored. A value from 0 to 255 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in Figure 9.4-16 is made, VSPD does not operate correctly.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	HCL_SIZE[12:0]	0000h	RW	Horizontal Clipping Size Setting When the HCEN bit is 1b, these bits specify the clipping size for horizontal clipping processing. Through this processing, the area of the horizontal size specified through the HCL_SIZE bits starting from the offset position specified through the HCL_OFST bits is determined as the valid image area. Accordingly, the right-side pixels beyond the (HCL_OFST + HCL_SIZE) size in the WPF0 input image are discarded. When the HCEN bit is 0b, this setting is ignored. A value from 1 to 8190 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in Figure 9.4-16 is made, VSPD does not operate correctly. When the WPF0 output format is YCbCr4:2:2 or YCbCr4:2:0, specify an even value in these bits.

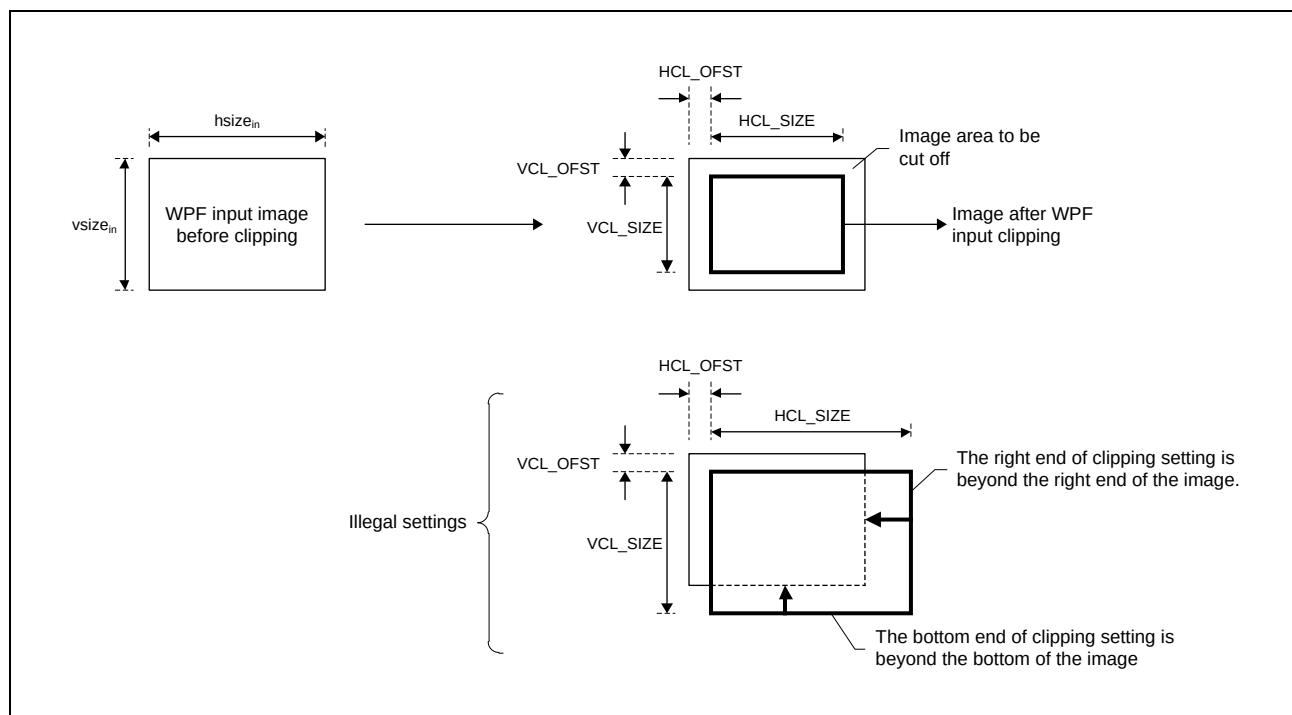


Figure 9.4-16 Image Clipping in WPF Input Section

(3) WPF0 Vertical Input Size Clipping Register (LCDC_VSPD_VI6_WPF0_VSZCLIP)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 1008h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VCEN	-	-	-	-	VCL_OFST[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VCL_SIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	VCEN	0h	RW	Vertical Size Clipping Enable/Disable Enables or disables clipping of the vertical size of the WPF0 input image. 0b: Vertical size clipping is disabled 1b: Vertical size clipping is enabled
27 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	VCL_OFST [7:0]	00h	RW	Vertical Size Clipping Offset Value Setting These bits specify the offset size (pixels) from the top end of the image in vertical size clipping when the VCEN bit is 1 (Figure 9.4-16). The top of the image input to the WPF is cut off for the size specified in these bits. When the VCEN bit is 0b, this setting is ignored. A value from 0 to 255 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in Figure 9.4-16 is made, VSPD does not operate correctly.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	VCL_SIZE [12:0]	0000h	RW	Vertical Clipping Size Setting When the VCEN bit is 1b, these bits specify the clipping size for vertical clipping processing. Through this processing, the area of the vertical size specified through the VCL_SIZE bits starting from the offset position specified through the VCL_OFST bits is determined as the valid image area. Accordingly, the bottom pixels beyond the (VCL_OFST + VCL_SIZE) size in the WPF0 input image are discarded. When the VCEN bit is 0b, this setting is ignored. A value from 1 to 8190 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in Figure 9.4-16 is made, VSPD does not operate correctly. When the WPF0 output format is YCbCr4:2:0, specify an even value in these bits.

(4) WPF0 Output Format Register (LCDC_VSPD_VI6_WPF0_OUTFMT)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 100Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PDV [7:0]								PXA	ODE	-	-	-	ROT[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPYCS	SPUVS	DITH[1:0]		WRTM[2:0]			CSC	-	WRFMT[6:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	PDV [7:0]	00h	RW	PAD Value in Output Packed Data These bits specify the value to be stored in the bit field indicated as PAD or P in the output formats shown in Table 9.4-18 . To store this value in PAD, specify 0 in the PXA bit. A value from 0 to 255 can be specified.
23	PXA	0h	RW	PAD Data Select Selects the value to be stored in the bit field indicated as PAD or P in the packed RGB output formats shown in Table 9.4-19 . Both the value specified in the PDV bits and the α data input from the DPR to WPF are 8 bits, but some of the PAD and P bit fields shown in Table 9.4-18 are four bits or one bit. When the target bit field is not 8 bits, the number of bits in the PDV value and the α data input from the DPR to WPF is reduced according to the VI6_WPF0_RNDCTRL.ABRM setting. For bit count reduction, refer to Figure 9.4-17 and the description of VI6_WPF0_RNDCTRL.ABRM. 0b: The value specified in the PDV bits is stored in the PAD shown in Table 9.4-18 . 1b: The α value output from DPR in pixel units is stored in the PAD shown in Table 9.4-18 .
22	ODE	0h	RW	Ordered Dither (mode A) Enable/Disable 0b: Ordered dither (mode A) is disabled. 1b: Ordered dither (mode A) is enabled. When the output format specified through the WRFMT bits is YCbCr, specify 0b in this bit. And when VI6_WPF0_OUTFMT.CSC is set to 1b, specify 0b in this bit even in the case that the output format specified through the WRFMT bits is RGB. Ordered dither is available only for 18bpp. So, when ODE bit is equal to 1b, set WRFMT at 18bpp format. When ODE bit is equal to 0, WPF dither method is specified by DITH[1:0] in the register Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp.
21 to 19	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18 to 16	ROT[2:0]	0h	RW	Rotation Processing Select These bits select the flipping processing to be applied to the WPF0 output image. Figure 9.4-18 shows the correspondence between the original image and the flipping result according to each setting. 0b: No flipping 1b: Vertical flipping Note that the destination address setting should be changed according to the setting of these bits. For details, refer to 9.4.3.2.7 (9) WPFn Destination Y/RGB Address Registers (VI6_WPFn_DSTN_ADDR_Y) . When the LIF module is used (VI6_LIF_CTRL.LIF_EN = 1b), set 0b to ROT[2:0].
15	SPYCS	0h	RW	WPF Output Mode Setting 1 When the output format is YUY2, set this bit to 1b and set the WRFMT bits to 71 (47h). When the output format is YYYU, set this bit and the SPUVS bit to 1b and set the WRFMT bits to 71 (47h). In other cases, set this bit to 0b.

Bit	Bit Name	Initial Value	R/W	Description
14	SPUVS	0h	RW	WPF Output Mode Setting 2 When the output format is NV61, set this bit to 1b and set the WRFMT bits to 65 (41h). When the output format is NV21, set this bit to 1b and set the WRFMT bits to 66 (42h). When the output format is YVYU, set this bit and the SPYCS bit to 1b and set the WRFMT bits to 71 (47h). In other cases, set this bit to 0b.
13,12	DITH[1:0]	0h	RW	Ordered Dither (mode B) Enable/Disable When the output format specified through the WRFMT bits is RGB with 18 bpp (260000 colors) or less, the color reduction processing is applied to match the number of colors. The color reduction processing may generate the artifacts of pseudo gradation, which can be suppressed through dithering. The DITH bits enable or disable dithering during color reduction. When the output format specified through the WRFMT bits is YCbCr, specify 0 in these bits. And when VI6_WPF0_OUTFMT.CSC is set to 1, specify 0 in these bits even in the case that the output format specified through the WRFMT bits is RGB. 0h: Dithering (mode B) is disabled 3h: Dithering (mode B) is enabled 1, 2h: Setting prohibited When ODE bit in the register is 1b, set this bit to 0h.
11 to 9	WRTM[2:0]	0h	RW	CSC Conversion Expression Setting These bits select the expression for color space conversion. The conversion direction is RGB to YCbCr when the format specified in the WRFMT bits is RGB, or YCbCr to RGB when the format is YCbCr. 0h: BT.601 YCbCr [16,235/240] ↔ RGB [0,255] 1h: BT.601 YCbCr [0,255] ↔ RGB [0,255] 2h: BT.709 YCbCr [16,235/240] ↔ RGB [0,255] 3h: BT.709 YCbCr [16,235/240] ↔ RGB [16,235] 4h to 7h: Setting prohibited
8	CSC	0h	RW	Color Space Conversion Setting Enables or disables YCbCr ↔ RGB color space conversion to be executed in the WPF0. The characteristics of color space conversion are determined by the WRTM setting. There are some points to be noted about the relationship between the CSC setting and output format (WRFMT). For details refer to (*) in 9.4.3.2.6 (3) RPFn Input Format Registers (VI6_RPFn_INFMT). 0b: Color space is not converted. 1b: Color space is converted.
7	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
6 to 0	WRFMT[6:0]	00h	RW	WPF Output Image Format Setting These bits select the format of the image output from the WPF0 to the external memory from among those listed in Table 9.4-18 and Table 9.4-19. [Note] Number of output pixels When YCbCr4:2:2 is specified through WRFMT, the horizontal size of the output image should be a multiple of 2 pixels. When YCbCr4:2:0 is specified, the vertical and horizontal sizes of the output image should be multiples of 2 pixels. Specify an appropriate data flow of the source RPF → DPR → target WPF so that the size of the image input to the target WPF satisfies the above restrictions. In particular, when the data flow includes a module or a function that modifies (up-scales, down-scales, or clips) the image size, take special care about the module or function settings. [Note] Output lines in YCbCr4:2:0 In the YCbCr4:2:0 output format, the number of chrominance lines in the vertical direction is one-half the number of luminance lines. For this reason, the WPF outputs only even-numbered chrominance lines (lines 0, 2, 4, 6, ...) (conversion from (A) to (B)) in. When vertical flipping is also specified through the ROT bits, the flipping processing is executed last and the chrominance line locations are inverted (lines 1, 3, 5, 7, ...) in the output image ((C) in Figure 9.4-19). [Note] Down sampling of CbCr in horizontal direction in YCbCr4:2:0 or YCbCr4:2:2 In the YCbCr4:2:2 or YCbCr4:2:0 output format, method of down sample of Cb/Cr in horizontal direction is average of neighbor two pixels.

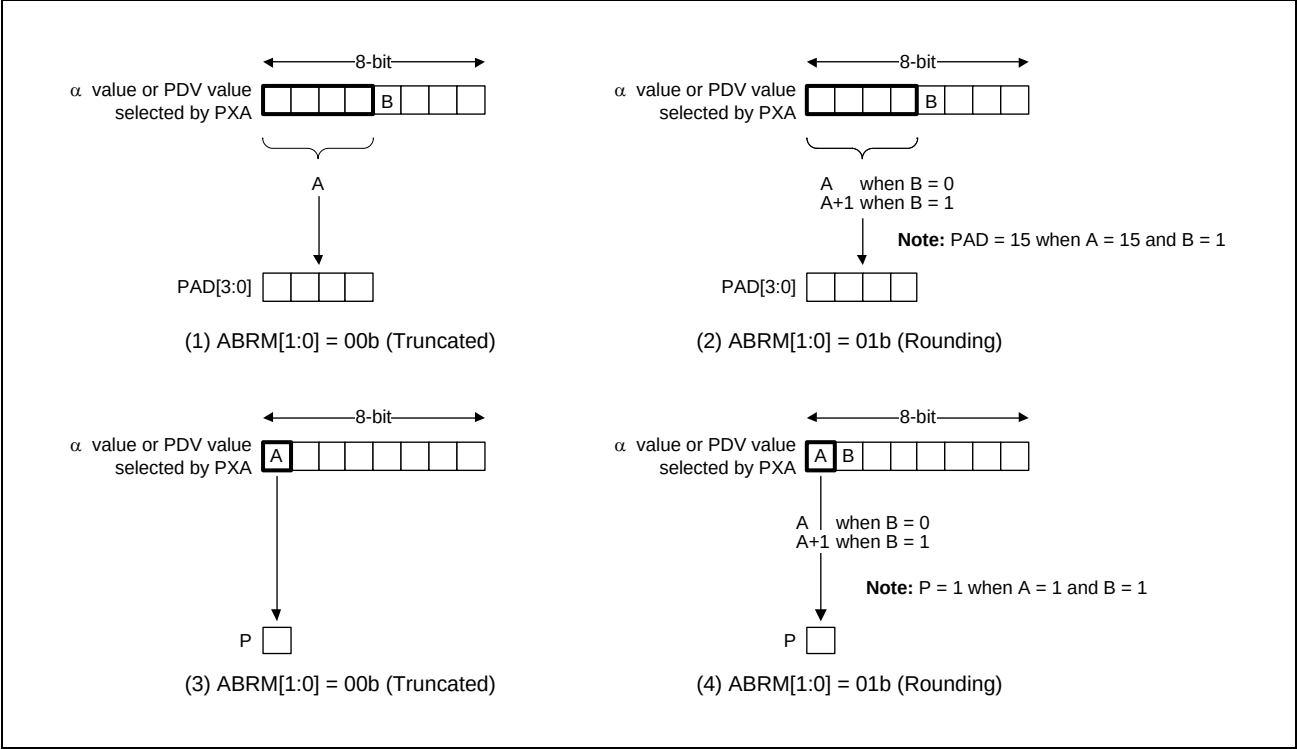


Figure 9.4-17 Selection of PAD Value and Reduction of Bit Count through PXA Setting

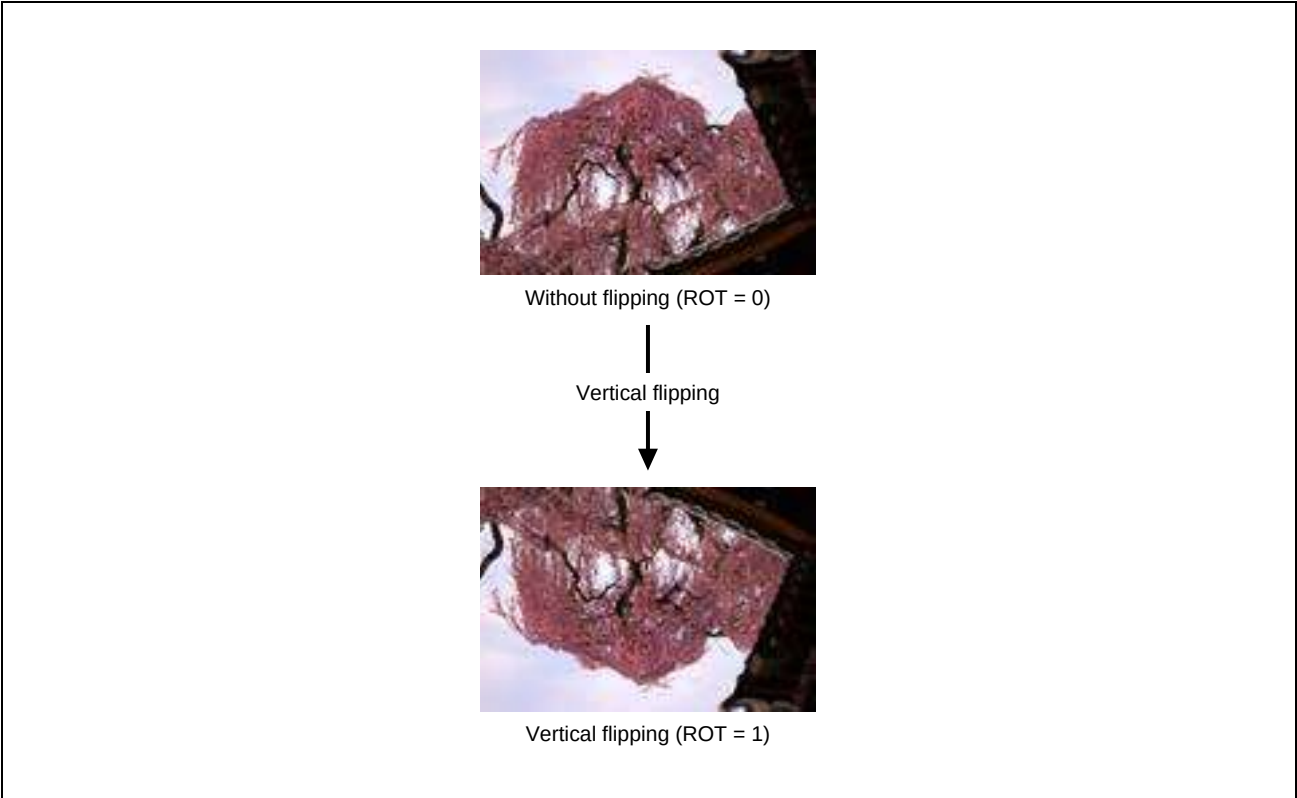


Figure 9.4-18 Correspondence between Original Image and Flipping Result according to ROT Setting

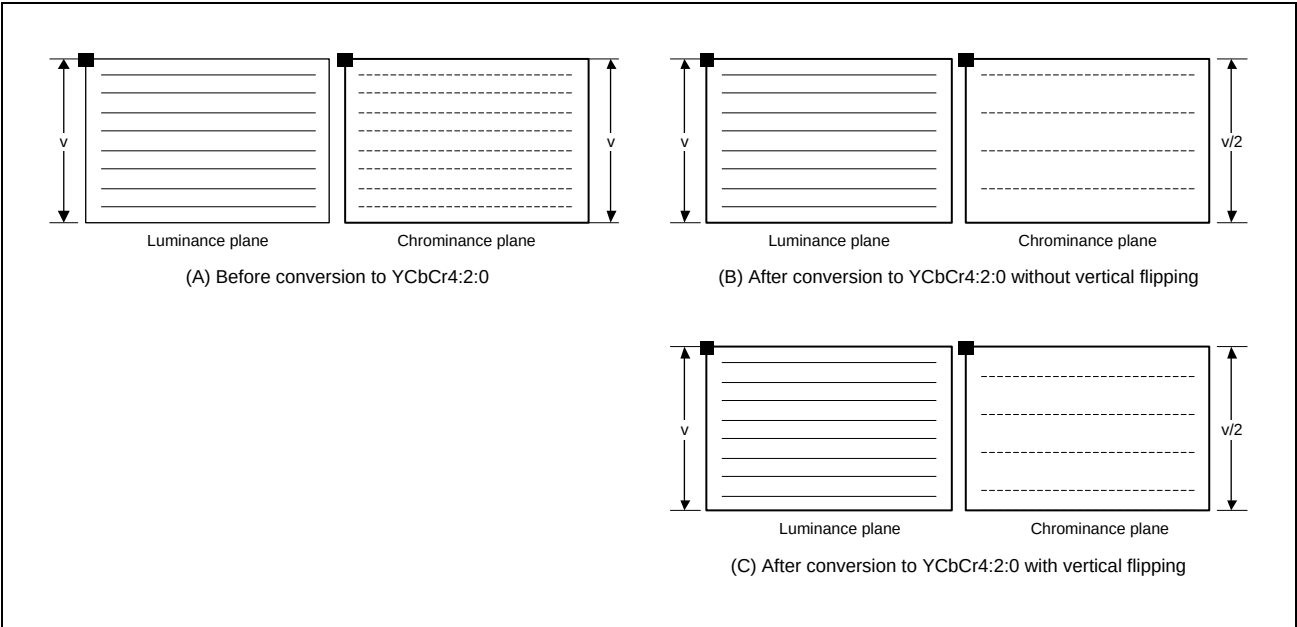


Figure 9.4-19 Chrominance Output Lines in YCbCr4:2:0 and Vertical Flipping Result

Table 9.4-18 Packed RGB Formats for WPF Output

WRFMT[6:0]	Bit per pixel	Phase	upper row - address / bottom row - bit field																																
			n								n+1								n+2								n+3								
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
00h	8	—	R0	R0	R0	G0	G0	G0	B0	B0	R1	R1	R1	G1	G1	G1	B1	B1	R2	R2	R2	G2	G2	G2	B2	B2	R3	R3	R3	G3	G3	G3	B3	B3	
01h	12	—	0	0	0	0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	0	0	0	0	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	
02h			R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	0	0	0	0	R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	0	0	0	0	
03h	—	—	Reserved								Reserved								Reserved								Reserved								
04h	15	—	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	
05h			R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	0	
06h	16	—	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	
07h	18	—	PAD								0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0		
08h			R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	0	0	0	PAD							
09h			0	0	0	0	0	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	PAD									
0Ah			PAD								R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0
0Bh			PAD								0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	B0
0Ch			0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	PAD								
0Dh			PAD								R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	0	0	0	0
0Eh			R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	PAD								
0Fh			0	0	0	0	0	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	0	R1	R1	
10h			1	R1	R1	R1	R1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	0	0	0	0	0	0	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2
		2	G2	G2	B2	B2	B2	B2	B2	B2	0	0	0	0	0	0	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3		
		0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	0	0	0	0	0	0	R1	R1	R1	R1	R1	R1	G1	G1			
H'11h		1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	0	0	0	0	0	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	B2	B2	B2	B2		
		2	B2	B2	0	0	0	0	0	0	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	0	0	0	0	0		
		0	0	0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	R1	R1	R1	R1	R1		
12h		1	0	0	G1	G1	G1	G1	G1	G1	0	0	B1	B1	B1	B1	B1	B1	0	0	R2	R2	R2	R2	R2	R2	0	0	G2	G2	G2	G2	G2		
		2	0	0	B2	B2	B2	B2	B2	B2	0	0	R3	R3	R3	R3	R3	R3	0	0	G3	G3	G3	G3	G3	G3	0	0	B3	B3	B3	B3	B3		
		0	R0	R0	R0	R0	R0	R0	0	0	G0	G0	G0	G0	G0	G0	0	0	B0	B0	B0	B0	B0	B0	0	0	R1	R1	R1	R1	R1	0	0		
13h	24	—	PAD								R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0			
14h			R0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	PAD									
15h			0	R0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0	B0	B0	B0	R1	R1	R1	R1	R1	R1	R1	R1		
		1	G1	G1	G1	G1	G1	G1	G1	G1	B1	B1	B1	B1	B1	B1	B1	R2	R2	R2	R2	R2	R2	R2	R2	G2	G2	G2	G2	G2	G2	G2	G2		
		2	B2	B2	B2	B2	B2	B2	B2	B2	R3	R3	R3	R3	R3	R3	R3	G3	G3	G3	G3	G3	G3	G3	G3	B3	B3	B3	B3	B3	B3	B3	B3		
16h	18	—	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	0	0	0	0	0	0	G0	G0	G0	B0	B0	B0	B0	B0	B0		
17h			0	0	0	0	0	0	0	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0		
18h	24	0	B0	B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0	B1	B1	B1	B1	B1	B1	B1	B1		
		1	G1	G1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	R1	R1	R1	B2	B2	B2	B2	B2	B2	B2	B2	G2	G2	G2	G2	G2	G2	G2	G2		
		2	R2	R2	R2	R2	R2	R2	R2	R2	B3	B3	B3	B3	B3	B3	B3	B3	G3	G3	G3	G3	G3	G3	G3	G3	R3	R3	R3	R3	R3	R3	R3	R3	
19h	12	—	PAD				R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	PAD				R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	
1Ah			R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	PAD				R1	R1	R1	R1	G1	G1	G1	G1	B1	B1	B1	B1	PAD				
1Bh	15	—	P	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	P	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1		
1Ch			—	R0	R0	R0	R0	R0	G0	G0	G0	G0	B0	B0	B0	B0	B0	P	R1	R1	R1	R1	R1	R1	G1	G1	G1	G1	G1	B1	B1	B1	B1	P	
1Dh	12	—	PAD				B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	PAD				B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1	
1Eh			—	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	PAD				B1	B1	B1	B1	G1	G1	G1	G1	R1	R1	R1	R1	PAD			
1Fh	15	—	P	B0	B0	B0	B0	B0	G0	G0	G0	G0	R0	R0	R0	R0	R0	P	B1	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	G1	R1	R1	R1	R1	
20h			—	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	P	B1	B1	B1	B1	B1	G1	G1	G1	G1	G1	R1	R1	R1	R1	P	
21h			0	0	0	B0	B0	B0	B0	B0	B0	0	0	G0	G0	G0	G0	G0	0	0	R0	R0	R0	R0	R0	R0	0	0	B1	B1	B1	B1	B1	B1	
		1	0	0	G1	G1	G1	G1	G1	G1	0	0	R1	R1	R1	R1	R1	R1	0	0	B2	B2	B2	B2	B2	B2	0	0	G2	G2	G2	G2	G2	G2	
		2	0	0	R2	R2	R2	R2	R2	R2	0	0	B3	B3	B3	B3	B3	B3	0	0	G3	G3	G3	G3	G3	G3	0	0	R3	R3	R3	R3	R3	R3	
22h	24	—	PAD								B0	B0	B0	B0	B0	B0	B0	G0	G0	G0	G0	G0	G0	G0	R0	R0	R0	R0	R0	R0	R0	R0			
23h	16	—	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R0	R0	R0	R0	R0	G0	G0	G0	G0	G0	G0	B0	B0	B0	B0	B0		
24h to 3Fh	—	—	Reserved								Reserved								Reserved								Reserved								

Table 9.4-19 Packed YCbCr Formats for WPF Output

WRFMT[6:0]	Packed YCbCr Output Format	Reference
40h	YCbCr4:4:4 semi-planar	Figure 9.4-9 ^{*4}
41h	YCbCr4:2:2 semi-planar (NV16, NV61 ^{*1})	
42h	YCbCr4:2:0 semi-planar (NV12, NV21 ^{*1})	
43h to 45h	Reserved	—
46h	YCbCr4:4:4 interleaved	Figure 9.4-10 ^{*4}
47h	YCbCr4:2:2 interleaved type 0 (UYVY, YUY2 ^{*2} , YVYU ^{*3})	
48h	YCbCr4:2:2 interleaved type 1	
49h	YCbCr4:2:0 interleaved ^{*5}	Figure 9.4-11 ^{*4}
4Ah	YCbCr4:4:4 planar	
4Bh	YCbCr4:2:2 planar (YV16)	
4Ch	YCbCr4:2:0 planar (YV12, I420)	
4Dh to 7Fh	Reserved	—

Note 1. When the output format is NV61 or NV21, set SPUVS (bit 14) to 1.

Note 2. When the output format is YUY2, set SPYCS (bit 15) to 1.

Note 3. When the output format is YVYU, set SPUVS (bit 14) to 1 and SPYCS (bit 15) to 1.

Note 4. **Figure 9.4-12** shows the definition of memory address for each pixel in **Figure 9.4-9** to **Figure 9.4-11**.

Note 5. Each line of plane is written twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).

For details of each YCbCr format, refer to **Figure 9.4-9** to **Figure 9.4-12**. In these figures, registers for the RPF are indicated; read them as registers for the WPF as follows.

(RPF registers in the figures)	→	(Corresponding WPF registers)
VI6_RPFn_SRCM_ADDR_Y.SRCM_ADDR_Y	→	VI6_WPFn_DSTN_ADDR_Y.DSTM_ADDR_Y
VI6_RPFn_SRCM_ADDR_C0.SRCM_ADDR_C0	→	VI6_WPFn_DSTN_ADDR_C0.DSTM_ADDR_C0
VI6_RPFn_SRCM_ADDR_C1.SRCM_ADDR_C1	→	VI6_WPFn_DSTN_ADDR_C1.DSTM_ADDR_C1

(5) WPF0 Data Swapping Register (LCDC_VSPD_VI6_WPF0_DSWAP)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 1010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	P_LLS	P_LWS	P_WDS	P_BTS
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	P_LLS	0h	RW	WPF Output Data Swapping in LONG LWORD Units The effect of this bit setting is the same as data swapping in the RPF; refer to Table 9.4-13 . 0b: Data swapping in LONG LWORD (64-bit) units is disabled 1b: Data swapping in LONG LWORD (64-bit) units is enabled
2	P_LWS	0h	RW	WPF Output Data Swapping in long word Units The effect of this bit setting is the same as data swapping in the RPF; refer to Table 9.4-13 . 0b: Data swapping in long word (32-bit) units is disabled 1b: Data swapping in long word (32-bit) units is enabled
1	P_WDS	0h	RW	WPF Output Data Swapping in Word Units The effect of this bit setting is the same as data swapping in the RPF; refer to Table 9.4-13 . 0b: Data swapping in word (16-bit) units is disabled 1b: Data swapping in word (16-bit) units is enabled
0	P_BTS	0h	RW	WPF Output Data Swapping in Byte Units The effect of this bit setting is the same as data swapping in the RPF; refer to Table 9.4-13 . 0b: Data swapping in byte (8-bit) units is disabled 1b: Data swapping in byte (8-bit) units is enabled

Table 9.4-13 shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

When data order in memory for each format is the same as in **Table 9.4-18** for RGB format and **Figure 9.4-9** to **Figure 9.4-12** for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is not the same as the definition, change data order within 16byte unit by these bits as shown in **Table 9.4-13**.

(6) WPF0 Rounding Control Register (LCDC_VSPD_VI6_WPF0_RNDCTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 1014h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	CBRM	-	-	ABRM[1:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CLMD[1:0]		-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	CBRM	0h	RW	Bit Count Reduction Method Selection for Data Storage in Packed RGB This bit specifies the method for reducing the number of bits when data is stored in the bit fields indicated as R, G, and B in Table 9.4-18 and the target bit fields are not 8 bits. 0b: Bit count conversion: The lower-order bits are truncated 1b: Bit count conversion: Rounding (rounding off)
27 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25, 24	ABRM[1:0]	0h	RW	Bit Count Reduction Method Selection for Data Storage in PAD These bits specify the method for reducing the number of bits when the data selected through VI6_WPF0_OUTFMT.PXA is stored in the bit fields indicated as PAD or P in Table 9.4-18 and the target bit field is four bits or one bit. A value of 10b can be specified only when the packed RGB format specified through VI6_WPF0_OUTFMT.WRFMT includes a 1-bit P field. In this case, when the data selected through VI6_WPF0_OUTFMT.PXA is greater than the ATHRESH value, 1 is stored in the P field; when the selected data is not greater than the ATHRESH value, 0 is stored. 00b: Bit count conversion: The lower-order bits are truncated 01b: Bit count conversion: Rounding (rounding off) 10b: Bit count conversion: Comparison with the threshold value (this setting is allowed only when the storage field is one bit) 11b: Setting prohibited
23 to 16	ATHRESH[7:0]	00h	RW	Threshold for Conversion to 1-Bit α Data These bits specify the threshold value used for conversion from 8-bit α data to one bit when the ABRM bits are set to 10b. When the 8-bit α value before bit count reduction is equal to or smaller than the ATHRESH value, 0 is stored as the reduced 1-bit α data. In other cases, 1 is stored as the 1-bit α data. A value from 0 to 255 can be specified.
15 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13, 12	CLMD[1:0]	0h	RW	Color Data Clipping These bits specify the method for clipping the YCbCr color data output from the WPF. When RGB color data is output from the WPF, specify 0b in these bits. 00b: Output value is not clipped (0-255) 01b: Output value is clipped: YCbCr mode 1 (16-235 (Y), 16-240 (Cb/Cr)) 10b: Output value is clipped: YCbCr mode 2 (Y/Cb/Cr = 1-254) 11b: Setting prohibited
11 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(7) WPF0 Destination Y Plane Memory Stride Register (LCDC_VSPD_VI6_WPF0_DSTM_STRIDE_Y)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 101Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PICT_STRD_Y[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	PICT_STRD_Y [15:0]	0000h	RW	Memory Stride of Destination Picture Y/RGB Plane These bits specify in 1-byte units the memory stride of the destination picture in the external memory to be written to by the WPF0 as shown in Figure 9.4-20 . A value from 0000h to FFFFh can be specified.

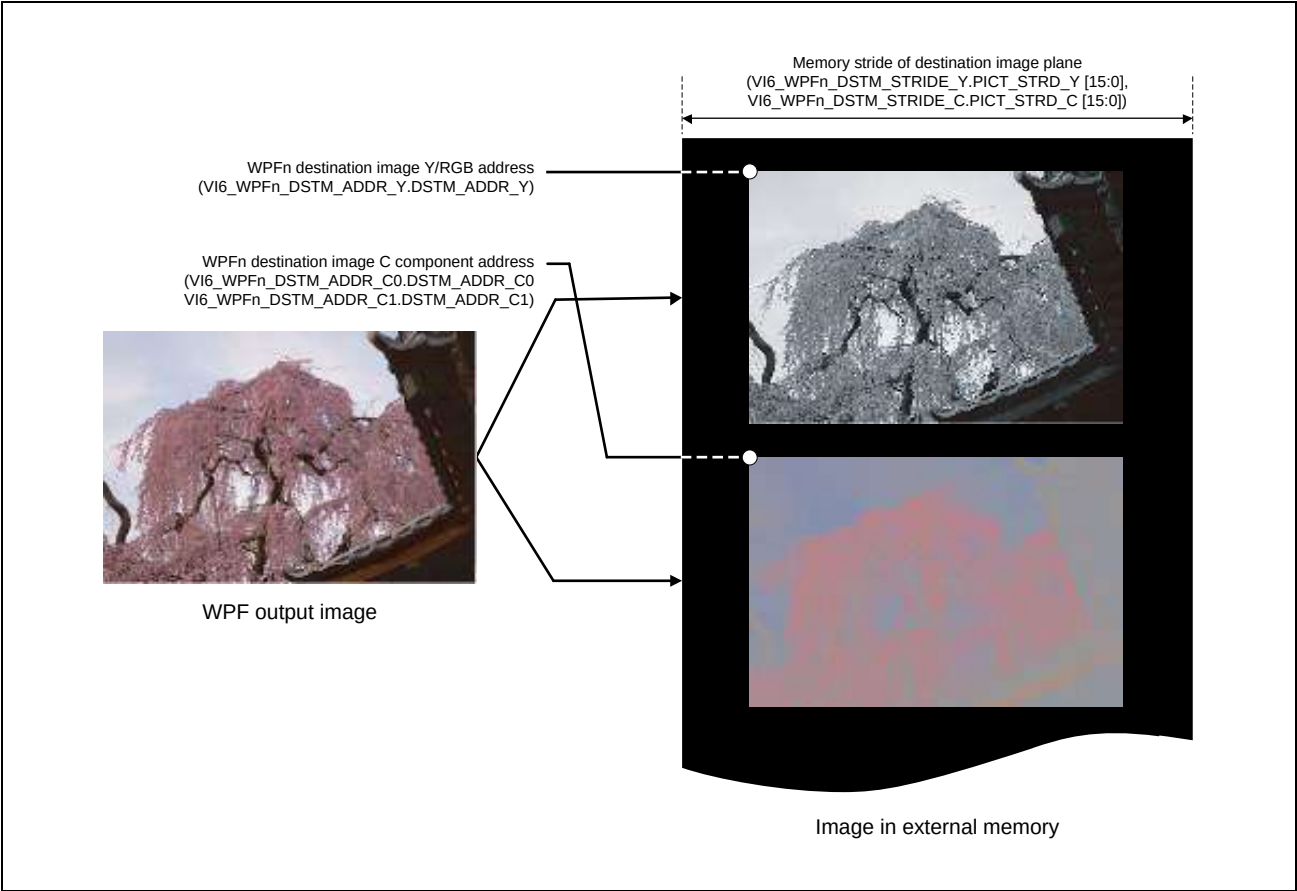


Figure 9.4-20 Writing Image Data to Destination Area in WPFn

(8) WPF0 Destination C Plane Memory Stride Register (LCDC_VSPD_VI6_WPF0_DSTM_STRIDE_C)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 1020h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PICT_STRD_C[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15 to 0	PICT_STRD_C [15:0]	0000h	RW	Memory Stride of Destination Picture C Plane These bits specify in 1-byte units the memory stride for the C plane of the destination picture in the external memory to be written to by the WPF0 as shown in Figure 9.4-20 . When the WPF0 outputs images in an RGB format, this setting is not used. When the WPF0 outputs images in YCbCr planar format, this setting is applied to both the Cb and Cr planes. A value from 0000h to FFFFh can be specified.

(9) WPF0 Destination Y/RGB Address Register (LCDC_VSPD_VI6_WPF0_DSTM_ADDR_Y)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 1024h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
DSTM_ADDR_Y[31:16]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DSTM_ADDR_Y[15:0]																	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DSTM_ADDR_Y [31:0]	0000_0000h	RW	Destination Image Y/RGB Plane Storing Address These bits specify in 1-byte units the address for storing the destination-image Y plane or packed RGB plane to be written to by the WPF0 in the method described later. A value from 0000 0000h to FFFF FFFFh can be specified.

[Destination Address Specification Method]

When flipping is not used, the start address of a frame (address FHA shown in **Figure 9.4-21**) should be specified as the destination address. When flipping is used, the destination address is not the frame start address (FHA); one of addresses A0 and A2 shown in **Figure 9.4-21** should be selected according to the combination of desired flipping (VI6_WPFn_OUTFMT.ROT setting).

To strictly define locations A0 and A2, let the horizontal size of the output image be H, the vertical size of the output image be V, and the memory stride (VI6_WPFn_DSTM_STRIDE_Y/C setting) be S as shown in **Figure 9.4-21**. Calculate the destination address (one of A0 and A2) using the formula shown in **Table 9.4-20** and specify it in the destination address storing register.

The values of variables L in **Table 9.4-20** depend on the other register settings and luminance and chrominance components. These values should be obtained by referring to **Table 9.4-21** when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_Y, or **Table 9.4-22** when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_C0 or VI6_WPFn_DSTM_ADDR_C1.

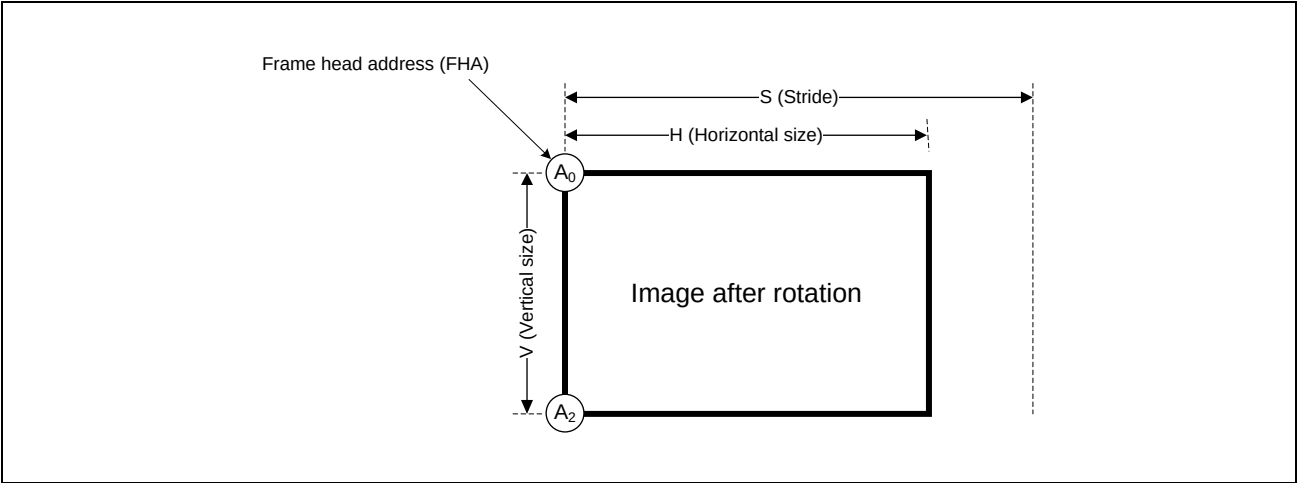


Figure 9.4-21 Location of Destination Address to be Specified for Flipping

Table 9.4-20 Destination Address A₀, A₁, A₂, and A₃ Calculation Formulas

VI6_WPFn_OUTFMT.ROT Setting	Formula for Calculating Address to be Set in VI6_WPFn_DSTM_ADDR_Y, VI6_WPFn_DSTM_ADDR_C0, and VI6_WPFn_DSTM_ADDR_C1
0	$A_0 = \text{FHA}$
1	$A_2 = \text{FHA} + (V \times L - 1) \times S$

Table 9.4-21 Value of L according to VI6_WPFn_OUTFMT.WRFMT Setting (for RGB and Luminance Y Address Calculation)

VI6_WPFn_OUTFMT.WRFMT	L
73	0.5
0 to 2, 4 to 35, 64 to 66, 70 to 72, or 74 to 76	1

Table 9.4-22 Value of L according to VI6_WPFn_OUTFMT.WRFMT Setting (for Chrominance C0 and C1 Address Calculation)

VI6_WPFn_OUTFMT.WRFMT	L
70 to 73	Not defined
66, 76	0.5
64 to 65, 74 to 75	1

(10) WPF0 Destination Chroma Address Register 0 (LCDC_VSPD_VI6_WPF0_DSTM_ADDR_C0)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 1028h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTM_ADDR_C0[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTM_ADDR_C0[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DSTM_ADDR_C0[31:0]	0000_0000h	RW	Destination Image C Plane Storing Address 0 These bits specify in 1-byte units the address for storing the destination-image C plane to be written to by the WPF0. Refer to the description of VI6_WPF0_DSTM_ADDR_Y for settings. Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in Table 9.4-19 or the Cb plane when a planar format is selected. When an interleaved format is selected or the output is in an RGB format, this setting is not used. A value from 0000 0000h to FFFF FFFFh can be specified. Refer to Figure 9.4-21 in 9.4.3.2.7 (9) WPFn Destination Y/RGB Address Registers (VI6_WPFn_DSTM_ADDR_Y) for settings.

(11) WPF0 Destination Chroma Address Register 1 (LCDC_VSPD_VI6_WPF0_DSTM_ADDR_C1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 102Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSTM_ADDR_C1[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSTM_ADDR_C1[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DSTM_ADDR_C1[31:0]	0000_0000h	RW	Destination Image C Plane Storing Address 1 These bits specify in 1-byte units the address for storing the Cr plane when the WPF0 outputs images to the external memory in YCbCr planar format shown in Table 9.4-19 . Refer to the description of VI6_WPF0_DSTM_ADDR_Y for settings. This setting is not used when the WPF outputs in YCbCr format that is not a planar format or in an RGB format. A value from 0000 0000h to FFFF FFFFh can be specified. Refer to Figure 9.4-21 in 9.4.3.2.7 (9) WPFn Destination Y/RGB Address Registers (VI6_WPFn_DSTM_ADDR_Y) for settings.

(12) WPF0 LIF Write Back Control Register (LCDC_VSPD_VI6_WPF0_WRBCK_CTRL)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 1034h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WBMD[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 2	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
1, 0	WBMD[1:0]		0h		RW		Display Data Write Back Control This bit is used for selecting the write back mode when the value of VI6_LIF0_CTRL.LIF_EN bit is set to 1b. 0b: Write Back Disabled 1b: Write Back Enabled									

9.4.3.2.8 DPR Control Registers

(1) Concept of DPR Settings

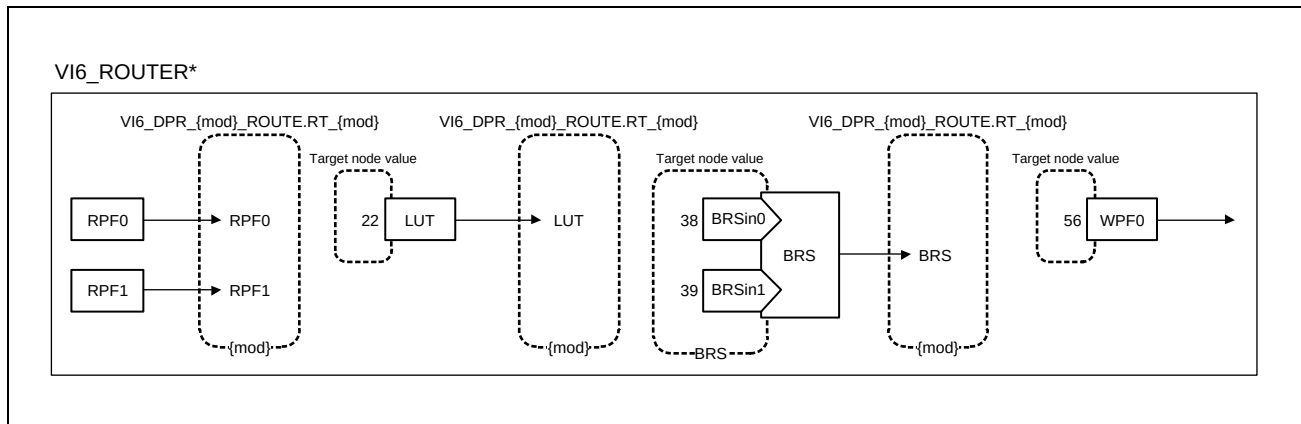


Figure 9.4-22 Node Register Names and Target Node Values on Data Path Router

In the VSPD internal data path, the order of processes can be specified as desired. The module for performing each process has a unique node value. Set each bit field in VI6_DPR_*_ROUTE to an appropriate node value shown in **Figure 9.4-22** to specify the target node to be connected behind each module.

For DPR settings, all of the following restrictions should be observed. If any of them is violated, even the WPF paths operating correctly at that time will be affected as well as the WPF paths connected through the DPR, and correct operation will not be guaranteed.

1. Specify 63 for the output node values of all RPFs and processing modules that are not used in the DPR. Here, make sure that no module is connected to a module for which 63 is specified as the node value. Note: VI6_DPR_{mod}_ROUTE_RT are not implemented in VSPD which doesn't have {mod}. See **Figure 9.4-1** whether VSPD has {mod} or not.
2. When specifying a value other than 63 for an output node value in the DPR, make sure that valid inputs (RPF0 to RPF1 or virtual RPF) and a target WPF are determined.
3. Only one module can be connected to each module; specifying the same target node value for two or more modules is prohibited.
4. Desired modules can be connected between each RPF and BRS input port, but all RPFs specified as the sources for a BRS input port should have the same target WPF.
5. Make appropriate routing or RPF register settings so that the color space formats (RGB/YCbCr) for all BRS input ports are the same.
6. Do not connect the output of any module as the input to the same module (in the BRS case, any input port) even when there is another module between the output and input (creating a loop is prohibited).
7. Each node can be used only once throughout all paths from RPFn to WPFn. When a module shown in **Figure 9.4-22** is assigned in one RPF to WPF path, it cannot be used in another RPF to WPF path.
8. While a WPF is operating, modifying the DPR connection settings in VI6_DPR_*_ROUTE is prohibited for modules used by the WPF but allowed for modules not used by the WPF. Be careful not to accidentally modify the settings of the modules included in the WPF path that is operating.
9. Be careful to do the following setting when BRS is used.
VI6_DPR_ILV_BRS_ROUTE.BRSSEL = 1

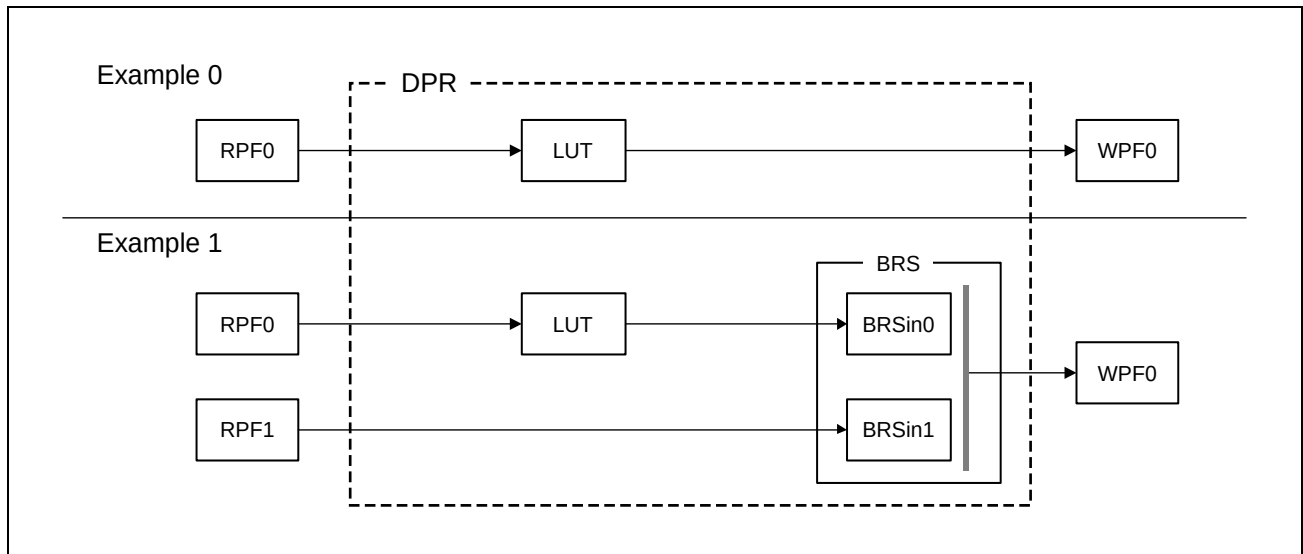


Figure 9.4-23 Examples of Internal Data Path Routing

Figure 9.4-23 shows examples of internal data path routing. Example 0 is WPF0 processing (RPF0 is the source RPF), and example 1 is also WPF0 (RPF0 to RPF1 are the source RPFs). Each example has the configuration shown in **Figure 9.4-23**. Example 0 performs LUT (e.g. γ correction) processing. Example 1 performs LUT processing (e.g. γ correction), for input 0 (RPF0) and then applies blending or raster operation between the resultant data and input data 1 (RPF1). The VI6_DPR_*_ROUTE settings for these examples are shown in **Table 9.4-23**. The bit fields for the modules that are not used in the examples should be set to 63.

Table 9.4-23 VI6_DPR_*_ROUTE Register Settings in Connection Examples Shown in **Figure 9.4-23**. Although both examples show in **Figure 9.4-23** include operation in image processing modules, connect the RPF to the WPF directly when only image format conversion or packed format conversion is required.

Table 9.4-23 Examples of Internal Data Path Routing

	Register Name	Setting	
Example 0	VI6_DPR_RPF0_ROUTE.RT_RPF0	22	(To LUT)
	VI6_DPR_RPF1_ROUTE.RT_RPF1	63	(UNUSED)
	VI6_DPR_LUT_ROUTE.RT	56	(To WPF0)
	VI6_DPR_ILV_BRS_ROUTE.RT	63	(UNUSED)
Example 1	VI6_DPR_RPF0_ROUTE.RT_RPF0	22	(To LUT)
	VI6_DPR_RPF1_ROUTE.RT_RPF1	39	(To BRSin1)
	VI6_DPR_LUT_ROUTE.RT	38	(To BRSin0)
	VI6_DPR_ILV_BRS_ROUTE.RT	56	(To WPF0)

(2) RPFn Routing Register (LCDC_VSPD_VI6_DPR_RPFn_ROUTE) (n = 0, 1)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 2000h (n=0) <LCDC_vspd_base> + 2004h (n=1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	RT_RPF0[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 6	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
5 to 0	RT_RPF0[5:0]		00h	RW	RPF0 Target Node Value These bits specify the target node value for RPF0. When using RPF0, refer to Figure 9.4-22 for settings. When RPF0 is not started through the VI6_WPF0_SRCRPF setting, specify 63.											

(3) WPF0 Timing Control Register (LCDC_VSPD_VI6_DPR_WPF0_FPORCH)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 2014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	FP_WPF0[5:0]						-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 8	FP_WPF0[5:0]	00h	RW	WPF0 Internal Operation Timing Setting Specify 5.
7 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

(4) LUT Routing Register (LCDC_VSPD_VI6_DPR_LUT_ROUTE)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 203Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	FXA[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	FP[5:0]						-	-	RT[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 16	FXA[7:0]	00h	RW	Fixed α Output Value for LUT The LUT does not support input/output of the α value. The α value input to the LUT is discarded, and the fixed α value specified in these bits is always output from the LUT. A value from 0 to 255 can be specified.
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 8	FP[5:0]	00h	RW	LUT Internal Operation Timing Setting Specify 0.
7, 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	RT[5:0]	00h	RW	LUT Target Node Value These bits specify the target node value for the LUT. When using the LUT, refer to Figure 9.4-22 for settings. When not using the LUT, specify 63.

(5) BRS Routing Register (LCDC_VSPD_VI6_DPR_BRS_ROUTE)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 2050h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	BRSSEL	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	FP[5:0]						-	-	RT[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	BRSSEL	0h	RW	Always specify 1.
27 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 8	FP[5:0]	00h	RW	BRS Internal Operation Timing Setting Specify 0.
7, 6	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
5 to 0	RT[5:0]	00h	RW	BRS Target Node Value These bits specify the target node value for the BRS. When using the BRS, refer to Figure 9.4-22 for settings. When not using the BRS, specify 63.

9.4.3.2.9 LUT Control Register

(1) LUT Control Register (LCDC_VSPD_VI6_LUT_CTRL)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 2800h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LUT_EN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	LUT_EN	0h	RW	1D-LUT Enable/Disable Enables or disables the 1D-LUT function by the LUT. When the 1D-LUT is used, the color component curve information needs to be set separately in the LUT table. For the LUT table settings, refer to 9.4.4.7.1 LUT . 0b: 1D-LUT function is disabled 1b: 1D-LUT function is enabled

In the LUT, various image processing, such as curves with high operation load (e.g., γ correction), negative-positive conversion, and gain adjustment of images, can be achieved by the data replacement processing by the 1D-LUT. As shown in **Figure 9.4-24**, the LUT replaces each component of the input pixel data using the set replacement table of 256 entries. For example, if the LUT is set as in **Figure 9.4-25**, when there is an input of 150, the data stored in address 150 of the 1D-LUT is read and output as the LUT output. **Figure 9.4-25** shows a case in which the input and output become equal for convenience in explaining.

In the LUT settings shown in **Figure 9.4-26**, the input bits are reversed. This has the effect of negative-positive flipping. In **Figure 9.4-27**, γ correction ($\gamma = 1.8$ is shown as an example) is possible. As described above, information to be set in the LUT indicates LUT processing characteristics. If the same value is set for each component in the LUT, an equal effect can be obtained for each component of the input image when it is processed. If the LUT is set with different characteristics for each component, the processing characteristics can be changed for each component.

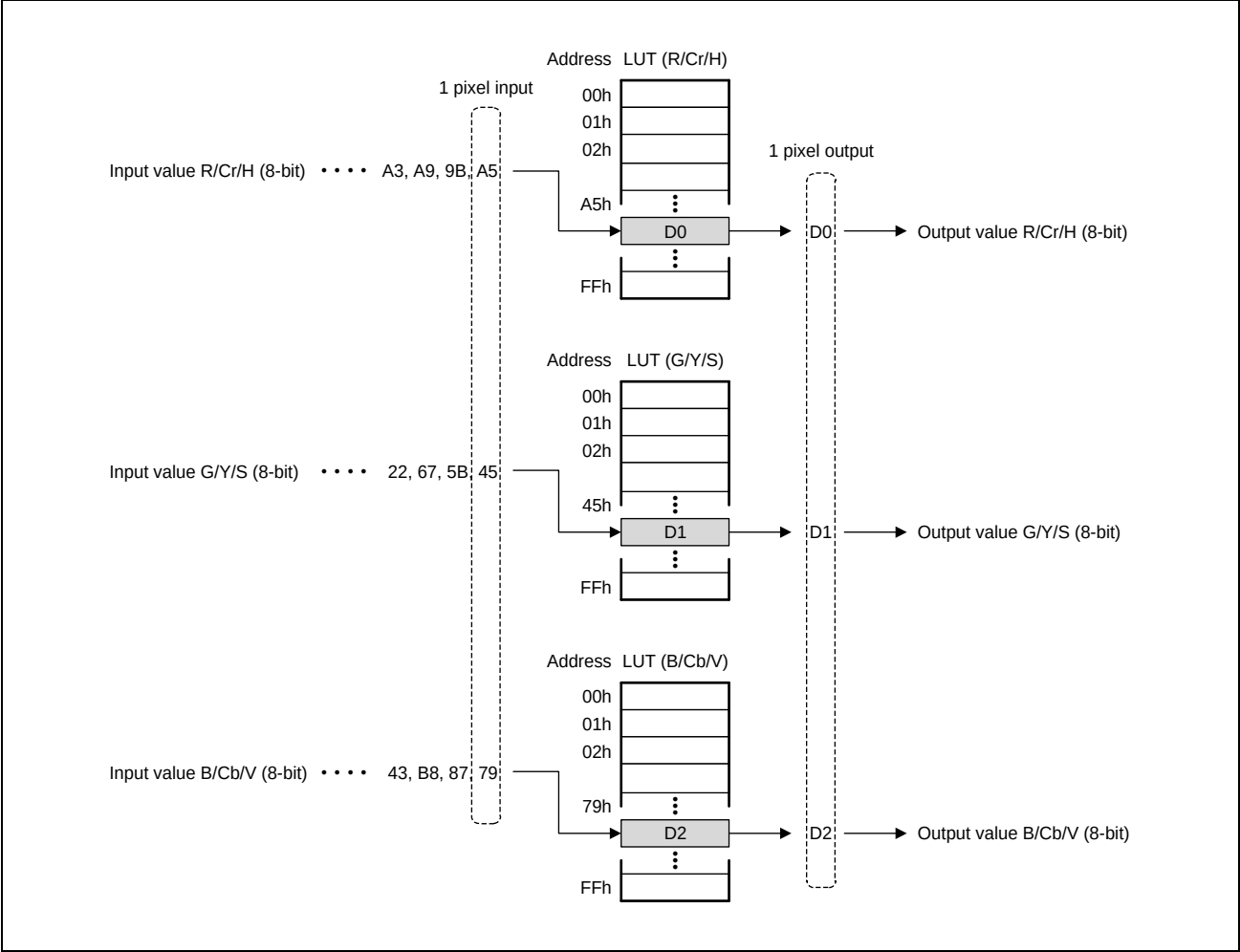


Figure 9.4-24 Relationship between Input and Output for 1D-LUT Table

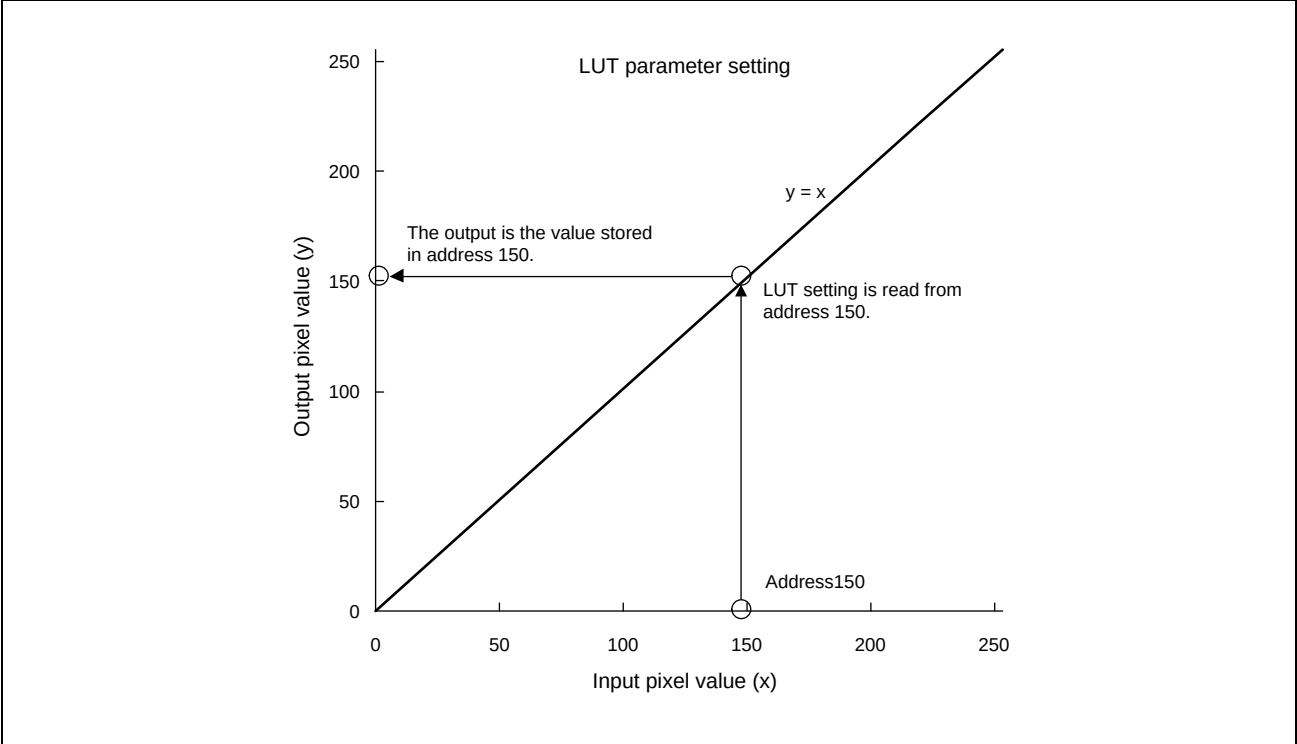


Figure 9.4-25 Setting Example in Which Output Becomes Equal to Input

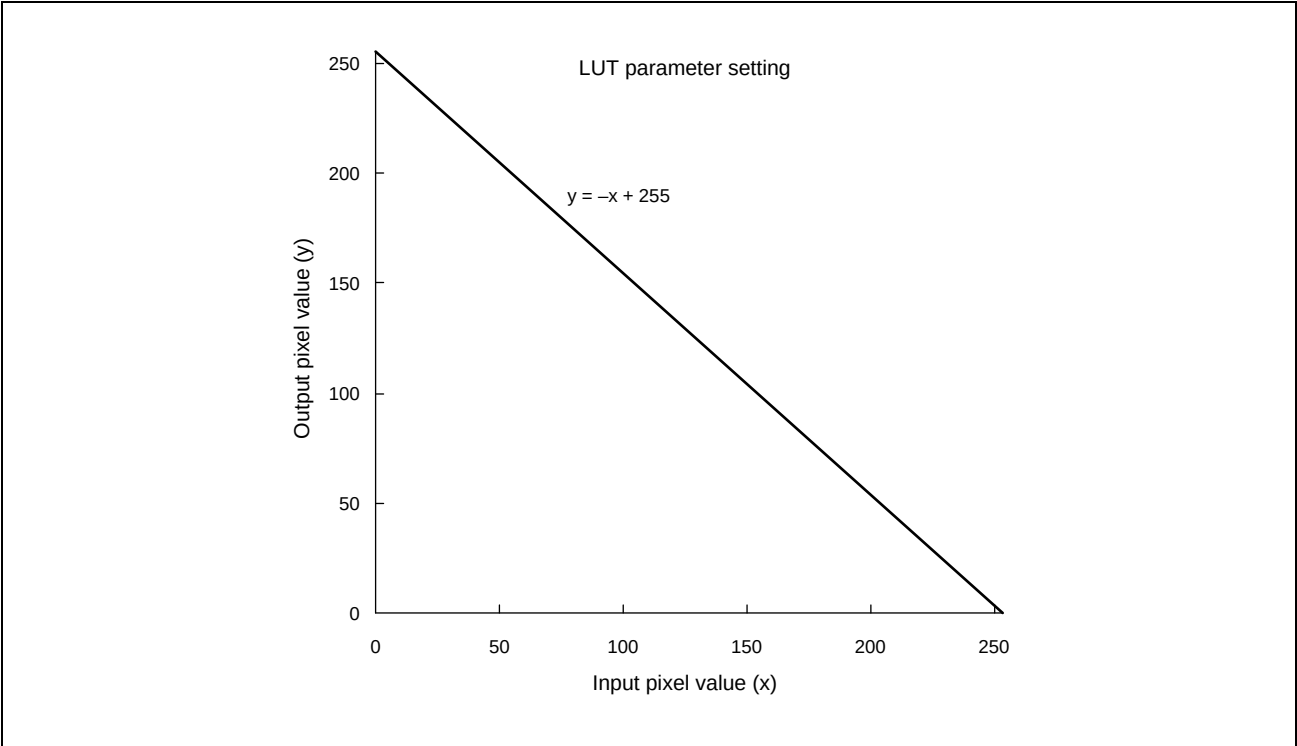


Figure 9.4-26 Setting Example of Negative-positive Conversion

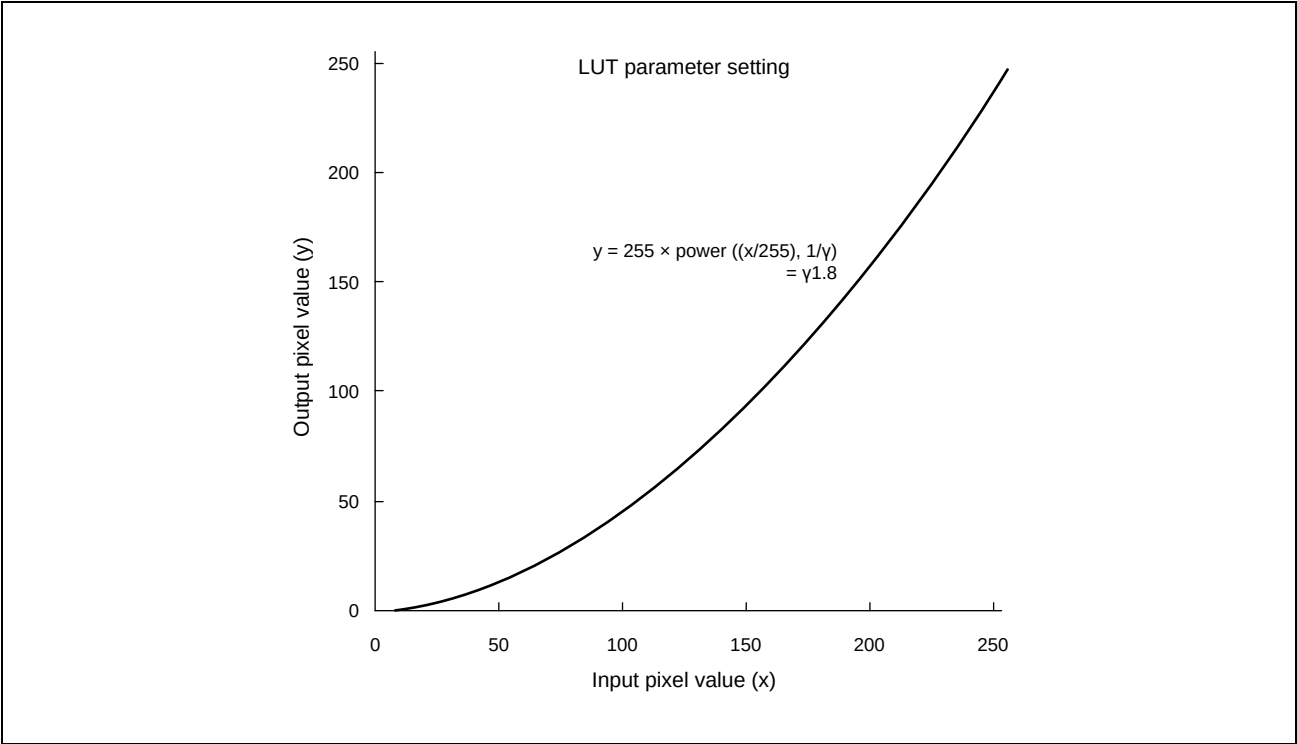


Figure 9.4-27 Setting Example of γ Correction

9.4.3.2.10 BRS Control Registers

(1) BRS Input Control Register (LCDC_VSPD_VI6_BRS_INCTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3900h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	NRM	-	-	-	-	-	-	-	-	-	-	D1ON	D0ON
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ODE1	DITH1[2:0]			ODE0	DITH0[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	NRM	0h	RW	Color Data Normalization Enables or disables division by the α value of the color data in BRS blending operation. This is used when converting the RGB color data format to which the α value is multiplied (pre-multiplied color) into the RGB color data format to which the α value is not multiplied (non pre-multiplied color). Do not use this for the YCbCr format. 0b: Divider (DIV unit in Figure 9.4-28) does not divide the color value by α 1b: Divider (DIV unit in Figure 9.4-28) divides the color value by α
27 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	D1ON	0h	RW	Ordered dither (mode B) Enable of BRS Input 1 Enables or disables dithering (color reduction) of BRS input 1 (BRSin1 in Figure 9.4-28). 0b: Dithering (mode B) of BRSin1 is disabled 1b: Dithering (mode B) of BRSin1 is enabled When ODE1 in this bit is set to 1, set 0 to this bit.
16	D0ON	0h	RW	Ordered dither (mode B) Enable of BRS Input 0 Enables or disables dithering (color reduction) of BRS input 0 (BRSin0 in Figure 9.4-28). 0b: Dithering (mode B) of BRSin0 is disabled 1b: Dithering (mode B) of BRSin0 is enabled When ODE0 in this bit is set to 1, set 0 to this bit.
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7	ODE1	0h	RW	Ordered Dither (mode A) of CH1 Input to BRS Enable/Disable 0b: Ordered dither (mode A) of BRSin1 is disabled. 1b: Ordered dither (mode A) of BRSin1 is enabled. Ordered dither is available only for 18bpp. So, when ODE1 bit is equal to 1, set DITH1 = 1. When ODE1 bit is equal to 0, BRSin1 dither method is specified by D1ON in the register Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp.
6 to 4	DITH1[2:0]	0h	RW	Dithering of CH1 Input to BRS These bits specify the number of colors for pixels after dithering (color reduction) when dithering (color reduction) for pixel information is enabled through the D1ON bit. When dithering (color reduction) for pixel information is disabled, specify 0h in these bits. 0h: Dithering of BRSin1 input image is disabled 1h: Dithering of BRSin1 input image at 18 bpp (RGB666: 260000 colors) 2h: Dithering of BRSin1 input image at 16 bpp (RGB565: 65535 colors) 3h: Dithering of BRSin1 input image at 15 bpp (RGB555: 32768 colors) 4h: Dithering of BRSin1 input image at 12 bpp (RGB444: 4096 colors) 5h: Dithering of BRSin1 input image at 8 bpp (RGB332: 256 colors) 6, 7h: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3	ODE0	0h	RW	Ordered Dither (mode A) of CH0 Input to BRS Enable/Disable 0b: Ordered dither (mode A) of BRSin0 is disabled. 1b: Ordered dither (mode A) of BRSin0 is enabled. Ordered dither is available only for 18bpp. So, when ODE0 bit is equal to 1b, set DITH0 = 1b. When ODE0 bit is equal to 0, BRSin0 dither method is specified by D0ON in the register Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp.
2 to 0	DITH0 [2:0]	0h	RW	Dithering of CH0 Input to BRS These bits specify how to perform dithering of the CH0 input to the BRS. The setting method is the same as that for the DITH1 bits. Read the description of the DITH1 bits with BRSin0 and D0ON replacing BRSin1 and D1ON, respectively.

Figure 9.4-28 shows the configuration of the BRS. For the BRS inputs, there are two inputs from the DPR and one internal input as a virtual RPF. BRSin0 to BRSin1 are input ports that have the target node values shown in **Figure 9.4-22**, and they can be connected to any module on the DPR. The same color space (YCbCr or RGB) has to be used for the two inputs from the DPR to the BRS.

The virtual RPF inside the BRS is an input unit not connected to the DPR. It is called the "virtual RPF" because it outputs images internally created by the BRS. Starting of the virtual RPF is controlled by VI6_WPFn_SRCRPF.VIR_ACT2, and the single-color data created at the virtual RPF can be used for blending or raster operation (ROP) with data from the other input units BRSin0 to BRSin1. The color space for the single color to be set for the virtual RPF needs to match the color space of the two inputs from the DPR to the BRS. For this setting method, see **(4) Color Information Register of BRS Input Virtual RPF (LCDC_VSPD_VI6_BRS_VIRRPF_COL)**.

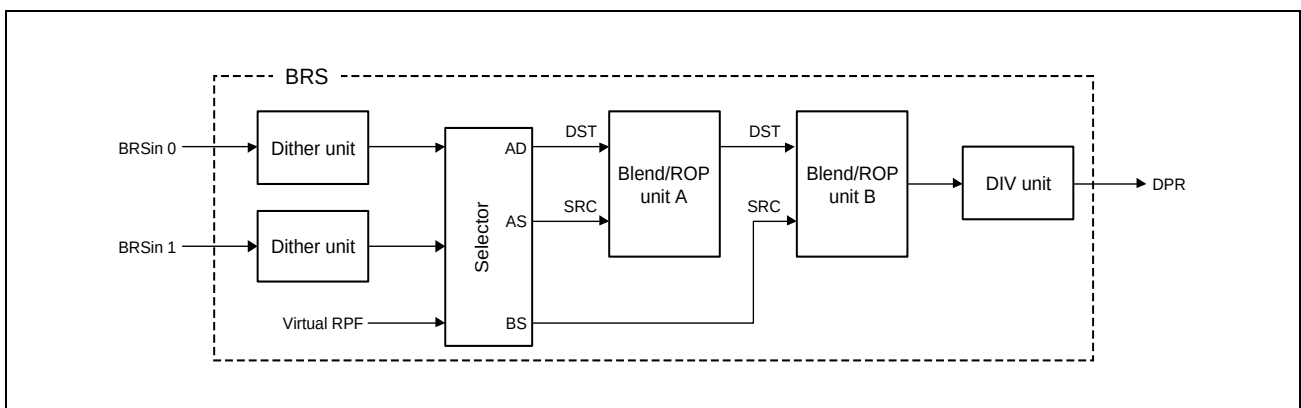


Figure 9.4-28 BRS Configuration

The selector in **Figure 9.4-28** is used to select the SRC and DST inputs to blending/ROP units A to B from BRSin0 to BRSin1 which are inputs from the DPR and the virtual RPF. The SRC and DST input sources for blending/ROP units A to B are either uniquely determined based on the configuration shown in **Figure 9.4-28** or selected as desired by registers. The input sources that can be arbitrarily selected by registers are AD, AS and BS, which correspond to the registers shown in **Table 9.4-24**.

Table 9.4-24 Correspondence between Selector Output Destinations and Register Bits for BRS

Selector Output	Output Destination	Register Bits
AD	Blending/ROP unit A - DST	VI6_BRSA_CTRL.DSTSEL
AS	Blending/ROP unit A - SRC	VI6_BRSA_CTRL.SRCSEL
BS	Blending/ROP unit B - SRC	VI6_BRSB_CTRL.SRCSEL

(2) Size Register of BRS Input Virtual RPF (LCDC_VSPD_VI6_BRS_VIRRPF_SIZE)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 3904h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VIR_HSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VIR_VSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	VIR_HSIZE [12:0]	0000h	RW	Virtual RPF Horizontal Size These bits set the horizontal size of an image from the virtual RPF shown in Figure 9.4-28 . A value from 1 to 8190 can be specified.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	VIR_VSIZE [12:0]	0000h	RW	Virtual RPF Vertical Size These bits set the vertical size of an image from the virtual RPF shown in Figure 9.4-28 . A value from 1 to 8190 can be specified.

The virtual RPF has only a function to output a fixed α value and a fixed pixel value. The virtual RPF can internally create a single-color image without accessing external memory via the MAU. Same as images from the other BRS input ports, a sublayer can be blended on an image created in this manner with the image used as the background (master layer). In turn, when using the image as a sublayer, it can be drawn on the master layer as a window.

(3) Display Location Register of BRS Input Virtual RPF (LCDC_VSPD_VI6_BRS_VIRRPF_LOC)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3908h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	HCOORD[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VCOORD[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	HCOORD [12:0]	0000h	RW	Horizontal Coordinate of Virtual RPF Location on Master Layer These bits specify the horizontal coordinate of where to locate the left-edge pixel of the virtual RPF's layer, with the left-edge pixel of the master layer set at coordinate 0. This setting should be made in pixel units. A value from 0 to 8189 can be specified. When the virtual RPF is specified as the master layer by VI6_WPF0_SRCRPF.VIR_ACT2, set these bits to 0b.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	VCOORD[12:0]	0000h	RW	Vertical Coordinate of Virtual RPF Location on Master Layer These bits specify the vertical coordinate of where to locate the top-edge pixel of the virtual RPF's layer, with the top-edge pixel of the master layer set at coordinate 0. This setting should be made in pixel units. A value from 0 to 8189 can be specified. When the virtual RPF is specified as the master layer by VI6_WPF0_SRCRPF.VIR_ACT2, set these bits to 0b.

(4) Color Information Register of BRS Input Virtual RPF (LCDC_VSPD_VI6_BRS_VIRRPF_COL)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 390Ch															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	COL_A[7:0]								COL_RCR[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	COL_GY[7:0]								COL_BCB[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	COL_A[7:0]	00h	RW	Fixed α of Virtual RPF These bits set the fixed α value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified.
23 to 16	COL_RCR[7:0]	00h	RW	Fixed R/Cr of Virtual RPF These bits set the fixed R/Cr value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified.
15 to 8	COL_GY[7:0]	00h	RW	Fixed G/Y of Virtual RPF These bits set the fixed G/Y value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified.
7 to 0	COL_BCB[7:0]	00h	RW	Fixed B/Cb of Virtual RPF These bits set the fixed B/Cb value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified.

The transparency information and color information of the single color that is created by the virtual RPF are set in the bits of this register. As described earlier, the color information is set for the YCbCr or RGB color space. The color space to be set in this register depends on the register settings of the environment and other modules to which the BRS is connected by the DPR. Two cases can be considered. Since the α value (COL_A) is transparency information and irrelevant to the concept of color space, the same setting is made for either the YCbCr or RGB color space.

[Case 1: When an input other than the virtual RPF is used]

When the source RPF is connected to any one of the BRS input ports (BRSin0 to BRSin1) other than the virtual RPF and valid data is being supplied, the same color space data as the color space for the BRS inputs should be set in this register as the color space for the virtual RPF's color information. This is based on the restriction of "all BRS inputs must have the same color space", as described in **(1) Concept of DPR Settings** or **(1) BRS Input Control Register (LCDC_VSPD_VI6_BRS_INCTRL)**.

[Case 2: When only the virtual RPF is used]

When only the virtual RPF is used as the source RPF of WPFn, RPFn is not connected to the BRS, as shown in **Figure 9.4-29**. Thus, there is no color space for another input that the color space for the virtual RPF has to follow, as in case 1. This means that the color space of the data output by the BRS is determined by the WPF setting.

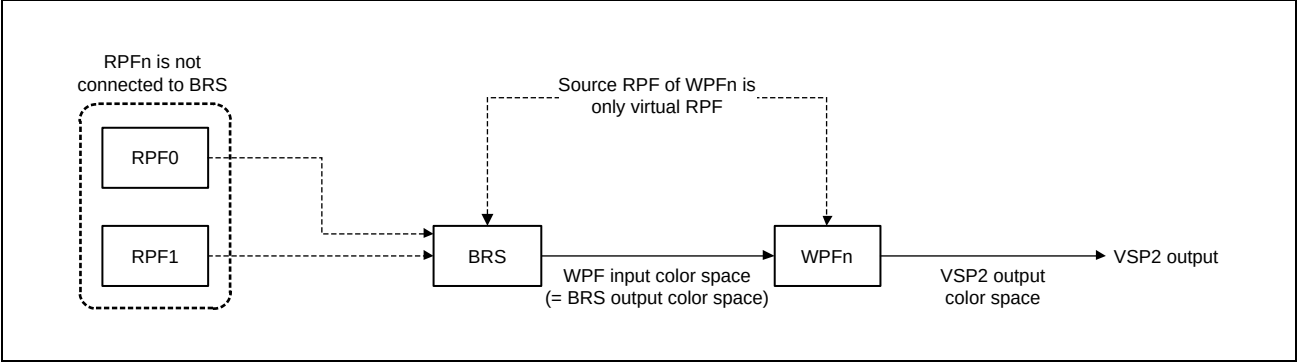


Figure 9.4-29 Relationship between DPR Connection and Color Space When Only Virtual RPF is Used

Figure 9.4-29, the output color space of the VSPD (= output color space of WPFn) is determined by VI6_WPFn_OUTFMT.WRFMT. When bit 6 in VI6_WPFn_OUTFMT.WRFMT (WRFMT [6]) is 0, the color space is RGB, while when it is 1, the color space is YCbCr. Next, the WPF input color space (= BRS output color space) is determined by the relationship between the WPF output color space and VI6_WPFn_OUTFMT.CSC. When VI6_WPFn_OUTFMT.CSC is 0, the WPF output color space and WPF input color space (= BRS output color space) are the same. When VI6_WPFn_OUTFMT.CSC is 1, the WPF output color space and WPF input color space (= BRS output color space) are the opposite. This relationship is summarized in **Table 9.4-25**.

The color space for the virtual RPF’s color information should be set in this register according to the “BRS output color space” shown in **Table 9.4-25**.

Table 9.4-25 Relationship between WPF Output Color Space and BRS Output Color Space

VI6_WPFn_OUTFMT Register Bit Settings				BRS Output Color Space (= WPF Input Color Space)
Bit 6 in WRFMT		CSC		
0	(WPF output is RGB)	0	(YCbCr to RGB conversion is disabled)	RGB
0	(WPF output is RGB)	1	(YCbCr to RGB conversion is enabled)	YCbCr
1	(WPF output is YCbCr)	0	(RGB to YCbCr conversion is disabled)	YCbCr
1	(WPF output is YCbCr)	1	(RGB to YCbCr conversion is enabled)	RGB

(5) BRS Control Register A (LCDC_VSPD_VI6_BRSA_CTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3910h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBC	-	-	-	-	-	-	-	-	DSTSEL[2:0]			-	SRCSEL[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	RW	RW	RW	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CROP[3:0]				AROP[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	RBC	0h	RW	Operation Type of Blending/ROP Unit A Specifies the operation type for blending/ROP unit A shown in Figure 9.4-28 . 0b: ROP (raster operation) 1b: Blending operation
30 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22 to 20	DSTSEL[2:0]	0h	RW	Input Selection for DST Side of Blending/ROP Unit A These bits select the input for the DST side of blending/ROP unit A shown in Figure 9.4-28 . These bits specify the connection between the BRS input port and the DST separately from the setting of connections between other modules and the BRS input port through the DPR. 000b: BRS input 0 (BRSin0) is input to DST 001b: BRS input 1 (BRSin1) is input to DST 100b: Virtual RPF is input to DST 010b, 011b, 101b to 111b: Setting prohibited [Note] The DSTSEL bits for blending/ROP unit B are reserved. The write value should always be 0b.
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18 to 16	SRCSEL[2:0]	0h	RW	Input Selection for SRC Side of Blending/ROP Unit A These bits select the input for the SRC side of blending/ROP unit A shown in Figure 9.4-28 . These bits specify the connection between the BRS input port and the SRC separately from the setting of connections between other modules and the BRS input port through the DPR. 000b: BRS input 0 (BRSin0) is input to SRC 001b: BRS input 1 (BRSin1) is input to SRC 100b: Virtual RPF is input to SRC 010b, 011b, 101b to 111b: Setting prohibited
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7 to 4	CROP[3:0]	0h	RW	Color Data ROP Operator These bits select the ROP operator of the color data in blending/ROP unit A. Select the opcode for ROP operation from Table 9.4-26 .
3 to 0	AROP[3:0]	0h	RW	α Data ROP Operator These bits select the ROP operator of the α data in blending/ROP unit A. Select the opcode for ROP operation from Table 9.4-26 .

Table 9.4-26 ROP Operator of Blending/ROP Unit m (m = A, B)

Opcode	Operator
0000b	NOP(D)
0001b	AND(S & D)
0010b	AND_REVERSE(S & ~D)
0011b	COPY(S)
0100b	AND_INVERTED(~S & D)
0101b	CLEAR(0)
0110b	XOR(S ^ D)
0111b	OR(S D)
1000b	NOR(~(S D))
1001b	EQUIV(~(S ^ D))
1010b	INVERT(~D)
1011b	OR_REVERSE(S ~D)
1100b	COPY_INVERTED(~S)
1101b	OR_INVERTED(~S D)
1110b	NAND(~(S & D))
1111b	SET(all1)

(6) BRS Blend Control Register A (LCDC_VSPD_VI6_BRSA_BLD)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3914h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBES	CCMDX[2:0]			-	CCMDY[2:0]			ABES	ACMDX[2:0]			-	ACMDY[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COEFX[7:0]								COEFY[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	CBES	0h	RW	Blending Expression Selection Selects the blending expression of the color data in the BRS (VI6_BRSA_CTRL.RBC = 1b). Blending coefficients are specified by the CCMDX and CCMDY bits. 0b: CCMDX × (DST color data) + CCMDY × (SRC color data) 1b: CCMDX × (DST color data) - CCMDY × (SRC color data)
30 to 28	CCMDX[2:0]	0h	RW	Blending Coefficient X Selection These bits specify coefficient X used in the blending expression determined by the CBES bit. 000b: DST α data is used as blending coefficient X 001b: 255 - (DST α data) is used as blending coefficient X 010b: SRC α data is used as blending coefficient X 011b: 255 - (SRC α data) is used as blending coefficient X 100b: Fixed α value 0 (COEFX setting)
27	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26 to 24	CCMDY[2:0]	0h	RW	Blending Coefficient Y Selection These bits specify coefficient Y used in the blending expression determined by the CBES bit. 000b: DST α data is used as blending coefficient Y 001b: 255 - (DST α data) is used as blending coefficient Y 010b: SRC α data is used as blending coefficient Y 011b: 255 - (SRC α data) is used as blending coefficient Y 100b: Fixed α value 1 (COEFY setting)
23	ABES	0h	RW	Blending α Creation Expression Specifies the expression for creating α data after blending by blending/ROP unit m (m = A, B). α creation coefficients are specified by the ACMDX and ACMDY bits. 0b: ACMDX × (DST α data) + ACMDY × (SRC α data) 1b: ACMDX × (DST α data) - ACMDY × (SRC α data)
22 to 20	ACMDX[2:0]	0h	RW	α Creation Coefficient X These bits specify α creation coefficient X used in the α creation expression determined by the ABES bit. 000b: (α creation coefficient X) = (DST α data) 001b: (α creation coefficient X) = 255 - (DST α data) 010b: (α creation coefficient X) = (SRC α data) 011b: (α creation coefficient X) = 255 - (SRC α data) 100b: (α creation coefficient X) = Fixed α value 0 (COEFX setting) 101b to 111b: Setting prohibited
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	ACMDY[2:0]	0h	RW	α Creation Coefficient Y These bits specify α creation coefficient Y used in the α creation expression determined by the ABES bit. 000b: (α creation coefficient Y) = (DST α data) 001b: (α creation coefficient Y) = 255 - (DST α data) 010b: (α creation coefficient Y) = (SRC α data) 011b: (α creation coefficient Y) = 255 - (SRC α data) 100b: (α creation coefficient Y) = Fixed α value 1 (COEFY setting) 101b to 111b: Setting prohibited
15 to 8	COEFX[7:0]	00h	RW	Fixed α Value 0 These bits specify fixed α value 0 used when the CCMDX or ACMDX bits are set to 100b. A value from 00h to FFh can be specified.
7 to 0	COEFY[7:0]	00h	RW	Fixed α Value 1 These bits specify fixed α value 1 used when the CCMDY or ACMDY bits are set to 100b. A value from 00h to FFh can be specified.

(7) BRS Control Register B (LCDC_VSPD_VI6_BRSB_CTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3918h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBC	-	-	-	-	-	-	-	-	-	-	-	-	SRCSEL[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CROP[3:0]				AROP[3:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	RBC	0h	RW	Operation Type of Blending/ROP Unit B Specifies the operation type for blending/ROP unit B shown in Figure 9.4-28 . 0b: ROP (raster operation) 1b: Blending operation
30 to 19	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18 to 16	SRCSEL[2:0]	0h	RW	Input Selection for SRC Side of Blending/ROP Unit B These bits select the input for the SRC side of blending/ROP unit A shown in Figure 9.4-28 . These bits specify the connection between the BRS input port and the SRC separately from the setting of connections between other modules and the BRS input port through the DPR. 000b: BRS input 0 (BRSin0) is input to SRC 001b, 010b, 011b, 100b, 101b to 111b: Setting prohibited
15 to 8	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
7 to 4	CROP[3:0]	0h	RW	Color Data ROP Operator These bits select the ROP operator of the color data in blending/ROP unit B. Select the opcode for ROP operation from Table 9.4-26 .
3 to 0	AROP[3:0]	0h	RW	α Data ROP Operator These bits select the ROP operator of the α data in blending/ROP unit B. Select the opcode for ROP operation from Table 9.4-26 .

(8) BRS Blend Control Register B (LCDC_VSPD_VI6_BRSB_BLD)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 391Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CBES	CCMDX[2:0]			-	CCMDY[2:0]			ABES	ACMDX[2:0]			-	ACMDY[2:0]		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COEFX[7:0]								COEFY[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	CBES	0h	RW	Blending Expression Selection Selects the blending expression of the color data in the BRS (VI6_BRSB_CTRL.RBC = 1). Blending coefficients are specified by the CCMDX and CCMDY bits. 0b: $CCMDX \times (DST \text{ color data}) + CCMDY \times (SRC \text{ color data})$ 1b: $CCMDX \times (DST \text{ color data}) - CCMDY \times (SRC \text{ color data})$
30 to 28	CCMDX[2:0]	0h	RW	Blending Coefficient X Selection These bits specify coefficient X used in the blending expression determined by the CBES bit. 000b: DST α data is used as blending coefficient X 001b: $255 - (DST \text{ } \alpha \text{ data})$ is used as blending coefficient X 010b: SRC α data is used as blending coefficient X 011b: $255 - (SRC \text{ } \alpha \text{ data})$ is used as blending coefficient X 100b: Fixed α value 0 (COEFX setting)
27	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26 to 24	CCMDY[2:0]	0h	RW	Blending Coefficient Y Selection These bits specify coefficient Y used in the blending expression determined by the CBES bit. 000b: DST α data is used as blending coefficient Y 001b: $255 - (DST \text{ } \alpha \text{ data})$ is used as blending coefficient Y 010b: SRC α data is used as blending coefficient Y 011b: $255 - (SRC \text{ } \alpha \text{ data})$ is used as blending coefficient Y 100b: Fixed α value 1 (COEFY setting)
23	ABES	0h	RW	Blending α Creation Expression Specifies the expression for creating α data after blending by blending/ROP unit m (m = A, B). α creation coefficients are specified by the ACMDX and ACMDY bits. 0b: $ACMDX \times (DST \text{ } \alpha \text{ data}) + ACMDY \times (SRC \text{ } \alpha \text{ data})$ 1b: $ACMDX \times (DST \text{ } \alpha \text{ data}) - ACMDY \times (SRC \text{ } \alpha \text{ data})$
22 to 20	ACMDX[2:0]	0h	RW	α Creation Coefficient X These bits specify α creation coefficient X used in the α creation expression determined by the ABES bit. 000b: (α creation coefficient X) = (DST α data) 001b: (α creation coefficient X) = $255 - (DST \text{ } \alpha \text{ data})$ 010b: (α creation coefficient X) = (SRC α data) 011b: (α creation coefficient X) = $255 - (SRC \text{ } \alpha \text{ data})$ 100b: (α creation coefficient X) = Fixed α value 0 (COEFX setting) 101b to 111b: Setting prohibited
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	ACMDY[2:0]	0h	RW	α Creation Coefficient Y These bits specify α creation coefficient Y used in the α creation expression determined by the ABES bit. 000b: (α creation coefficient Y) = (DST α data) 001b: (α creation coefficient Y) = 255 - (DST α data) 010b: (α creation coefficient Y) = (SRC α data) 011b: (α creation coefficient Y) = 255 - (SRC α data) 100b: (α creation coefficient Y) = Fixed α value 1 (COEFY setting) 101b to 111b: Setting prohibited
15 to 8	COEFX[7:0]	00h	RW	Fixed α Value 0 These bits specify fixed α value 0 used when the CCMDX or ACMDX bits are set to 100b. A value from 00h to FFh can be specified.
7 to 0	COEFY[7:0]	00h	RW	Fixed α Value 1 These bits specify fixed α value 1 used when the CCMDY or ACMDY bits are set to 100b. A value from 00h to FFh can be specified.

9.4.3.2.11 LIF Control Registers

(1) LIF0 Control Register (LCDC_VSPD_VI6_LIF0_CTRL)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3B00h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	OBTH [11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PADL	-	-	-	CFMT	-	-	REQSEL	LIF_EN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	RW	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 16	OBTH [11:0]	000h	RW	Buffer Threshold for Start Ready Notification to Display Module Always specify 1500, when LIF_EN is set to 1b.
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	PADL	0h	RW	Enable/Disable of padding with dummy lines to output-image to the DU. Enables or disables of padding with dummy lines to output-image to the DU. 0b: Padding with dummy lines to output image to DU is disabled. 1b: Padding with dummy lines to output image to DU is enabled.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	CFMT	0h	RW	Chroma Format This bit selects the output format from the LIF module to the display module. When RGB format is used, this bit shall be set to 0b. 0b: YCbCr444 or RGB Format 1b: YCbCr422 Format [Note] The DU cannot receive YCbCr422 format. Therefore, when CFMT is 1, YCbCr444 data which information contents is equal to YCbCr422 is output to DU as shown in Figure 9.4-30 .
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	REQSEL	0h	RW	External Display Module Selection 0b: This value is setting prohibited when 1b is set to LIF_EN 1b: DU is selected as the destination external display module.
0	LIF_EN	0h	RW	Enable/Disable of Data Output to External Display Module Enables or disables data output from the LIF to the external display module (DU). 0b: Data output to the external display module is disabled. 1b: Data output to the external display module is enabled.

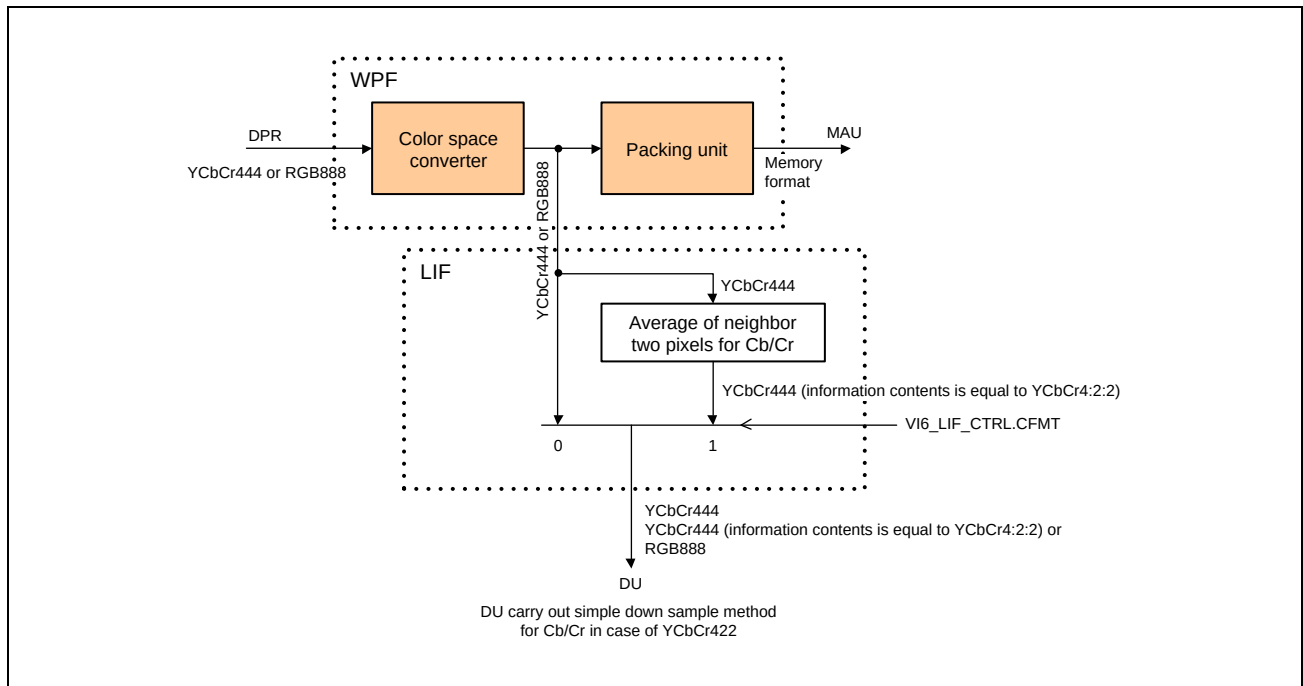


Figure 9.4-30 Data flow between LIF and DU

(2) LIF0 Clock Stop Buffer Control Register (LCDC_VI6_LIF0_CSBTH)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 3B04h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	HBTH[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	LBTH[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name			Initial Value	R/W	Description										
31 to 28	-			All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
27 to 16	HBTH[11:0]			000h	RW	Buffer Threshold for Clock Stop in Dynamic Clock Control Set HBTH[11:0] = 0 (fixed value)										
15 to 12	-			All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
11 to 0	LBTH[11:0]			000h	RW	Buffer Threshold for Clock Start in Dynamic Clock Control Set LBTH[11:0] = 0 (fixed value)										

(3) LIF0 Buffer Attribute Register (LCDC_VI6_LIF0_LBA)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 3B0Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LBA0	-	-	-	LBA1[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	LBA2[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	LBA0	0h	RW	LIF Buffer Attribute Register0 Always specify 1b, when LIF_EN is set to 1b
30 to 28	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
27 to 16	LBA1[11:0]	000h	RW	LIF Buffer Attribute Register1 Always specify 1536, when LIF_EN is set to 1b
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	LBA2[11:0]	000h	R	LIF Buffer Attribute Register2 These bits are internal status for purpose of h/w debugging.

(4) LIF0 Padding Line Cycle Register (LCDC_VI6_LIF0_PADLN_CYC)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 3B30h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PADLN_CYC[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	-	All 0	R	Reserved Whenever it is read, 0 is read. The written value will be ignored.
5 to 0	PADLN_CYC [5:0]	00h	RW	Cycle of padding pattern. These bits specify the minimum number of cycles of the padding pattern of valid lines and dummy lines. For example, if PADLN_CYC[5:0] is 4, lower 4 bits of PADLN_PATTERN[3:0] indicates padding pattern. A value from 1 to 32 can be specified.

(5) LIF0 Padding Line Pattern Register (LCDC_VI6_LIF0_PADLN_PT)

Access Size :		32 bits														
Address :		<LCDC_vspd_base> + 3B34h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PADLN_PATTERN[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PADLN_PATTERN[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	PADLN_PATTE RN[31:0]	0000_0000h	RW	Pattern of padding with dummy lines These bits specify the padding pattern of valid lines and dummy lines. In each bit, 0b indicates an active line, 1b indicates a dummy line. It starts from the lower bit (Bit 0). First line should be valid line (Always set 0 to PADLN_PATTERN[0]).

(6) LIF0 Padding Line Value Register (LCDC_VI6_LIF0_PADLN_VAL)

Access Size :		32 bits															
Address :		<LCDC_vspd_base> + 3B38h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
PADA[7:0]								PADR[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PADG[7:0]								PADB[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	PADA[7:0]							PADR[7:0]									
31 to 24	PADA[7:0]							PADR[7:0]									
23 to 16	PADA[7:0]							PADR[7:0]									
15 to 8	PADG[7:0]							PADB[7:0]									
7 to 0	PADG[7:0]							PADB[7:0]									

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	PADA[7:0]	00h	RW	Padding data of EDC code for dummy line.
23 to 16	PADR[7:0]	00h	RW	Padding data of R/Cr for dummy line.
15 to 8	PADG[7:0]	00h	RW	Padding data of G/Y for dummy line.
7 to 0	PADB[7:0]	00h	RW	Padding data of B/Cb for dummy line.

Note: Set the register to 24FF_FFFFh when the output format is YUV422.
Otherwise, set it 2EFF_FFFFh.

(7) LIF0 Padding Line Size Register (LCDC_VI6_LIF0_PADLN_SIZE)

Access Size : 32 bits

Address : <LCDC_vspd_base> + 3B3Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PADLN_VSIZE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	PADLN_VSIZE [12:0]	0000h	RW	Vertical Size of LIF output in case of enabling Padding line When VI6_LIF0_CTRL.PADL is 1b, dummy lines are included in the output data from LIF, so set the LIF output VSIZE including dummy lines to these bits. If the number of output lines does not reach the value of PADLN_VSIZE [12:0] even if all valid lines are output according to padding pattern, dummy lines are inserted until the number of output lines reaches PADLN_VSIZE [12:0]. - Output vsize of LIF is same with input vsize to LIF when VI6_LIF0_CTRL.PADL is 0b - Output vsize of LIF is PADLN_VSIZE [12:0] when VI6_LIF0_CTRL.PADL is 1b

9.4.4 DU Registers

The DU contains the registers which can directly be read and written to by the host CPU. The host can access any long word (32 bits), word (16 bits), and byte (8 bits) in the registers. Therefore, read-modify-write is needed to change partial bits not more than 8 bits in the 32-bit register.

Never access the registers when the DU is in the module standby mode (with the operation clock placed in hold status).

Table 9.4-27 DU Register Base Address

Base Register Name	Base Address
<LCDC_du_base>	0_1646_0000h (5646_0000h* ¹ , 4646_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.4.4.1 List of DU Registers

The registers are listed below.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
DU Module Control Register 0	LCDC_DU_MCR0	0000_0000h	0000h	32, 16, 8
DU Module Status Register 0	LCDC_DU_MSR0	0054_0000h	0004h	32, 16, 8
DU Module Status Register 1	LCDC_DU_MSR1	0000_0000h	0008h	32, 16, 8
DU Interrupt Mask Register 0	LCDC_DU_IMR0	0000_0000h	000Ch	32, 16, 8
DU Display I/F Timing Register 0	LCDC_DU_DITR0	0003_0300h	0010h	32, 16, 8
DU Display I/F Timing Register 1	LCDC_DU_DITR1	0000_0000h	0014h	32, 16, 8
DU Display I/F Timing Register 2	LCDC_DU_DITR2	0000_0000h	0018h	32, 16, 8
DU Display I/F Timing Register 3	LCDC_DU_DITR3	0000_0000h	001Ch	32, 16, 8
DU Display I/F Timing Register 4	LCDC_DU_DITR4	0000_0000h	0020h	32, 16, 8
Reserve	-	-	0024h to 003Fh	-
DU Module Control Register 1	LCDC_DU_MCR1	0000_0000h	0040h	32, 16, 8
Reserve	-	-	0044h to 004Bh	-
DU PBUF Control Register 0	LCDC_DU_PBCR0	0000_001Fh	004Ch	32, 16, 8
DU PBUF Control Register 1	LCDC_DU_PBCR1	0000_0000h	0050h	32, 16, 8
DU PBUF Control Register 2	LCDC_DU_PBCR2	0000_0000h	0054h	32, 16, 8

9.4.4.2 DU Register Description

The prefix (LCDC_) of the register names is omitted in this and subsequent sections.

9.4.4.2.1 DU Module Control Register 0 (LCDC_DU_MCR0)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 0000h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB_CLR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DI_EN	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	PB_CLR	0h	RW	Clear PBUF pointers. This parameter must be changed while VSPD is not running and DU_MSR0.ST_DIF_BSY is '0b'. Writing '1b' to this parameter starts clearing the PBUF pointers. After clearing the PBUF pointers is completed, DU_MSR0.ST_PB_WINIT and DU_MSR0.ST_PB_RINIT are read as 1b, this parameter must be written back to '0b'.
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	DI_EN	0h	RW	Display enable. 1b: Enabled 0b: Disabled
7 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.

9.4.4.2.2 DU Module Status Register 0 (LCDC_DU_MSR0)

Access Size : 32, 16, 8 bits

Address : <LCDC_du_base> + 0004h

Initial Value : 0054_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	ST_PB_RINIT	ST_PB_RUF	ST_PB_REMPTY	-	ST_PB_WINIT	-	ST_PB_WFULL
Initial Value	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ST_DI_BSY	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	ST_PB_RINIT	1h	R	PBUF FIFO read pointer cleared status. 1b: Cleared (initial value) 0b: Not cleared
21	ST_PB_RUF	0h	R	PBUF read underflow status. This parameter clears when PBUF pointers are cleared. 1b: Occur PBUF underflow 0b: Not occur PBUF underflow
20	ST_PB_REMPTY	1h	R	PBUF read empty status. 1b: Empty 0b: Not empty
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18	ST_PB_WINIT	1h	R	PBUF FIFO write pointer cleared status. 1b: Cleared (initial value) 0b: Not cleared
17	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	ST_PB_WFULL	0h	R	PBUF write full status. 1b: Full 0b: Not full
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	ST_DI_BSY	0h	R	Display I/F status. 1b: Busy 0b: Idle
7 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.4.4.2.3 DU Module Status Register 1 (LCDC_DU_MSR1)

Access Size : 32, 16, 8 bits

Address : <LCDC_du_base> + 0008h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	UF_VACT[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	UF_HACT[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28 to 16	UF_VACT[12:0]	0000h	R	Vactive counter when the PBUF underflow occurs. This parameter is value latched the down counter starting from VACTIVE. This parameter is cleared when DU_MSR0.ST_PB_RUF is cleared.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12 to 0	UF_HACT[12:0]	0000h	R	Hactive counter when the PBUF underflow occurs. This parameter is value latched the down counter starting from HACTIVE. This parameter is cleared when DU_MSR0.ST_PB_RUF is cleared.

9.4.4.2.4 DU Interrupt Mask Register 0 (LCDC_DU_IMR0)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 000Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IM_PB_RUF
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	IM_PB_RUF		0h		RW		Mask PBUF read underflow interrupt. 1b: Mask interrupt 0b: Not mask interrupt									

9.4.4.2.5 DU Display I/F Timing Register 0 (LCDC_DU_DITR0)

Access Size : 32, 16, 8 bits

Address : <LCDC_du_base> + 0010h

Initial Value : 0003_0300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	HSPOL	VSPOL
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DEMD[1:0]		-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17	HSPOL	1h	RW	hsync polarity. 1b: High active 0b: Low active
16	VSPOL	1h	RW	vsync polarity. 1b: High active 0b: Low active
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9, 8	DEMD[1:0]	3h	RW	de output mode. 00b: Reserved (Fixed to High) 01b: Fixed to Low 10b: Reserved (Low active) 11b: Data enable (High active)
7 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	-	0h	RW	Reserved These bits are read as 0b. The write value should always be 0b.

9.4.4.2.6 DU Display I/F Timing Register 1 (LCDC_DU_DITR1)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VACTIVE[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	VSA[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	VACTIVE[12:0]		0000h	RW	The number of lines in the Vactive period. This parameter should be set from 1 to 8190.											
15 to 12	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
11 to 0	VSA[11:0]		000h	RW	The number of lines in the Vsync period. This parameter should be set from 0 to 4095, and DU_DITR1.VSA + DU_DITR2.VBP should be set more than 0. If set to 0, vsync is not asserted to DU_DITR0.VSPOL.											

Figure 9.4-31 shows the relationship between the parameters and the video output timings.

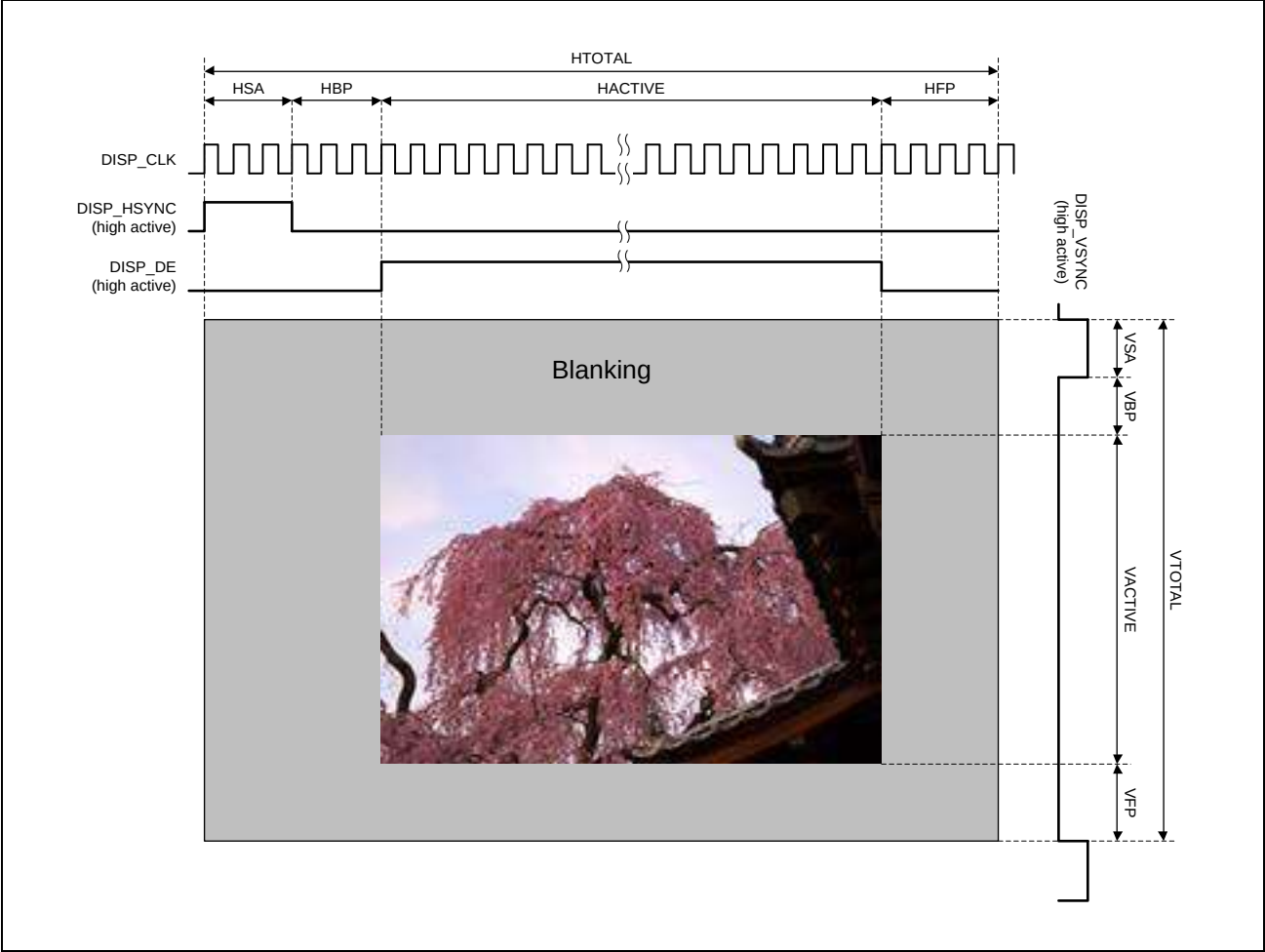


Figure 9.4-31 Video Output Timings

9.4.4.2.7 DU Display I/F Timing Register 2 (LCDC_DU_DITR2)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 0018h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VFP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VBP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	VFP[12:0]		0000h	RW	The number of lines in the Vfront period. This parameter should be set from 0 to 8191.											
15 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	VBP[12:0]		0000h	RW	The number of lines in the Vback period. This parameter should be set from 0 to 8191, and DU_DITR1.VSA + DU_DITR2.VBP should be set more than 0.											

9.4.4.2.8 DU Display I/F Timing Register 3 (LCDC_DU_DITR3)

Access Size :		32, 16, 8 bits															
Address :		<LCDC_du_base> + 001Ch															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	HACTIVE[12:0]													
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	HSA[11:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	Bit Name		Initial Value	R/W	Description												
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												
28 to 16	HACTIVE[12:0]		0000h	RW	The number of cycles (pixels) in the Hactive period. This parameter should be set from 1 to 8190.												
15 to 12	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												
11 to 0	HSA[11:0]		000h	RW	The number of cycles in the Hsync period. This parameter should be set from 0 to 4095, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be set 3 or more. If set to 0, hsync is not asserted to DU_DITR0.HSPOL.												

9.4.4.2.9 DU Display I/F Timing Register 4 (LCDC_DU_DITR4)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	HFP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	HBP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	HFP[12:0]		0000h	RW	The number of cycles in the Hfront period. This parameter should be set from 1 to 8191, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be set 3 or more.											
15 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	HBP[12:0]		0000h	RW	The number of cycles in the Hback period. This parameter should be set from 0 to 8191, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be set 3 or more.											

9.4.4.2.10 DU Module Control Register 1 (LCDC_DU_MCR1)

Access Size : 32, 16, 8 bits

Address : <LCDC_du_base> + 0040h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB_AUT OCLR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	OPMD[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	PB_AUTOCLR	0h	RW	PBUF pointers auto clear enable. Specify timing to clear the PBUF pointers. 0b: Not clear until DU_MCR0.PB_CLR is asserted 1b: Clear during blanking (according to DU_MCR1.OPMD)
15 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	-	0h	RW	Reserved Whenever it is read, 0b is read. The written value should always be 0b.
7 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	OPMD[1:0]	0h	RW	Operation mode. Specify behavior of LIFC and PBUF during Vfront and Hfront period. This parameter must be changed while VSPD is not running. 00b: After frame end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to '1b'. 01b: After line end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to '1'. 10b: Reserved 11b: LIFC always request to VSPD. PBUF pointers are not cleared until DU_MCR0.PB_CLR is set to '1b'.

9.4.4.2.11 DU PBUF Control Register 0 (LCDC_DU_PBCR0)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 004Ch														
Initial Value :		0000_001Fh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PB_DEP[4:0]				
Initial Value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 5	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
4 to 0	PB_DEP[4:0]		1Fh		RW		Valid PBUF depth. This parameter should be set 1Fh. This parameter should be set ((2 ⁿ) - 1).									

9.4.4.2.12 DU PBUF Control Register 1 (LCDC_DU_PBCR1)

Access Size :		32, 16, 8 bits														
Address :		<LCDC_du_base> + 0050h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB_RU FOP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	PB_RUFOP		0h		RW	Specify behavior when the PBUF underflow occurs. 0b: Continue outputting invalid data 1b: Output DU_PBCR2.PB_RUFDAT										

9.4.4.2.13 DU PBUF Control Register 2 (LCDC_DU_PBCR2)

Access Size :		32, 16, 8 bits																														
Address :		<LCDC_du_base> + 0054h																														
Initial Value :		0000_0000h																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
	-	-	-	-	-	-	-	-	PB_RUFDAT[23:16]																							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
	PB_RUFDAT[15:0]																															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW																
Bit	Bit Name																															
31 to 24	-																															
	Initial Value																															
	All 0																															
	R/W																															
	R																															
Description																																
Reserved																																
Whenever it is read, 0b is read. The written value will be ignored.																																
23 to 0	PB_RUFDAT [23:0]	00_0000h																														
		RW																														
Read data when the PBUF underflow occurs.																																

9.4.5 Operation

9.4.5.1 Operation Control Setting

(a) To start image process

FCPVD registers do not need to be set when starting an image process. Therefore, refer **9.4.5.4 Concept of VSPD Operation Starting and Stopping** for the guidance of starting an image process.

9.4.5.2 Reset Operation

This unit can be reset by hardware reset. The hardware reset is controlled by LCDC_0_RESET_N of CPG, and this reset initializes the whole this unit immediately.

9.4.5.3 Registers to set fixed value

Set fixed value to following registers in any case.

- [1] Always set 0000 0808h to VI6_CLK_DCSWT
- [2] Always set 256 to VI6_DL_CTRL.AR_WAIT [7:0] in case of using display list.
- [3] Always set 2 to VI6_DL_EXT_CTRLn.POLINT [5:0] in case of using extended display list of WPFn.
- [4] Always set 1 to VI6_DL_EXT_CTRLn.DLPRI in case of using extended display list of WPFn.
- [5] Always set 0 to VI6_DL_EXT_CTRLn.EXPRI in case of using extended display list of WPFn.
- [6] Always set 0000 0500h to VI6_DPR_WPFn_FPORCH (n = 0, 1) in case of using WPFn.
- [7] Always set 1500 to VI6_LIF0_CTRL.OBTH [11:0] in case of using LIF (VI6_LIF0_CTRL.LIF_EN = 1).
- [8] Always set 8600 0000h to VI6_LIF0_LBA in case of using LIF0 (VI6_LIF0_CTRL.LIF_EN = 1).
- [9] Always set 1 to VI6_LIFn_CTRL.REQSEL in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).
- [10] Always set 0 to VI6_LIFn_CSBTH in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).
- [11] Always set 1Fh to DU_PBCR0 in case of using DU (DU_MCR0.DI_EN = 1).

9.4.5.4 Concept of VSPD Operation Starting and Stopping

The VSPD provides one channel of image processing. Each channel is started by setting the corresponding start register. Here, starting a processing channel means starting one of WPF0, which are output modules of the VSPD. Use the start registers shown in **Table 9.4-28** to start WPF modules.

After a WPF module is started and specified processing is completed, the corresponding channel stops operation and notifies the end of processing through an end interrupt. End interrupts are generated through the end interrupt source registers shown in **Table 9.4-28**; clearing a source register cancels the corresponding interrupt signal. Each of the operating status registers shown in the table indicates the busy state after the corresponding channel is started through the start register until processing is completed and operation stops. **Figure 9.4-33** shows these operation timings.

Table 9.4-28 Target Module and Corresponding Registers for Starting and Stopping Operation

Target Module	Start Register	End Interrupt Source Register	Operating Status Register
WPF0	VI6_CMD0.STRCMD	VI6_WPF0_IRQ_STA.FRE	VI6_STATUS.SYS0_ACT

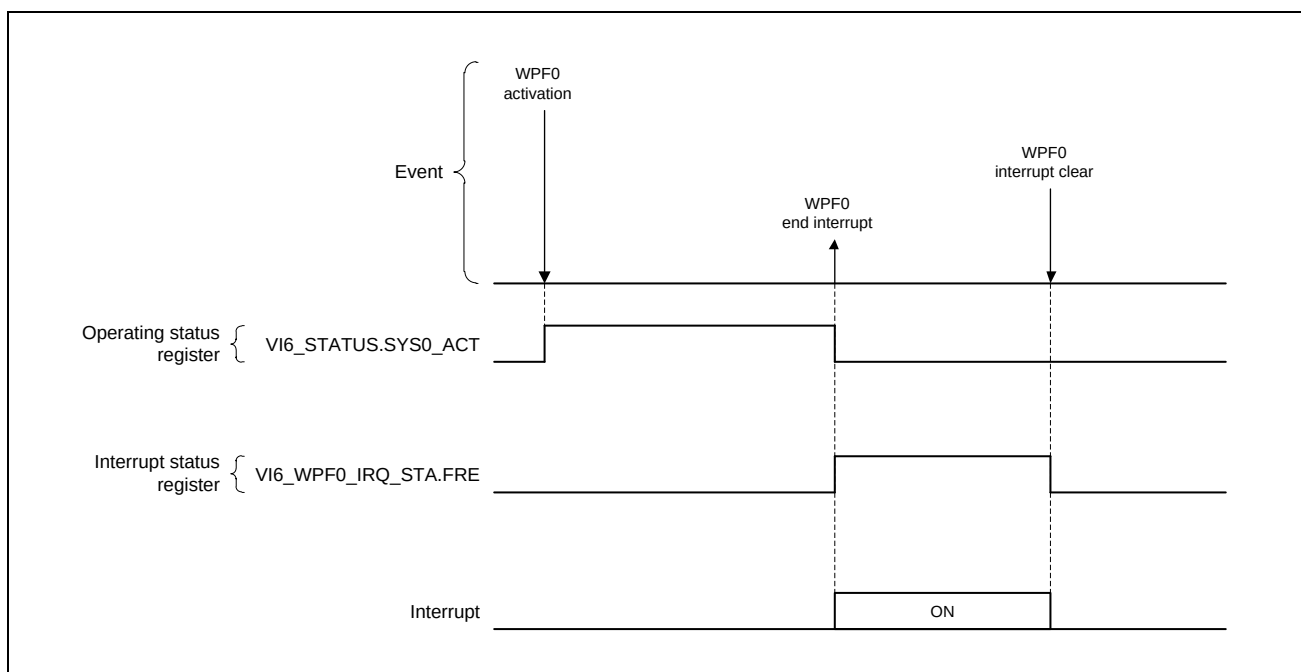


Figure 9.4-32 VSPD Startup and Status of Each Register and Interrupt

The following describes the operating states (operating or stopped) of VSPD internal modules. As described in **9.4.3.2 VSPD Register Description**, the VSPD has several image processing modules and the connections between modules are determined by the DPR. Accordingly, the operating state of a module is the same as that of the target WPF for that module. For example, when the target WPF for the BRS is WPF0, the BRS operating state is the same as that of WPF0; that is, the BRS operating state is indicated by the VI6_STATUS.SYS0_ACT as shown in **Table 9.4-28** and the status change timing is shown as VI6_STATUS.SYS0_ACT in **Figure 9.4-32**. Likewise, the operating states of all modules connected to WPF0 is indicated by VI6_STATUS.SYS0_ACT.

Connections should be changed through the DPR-related registers (described later) while all modules to be affected by any change in connections are stopped. If connections through the DPR are changed during operation, the VSPD will hang.

Figure 9.4-33 shows relation between start reservation and operating status. When start is reserved while VSPD is operating, the reservation is accepted after VSPD status is moved from “operating” into “idle” as shown in **Figure 9.4-33**.

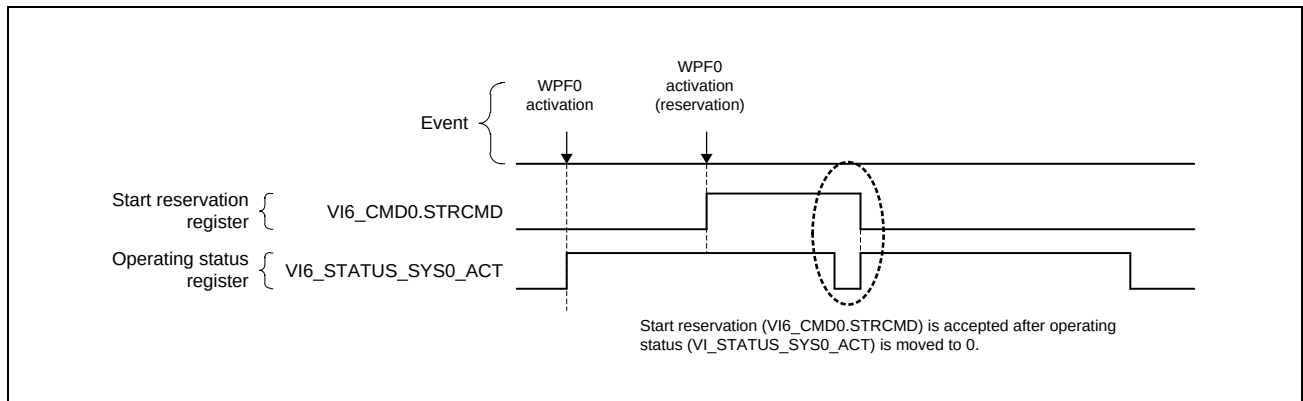


Figure 9.4-33 VSPD Start Reservation and Operating Status

9.4.5.5 Display List

9.4.5.5.1 Functional description

The VSPD provides the display list function. As a display list, the VSPD automatically downloads the register settings except for the control registers (**9.4.3.2.4 General Control Registers** and **9.4.3.2.5 Display List Control Registers**) from external memory and stores the settings in the VSPD registers. This function is advantageous in that the interrupt processing or register setting modification processing can be executed without CPU intervention during multiple-frame processing because the register settings used for VSPD processing are prepared in advance in external memory such as SDRAM.

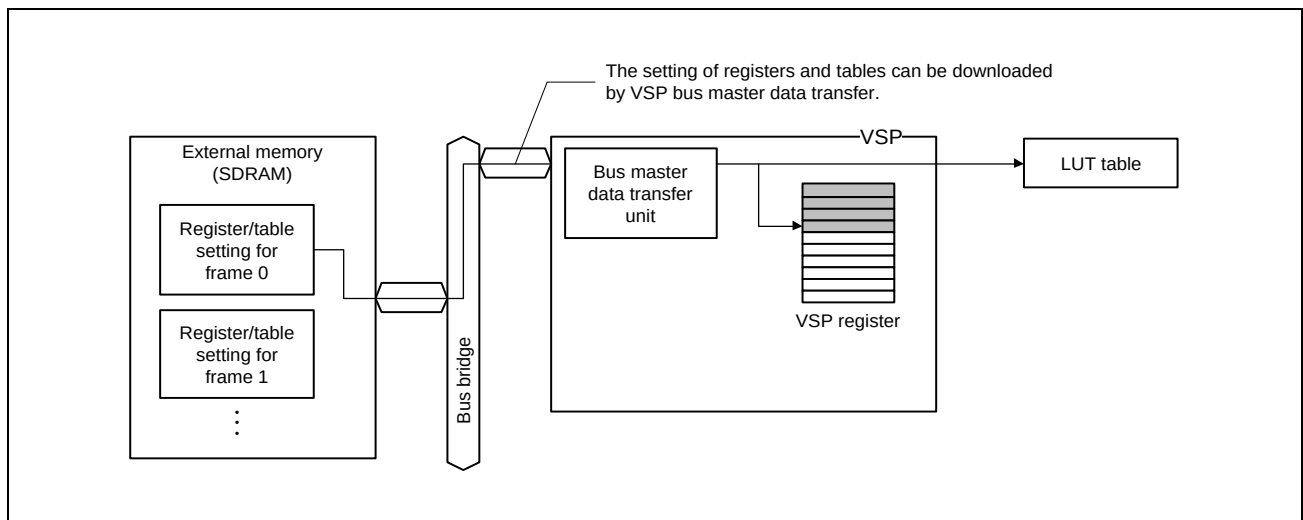


Figure 9.4-34 Concept of Display List

To use display lists, specify the external memory addresses to the display list control registers described in **9.4.3.2.5 Display List Control Registers**. The register settings or various types of information should be stored in external memory in the format described in **9.4.5.5.2 Normal display list mode**. **Figure 9.4-35** shows the difference between VSPD operation through normal register settings and through display lists.

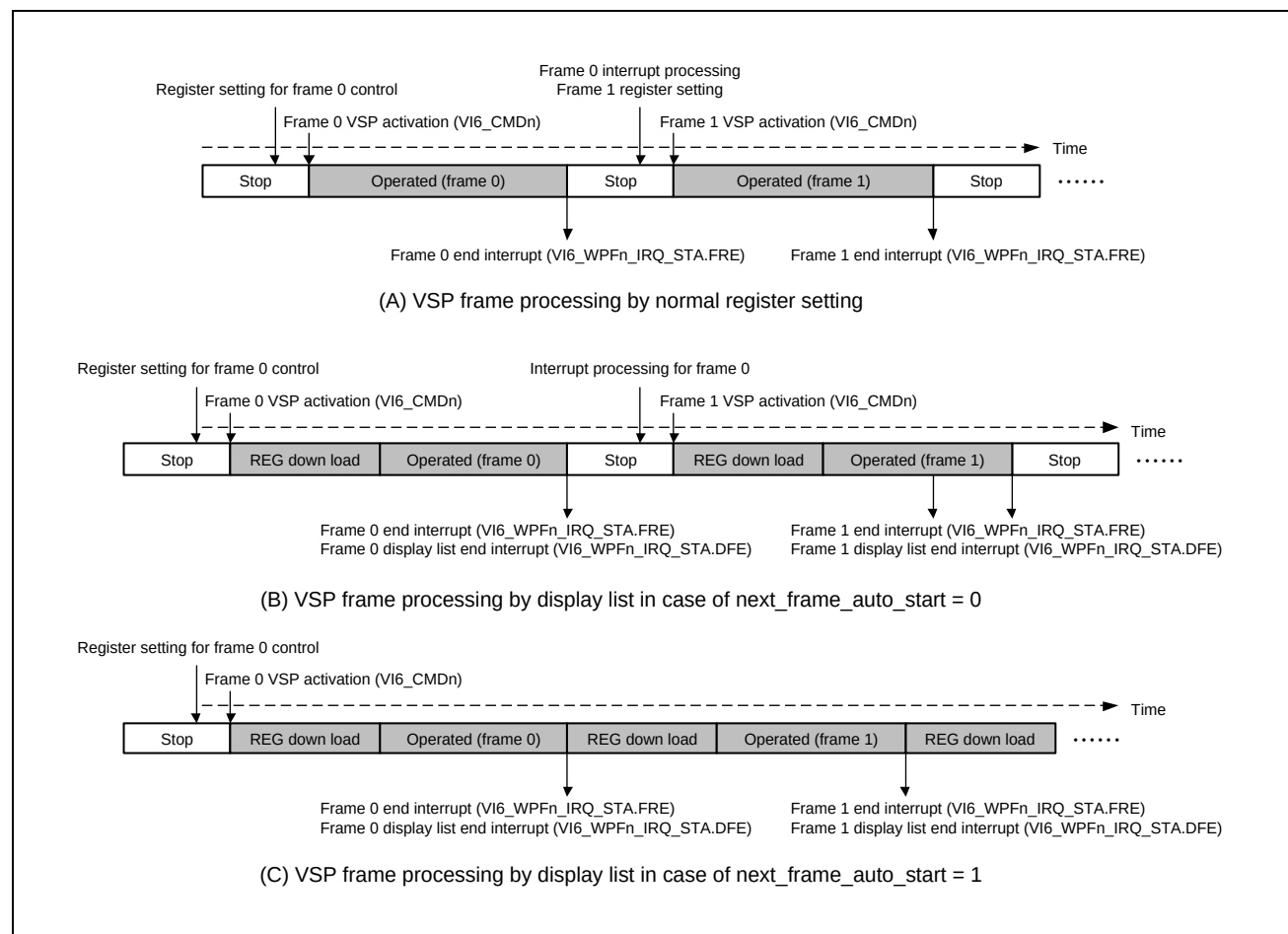


Figure 9.4-35 Comparison of VSPD Operation between Normal Register Settings and Display Lists

As shown in **Figure 9.4-35 (A)**, in the VSPD processing through normal register settings, all registers should be set up before the VSPD is started for each frame. After the VSPD processing is completed, the VSPD outputs a frame end interrupt (VI6_WPFn_IRQ_STA.FRE). This method requires a certain amount of time for register settings or interrupt processing by the CPU between frames. In contrast, when display lists are used, the VSPD automatically downloads register settings from external memory as shown in **Figure 9.4-35 (B)** and **(C)**, which reduces the load on the CPU between frames.

Figure 9.4-35 (B) shows the display list usage where the VSPD stops at the end of every frame; only the VSPD start processing for each frame is done by the CPU. This is suitable for the cases when the CPU controls synchronization of frame processing in frame buffer management or when the amount of register values or table data to be set in the VSPD is large. In the case shown in **Figure 9.4-35 (C)**, as soon as the frame processing ends, the VSPD automatically begins next frame operation and starts downloading new register settings. This is the fastest operation using display lists.

Table 9.4-29 shows the modes of the display list and the supported functions for each mode. The detail of each mode is described in the following sections.

Table 9.4-29 Display list mode and supported functions

Mode	Extended Display List	Continuous Frames
Normal Display List Mode	Supported	Controlled by “next frame auto start” in the header of the display list.
Header-less Display List Mode	Not Supported	Controlled by VI6_DL_CTRL.CFM0 bit.

9.4.5.5.2 Normal display list mode

The VSPD display lists include control information as well as simple register settings in order to control multiple-frame processing in an optimum way for each application. **Figure 9.4-36** shows the display list structure.

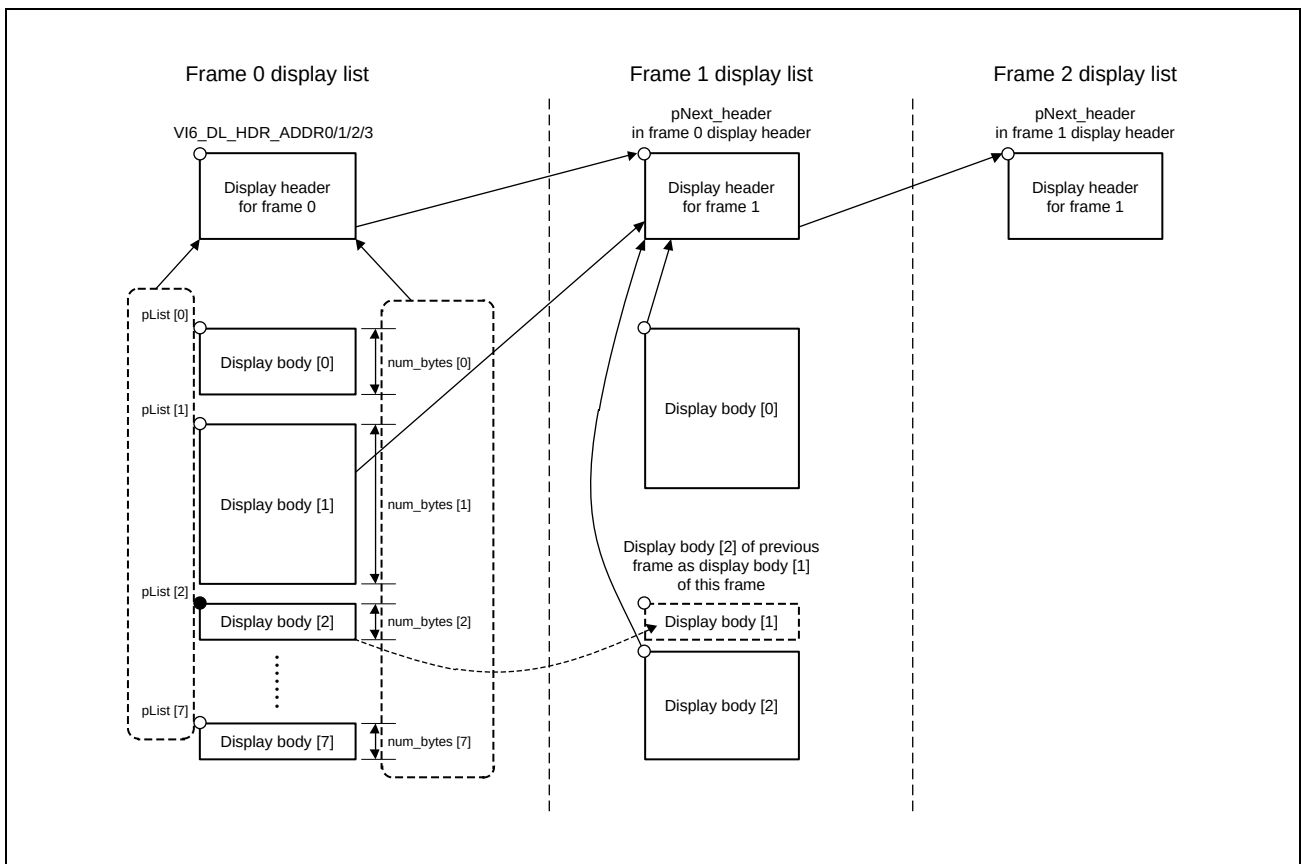


Figure 9.4-36 Structure and Concept of VSPD Display List

A VSPD display list consists of two sections; a header section for storing various information and control flags and a body section for storing register and table settings. A combination of these two sections is defined as a display list for a frame. The register and table settings can be divided and stored in up to eight separate bodies allocated in memory. Therefore, the separate bodies storing the register settings for one frame (for example, frame 0) can have non-sequential start addresses; that is, the bodies for one frame can be allocated to areas distant from each other in memory. To gather these bodies and configure the register settings for one frame, a header is used. A display header stores the number of bodies linked with the header and the start address and size of each body.

The VSPD analyzes the header to gather register and table settings stored in separate memory areas and reconfigures the complete register settings.

The addresses of display headers should be specified in the VSPD registers described in **9.4.3.2.5 Display List Control Registers**. When activated in a mode that uses display lists, the VSPD downloads display headers from the

addresses specified in VI6_DL_HDR_ADDR0 (number 0 correspond to the WPF channel index numbers), analyzes the numbers of bodies and the address and data size of each body, downloads the bodies, and completes register and table settings. After display list downloading is completed, the VSPD becomes ready for frame image processing; the VSPD then starts the actual frame processing.

After processing of a frame ends, the VSPD proceeds to the next frame processing. Here, there are two modes for starting the next frame processing as shown in **Figure 9.4-35**. In one mode, the VSPD stops operation and waits for the next activation by the CPU in the same way as when display lists are not used. In this mode, the address used for downloading the display header of the next frame is kept by the internal hardware. Therefore, if valid address information is not stored in the display header for the previous frame, a correct value should be specified in VI6_DL_HDR_ADDR0 while the VSPD is stopped. When the VSPD is started after a correct value is specified, the VSPD starts next frame processing with the same procedure for the previous frame. In contrast to this mode, which stops the VSPD after the end of one-frame processing, there is another mode for automatically starting next frame processing. In automatic start mode, the VSPD downloads the next display header as soon as the previous frame processing ends. After downloading ends, the VSPD starts image processing. The information regarding mode selection, that is, whether to automatically start next frame processing, should be stored in the display header downloaded for the previous frame.

To strictly define the display list format described above, the following shows the grammatical structure of a display list using pseudo-code. First, to simplify the description of the display list format in the following pages, **Table 9.4-30** defines a function. Function zero bits (num bits) generates a string of one-bit 0s for the number of bits specified by the parameter for the function. By using this function, **Table 9.4-31** defines the header section format of a display list and **Table 9.4-32** defines the body section format and **Table 9.4-33** defines the extended display list body section format.

Data order of Display List Header Section, Display List Body Section and Extended Display List Body Section are assumed as the data is stored in SDRAM by big endian. **Table 9.4-34** shows an example of data order. If data order is not same as the definition, data order within 8byte unit can be adjusted by setting VI6_DL_SWAP (see **(3) Display List-0 Data Swapping Register (LCDC_VSPD_VI6_DL_SWAP0)**).

Table 9.4-30 Definition of a Function for Simple Description

Syntax	Bit Count
zero_bits (num_bits)	
{	
for (i = 0; i<num_bits; i++) {	
zero_bit	1
}	
}	
Bit String	Contents
zero_bit	zero_bit indicates a 1-bit integer having a value of 0.

Table 9.4-31 Format of Display List Header Section

Syntax	Bit Count
display_header () /* Fixed length */	
{	
zero_bits (29)	
num_list_minus1	3
for (i=0; i<8; i++)	
zero_bits (15)	
num_bytes [i]	17
pList [i]	32
}	
pNext_header	32
zero_bits (23)	
wait_wup	1
zero_bits (3)	
ignore_upd_dl	1
zero_bits (2)	
current_frame_int_enable	1
next_frame_auto_start /* 76 bytes from the beginning of this header*/	1
if (VI6_DL_EXT_CTRL.EXT) {	
zero_bits (32) /* padding zero 4 bytes for alignment */	
zero_bits (6)	
pre_ext_dl_exec	1
post_ext_dl_exec	1
zero_bits (8)	
pre_ext_dl_num_cmd	16
pre_ext_dl_pList	32
zero_bits (16)	
post_ext_dl_num_cmd	16
post_ext_dl_pList /* 96 bytes from the beginning of this header*/	32
}	
}	

Bit String	Contents
num_list_minus1	Specifies the value obtained by subtracting 1 from the total number of display list bodies linked with the display header. For example, when this bit field is set to 0, this display list uses one body.
num_bytes [i]	Specifies the number of bytes in the i-th display list body (indicated by index i). Be sure to specify a multiple of eight bytes. For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), specify 0.
pList [i]	Specifies the start address of the i-th display list body (indicated by index i). Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0). For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), specify 0.
pNext_header	Specifies the address of the display list header for the next frame. After display list downloading ends, the VSPD keeps its value in the internal memory and uses it in the next display list header downloading. Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0).
wait_wup	When this bit is 1, VSPD starts reading image data from external memory after WUP (Wake Up) signal from any channel of VIN asserts. This bit is available for WPF0.
ignore_upd_dl	If this bit is set to 1, new display list pointed by VI6_DL_HDR_ADDR0 is not downloaded in VSPD H/W for the next frame, even if S/W sets VI6_DL_HDR_ADDR0 during the frame. New display list is downloaded in VSPD H/W at next frame of the processing which this bit is 0. This bit is available for WPF0.
current_frame_int_enable	This is a flag that indicates whether to set the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) to 1 when the current frame processing ends. If this flag is set to 0, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is not set to 1 when one-frame processing by this display header ends. In this state, even if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), no interrupt will be generated. If this flag is set to 1, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is set to 1 when one-frame processing by this display header ends. In this state, if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), the VSPD generates an interrupt.
Next_frame_auto_start	Enables or disables automatic start of next frame processing when one-frame processing by this display header ends. If this bit is set to 1, the VSPD starts next frame processing as soon as one-frame processing by this display header ends, and starts downloading the next frame display header from the pNext_header address specified in this display header. If this bit is set to 0, the VSPD stops operation when one-frame processing by this display header ends. In this case, start the VSPD through VI6_CMDn to process the next frame.
pre_ext_dl_exec	Enables execution of the extended display list for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. If this bit is set to 1, the VSPD executes the extended display list for frame preprocessing. The VSPD does not execute it if this bit is set to 0. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.
post_ext_dl_exec	Set 0 to this bit.
pre_ext_dl_num_cmd	Specifies the number of commands in the extended display list body section for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. The number of commands that can be specified is 1, and a command is 16 bytes. When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; specify 0 in this bit. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.
pre_ext_dl_pList	Specifies the start address of the area where the extended display list body section for frame preprocessing is stored when VI6_DL_EXT_CTRL.EXT is 1. Be sure to specify an address aligned with a 16-byte boundary (lower-order four bits are 0). When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; specify 0 in this bit. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0.
post_ext_dl_num_cmd	Set 0 to this bit.
post_ext_dl_pList	Set 0 to this bit.

Table 9.4-32 Format of Display List Body Section

Syntax	Bit Count
display_list (num_bytes) /* Variable length (num_bytes) */	
{	
for (i=0; i<num_bytes; i+=8) {	
set_address	32
set_data	32
}	
}	
Bit String	Contents
set_address	Specifies the address where the value specified by set_data is to be stored. Specify a register address. <i>Note:</i> Register space of VSPD is 32 Kbyte (0000h - 7FFCh). So, set 0 to upper 17-bit of set_address in Display List Body Section.
set_data	Specifies the value to be stored in the address specified by set_address. Specify a value to be set in a register.

Table 9.4-33 Format of Extended Display List Body Section

Syntax	Bit Count
ext_dl_display_list (num_llw) /* Variable length (pre/post_ext_dl_num_bytes) */	
{	
for (i=0; i<num_llw; i+=2) {	
ext_dl_cmd	64
ext_dl_data	64
}	
}	

Table 9.4-34 Data order of display list body

Address	Syntax
pList	set_address [0] (bit31-24)
pList + 1	set_address [0] (bit23-16)
pList + 2	set_address [0] (bit15-8)
pList + 3	set_address [0] (bit7-0)
pList + 4	set_data [0] (bit31-24)
pList + 5	set_data [0] (bit23-16)
pList + 6	set_data [0] (bit15-8)
pList + 7	set_data [0] (bit7-0)
pList + 8	set_address [1] (bit31-24)
pList + 9	set_address [1] (bit23-16)
pList + 10	set_address [1] (bit15-8)
pList + 11	set_address [1] (bit7-0)
pList + 12	set_data [1] (bit31-24)
pList + 13	set_data [1] (bit23-16)
pList + 14	set_data [1] (bit15-8)
pList + 15	set_data [1] (bit7-0)
~~	~~~

9.4.5.5.3 Header-less display list mode

The header-less display list does not have the display header listed in **Table 9.4-31**, and it has the simplest structure that has only single body. The extended display list function is not available in case of the header-less display list mode. Set the value 1 to VI6_DL_CTRL.NH0 (see **(1) Display List Control Register (LCDC_VSPD_VI6_DL_CTRL)**) to use the header-less display list. The start address downloaded by the header-less display list should be set to VI6_DL_HDR_ADDR0. And the size of the display body which is originally defined in the display header should be set to VI6_DL_BODY_SIZE0. The header-less display list is available only in WPF0.

9.4.5.5.4 Restrictions on display list usage

Access to the general control registers and display list control registers through a display list is prohibited. When using display lists, be sure to observe the following restrictions on register access by the CPU.

1. Do not execute write access to the same register (same address) from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed.
2. Do not execute write access to the same LUT lookup table from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed. Here, the same lookup table means the address space having the same space name shown in **Table 9.4-35**.
3. When read access by the CPU and write access through a display list to the same register (same address) occur at the same time, the read value returned to the CPU is not guaranteed.
4. When read access by the CPU and write access through a display list to the same LUT lookup table occur at the same time, the read value returned to the CPU is not guaranteed. Here, the same lookup table means the address space having the same space name shown in **Table 9.4-35**.
5. For other restrictions on values and timing of register setting through display lists, refer to the restrictions on normal register settings described in **9.4.3.2.3 Restrictions on Access to Registers and Lookup Tables**.
6. Manipulation and setting of the registers described in **9.4.3.2.4 General Control Registers** and **9.4.3.2.5 Display List Control Registers** through a display list is prohibited.
7. Do not use extended display lists.

9.4.5.6 Interrupt Processing

9.4.5.6.1 vs1o_intreq_n

When a WPF module generates an internal source that should be notified, an interrupt signal is output. As internal sources are generated in WPF independently, the VSPD has the following registers to control interrupts in WPF.

- WPFn Interrupt Enable Registers (VI6_WPFn_IRQ_ENB: n = 0) (refer to **(12) WPF0 Interrupt Enable Register (LCDC_VSPD_VI6_WPF0_IRQ_ENB)**)
- WPFn Interrupt Status Registers (VI6_WPFn_IRQ_STA: n = 0) (refer to **(13) WPF0 Interrupt Status Register (LCDC_VSPD_VI6_WPF0_IRQ_STA)**)

CAUTION

To use interrupts, enable them through WPF interrupt enable registers. If an interrupt source register has already been set to 1 for some reason, an unintended interrupt will occur as soon as the corresponding interrupt enable register is set as enabled. To avoid this, before enabling interrupts through WPF interrupt enable registers, be sure to clear all WPF interrupt sources to be enabled to 0. Be careful about this procedure when setting up registers before starting the VSPD.

9.4.5.6.2 duo_intr_n

When the PBUF will detect a FIFO underflow, an interrupt signal is output. The DU has the following registers to control the interrupt.

- DU Module Status Register 0 (DU_MSR0) (refer to **9.4.4.2.2 DU Module Status Register 0 (LCDC_DU_MSR0)**)
- DU Interrupt Mask Register 0 (DU_IMR0) (refer to **9.4.4.2.4 DU Interrupt Mask Register 0 (LCDC_DU_IMR0)**)

CAUTION

To use interrupts, enable them through DU Interrupt Mask Register. If an interrupt is asserted, the PBUF is underflowing and invalid display data is being output. When an interrupt occurs, the interrupt can only be cleared by DU_MCR0.PB_CLR register.

If DU_MCR1.PB_AUTOCLR is set a value other than 0, an interrupt is cleared according to the DU_MCR1.OPMD register.

9.4.5.7 Lookup Table Settings

9.4.5.7.1 LUT

For a single entry to the LUT space (see **Table 9.4-35**) of the VSPD, the LUT data is set by a write access in the format shown in **Figure 9.4-37**.

The entry address of each space is (start address of the space) + (entry number counting from the base point 0 × 4). For example, the address of entry 7 to the LUT space is (offset address) + 7000h + 7 × 4 = (offset address) + 701Ch.

Table 9.4-35 shows the spaces for which entries can be made. Note that if the module that references that space is operating, write accesses to the relevant space are prohibited. For example, while RPF2 is operating, accesses to the entire space of VI6_CLUT2_TBL are prohibited. When a read access is made to the relevant space during operation of the referencing module, undefined values will be read out.

The operating/stopped state of each module in **Table 9.4-35** is the operating state of the WPF to which each module is connected. Determine whether the module is operating or stopped using each WPF operating status bit in the VI6_STATUS register.

Table 9.4-35 LUT Space Addresses

Space Name	Space Addresses	Entry Count	Module that References the Space in the Left Column
VI6_LUT_TBL	(offset address) + 7000h to (offset address) + 73FCh	256	LUT

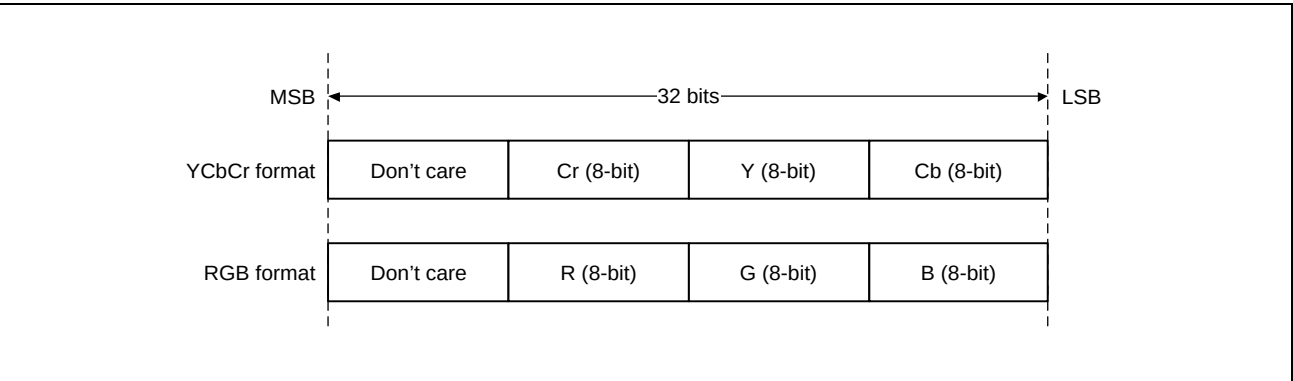


Figure 9.4-37 VI6_LUT_TBL Formats

9.4.5.8 Linked with DU

9.4.5.8.1 Operation flow of VSPD and DU

This section shows operation flow of VSPD by using Normal Display List Mode. For details of Normal Display List Mode, refer to **9.4.5.5.2 Normal display list mode**.

Figure 9.4-38 and **Figure 9.4-39** shows a procedure (register setting flow) to display image data to display panel.

Figure 9.4-38 shows setting flow to start VSPD/LIF0 linked with DU. **Figure 9.4-39** shows setting flow to update display image.

Using display list with enabling next_frame_auto_start shown in **9.4.5.5 Display List** is required to take the flow from **Figure 9.4-38** and **Figure 9.4-39**.

NOTE

If display sync operation ([1-7] step in **Figure 9.4-38**) is set before activating the VSPD, invalid data is displayed until image data read from external memory is ready to be transferred to DU. In this case, image data from external memory is displayed late by one vsync or more.

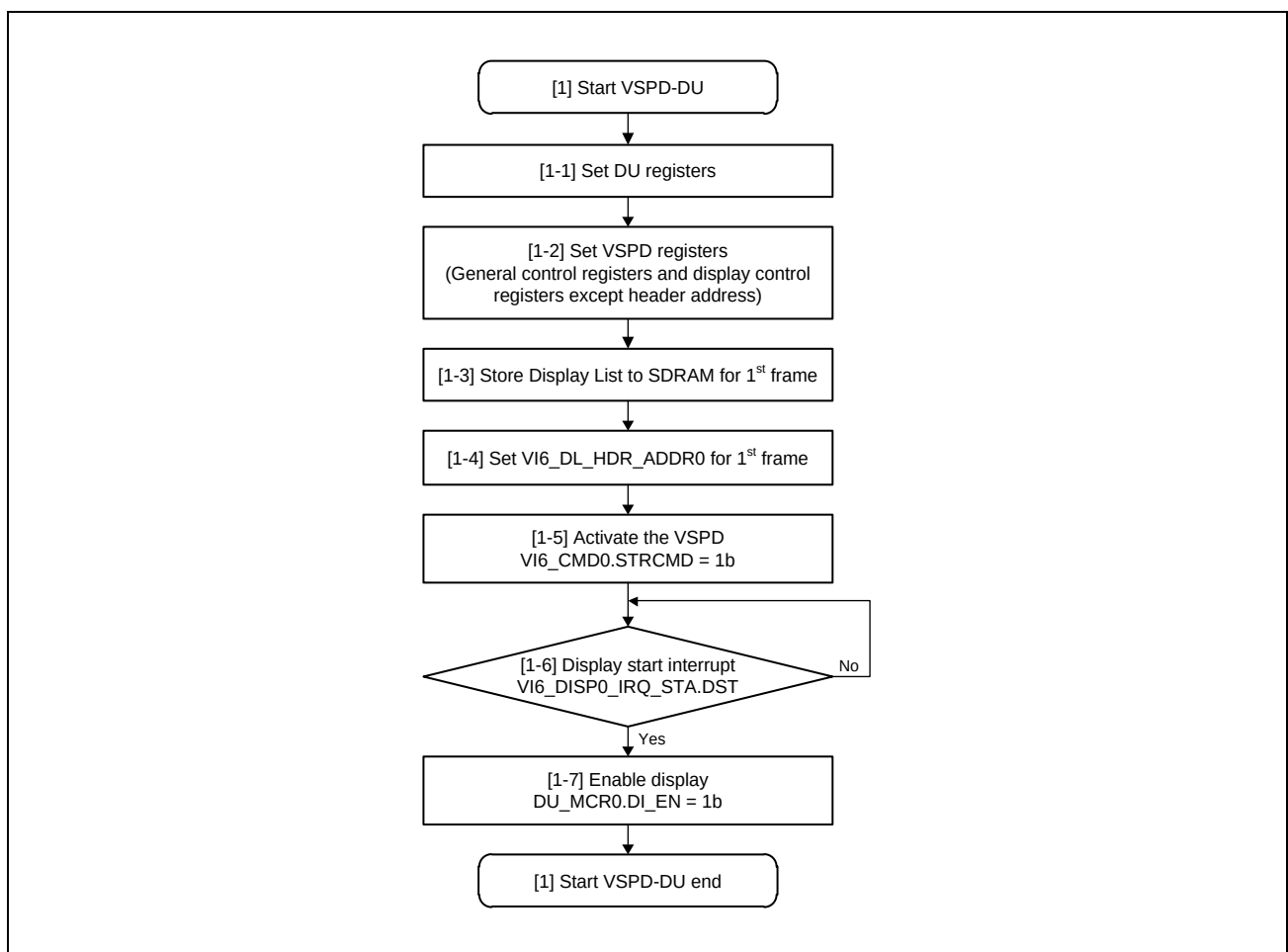


Figure 9.4-38 Setting Flow to Start VSPD Linked with DU

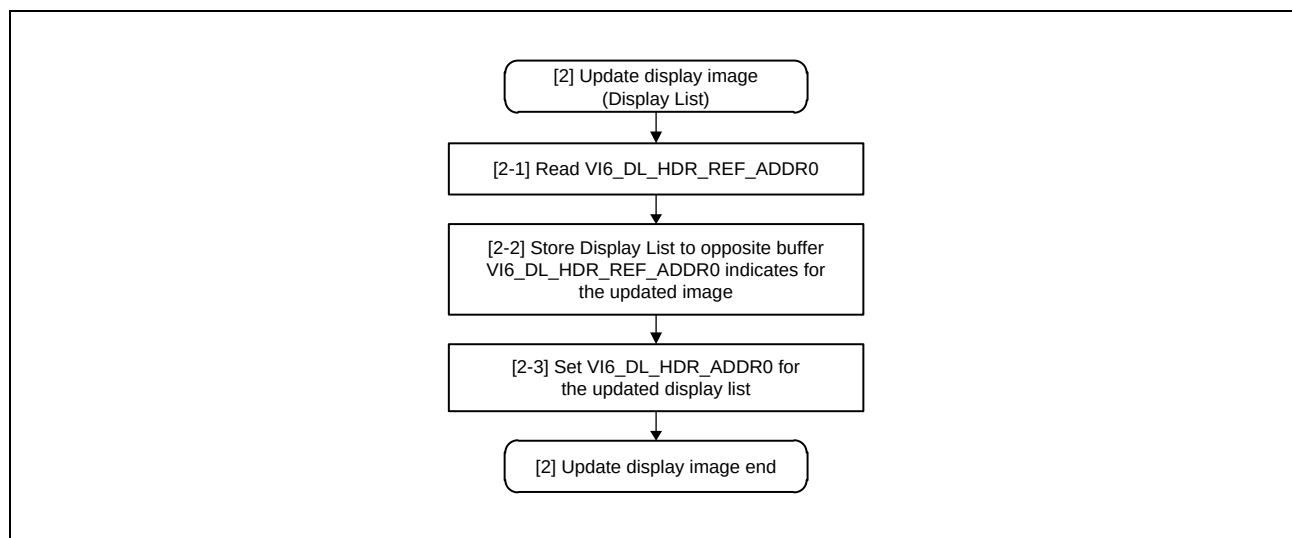


Figure 9.4-39 Setting flow to update display image

It is possible to update display list at any time (arbitrary timing) as shown in **Figure 9.4-39**. Don't overwrite display list in external memory pointed by VI6_DL_HDR_REF_ADDR. It is because VSPD is reading the display list pointed by VI6_DL_HDR_REF_ADDR or will be reading the display list soon.

Detail explanation is shown below.

As shown in **(6) Display List-0 Header Reference Address Register**

(LCDC_VSPD_VI6_DL_HDR_REF_ADDR0), when VSPD is reading display list from external memory (Period [A] in **Figure 9.4-40**), VI6_DL_HDR_REF_ADDRn indicates header address of the display list referred by VSPD. When VSPD is not reading display list from external memory (Period [B] in **Figure 9.4-40**),

VI6_DL_HDR_REF_ADDRn indicates the value of VI6_DL_HDR_ADDRn, and there is possibility that the display list is being read soon by VSPD at next frame start timing. Therefore, keep display list in external memory pointed by VI6_DL_HDR_REF_ADDR. Detail software sequence to update display list and VSPD -H/W behavior are shown below.

[1] Store latest display list in external memory area different from the area pointed by VI6_DL_HDR_REF_ADDR, because the area pointed by VI6_DL_HDR_REF_ADDR is being read or is being read soon by VSPD.

[2] Set header address of new display list into VI6_DL_HDR_ADDRn.

[3] VSPD reads display list from the external memory area pointed by VI6_DL_HDR_ADDRn at frame start timing.

Refer also to **(2) Display List-0 Header Address Register (LCDC_VSPD_VI6_DL_HDR_ADDR0)** and **(6) Display List-0 Header Reference Address Register (LCDC_VSPD_VI6_DL_HDR_REF_ADDR0)**.

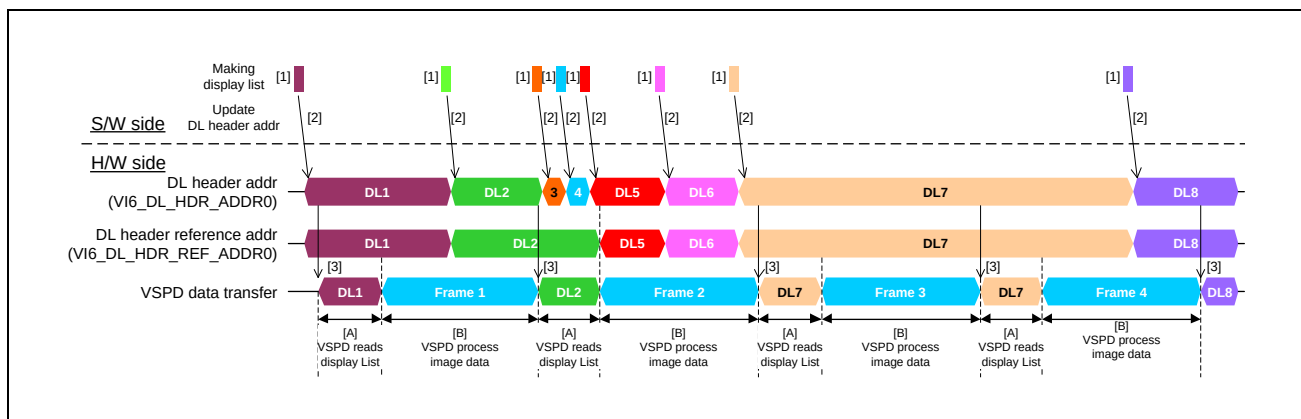


Figure 9.4-40 Updating display list at arbitrary timing

9.4.5.8.2 Controlling two register planes using display lists

This section shows operation flow of VSPD by using Header-less Display List Mode. Start procedure and stop procedure are same with Normal Display List Mode in **9.4.5.8.1 Operation flow of VSPD and DU** except for the point that VI6_DL_BODY_SIZE0 should be set before step [1-5] in start procedure shown in **Figure 9.4-38**. The procedure to update display list is different with Normal Display List Mode, and is described later. **Figure 9.4-41** shows the control of two register planes using header-less display lists (see **9.4.5.5.3 Header-less display list mode**) and its timing. In the description hereafter, the use of header-less display lists is always assumed and they are simply called display lists.

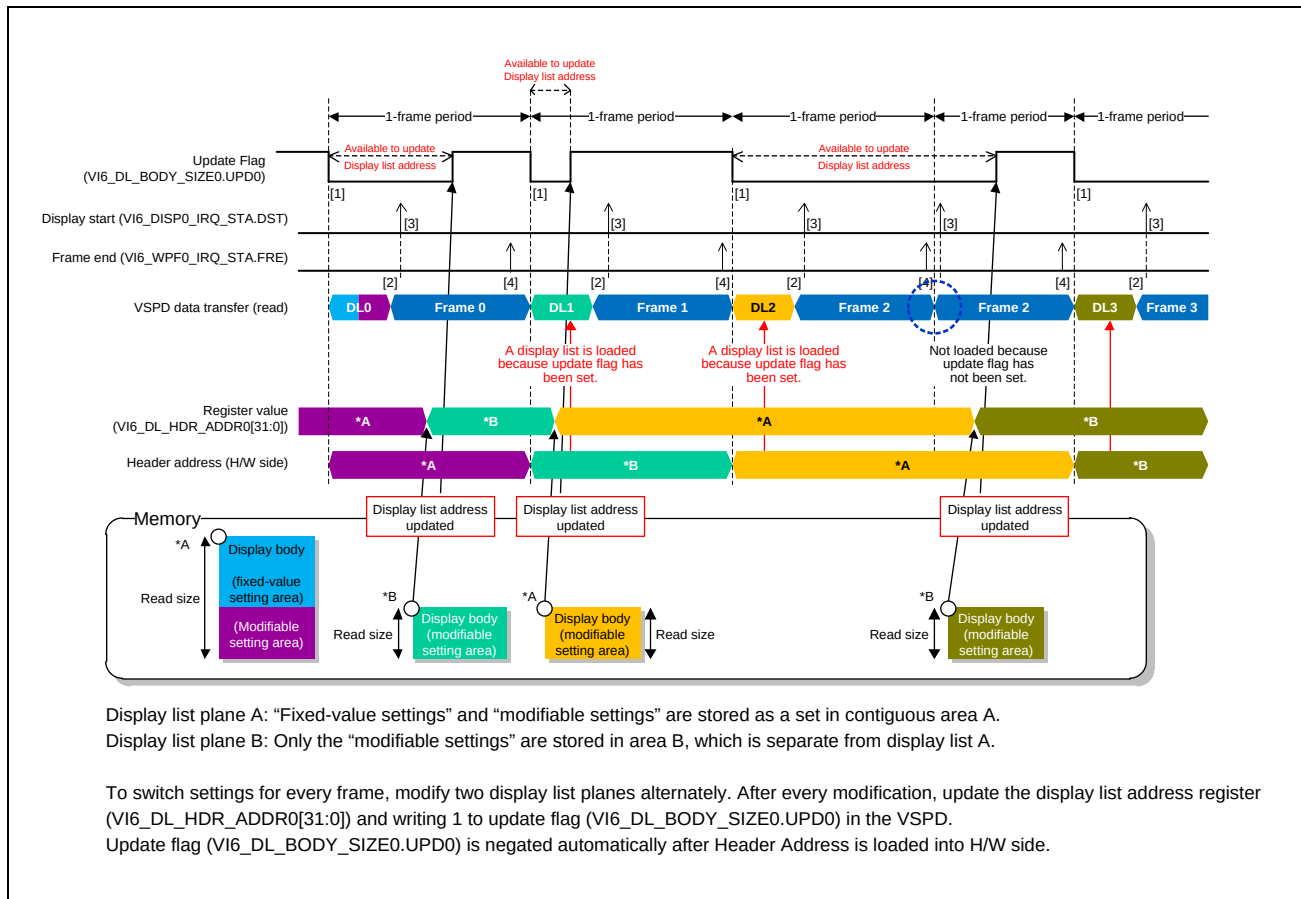


Figure 9.4-41 Controlling Two Register Planes Using a Display List

The VSPD downloads a display list immediately after activation. In the start frame, download the display list that contains all necessary settings. From the next frame on, only the necessary register or table values should be specified in a display list.

When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 1, VSPD downloads a new display list at the start of the next frame. When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 0, the register settings acquired from the display list previously downloaded are retained and used for the next operation without downloading a new display list.

Order of each VSPD -H/W Events mentioned as [1], [2], [3] and [4] in **Figure 9.4-41** are shown below.

[1] VI6_DL_BODY_SIZE0.BS0 and VI6_DL_HDR_ADDR0 are downloaded in H/W side. And Update Flag (VI6_DL_BODY_SIZE0.UPD0) is negated automatically.

[2] Reading DisplayList from external memory into VSPD -H/W is finished through master access.

[3] Display start interrupt status (VI6_DISP_IRQ_STA.DST) asserts.

[4] Frame end interrupt status (VI6_WPF0_IRQ_STA.FRE) asserts.

Don't overwrite the display list in external memory for a period from "[4] VI6_WPF0_IRQ_STA.FRE" to next "[3] VI6_DISP_IRQ_STA.DST", because VSPD is reading the display list for the period.

9.4.6 Usage Note

9.4.6.1 Assignment in Memory Space

Make sure that VSPD memory space shall be mapped to Non-Cache region.

9.4.6.2 Limitations on Software Reset and Module Standby

Stop LCDC with the software reset sequence following to the guidance in **9.4.5.1 Operation Control Setting**.

9.4.6.3 Input Image Size

Table 9.4-36 is a list of input size specifications.

Table 9.4-36 List of Input Size Specifications

Module	Min. Input Size	Max. Input Size	Restriction on Setting Unit
RPF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1200 (vertical) pixels	YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. YCbCr420: 2-pixel units both horizontally and vertically. Other formats: 1-pixel units both horizontally and vertically. <i>Note 1.</i> When the 1-bpp alpha plane*1 is input, the size can always be specified in 8-pixel units both horizontally and vertically regardless of the input format. <i>Note 2.</i> These restrictions including note 1 are applied to the following. VI6_RPFn_SRC_BSIZE VI6_RPFn_SRC_ESIZE
LUT	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1200 (vertical) pixels	1-pixel units both horizontally and vertically.
BRS	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1200 (vertical) pixels	1-pixel units both horizontally and vertically.
LIF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1200 (vertical) pixels	1-pixel units both horizontally and vertically.
WPF	1 (horizontal) × 1 (vertical) pixel	1920 (horizontal) × 1200 (vertical) pixels	YCbCr422: 2-pixel units horizontally and 1-pixel units vertically. YCbCr420: 2-pixel units both horizontally and vertically. Other formats: 1-pixel units both horizontally and vertically. <i>Note 1.</i> This restriction on the WPF only applies to the WPF output size. The WPF input size should be specified in 1-pixel units.

Note 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 011b.

The most important restriction shown in **Table 9.4-36** is the setting unit. Make appropriate register settings so that the size of the image input to each module does comply with setting units and does not exceed the limits shown in **Table 9.4-36**.

9.4.6.4 Output Image Size

The size of the output from each WPF is determined by the results of processing in the modules connected with the DPR. As shown in **Figure 9.4-1**, the data input to the VSPD is sent to the WPF output modules through the RPF and the modules connected with the DPR. When there is no processing that changes the image size through this data path, the WPF output size is the same as the RPF input size. **Table 9.4-37** is a list of the processing that changes image size.

Table 9.4-37 Image Processing that changes image size

Module	Function*1	Related Register	Size of Output from Module
WPF	Input size clipping	VI6_WPFn_HSZCLIP VI6_WPFn_VSZCLIP	When this function is disabled, the input size and the output size are the same. When this function is enabled, the output is in the following size. Horizontal output size: VI6_WPFn_HSZCLIP.HCL_SIZE × 2 setting Vertical output size: VI6_WPFn_VSZCLIP.VCL_SIZE × 2 setting
LIF	Padding line	VI6_LIFn_CTRL	When padding with dummy lines is disabled, output size is same with Input size. When padding with dummy lines is enabled: Horizontal output size: input size Vertical output size: VI6_LIF0_PADLN_SIZE setting

Note 1. For details of each function, refer to the descriptions of the related registers.

Note 2. Refer to **9.4.3.2.1 Notational Conventions for Registers and Bit Fields** for explanation of register bit field.

The input image size can be changed only with the modules and functions shown in **Table 9.4-37**. With the other modules and functions, the input size and output size are the same. Accordingly, after module connections with the DPR are determined, the VSPD output size can also be determined through the following steps.

1. The image size (VI6_RPFn_SRC_ESIZE) read from the external memory and sent to the DPR by the RPFn is the initial value.
2. When the module connected with the DPR is not a module (function) shown in **Table 9.4-37**, the output size from the module is the same as the input size; the image size does not need to be updated.
3. When a module connected with the DPR is a module (function) shown in **Table 9.4-37**, the output image size should be updated to the size shown in the table, which should be used as the input image size for the module connected behind it.
4. When the final image size at the WPFn is determined, this size is the VSPD output size for that WPFn path.

Figure 9.4-42 shows how to determine the image size in a sample DPR connection through the above steps. The related conditions that determine the image size are also shown.

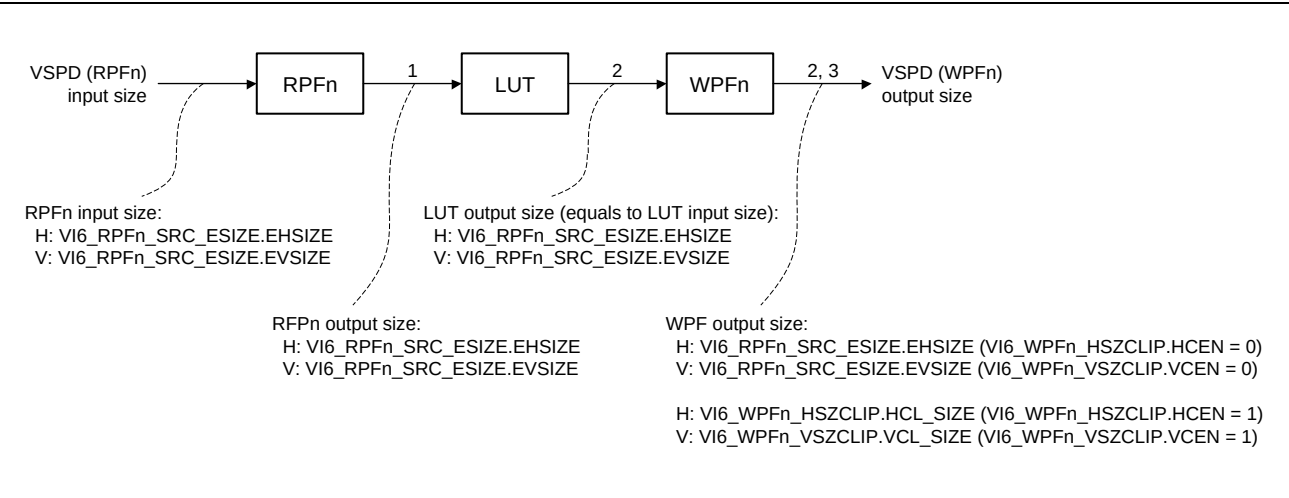


Figure 9.4-42 Input/output Size for Each Module in a Sample DPR Connection

Make appropriate register settings in each module so that the VSPD output image size determined as shown in the figure does not violate the restrictions shown in **Table 9.4-37**.

9.4.6.5 Restriction

When the data output from this unit is written back to the same memory area where the input data for this unit has been read, this unit has the following restrictions.

1. The access order and format on the frame memory are the same between the input pixels and output pixels.
2. Specifying a larger output image size than the input image size, either vertically or horizontally or both vertically and horizontally, is prohibited.
3. When the YCbCr4:2:0 format is input, operation between color components is prohibited.

These restrictions are summarized in. Refer to the descriptions of the registers related to each restriction. In the table, RPFm indicates the RPFn that inputs the master layer, and WPFwb indicates the WPFn that writes back the output image to the source image area for the master layer.

Table 9.4-38 Restrictions on Use when Output Data is Written Back to Input Data Area

No.	Restriction	Related Registers
Restriction 1	The RPFm input format and the WPFwb output format should be the same.	VI6_RPFn_INFMT.RDFMT VI6_RPFn_INFMT.VIR
	The RPFm source image storing address and the WPFwb destination address should be the same.	VI6_RPFn_SRCM_ADDR_*
	The RPFm source picture memory stride and the WPFwb destination memory stride should be the same.	VI6_RPFn_SRCM_PSTRIDE VI6_RPFn_SRCM_ASTRIDE
	The RPFm and WPFwb data swapping settings should be the same.	VI6_WPFn_DSWAP
Restriction 2	The RPFm basic read size and extended read size should be the same.	VI6_RPFn_SRC_BSIZE VI6_RPFn_SRC_ESIZE
	Color space conversion is prohibited in RPFm and WPFwb.	VI6_RPFn_INFMT.CSC VI6_WPFn_OUTFMT.CSC
Restriction 3*1	NOP should be specified for IROP operation.	VI6_RPFn_ALPH_SEL.IROP
	Color keying is prohibited.	VI6_RPFn_CKEY_CTRL.CV

Note 1. When the input format is not YCbCr4:2:0, restriction 3 is not applied.

SECTION 9 IMAGE

9.5 MIPI DSI Interface (DSI)

9.5.1 Overview

This section describes the features of the MIPI-DSI unit of this LSI.

This unit is the MIPI-DSI Tx Module; it is composed of the MIPI DSI-2 host controller (LINK) and the MIPI D-PHY Tx (D-PHY).

This unit supports the MIPI Alliance Specification for Display Serial Interface (DSI) Specification. This unit provides a solution for transmitting MIPI DSI compliant digital video and packets. The normative references are listed below.

- MIPI Alliance Specification for Display Serial Interface Version 1.3.1
- MIPI Alliance Specification for D-PHY Version 1.2

9.5.1.1 Features

The following is key features of this unit.

In this document the word “DSI” means “Display Serial Interface Version 1.3.1”.

Overview

- 1 channel
- The number of lanes: 4 lanes
- Supports up to Full HD (1920 × 1200), 60 fps (RGB888)
- Maximum bandwidth: 1.5 Gbps per lane
- Supports the output data format: RGB666 / RGB888

Protocol Interface

- PPI interface (8 bits)
- Packet generation from Video Input signal for Video Mode (Video-Input Operation)
 - Supports the RGB format (18-bit, 18-bit loosely and 24-bit)
 - Not supports the RGB format (30-bit and 36-bit)
 - Not supports the YCbCr 4:2:2 format (16-bit, 20-bit loosely and 24-bit)
 - Not supports the YCbCr 4:2:0 format (12-bit)
 - Not supports Compressed Pixel Stream
 - Supports Blanking Packet or LP-11 selection during each of HSA, HBP, and HFP blanking interval
 - Supports “Non-Burst Mode with Sync Pulse”, “Non-Burst Mode with Sync Event”, and “Burst Mode”
 - Not supports interlaced video
 - Not supports deskew pattern insertion after vertical video data timing

- LP only packet generation and LP packet reception from descriptor list (Sequence Operation channel 0)
 - Supports one sequence input channel
- HS or LP packet generation and LP packet reception from descriptor list (Sequence Operation channel 1)
 - Supports one sequence input channel

Link Layer

- Supports 1, 2, 3, and 4 lane configurations
- Supports unidirectional high-speed mode Tx
- Supports bidirectional LP mode Tx/Rx (Only Lane0)
- Supports ECC/Checksum generation for Tx packet
- Supports ECC/Checksum verification and ECC error correction for Rx packet
- Supports automated insertion of EoTp in HS mode
- Supports ULPS (Tx)
- Supports automated power change to LP mode and return to HS mode
- Supports automated clock stop and resume (non-continuous clock mode)
- Supports assignment for Virtual Channel for video input channel in Video-Input Operation
- Supports assignment for individual Virtual Channel for each packet in Sequence Operation
- Supports detection for PHY contention error and timeout error
- Supports generation of scrambled packets
- Not supports input of TE signal

PHY Layer

- Synchronous link between Master (data source) and Slave (data sink)
- All lanes support high-speed transmission in the forward direction.
- Bi-directional data transmission in Low-Power mode at the Master Data Lane 0 only
- Uses token passing to control the communication direction of the link.
- High-Speed mode for fast data traffic and Low-Power mode for controls and low-speed data transmission.
- High-Speed mode: Differential and terminated, 200 mV swing, and 80 - 1500 Mbps operation speed.
- Low-Power mode: Single-ended and non-terminated, 1.2 V swing, and maximum 10 Mbps speed.
- Number of lanes
 - Master data lane: 4 lanes
 - Master clock lane: 1 lane
- Lane module functions
 - Master data lane 0: CIL-MFAA (HS-TX, LP-TX, LP-RX, LP-CD)
 - Master data lanes 1 to 3: CIL-MFEN (HS-TX, LP-TX)
 - Master clock lane: CIL-MCEN (HS-TX, LP-TX)

- Optional functions

[Escape mode]

- LPDT

- LP transmit: Supported
- LP receive: Supported (only Master Data Lane 0)
- ULPS: Supported
- Reset trigger: Supported

- Bi-direction

- Bus turn-around: Supported
- Contention detection: Supported

- Calibration

- Deskew calibration / Preamble: Not-supported
- Pre/De-Emphasis: Not-supported
- Equalization: Not-Supported

9.5.1.2 Block Diagram

Figure 9.5-1 shows a block diagram.

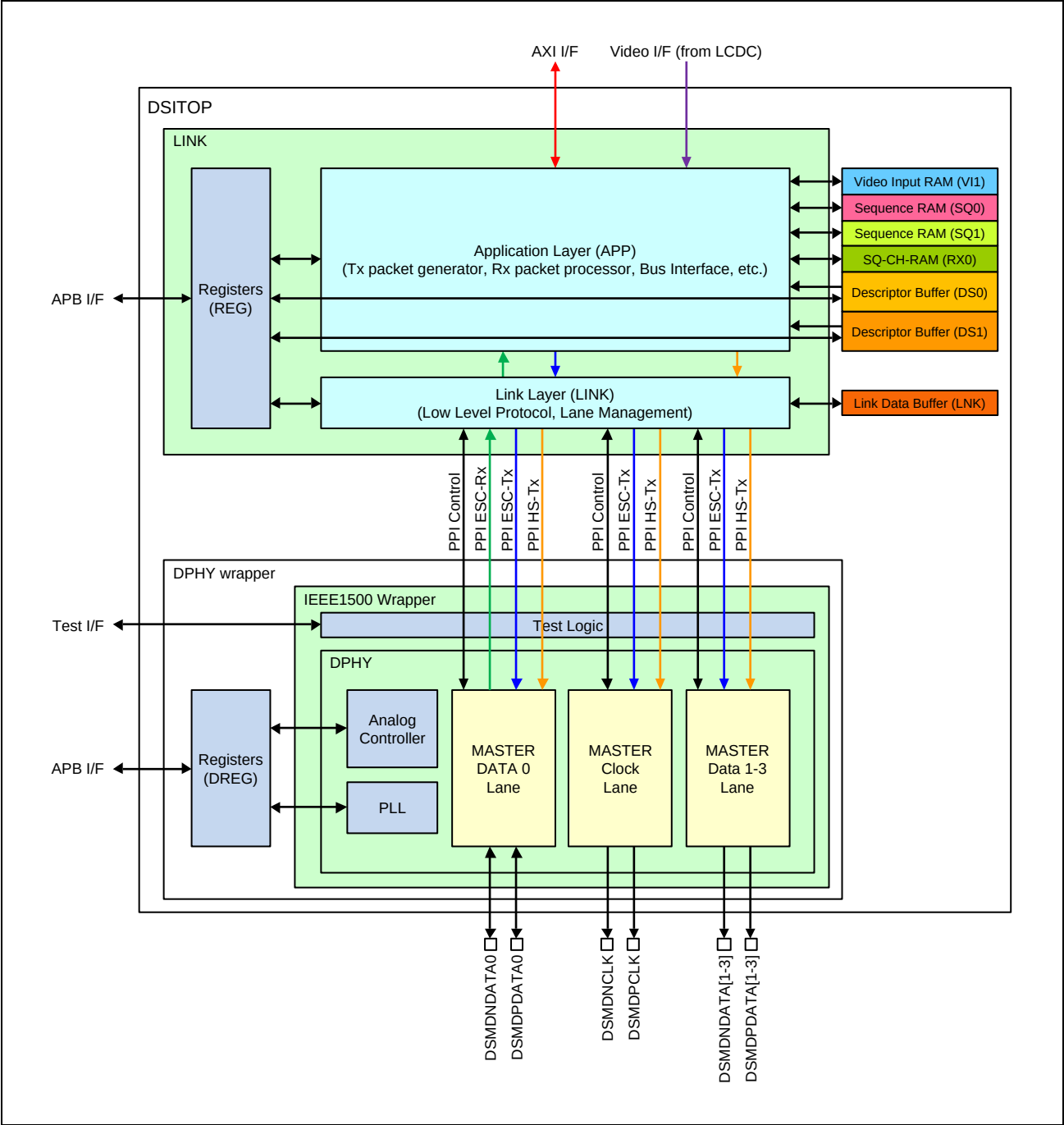


Figure 9.5-1 Block Diagram

9.5.1.3 External Pins

Table 9.5-1 shows the pin configuration.

Table 9.5-1 External Pins

Pin Name	Input/Output	Function
DSI_DPCLK	Output	MIPI DSI Tx clock lane (pos)
DSI_DNCLK	Output	MIPI DSI Tx clock lane (neg)
DSI_DPDATA0	Output	MIPI DSI Tx data 0 lane (pos)
DSI_DNDATA0	Output	MIPI DSI Tx data 0 lane (neg)
DSI_DPDATA1	Output	MIPI DSI Tx data 1 lane (pos)
DSI_DNDATA1	Output	MIPI DSI Tx data 1 lane (neg)
DSI_DPDATA2	Output	MIPI DSI Tx data 2 lane (pos)
DSI_DNDATA2	Output	MIPI DSI Tx data 2 lane (neg)
DSI_DPDATA3	Output	MIPI DSI Tx data 3 lane (pos)
DSI_DNDATA3	Output	MIPI DSI Tx data 3 lane (neg)

9.5.1.4 Clocks

The following shows a list of clocks for this unit.

- T_{LPX} : Transmitted length of any low-power state period (Min: 50 ns)
- $T_{LPX(H)}$: TLPX of the host processor (the DSI-2 TX module)
- $T_{LPX(P)}$: TLPX of the peripheral device (connected device)

Table 9.5-2 Clocks and the Frequency Range

Clock Name	CPG Clock Name	Input/ Output	Description	Frequency (MHz)
pclk	DSI_0_pclk	Input	APB clock	100
acclk	DSI_0_acclk	Input	AXI clock	400/200/100/50/13
vcclk1	DSI_0_vcclk1	Input	Video clock	187.5 to 5.44
lpclk	DSI_0_lpclk	Input	DSI Escape mode Transmit Clock (TxClkEsc) Set D-PHY TxClkEsc ($1/T_{LPX(H)}$) frequency. $T_{LPX(H)}/T_{LPX(P)}$ needs to set between 2/3 to 3/2.	16/8/4/2
PLLREFCLK	DSI_0_PLLREFCLK	Input	PLL reference input clock	24
hscclk	—	—	DSI High-Speed Transmit Word Clock (TxWordClkHS) Upper limit for D-PHY 1.5 Gbps Lower limit for D-PHY 80 Mbps This clock is supplied from D-PHY directly, and its frequency is 1/8 of the PLL clock frequency (F_{FOUT}).	10 to 187.5
lppclk	—	—	DSI Escape mode Receive Clock (RxClkEsc) This clock is supplied from D-PHY directly, and its frequency is ($1/T_{LPX(P)}$). This clock is generated dividing in D-PHY.	2 to 20

Video clock and DSI HS Byte clock must follow the relationship.

$$\begin{aligned} & \text{Video clock Frequency [Hz]} \times \text{Video Pixel Bit Depth [bit]} \\ & \leq \text{DSI HS Byte clock Frequency [Hz]} \times 8 [\text{bit}] \times \text{Number of DSI HS Data Lane} \end{aligned}$$

9.5.2 DSI Registers

The base addresses for the DSI are as follows.

Table 9.5-3 Register Base Addresses

Base Register Name	Unit Name	Base Address
<DSI_LINK_base>	DSI (LINK)	0_1643_0000h (5643_0000h* ¹ , 4643_0000h* ²)
<DSI_DPHY_base>	DSI (DPHY)	0_1644_0000h (5644_0000h* ¹ , 4644_0000h* ²)

Note 1. CM33 address space (non-secure)

Note 2. CM33 address space (secure)

9.5.2.1 List of Registers

9.5.2.1.1 Registers in LINK

This section describes the registers in LINK. All registers can be accessed in 8/16/32 bits.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Interrupt Status Register	DSI_LINK_ISR	0000_0000h	0000h	8, 16, 32
Reserve	-	-	0004h to 000Fh	-
Link Status Register	DSI_LINK_LINKSR	0000_0000h	0010h	8, 16, 32
Reserve	-	-	0014h to 00FFh	-
Tx Set Register	DSI_LINK_TXSETR	0003_0003h	0100h	8, 16, 32
HS Clock Set Register	DSI_LINK_HSCLKSETR	0000_0000h	0104h	8, 16, 32
ULPS Set Register	DSI_LINK_ULPSSETR	0000_00A0h	0108h	8, 16, 32
ULPS Control Register	DSI_LINK_ULPSCLR	0000_0000h	010Ch	8, 16, 32
Reset Control Register	DSI_LINK_RSTCR	0000_0000h	0110h	8, 16, 32
Reset Status Register	DSI_LINK_RSTSR	0000_0F00h	0114h	8, 16, 32
Reserve	-	-	0118h to 011Fh	-
DSI Set Register	DSI_LINK_DSISETR	80F1_0001h	0120h	8, 16, 32
Receive Buffer Size Register	DSI_LINK_RXBUFSZR	0000_0000h	0124h	8, 16, 32
Reserve	-	-	0128h to 015Fh	-
Tx Packet Payload Data 0 Register	DSI_LINK_TXPPD0R	0000_0000h	0160h	8, 16, 32
Tx Packet Payload Data 1 Register	DSI_LINK_TXPPD1R	0000_0000h	0164h	8, 16, 32
Tx Packet Payload Data 2 Register	DSI_LINK_TXPPD2R	0000_0000h	0168h	8, 16, 32
Tx Packet Payload Data 3 Register	DSI_LINK_TXPPD3R	0000_0000h	016Ch	8, 16, 32
Reserve	-	-	0170h to 01FFh	-
Rx Status Register	DSI_LINK_RXSR	0000_0000h	0200h	8, 16, 32
Rx Status Clear Register	DSI_LINK_RXSCR	0000_0000h	0204h	8, 16, 32
Rx Interrupt Enable Register	DSI_LINK_RXIER	0000_0000h	0208h	8, 16, 32
Reserve	-	-	020Ch to 020Fh	-
Peripheral Response Timeout BTA Set Register	DSI_LINK_PRESPTOBTASETR	0000_0000h	0210h	8, 16, 32
Peripheral Response Timeout LP Set Register	DSI_LINK_PRESPTOLPSETR	0000_0000h	0214h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Peripheral Response Timeout HS Set Register	DSI_LINK_PRESPTOHSSETR	0000_0000h	0218h	8, 16, 32
Reserve	-	-	021Ch to 021Fh	-
Acknowledge and Error Report Packet Parameter Latest Info Register	DSI_LINK_AKEPLATIR	0000_0000h	0220h	8, 16, 32
Acknowledge and Error Report Packet Parameter Accumulate Status Register	DSI_LINK_AKEPACMSR	0000_0000h	0224h	8, 16, 32
Acknowledge and Error Report Packet Parameter Status Clear Register	DSI_LINK_AKEPSCR	0000_0000h	0228h	8, 16, 32
Reserve	-	-	022Ch to 022Fh	-
Rx Result Saved Status Register	DSI_LINK_RXRSSR	0000_0000h	0230h	8, 16, 32
Rx Result Saved Status Clear Register	DSI_LINK_RXRSSCR	0000_0000h	0234h	8, 16, 32
Rx Result Info Overwrite Status Register	DSI_LINK_RXRINFOOWSR	0000_0000h	0238h	8, 16, 32
Rx Result Info Overwrite Status Clear Register	DSI_LINK_RXRINFOOWSCR	0000_0000h	023Ch	8, 16, 32
Rx Result Save Slot 0 Register	DSI_LINK_RXRSS0R	0000_0000h	0240h	8, 16, 32
Rx Result Save Slot 1 Register	DSI_LINK_RXRSS1R	0000_0000h	0244h	8, 16, 32
Rx Result Save Slot 2 Register	DSI_LINK_RXRSS2R	0000_0000h	0248h	8, 16, 32
Rx Result Save Slot 3 Register	DSI_LINK_RXRSS3R	0000_0000h	024Ch	8, 16, 32
Reserve	-	-	0250h to 02BFh	-
Rx Packet Payload Data 0 Register	DSI_LINK_RXPPD0R	0000_0000h	02C0h	8, 16, 32
Rx Packet Payload Data 1 Register	DSI_LINK_RXPPD1R	0000_0000h	02C4h	8, 16, 32
Rx Packet Payload Data 2 Register	DSI_LINK_RXPPD2R	0000_0000h	02C8h	8, 16, 32
Rx Packet Payload Data 3 Register	DSI_LINK_RXPPD3R	0000_0000h	02CCh	8, 16, 32
Reserve	-	-	02D0h to 02DFh	-
HSTX Timeout Set Register	DSI_LINK_HSTXTOSETR	0000_0000h	02E0h	8, 16, 32
LRX-H Timeout Set Register	DSI_LINK_LRXHTOSETR	0000_0000h	02E4h	8, 16, 32
TA Timeout Set Register	DSI_LINK_TATOSETR	0000_0000h	02E8h	8, 16, 32
Reserve	-	-	02ECh to 02FFh	-
Fatal Error Status Register	DSI_LINK_FERRSR	0000_0000h	0300h	8, 16, 32
Fatal Error Status Clear Register	DSI_LINK_FERRSCR	0000_0000h	0304h	8, 16, 32
Fatal Error Interrupt Enable Register	DSI_LINK_FERRIER	0000_0000h	0308h	8, 16, 32
Reserve	-	-	030Ch to 0313h	-
Clock Lane Stop Time Set Register	DSI_LINK_CLSTPTSETR	0000_0000h	0314h	8, 16, 32
LP Transition Time Set Register	DSI_LINK_LPTRNSTSETR	0000_0000h	0318h	8, 16, 32
Reserve	-	-	031Ch to 031Fh	-
Physical Lane Status Register	DSI_LINK_PLSR	0000_0FF3h	0320h	8, 16, 32
Physical Lane Status Clear Register	DSI_LINK_PLSCR	0000_0000h	0324h	8, 16, 32
Physical Lane Interrupt Enable Register	DSI_LINK_PLIER	0000_0000h	0328h	8, 16, 32
Reserve	-	-	032Ch to 03FFh	-
Video-Input Channel 1 Set 0 Register	DSI_LINK_VICH1SET0R	0000_0000h	0400h	8, 16, 32
Video-Input Channel 1 Set 1 Register	DSI_LINK_VICH1SET1R	1093_0000h	0404h	8, 16, 32
Reserve	-	-	0408h to 040Fh	-
Video-Input Channel 1 Status Register	DSI_LINK_VICH1SR	0000_0000h	0410h	8, 16, 32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Video-Input Channel 1 Status Clear Register	DSI_LINK_VICH1SCR	0000_0000h	0414h	8, 16, 32
Video-Input Channel 1 Interrupt Enable Register	DSI_LINK_VICH1IER	0000_0000h	0418h	8, 16, 32
Reserve	-	-	041Ch to 041Fh	-
Video-Input Channel 1 Pixel Packet Set Register	DSI_LINK_VICH1PPSETR	000E_0000h	0420h	8, 16, 32
Reserve	-	-	0424h to 0427h	-
Video-Input Channel 1 Vertical Size Set Register	DSI_LINK_VICH1VSSETR	0000_0000h	0428h	8, 16, 32
Video-Input Channel 1 Vertical Porch Set Register	DSI_LINK_VICH1VPSETR	0000_0000h	042Ch	8, 16, 32
Video-Input Channel 1 Horizontal Size Set Register	DSI_LINK_VICH1HSSETR	0000_0000h	0430h	8, 16, 32
Video-Input Channel 1 Horizontal Porch Set Register	DSI_LINK_VICH1HPSETR	0000_0000h	0434h	8, 16, 32
Reserve	-	-	0438h to 05BFh	-
Sequence Channel 0 Set 0 Register	DSI_LINK_SQCH0SET0R	0080_0000h	05C0h	8, 16, 32
Sequence Channel 0 Set 1 Register	DSI_LINK_SQCH0SET1R	0800_0000h	05C4h	8, 16, 32
Reserve	-	-	05C8h to 05CFh	-
Sequence Channel 0 Status Register	DSI_LINK_SQCH0SR	0000_0000h	05D0h	8, 16, 32
Sequence Channel 0 Status Clear Register	DSI_LINK_SQCH0SCR	0000_0000h	05D4h	8, 16, 32
Sequence Channel 0 Interrupt Enable Register	DSI_LINK_SQCH0IER	0000_0000h	05D8h	8, 16, 32
Reserve	-	-	05DCh to 05FFh	-
Sequence Channel 1 Set 0 Register	DSI_LINK_SQCH1SET0R	0000_0000h	0600h	8, 16, 32
Sequence Channel 1 Set 1 Register	DSI_LINK_SQCH1SET1R	0830_0000h	0604h	8, 16, 32
Reserve	-	-	0608h to 060Fh	-
Sequence Channel 1 Status Register	DSI_LINK_SQCH1SR	0000_0000h	0610h	8, 16, 32
Sequence Channel 1 Status Clear Register	DSI_LINK_SQCH1SCR	0000_0000h	0614h	8, 16, 32
Sequence Channel 1 Interrupt Enable Register	DSI_LINK_SQCH1IER	0000_0000h	0618h	8, 16, 32
Reserve	-	-	061Ch to 077Fh	-
Sequence Channel 0 Descriptor 00-A Register	DSI_LINK_SQCH0DSC00AR	xxxx_xxxxh	0780h	32
Sequence Channel 0 Descriptor 00-B Register	DSI_LINK_SQCH0DSC00BR	xxxx_xxxxh	0784h	32
Sequence Channel 0 Descriptor 00-C Register	DSI_LINK_SQCH0DSC00CR	xxxx_xxxxh	0788h	32
Sequence Channel 0 Descriptor 00-D Register	DSI_LINK_SQCH0DSC00DR	xxxx_xxxxh	078Ch	32
Sequence Channel 0 Descriptor 01-A Register	DSI_LINK_SQCH0DSC01AR	xxxx_xxxxh	0790h	32
Sequence Channel 0 Descriptor 01-B Register	DSI_LINK_SQCH0DSC01BR	xxxx_xxxxh	0794h	32
Sequence Channel 0 Descriptor 01-C Register	DSI_LINK_SQCH0DSC01CR	xxxx_xxxxh	0798h	32
Sequence Channel 0 Descriptor 01-D Register	DSI_LINK_SQCH0DSC01DR	xxxx_xxxxh	079Ch	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Sequence Channel 0 Descriptor 02-A Register	DSI_LINK_SQCH0DSC02AR	xxxx_xxxxh	07A0h	32
Sequence Channel 0 Descriptor 02-B Register	DSI_LINK_SQCH0DSC02BR	xxxx_xxxxh	07A4h	32
Sequence Channel 0 Descriptor 02-C Register	DSI_LINK_SQCH0DSC02CR	xxxx_xxxxh	07A8h	32
Sequence Channel 0 Descriptor 02-D Register	DSI_LINK_SQCH0DSC02DR	xxxx_xxxxh	07ACh	32
Sequence Channel 0 Descriptor 03-A Register	DSI_LINK_SQCH0DSC03AR	xxxx_xxxxh	07B0h	32
Sequence Channel 0 Descriptor 03-B Register	DSI_LINK_SQCH0DSC03BR	xxxx_xxxxh	07B4h	32
Sequence Channel 0 Descriptor 03-C Register	DSI_LINK_SQCH0DSC03CR	xxxx_xxxxh	07B8h	32
Sequence Channel 0 Descriptor 03-D Register	DSI_LINK_SQCH0DSC03DR	xxxx_xxxxh	07BCh	32
Sequence Channel 0 Descriptor 04-A Register	DSI_LINK_SQCH0DSC04AR	xxxx_xxxxh	07C0h	32
Sequence Channel 0 Descriptor 04-B Register	DSI_LINK_SQCH0DSC04BR	xxxx_xxxxh	07C4h	32
Sequence Channel 0 Descriptor 04-C Register	DSI_LINK_SQCH0DSC04CR	xxxx_xxxxh	07C8h	32
Sequence Channel 0 Descriptor 04-D Register	DSI_LINK_SQCH0DSC04DR	xxxx_xxxxh	07CCh	32
Sequence Channel 0 Descriptor 05-A Register	DSI_LINK_SQCH0DSC05AR	xxxx_xxxxh	07D0h	32
Sequence Channel 0 Descriptor 05-B Register	DSI_LINK_SQCH0DSC05BR	xxxx_xxxxh	07D4h	32
Sequence Channel 0 Descriptor 05-C Register	DSI_LINK_SQCH0DSC05CR	xxxx_xxxxh	07D8h	32
Sequence Channel 0 Descriptor 05-D Register	DSI_LINK_SQCH0DSC05DR	xxxx_xxxxh	07DCh	32
Sequence Channel 0 Descriptor 06-A Register	DSI_LINK_SQCH0DSC06AR	xxxx_xxxxh	07E0h	32
Sequence Channel 0 Descriptor 06-B Register	DSI_LINK_SQCH0DSC06BR	xxxx_xxxxh	07E4h	32
Sequence Channel 0 Descriptor 06-C Register	DSI_LINK_SQCH0DSC06CR	xxxx_xxxxh	07E8h	32
Sequence Channel 0 Descriptor 06-D Register	DSI_LINK_SQCH0DSC06DR	xxxx_xxxxh	07ECh	32
Sequence Channel 0 Descriptor 07-A Register	DSI_LINK_SQCH0DSC07AR	xxxx_xxxxh	07F0h	32
Sequence Channel 0 Descriptor 07-B Register	DSI_LINK_SQCH0DSC07BR	xxxx_xxxxh	07F4h	32
Sequence Channel 0 Descriptor 07-C Register	DSI_LINK_SQCH0DSC07CR	xxxx_xxxxh	07F8h	32
Sequence Channel 0 Descriptor 07-D Register	DSI_LINK_SQCH0DSC07DR	xxxx_xxxxh	07FCh	32
Sequence Channel 1 Descriptor 00-A Register	DSI_LINK_SQCH1DSC00AR	xxxx_xxxxh	0800h	32
Sequence Channel 1 Descriptor 00-B Register	DSI_LINK_SQCH1DSC00BR	xxxx_xxxxh	0804h	32
Sequence Channel 1 Descriptor 00-C Register	DSI_LINK_SQCH1DSC00CR	xxxx_xxxxh	0808h	32
Sequence Channel 1 Descriptor 00-D Register	DSI_LINK_SQCH1DSC00DR	xxxx_xxxxh	080Ch	32
Sequence Channel 1 Descriptor 01-A Register	DSI_LINK_SQCH1DSC01AR	xxxx_xxxxh	0810h	32
Sequence Channel 1 Descriptor 01-B Register	DSI_LINK_SQCH1DSC01BR	xxxx_xxxxh	0814h	32
Sequence Channel 1 Descriptor 01-C Register	DSI_LINK_SQCH1DSC01CR	xxxx_xxxxh	0818h	32
Sequence Channel 1 Descriptor 01-D Register	DSI_LINK_SQCH1DSC01DR	xxxx_xxxxh	081Ch	32
Sequence Channel 1 Descriptor 02-A Register	DSI_LINK_SQCH1DSC02AR	xxxx_xxxxh	0820h	32

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
Sequence Channel 1 Descriptor 02-B Register	DSI_LINK_SQCH1DSC02BR	xxx_xxxxh	0824h	32
Sequence Channel 1 Descriptor 02-C Register	DSI_LINK_SQCH1DSC02CR	xxx_xxxxh	0828h	32
Sequence Channel 1 Descriptor 02-D Register	DSI_LINK_SQCH1DSC02DR	xxx_xxxxh	082Ch	32
Sequence Channel 1 Descriptor 03-A Register	DSI_LINK_SQCH1DSC03AR	xxx_xxxxh	0830h	32
Sequence Channel 1 Descriptor 03-B Register	DSI_LINK_SQCH1DSC03BR	xxx_xxxxh	0834h	32
Sequence Channel 1 Descriptor 03-C Register	DSI_LINK_SQCH1DSC03CR	xxx_xxxxh	0838h	32
Sequence Channel 1 Descriptor 03-D Register	DSI_LINK_SQCH1DSC03DR	xxx_xxxxh	083Ch	32
Sequence Channel 1 Descriptor 04-A Register	DSI_LINK_SQCH1DSC04AR	xxx_xxxxh	0840h	32
Sequence Channel 1 Descriptor 04-B Register	DSI_LINK_SQCH1DSC04BR	xxx_xxxxh	0844h	32
Sequence Channel 1 Descriptor 04-C Register	DSI_LINK_SQCH1DSC04CR	xxx_xxxxh	0848h	32
Sequence Channel 1 Descriptor 04-D Register	DSI_LINK_SQCH1DSC04DR	xxx_xxxxh	084Ch	32
Sequence Channel 1 Descriptor 05-A Register	DSI_LINK_SQCH1DSC05AR	xxx_xxxxh	0850h	32
Sequence Channel 1 Descriptor 05-B Register	DSI_LINK_SQCH1DSC05BR	xxx_xxxxh	0854h	32
Sequence Channel 1 Descriptor 05-C Register	DSI_LINK_SQCH1DSC05CR	xxx_xxxxh	0858h	32
Sequence Channel 1 Descriptor 05-D Register	DSI_LINK_SQCH1DSC05DR	xxx_xxxxh	085Ch	32
Sequence Channel 1 Descriptor 06-A Register	DSI_LINK_SQCH1DSC06AR	xxx_xxxxh	0860h	32
Sequence Channel 1 Descriptor 06-B Register	DSI_LINK_SQCH1DSC06BR	xxx_xxxxh	0864h	32
Sequence Channel 1 Descriptor 06-C Register	DSI_LINK_SQCH1DSC06CR	xxx_xxxxh	0868h	32
Sequence Channel 1 Descriptor 06-D Register	DSI_LINK_SQCH1DSC06DR	xxx_xxxxh	086Ch	32
Sequence Channel 1 Descriptor 07-A Register	DSI_LINK_SQCH1DSC07AR	xxx_xxxxh	0870h	32
Sequence Channel 1 Descriptor 07-B Register	DSI_LINK_SQCH1DSC07BR	xxx_xxxxh	0874h	32
Sequence Channel 1 Descriptor 07-C Register	DSI_LINK_SQCH1DSC07CR	xxx_xxxxh	0878h	32
Sequence Channel 1 Descriptor 07-D Register	DSI_LINK_SQCH1DSC07DR	xxx_xxxxh	087Ch	32

9.5.2.1.2 Registers in D-PHY

This section describes the registers in D-PHY. All registers can be accessed in 8/16/32 bits.

Register Name	Abbreviation	Initial Value	Offset Address	Access Size [bits]
DSI PLL Enable Register	DSI_DPHY_PLENR	0000_0000h	0000h	8, 16, 32
DSI PHY Reset Register	DSI_DPHY_PHYRSTR	0000_0000h	0004h	8, 16, 32
Reserve	-	-	0008h to 000Fh	-
DSI PLL Clock Setting 0 Register	DSI_DPHY_PLLCLKSET0R	0000_0000h	0010h	8, 16, 32
DSI PLL Clock Setting 1 Register	DSI_DPHY_PLLCLKSET1R	0000_0000h	0014h	8, 16, 32
Reserve	-	-	0018h to 001Fh	-
DSI PHY TCLK Setting Register	DSI_DPHY_PHYTCLKSETR	0000_0000h	0020h	8, 16, 32
DSI PHY THS Setting Register	DSI_DPHY_PHYTHSSETR	0000_0000h	0024h	8, 16, 32
DSI PHY TLPX Setting Register	DSI_DPHY_PHYTLPXSETR	0000_0000h	0028h	8, 16, 32
Reserve	-	-	002Ch to 002Fh	-
DSI PHY Control Register	DSI_DPHY_PHYCR	0000_0000h	0030h	8, 16, 32
DSI PHY Control 1 Register	DSI_DPHY_PHYC1R	0000_0000h	0034h	8, 16, 32
DSI PHY Control 2 Register	DSI_DPHY_PHYC2R	0000_0000h	0038h	8, 16, 32
DSI PHY Control 3 Register	DSI_DPHY_PHYC3R	0000_0000h	003Ch	8, 16, 32

9.5.2.2 MIPI DSI Register Descriptions

The prefix (DSI_LINK_, DSI_DPHY_) of the register names is omitted in this and subsequent sections.

9.5.2.2.1 Interrupt Status Register (DSI_LINK_ISR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 0000h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	PPI	-	-	-	FERR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	RCV	-	-	-	VIN1	-	-	-	SQ1	-	-	-	SQ0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	PPI	0h	R	PPI Interrupt When this value is 1b, a DSI D-PHY PPI related interrupt exists. Refer to the description of the PLSR register for more information.
19 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	FERR	0h	R	FERR Interrupt When this value is 1b, a DSI Fatal Error interrupt exists. Refer to the description of the FERRSR register for more information.
15 to 13	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
12	RCV	0h	R	RCV Interrupt When this value is 1b, a DSI Packet Receive interrupt exists. Refer to the description of the RXSR register for more information.
11 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	VIN1	0h	R	VIN1 Interrupt When this value is 1b, a Video Input-Operation channel 1 interrupt exists. Refer to the description of the VICH1SR register for more information.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	SQ1	0h	R	SQ1 Interrupt When this value is 1b, a Sequence Operation channel 1 interrupt exists. Refer to the description of the SQCH1SR register for more information.
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	SQ0	0h	R	SQ0 Interrupt When this value is 1b, a Sequence Operation channel 0 interrupt exists. Refer to the description of the SQCH0SR register for more information.

9.5.2.2.2 Link Status Register (DSI_LINK_LINKSR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 0010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	LPBUS Y	HSBUS Y	-	-	-	VICHR UN1	-	-	-	SQCH RUN1	-	-	-	SQCH RUN0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13	LPBUSY	0h	R	LP Operation Busy The setting 1b indicates that an operation related to LP is in progress.
12	HSBUSY	0h	R	HS Operation Busy The setting 1b indicates that an operation related to HS is in progress.
11 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	VICHRUN1	0h	R	Video-Input Channel 1 Running This bit is a copy of VICH1SR.RUNNING.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	SQCHRUN1	0h	R	Sequence Channel 1 Running This bit is a copy of SQCH1SR.RUNNING.
3 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	SQCHRUN0	0h	R	Sequence Channel 0 Running This bit is a copy of SQCH0SR.RUNNING.

9.5.2.2.3 Tx Set Register (DSI_LINK_TXSETR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 0100h

Initial Value : 0003_0003h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	NUMLANE CAP[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DLEN	CLEN	-	-	-	-	-	-	NUMLANE USE[1:0]	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	RW	RW	R	R	R	R	R	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17, 16	NUMLANE CAP[1:0]	3h	R	Number of Lanes Capability 3h: The maximum Lane count is 4 (Lanes 0 to 3 can be used)
15 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9	DLEN	0h	RW	Data Lane Enable When this bit is 1b, Data Lanes are enabled by the PHY interface. Lanes to be enabled are determined by NUMLANEUSE.
8	CLEN	0h	RW	Clock Lane Enable When this bit is 1b, Clock Lane is enabled by the PHY interface.
7 to 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1, 0	NUMLANE USE[1:0]	3h	RW	Number of Lanes for Use Set the Lane count for use. Selected Data Lanes are enabled if DLEN is "1". 0h: 1 Lane (use of Lane 0) 1h: 2 Lane (use of Lanes 0 and 1) 2h: 3 Lane (use of Lanes 0 to 2) 3h: 4 Lane (use of Lanes 0 to 3) Setting a value over NUMLANECAP is prohibited.

9.5.2.2.4 HS Clock Set Register (DSI_LINK_HSCLKSETR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_LINK_base> + 0104h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	HSCLK MO DE	HSCLK RUN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 2	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
1	HSCLKMO DE		0h		RW		HS Clock running Mode 0b: Requests Clock Lane HS transmission when HS transmission is scheduled (non-continuous clock mode). 1b: Keeps Clock Lane HS transmission (continuous clock mode). If HSCLKRUN = 0, this field has no meaning. Changing the value of this bit while HSCLKRUN = 1 is prohibited.									
0	HSCLKRUN		0h		RW		Start HS Clock running on Clock Lane 0b: No HS transmission (keep LP) 1b: Allows HS transmission according to HSCLKMODE Confirm stable TxWordClkHS (hscIk) before setting this field to 1. Setting this field to “1” while TXSETR.CLEN = 0 is prohibited.									

9.5.2.2.5 ULPS Set Register (DSI_LINK_ULPSSETR)

Access Size :		8, 16, 32 bits																
Address :		<DSI_LINK_base> + 0108h																
Initial Value :		0000_00A0h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	ULPSWKUP[7:0]									
Initial Value	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	Bit Name		Initial Value		R/W		Description											
31 to 8	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.											
7 to 0	ULPSWKUP [7:0]		A0h		RW		Twakeup period of ULPS This period is specified by the count value of lpcik *128. This period should be longer than 1 ms. For example, - In the case lpcik is 4/3 MHz and lpcik*128 = 96 μs, set ULPSWKUP = 0Bh and Twakeup period = 1.06 ms (11*96 μs) - In the case lpcik is 20 MHz and lpcik*128 = 6.4 μs, set ULPSWKUP = A0h and Twakeup period = 1.02 ms (160*6.4 μs)											

9.5.2.2.6 ULPS Control Register (DSI_LINK_ULPSCR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 010Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	DLULP SEX T	DLULP SEN T	-	-	CLULP SEX T	CLULP SEN T	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R0W	R0W	R	R	R0W	R0W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29	DLULPSEX T	0h	R0W	The read value is always 0b. DL ULPS Exit Writing 1b to this bit issues a request for all enabled Data Lanes to exit the Ultra-Low Power State (ULPS). Writing 1b while enabled Data Lanes are not in ULPS is prohibited. Writing 0b has no effect.
28	DLULPSEN T	0h	R0W	The read value is always 0b. DL ULPS Enter Writing 1b to this bit issues a request for enabled Data Lanes to enter the Ultra- Low Power State (ULPS). Writing 1b while the enabled Data Lanes are in ULPS is prohibited. Writing 0b has no effect.
27, 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25	CLULPSEX T	0h	R0W	The read value is always 0b. CL ULPS Exit Writing 1b to this bit issues a request for Clock Lane to exit the Ultra-Low Power State (ULPS). Writing 1b while Clock Lane is not in ULPS is prohibited. Writing 0b has no effect.
24	CLULPSEN T	0h	R0W	The read value is always 0b. CL ULPS Enter Writing 1b to this bit issues a request for Clock Lane to enter the Ultra-Low Power State (ULPS). Before setting this field, HSCLKSETR.HSCLKRUN must be set to 0b. Writing 1b while Clock Lane is in ULPS is prohibited. Writing 0b has no effect.
23 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.7 Reset Control Register (DSI_LINK_RSTCR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 0110h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FCETX STP
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	RW	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SWRST
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
23 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	FCETXSTP	0h	RW	Force Tx Stop Mode When this bit is 1b, change to Transmit mode and Stop State is forced by ForceTxStopmode on all PPI Data Lanes. Setting this bit to 1b when SWRST = 0 is prohibited.
15 to 1	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
0	SWRST	0h	RW	Request for Software Reset Set this bit to 1b to request a software reset. Then, wait until you confirm the completion of the Reset procedure by RSTSR.SWRSTx = 1. And set to 0b after that.

9.5.2.2.8 Reset Status Register (DSI_LINK_RSTSR)

Access Size : 8, 16, 32 bits

Address : <DSI_LINK_base> + 0114h

Initial Value : 0000_0F00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DL0DIR	-	-	-	DLSTPST[3:0]				-	-	-	SWRS TV1	SWRS TIB	SWRS TAPB	SWRS TLP	SWRS THS
Initial Value	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15	DL0DIR	0h	R	Direction (Tx or Rx) of PPI Data Lane 0 0b: PHY is in Tx Mode 1b: PHY is in Rx Mode This field is a copy of PLSR.DL0DIR.
14 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 8	DLSTPST[3:0]	Fh	R	Status of Stop State on Data Lane Bit 3: Stop State of Data Lane 3 Bit 2: Stop State of Data Lane 2 Bit 1: Stop State of Data Lane 1 Bit 0: Stop State of Data Lane 0 This field is a copy of PLSR.DLSTPST.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	SWRSTV1	0h	R	Status of Software Reset by RSTCR.SWRST at DSI_0_vclk1 0b: Reset procedure is not underway at DSI_0_vclk1. 1b: Reset procedure is underway DSI_0_vclk1.
3	SWRSTIB	0h	R	Status of Software Reset by RSTCR.SWRST at DSI_0_aclk 0b: Reset procedure is not underway at DSI_0_aclk. 1b: Reset procedure is underway at DSI_0_aclk.
2	SWRSTAPB	0h	R	Status of Software Reset by RSTCR.SWRST at DSI_0_pclk 0b: Reset procedure is not underway at DSI_0_pclk. 1b: Reset procedure is underway at DSI_0_pclk.
1	SWRSTLP	0h	R	Status of Software Reset by RSTCR.SWRST at DSI_0_lpcik 0b: Reset procedure is not underway at DSI_0_lpcik. 1b: Reset procedure is underway at DSI_0_lpcik.
0	SWRSTHS	0h	R	Status of Software Reset by RSTCR.SWRST at hscik 0b: Reset procedure is not underway at hscik. 1b: Reset procedure is underway at hscik.

9.5.2.2.9 DSI Set Register (DSI_LINK_DSISETR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0120h

Initial Value : 80F1_0001h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EOTPEN	TEDIR	SCREN	-	-	-	-	-	CRCEN[3:0]				-	-	-	ECCEN
Initial Value	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1
R/W	RW	RW	RW	R	R	R	R	R	RW	RW	RW	RW	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MRPSZ[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	EOTPEN	1h	RW	HS Transfer EoTp Enable 0b: Transfer of EoTp is disabled. 1b: Transfer of EoTp is enabled. EoTp is always disabled in LP Transfer.
30	TEDIR	0h	RW	Tearing Effect Input Direction External Tearing Effect Setting 0b: Rise edge detection. 1b: Fall edge detection.
29	SCREN	0h	RW	Scramble Enable 0b: Scrambling data is disabled. 1b: Scrambling data is enabled. This bit should not be set to "1" if the peer-device does not support a data scrambling function.
28 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 20	CRCEN[3:0]	Fh	RW	Enable Received CRC check for Virtual Channel Bit 3: Enable Received CRC check for Virtual Channel 3 (0b: Disabled, 1b: Enabled). Bit 2: Enable Received CRC check for Virtual Channel 2 (0b: Disabled, 1b: Enabled). Bit 1: Enable Received CRC check for Virtual Channel 1 (0b: Disabled, 1b: Enabled). Bit 0: Enable Received CRC check for Virtual Channel 0 (0b: Disabled, 1b: Enabled).
19 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16	ECCEN	1h	RW	Enable Received ECC Check 0b: Received ECC check is disabled. 1b: Received ECC check is enabled.
15 to 0	MRPSZ[15:0]	1h	RW	Maximum Return Packet Size If the returned Long Packet's WC is over this value, the DSI-Tx module asserts the RXSR.MAXRPSZERR error. In that case, data will not be saved to the memory space. Setting this field to 0b is prohibited. Set this field to a value greater than or equal to the peripheral setting.

9.5.2.2.10 Receive Buffer Size Register (DSI_LINK_RXBUFSZR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0124h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	RXBUFSZ[3:0]			-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23 to 20	RXBUFSZ[3:0]	0h	R	Receive Buffer Size 0h: 128 B Buffer size = 128 * 2 RXBUFSZ
19 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.11 Tx Packet Payload Data 0 Register (DSI_LINK_TXPPD0R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0160h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	RW	Tx Packet Payload Data 0 When Sequence Operation Long Packet is set (SQCHxDSCyAR.FMT = 1) and the Packet Payload Data register is selected (SQCHxDSCyBR.DTSEL = 0h) for payload data access, the values in this register are used for Long Packet Payload. [7:0]: Data 0 [15:8]: Data 1 [23:16]: Data 2 [31:24]: Data 3

9.5.2.2.12 Tx Packet Payload Data 1 Register (DSI_LINK_TXPPD1R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0164h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	RW	Tx Packet Payload Data 1 When Sequence Operation Long Packet is set (SQCHxDSCyAR.FMT = 1) and the Packet Payload Data register is selected (SQCHxDSCyBR.DTSEL = 0h) for payload data access, the values in this register are used for Long Packet Payload. [7:0]: Data 4 [15:8]: Data 5 [23:16]: Data 6 [31:24]: Data 7

9.5.2.2.13 Tx Packet Payload Data 2 Register (DSI_LINK_TXPPD2R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0168h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	RW	Tx Packet Payload Data 2 When Sequence Operation Long Packet is set (SQCHxDSCyAR.FMT = 1) and the Packet Payload Data register is selected (SQCHxDSCyBR.DTSEL = 0h) for payload data access, the values in this register are used for Long Packet Payload. [7:0]: Data 8 [15:8]: Data 9 [23:16]: Data 10 [31:24]: Data 11

9.5.2.2.14 Tx Packet Payload Data 3 Register (DSI_LINK_TXPPD3R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 016Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	RW	Tx Packet Payload Data 3 When Sequence Operation Long Packet is set (SQCHxDSCyAR.FMT = 1) and the Packet Payload Data register is selected (SQCHxDSCyBR.DTSEL = 0h) for payload data access, the values in this register are used for Long Packet Payload. [7:0]: Data 12 [15:8]: Data 13 [23:16]: Data 14 [31:24]: Data 15

9.5.2.2.15 Rx Status Register (DSI_LINK_RXSR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0200h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	-	ECCERR1B	-	MAXRPSZERR	NORETERR	PRESPTOERR	RXOVFERR	IBERR	CRCERR	WCERR	-	UEXPKTERR	ECCERR	MLFER
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXTTED	RXACK	RXTE	RXRTRG	RXUK5TRG	RXEOTP	-	RXRESP	-	-	-	-	-	TATO	LRXHTO	BTAREQND
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	R	Rx Acknowledge and Error Report When this value is 1b, the Acknowledge and Error Report Packet is received. Write 1b to the RXSCR.RXAKE register to clear this field.
29	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	ECCERR1B	0h	R	ECC Error (1 bit) When this value is 1b, the Rx ECC Error (1 bit) is detected. Write 1b to the RXSCR.ECCERR1B register to clear this field.
27	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26	MAXRPSZERR	0h	R	Maximum Return Packet Size Error When this value is 1b, the WC value of Rx Long Packet is larger than DSISETR.MRPSZ. Write 1b to the RXSCR.MAXRPSZERR register to clear this field.
25	NORETERR	0h	R	Nothing Return Error When this value is 1b, no trigger or packet is returned in BTA period. Write 1b to the RXSCR.NORETERR register to clear this field.
24	PRESPTOERR	0h	R	Peripheral Response Timeout Error When this value is 1b, the Rx Peripheral Response Timeout is detected after the transition to Rx mode along with Bus Turn-Around and within the time specified by PRESPTOBTASETR.PRESPTOBTAT, PRESPTOLPSETR.PRESPTOLPR, PRESPTOLPSETR.PRESPTOLPW, PRESPTOLPSETR.PRESPTOHSR or PRESPTOLPSETR.PRESPTOHSW. The value is selected from kind of BTA. Write 1b to the RXSCR.PRESPTOERR register to clear this field.
23	RXOVFERR	0h	R	Receive Buffer Overflow Error When this value is 1b, the Buffer Overflow Error is detected at Rx Long Packet. Write 1b to the RXSCR.RXOVFERR register to clear this field.
22	IBERR	0h	R	Internal Bus Error When this value is 1b, the Internal Bus Write had failed to respond. Write 1b to the RXSCR.IBERR register to clear this field.
21	CRCERR	0h	R	CRC Error When this value is 1b, the Rx CRC Error is detected. Write 1b to the RXSCR.CRCERR register to clear this field.
20	WCERR	0h	R	Word Count Error When this value is 1b, the Rx packet's actual WC length is shorter than the packet header's WC. Write 1b to the RXSCR.WCERR register to clear this field.
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18	UEXPKTERR	0h	R	Unexpected Packet Error When this value is 1b, an unexpected packet is received. "Unexpected Packet" means unexpected DT for receive or an unexpected response for receive. Write 1b to the RXSCR.UEXPKTERR register to clear this field.

Bit	Bit Name	Initial Value	R/W	Description
17	ECCERR	0h	R	ECC Error (over 1 bit) When this value is 1b, the Rx ECC Error (over 1 bit) is detected. Write 1b to the RXSCR.ECCERR register to clear this field.
16	MLFERR	0h	R	Malform Error When this value is 1b, the Rx packet shorter than 4 bytes is received. Write 1b to the RXSCR.MLFERR register to clear this field.
15	EXTTEDET	0h	R	External Tearing Effect Detect When this value is 1b, External Tearing Effect is detected. Write 1b to the RXSCR.EXTTEDET register to clear this field.
14	RXACK	0h	R	Rx ACK Trigger When this value is 1b, the ACK trigger is received. Write 1b to the RXSCR.RXACK register to clear this field.
13	RXTE	0h	R	Rx Tearing Effect Trigger When this value is 1b, Tearing Effect Trigger is received. Write 1b to the RXSCR.RXTE register to clear this field.
12	RXRTRG	0h	R	Rx Reset Trigger When this value is 1b, Reset Trigger is received. Normally, the DSI-Tx module does not receive Reset Trigger. Write 1b to the RXSCR.RXRTRG register to clear this field.
11	RXUK5TRG	0h	R	Rx Unknown-5 Trigger When this value is 1b, Unknown-5 Trigger is received. Normally, the DSI-Tx module does not receive Unknown-5 Trigger. Write 1b to the RXSCR.RXUK5TRG register to clear this field.
10	RXEOTP	0h	R	Rx EoTp When this value is 1b, the EoTp is received. Write 1b to the RXSCR.RXEOTP register to clear this field.
9	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	RXRESP	0h	R	Rx Response Packet When this value is 1b, the Response Packet is received. Write 1b to the RXSCR.RXRESP register to clear this field.
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	R	Turnaround Acknowledge Timeout When this value is 1b, the Turnaround Acknowledge Timeout (TA_TO) is detected. Write 1b to the RXSCR.TATO register to clear this field.
1	LRXHTO	0h	R	LP-RX Host Processor Timeout When this value is 1b, the LP-RX Host Processor Timeout (LRX-H_TO) is detected. Write 1b to RXSCR.LRXHTO register to clear this field.
0	BTAREQEN D	0h	R	BTA Request End When this value is 1b, the completion of the transmission request which includes Rx is detected. This bit is set if all BTA requests are finished including error. The BTA request status is indicated by RXSR[31:1]. Write 1b to the RXSCR.BTAREQEND register to clear this field.

9.5.2.2.16 Rx Status Clear Register (DSI_LINK_RXSCR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0204h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	-	ECCERR1B	-	MAXRPSZERR	NORETERR	PRESPTOERR	RXOVFERR	IBERR	CRCERR	WCERR	-	UEXPKTERR	ECCERR	MLFER
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R0W	R	R0W	R	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R	R0W	R0W	R0W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXTTED	RXACK	RXTE	RXRTRG	RXUK5TRG	RXEOTP	-	RXRESP	-	-	-	-	-	TATO	LRXHTO	BTAREQEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0W	R0W	R0W	R0W	R0W	R0W	R	R0W	R	R	R	R	R	R0W	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	R0W	The read value is always 0b. RXSR.RXAKE Clear Set to 1b to clear RXSR.RXAKE. Writing 0b has no effect.
29	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	ECCERR1B	0h	R0W	The read value is always 0b. RXSR.ECCERR1B Clear Set to 1b to clear RXSR.ECCERR1B. Writing 0b has no effect.
27	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26	MAXRPSZERR	0h	R0W	The read value is always 0b. RXSR.MAXRPSZERR Clear Set to 1b to clear RXSR.MAXRPSZERR. Writing 0b has no effect.
25	NORETERR	0h	R0W	The read value is always 0b. RXSR.NORETERR Clear Set to 1b to clear RXSR.NORETERR. Writing 0b has no effect.
24	PRESPTOERR	0h	R0W	The read value is always 0b. RXSR.PRESPTOERR Clear Set to 1b to clear RXSR.PRESPTOERR. Writing 0b has no effect.
23	RXOVFERR	0h	R0W	The read value is always 0b. RXSR.RXOVFERR Clear Set to 1b to clear RXSR.RXOVFERR. Writing 0b has no effect.
22	IBERR	0h	R0W	The read value is always 0b. RXSR.IBERR Clear Set to 1b to clear RXSR.IBERR. Writing 0b has no effect.
21	CRCERR	0h	R0W	The read value is always 0b. RXSR.CRCERR Clear Set to 1b to clear RXSR.CRCERR. Writing 0b has no effect.
20	WCERR	0h	R0W	The read value is always 0b. RXSR.WCERR Clear Set to 1b to clear RXSR.WCERR. Writing 0b has no effect.
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
18	UEXPKTERR R	0h	R0W	The read value is always 0b. RXSR.UEXPKTERR Clear Set to 1b to clear RXSR.UEXPKTERR. Writing 0b has no effect.
17	ECCERR	0h	R0W	The read value is always 0b. RXSR.ECCERR Clear Set to 1b to clear RXSR.ECCERR. Writing 0b has no effect.
16	MLFERR	0h	R0W	The read value is always 0b. RXSR.MLFERR Clear Set to 1b to clear RXSR.MLFERR. Writing 0b has no effect.
15	EXTTDEDET	0h	R0W	The read value is always 0b. RXSR.EXTTDEDET Clear Set to 1b to clear RXSR.EXTTDEDET. Writing 0b has no effect.
14	RXACK	0h	R0W	The read value is always 0b. RXSR.RXACK Clear Set to 1b to clear RXSR.RXACK. Writing 0b has no effect.
13	RXTE	0h	R0W	The read value is always 0b. RXSR.RXTE Clear Set to 1b to clear RXSR.RXTE. Writing 0b has no effect.
12	RXRTRG	0h	R0W	The read value is always 0b. RXSR.RXRTRG Clear Set to 1b to clear RXSR.RXRTRG. Writing 0b has no effect.
11	RXUK5TRG	0h	R0W	The read value is always 0b. RXSR.RXUK5TRG Clear Set to 1b to clear RXSR.RXUK5TRG. Writing 0b has no effect.
10	RXEOTP	0h	R0W	The read value is always 0b. RXSR.RXEOTP Clear Set to 1b to clear RXSR.RXEOTP. Writing 0b has no effect.
9	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	RXRESP	0h	R0W	The read value is always 0b. RXSR.RXRESP Clear Set to 1b to clear RXSR.RXRESP. Writing 0b has no effect.
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	R0W	The read value is always 0b. RXSR.TATO Clear Set to 1b to clear RXSR.TATO. Writing 0b has no effect.
1	LRXHTO	0h	R0W	The read value is always 0b. RXSR.LRXHTO Clear Set to 1b to clear RXSR.LRXHTO. Writing 0b has no effect.
0	BTAREQEN D	0h	R0W	The read value is always 0b. RXSR.BTAREQEND Clear Set to 1b to clear RXSR.BTAREQEND. Writing 0b has no effect.

9.5.2.2.17 Rx Interrupt Enable Register (DSI_LINK_RXIER)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0208h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	-	ECCERR1B	-	MAXRPSZERR	NORETERR	PRESPTOERR	RXOVFERR	IBERR	CRCERR	WCERR	-	UEXPKTERR	ECCERR	MLFERR
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	R	RW	R	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXTTEDET	RXACK	RXTE	RXRTRG	RXUK5TRG	RXEOTP	-	RXRESP	-	-	-	-	-	TATO	LRXHTO	BTAREQEND
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	R	RW	R	R	R	R	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	RW	Interrupt enable for RXSR.RXAKE 0b: Disables interrupt dsi_int_rcv when RXSR.RXAKE = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXAKE = 1.
29	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	ECCERR1B	0h	RW	Interrupt enable for RXSR.ECCERR1B 0b: Disables interrupt dsi_int_rcv when RXSR.ECCERR1B = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.ECCERR1B = 1.
27	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
26	MAXRPSZERR	0h	RW	Interrupt enable for RXSR.MAXRPSZERR 0b: Disables interrupt dsi_int_rcv when RXSR.MAXRPSZERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.MAXRPSZERR = 1.
25	NORETERR	0h	RW	Interrupt enable for RXSR.NORETERR 0b: Disables interrupt dsi_int_rcv when RXSR.NORETERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.NORETERR = 1.
24	PRESPTOERR	0h	RW	Interrupt enable for RXSR.PRESPTOERR 0b: Disables interrupt dsi_int_rcv when RXSR.PRESPTOERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.PRESPTOERR = 1.
23	RXOVFERR	0h	RW	Interrupt enable for RXSR.RXOVFERR 0b: Disables interrupt dsi_int_rcv when RXSR.RXOVFERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXOVFERR = 1.
22	IBERR	0h	RW	Interrupt enable for RXSR.IBERR 0b: Disables interrupt dsi_int_rcv when RXSR.IBERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.IBERR = 1.
21	CRCERR	0h	RW	Interrupt enable for RXSR.CRCERR 0b: Disables interrupt dsi_int_rcv when RXSR.CRCERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.CRCERR = 1.
20	WCERR	0h	RW	Interrupt enable for RXSR.WCERR 0b: Disables interrupt dsi_int_rcv when RXSR.WCERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.WCERR = 1.
19	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
18	UEXPKTERR	0h	RW	Interrupt enable for RXSR.UEXPKTERR 0b: Disables interrupt dsi_int_rcv when RXSR.UEXPKTERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.UEXPKTERR = 1.
17	ECCERR	0h	RW	Interrupt enable for RXSR.ECCERR 0b: Disables interrupt dsi_int_rcv when RXSR.ECCERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.ECCERR = 1.
16	MLFERR	0h	RW	Interrupt enable for RXSR.MLFERR 0b: Disables interrupt dsi_int_rcv when RXSR.MLFERR = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.MLFERR = 1.

Bit	Bit Name	Initial Value	R/W	Description
15	EXTTEDET	0h	RW	Interrupt enable for RXSR.EXTTEDET 0b: Disables interrupt dsi_int_rcv when RXSR.EXTTEDET = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.EXTTEDET = 1.
14	RXACK	0h	RW	Interrupt enable for RXSR.RXACK 0b: Disables interrupt dsi_int_rcv when RXSR.RXACK = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXACK = 1.
13	RXTE	0h	RW	Interrupt enable for RXSR.RXTE 0b: Disables interrupt dsi_int_rcv when RXSR.RXTE = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXTE = 1.
12	RXRTRG	0h	RW	Interrupt enable for RXSR.RXRTRG 0b: Disables interrupt dsi_int_rcv when RXSR.RXRTRG = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXRTRG = 1.
11	RXUK5TRG	0h	RW	Interrupt enable for RXSR.RXUK5TRG 0b: Disables interrupt dsi_int_rcv when RXSR.RXUK5TRG = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXUK5TRG = 1.
10	RXEOTP	0h	RW	Interrupt enable for RXSR.RXEOTP 0b: Disables interrupt dsi_int_rcv when RXSR.RXEOTP = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXEOTP = 1.
9	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	RXRESP	0h	RW	Interrupt enable for RXSR.RXRESP 0b: Disables interrupt dsi_int_rcv when RXSR.RXRESP = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.RXRESP = 1.
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	RW	Interrupt enable for RXSR.TATO 0b: Disables interrupt dsi_int_rcv when RXSR.TATO = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.TATO = 1.
1	LRXHTO	0h	RW	Interrupt enable for RXSR.LRXHTO 0b: Disables interrupt dsi_int_rcv when RXSR.LRXHTO = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.LRXHTO = 1.
0	BTAREQEND	0h	RW	Interrupt enable for RXSR.BTAREQEND 0b: Disables interrupt dsi_int_rcv when RXSR.BTAREQEND = 1. 1b: Enables interrupt dsi_int_rcv when RXSR.BTAREQEND = 1.

9.5.2.2.18 Peripheral Response Timeout BTA Set Register (DSI_LINK_PRESPTOBTASETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0210h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESPTOBTA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRESPTOBTA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

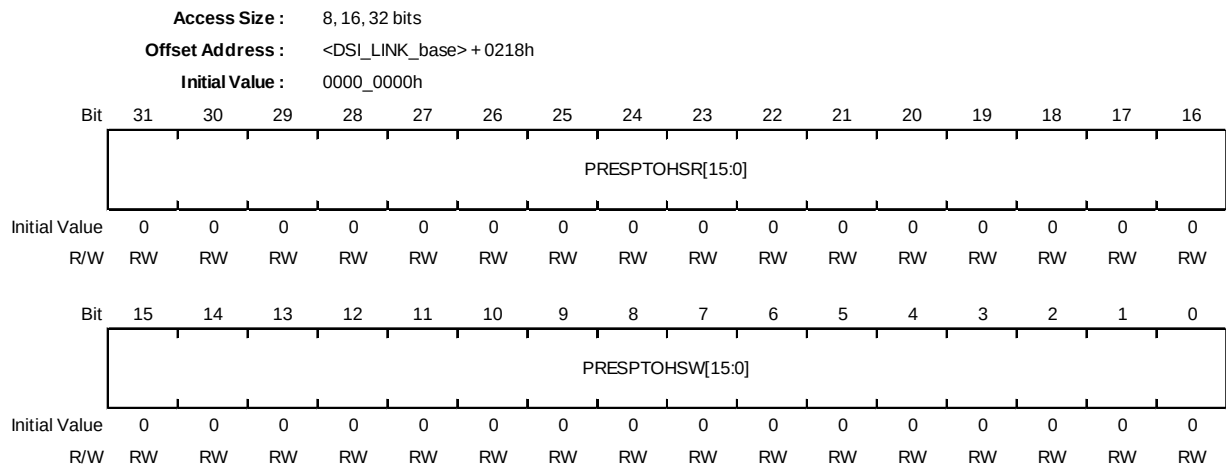
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	PRESPTOBTA [31:0]	0h	RW	Peripheral Response Timeout (Bus Turn Around) Count Set the timeout value to measure the interval between the start of Rx mode and the start of packet reception along with Bus Turn-Around while SQCHxDSCyAR.BTA = 3h. The limit time for timeout is calculated by the formula of PRESPTOBTA * (the period of DSI_0_lpcclk). When RXSR.PRESPTOERR is 1b, a timeout is detected. Note that no timeout is detected when PRESPTOBTA = 0. Since the counting clock is asynchronous with packet reception, there is some jitter in the timeout.

9.5.2.2.19 Peripheral Response Timeout LP Set Register (DSI_LINK_PRESPOLPSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0214h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESPTOLPR[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRESPTOLPW[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	PRESPTOLPR [15:0]	0h	RW	Peripheral Response Timeout (LPDT READ Request) Count Set the timeout value to measure the interval between the start of Rx mode and the start of packet reception along with Bus Turn-Around while SQCHxDSCyAR.BTA = 2h and SQCHxDSCyAR.SPD = 1b. The limit time for timeout is calculated by the formula of PRESPTOLPR * (the period of DSI_0_lpcclk). When RXSR.PRESPTOERR is 1b, a timeout is detected. Note that no timeout is detected when PRESPTOLPR = 0. Since the counting clock is asynchronous with packet reception, there is some jitter in the timeout.
15 to 0	PRESPTOLPW [15:0]	0h	RW	Peripheral Response Timeout (LPDT WRITE Request) Count Set the timeout value to measure the interval between the start of Rx mode and the start of packet reception along with Bus Turn-Around while SQCHxDSCyAR.BTA = 01b and SQCHxDSCyAR.SPD = 1b. The limit time for timeout is calculated by the formula of PRESPTOLPW * (the period of DSI_0_lpcclk). When RXSR.PRESPTOERR is 1b, a timeout is detected. Note that no timeout is detected when PRESPTOLPW = 0. Since the counting clock is asynchronous with packet reception, there is some jitter in the timeout.

9.5.2.2.20 Peripheral Response Timeout HS Set Register (DSI_LINK_PRESPTOHSSETR)



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	PRESPTOHSR [15:0]	0h	RW	Peripheral Response Timeout (HS READ Request) Count Set the timeout value to measure the interval between the start of Rx mode and the start of packet reception along with Bus Turn-Around while SQCHxDSCyAR.BTA = 2h and SQCHxDSCyAR.SPD = 0b. The limit time for timeout is calculated by the formula of PRESPTOHSR * (the period of DSI_0_lpcclk). When RXSR.PRESPTOERR is 1b, a timeout is detected. Note that no timeout is detected when PRESPTOHSR = 0. Since the counting clock is asynchronous with packet reception, there is some jitter in the timeout.
15 to 0	PRESPTOHSW [15:0]	0h	RW	Peripheral Response Timeout (HS WRITE Request) Count Set the timeout value to measure the interval between the start of Rx mode and the start of packet reception along with Bus Turn-Around while SQCHxDSCyAR.BTA = 1h and SQCHxDSCyAR.SPD = 0b. The limit time for timeout is calculated by the formula of PRESPTOHSW * (the period of DSI_0_lpcclk). When RXSR.PRESPTOERR is 1b, a timeout is detected. Note that no timeout is detected when PRESPTOHSW = 0. Since the counting clock is asynchronous with packet reception, there is some jitter in the timeout.

9.5.2.2.21 Acknowledge and Error Report Packet Parameter Latest Info Register (DSI_LINK_AKEPLATIR)

Access Size :		8, 16, 32 bits															
Offset Address :		<DSI_LINK_base> + 0220h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	VC3	VC2	VC1	VC0	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	ERRRPTLAT[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	VC3	0h	R	Virtual Channel Identifier 3 Latest This field indicates the Virtual Channel identifier of a received Acknowledge and Error Report Packet. 0b: Virtual Channel identifier is not 3. 1b: Virtual Channel identifier is 3. This field indicates the latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated.
18	VC2	0h	R	Virtual Channel Identifier 2 Latest This field indicates the Virtual Channel identifier of a received Acknowledge and Error Report Packet. 0b: Virtual Channel identifier is not 2. 1b: Virtual Channel identifier is 2. This field indicates the latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated.
17	VC1	0h	R	Virtual Channel Identifier 1 Latest This field indicates the Virtual Channel identifier of a received Acknowledge and Error Report Packet. 0b: Virtual Channel identifier is not 1. 1b: Virtual Channel identifier is 1. This field indicates the latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated.
16	VC0	0h	R	Virtual Channel Identifier 0 Latest This field indicates the Virtual Channel identifier of a received Acknowledge and Error Report Packet. 0b: Virtual Channel identifier is not 0. 1b: Virtual Channel identifier is 0. This field indicates the latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated.
15 to 0	ERRRPTLAT [15:0]	0h	R	Error Report Latest This field indicates the Error Report of a received Acknowledge and Error Report Packet. This field indicates the latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated.

9.5.2.2.22 Acknowledge and Error Report Packet Parameter Accumulate Status Register (DSI_LINK_AKEPACMSR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0224h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	VC3	VC2	VC1	VC0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERRRPTACM[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	VC3	0h	R	Virtual Channel Identifier 3 Accumulate This field is set to 1b when an Acknowledge and Error Report Packet whose Virtual Channel identifier = 3 is received. This field indicates accumulated information. This bit is cleared by AKEPSCR.VC3.
18	VC2	0h	R	Virtual Channel Identifier 2 Accumulate This field is set to 1b when an Acknowledge and Error Report Packet whose Virtual Channel identifier = 2 is received. This field indicates accumulated information. This bit is cleared by AKEPSCR.VC2.
17	VC1	0h	R	Virtual Channel Identifier 1 Accumulate This field is set to 1b when an Acknowledge and Error Report Packet whose Virtual Channel identifier = 1 is received. This field indicates accumulated information. This bit is cleared by AKEPSCR.VC1.
16	VC0	0h	R	Virtual Channel Identifier 0 Accumulate This field is set to 1b when an Acknowledge and Error Report Packet whose Virtual Channel identifier = 0 is received. This field indicates accumulated information. This bit is cleared by AKEPSCR.VC0.
15 to 0	ERRRPTACM [15:0]	0h	R	Error Report Accumulate Each bit of this field is set to 1b, when an Acknowledge and Error Report Packet whose each Error Report bit = 1 is received. This field indicates accumulated information. Each bit is cleared by the corresponding AKESCR.ERRRPTACM.

9.5.2.2.23 Acknowledge and Error Report Packet Parameter Status Clear Register (DSI_LINK_AKEPSCR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0228h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	VC3	VC2	VC1	VC0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R0W	R0W	R0W	R0W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERRRPTACM[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	VC3	0h	R0W	The read value is always 0b. AKEPACMSR.VC3 Clear Set to 1b to clear AKEPACMSR.VC3. Writing 0b has no effect.
18	VC2	0h	R0W	The read value is always 0b. AKEPACMSR.VC2 Clear Set to 1b to clear AKEPACMSR.VC2. Writing 0b has no effect.
17	VC1	0h	R0W	The read value is always 0b. AKEPACMSR.VC1 Clear Set to 1b to clear AKEPACMSR.VC1. Writing 0b has no effect.
16	VC0	0h	R0W	The read value is always 0b. AKEPACMSR.VC0 Clear Set to 1b to clear AKEPACMSR.VC0. Writing 0b has no effect.
15 to 0	ERRRPTACM [15:0]	0h	R0W	The read value is always 0b. Error Report Accumulate Clear Set to 1b to clear each AKEPACMSR.ERRRPTACM. Writing 0b has no effect.

9.5.2.2.24 Rx Result Saved Status Register (DSI_LINK_RXRSSR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0230h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SLT3V LD	SLT2V LD	SLT1V LD	SLT0V LD
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	SLT3VLD	0h	R	Slot 3 Valid Response packet is received, or information is saved to RXRSS3R. Write 1b to RXRSSCR.SLT3C to clear this bit. This bit and RXRSS3R are set if the next receive action is caused with SQCHxDSCyCR.ACTCODE = 03 again.
2	SLT2VLD	0h	R	Slot 2 Valid Response packet is received, or information is saved to RXRSS2R. Write 1b to RXRSSCR.SLT2C to clear this bit. This bit and RXRSS2R are set if the next receive action is caused with SQCHxDSCyCR.ACTCODE = 02 again.
1	SLT1VLD	0h	R	Slot 1 Valid Response packet is received, or information is saved to RXRSS1R. Write 1b to RXRSSCR.SLT1C to clear this bit. This bit and RXRSS1R are set if the next receive action is caused with SQCHxDSCyCR.ACTCODE = 01 again.
0	SLT0VLD	0h	R	Slot 0 Valid Response packet is received, or information is saved to RXRSS0R. Write 1b to RXRSSCR.SLT0C to clear this bit. This bit and RXRSS0R are set if the next receive action is caused with SQCHxDSCyCR.ACTCODE = 00 again.

9.5.2.2.25 Rx Result Saved Status Clear Register (DSI_LINK_RXRSSCR)

Access Size :		8, 16, 32 bits															
Offset Address :		<DSI_LINK_base> + 0234h															
Initial Value :		0000_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	SLT3C	SLT2C	SLT1C	SLT0C	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R0W	R0W	R0W	R0W	

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	SLT3C	0h	R0W	The read value is always 0b. RXRSSR.SLT3VLD Clear Write 1b to clear RXRSSR.SLT3VLD. Writing 0b has no effect.
2	SLT2C	0h	R0W	The read value is always 0b. RXRSSR.SLT2VLD Clear Write 1b to clear RXRSSR.SLT2VLD. Writing 0b has no effect.
1	SLT1C	0h	R0W	The read value is always 0b. RXRSSR.SLT1VLD Clear Write 1b to clear RXRSSR.SLT1VLD. Writing 0b has no effect.
0	SLT0C	0h	R0W	The read value is always 0b. RXRSSR.SLT0VLD Clear Write 1b to clear RXRSSR.SLT0VLD. Writing 0b has no effect.

9.5.2.2.26 Rx Result Info Overwrite Status Register (DSI_LINK_RXRINFOOWSR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0238h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SLT3IN FOO W	SLT2IN FOO W	SLT1IN FOO W	SLT0IN FOO W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	SLT3INFOOW	0h	R	Slot 3 Information Overwrite This bit is a copy of RXRSS3R.INFOOW.
2	SLT2INFOOW	0h	R	Slot 2 Information Overwrite This bit is a copy of RXRSS2R.INFOOW.
1	SLT1INFOOW	0h	R	Slot 1 Information Overwrite This bit is a copy of RXRSS1R.INFOOW.
0	SLT0INFOOW	0h	R	Slot 0 Information Overwrite This bit is a copy of RXRSS0R.INFOOW.

9.5.2.2.27 Rx Result Info Overwrite Status Clear Register (DSI_LINK_RXRINFOOWSCR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 023Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SLT3IN FOOW C	SLT2IN FOOW C	SLT1IN FOOW C	SLT0IN FOOW C
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R0W	R0W	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	SLT3INFOOWC	0h	R0W	The read value is always 0b. RXRSS3R.INFOOW Clear Write 1b to clear RXRSS3R.INFOOW. Writing 0b has no effect. To prevent missing of RXRSS3R.INFOOW, clear RXRSS3R.INFOOW before clearing RXRSSR.SLT3VLD.
2	SLT2INFOOWC	0h	R0W	The read value is always 0b. RXRSS2R.INFOOW Clear Write 1b to clear RXRSS2R.INFOOW. Writing 0b has no effect. To prevent missing of RXRSS2R.INFOOW, clear RXRSS2R.INFOOW before clearing RXRSSR.SLT2VLD.
1	SLT1INFOOWC	0h	R0W	The read value is always 0b. RXRSS1R.INFOOW Clear Write 1b to clear RXRSS1R.INFOOW. Writing 0b has no effect. To prevent missing of RXRSS1R.INFOOW, clear RXRSS1R.INFOOW before clearing RXRSSR.SLT1VLD.
0	SLT0INFOOWC	0h	R0W	The read value is always 0b. RXRSS0R.INFOOW Clear Write 1b to clear RXRSS0R.INFOOW. Writing 0b has no effect. To prevent missing of RXRSS0R.INFOOW, clear RXRSS0R.INFOOW before clearing RXRSSR.SLT0VLD.

9.5.2.2.28 Rx Result Save Slot x Register (DSI_LINK_RXRSSxR) (x = 0, 1, 2, 3)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0240h + 0004h × (x)

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INFOOW	RXAKE	RXCORERR	RXPKTDFAIL	RXFAIL	RXFATALERR	RXSUC	FMT	VC[1:0]		DT[5:0]					
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA1[7:0]								DATA0[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	INFOOW	0h	R	Information Overwrite This bit is set if the information of RXRSSxR is updated when RXRSSR.SLTxVLD is 1b. Write 1b to RXRINFOOWSCR.SLTxINFOOWC to clear this bit.
30	RXAKE	0h	R	Receive Acknowledge and Error Report Packet RXSR.RXAKE is also reported.
29	RXCORERR	0h	R	Receive Correctable Error The received packet has a correctable error. RXSR.ECCERR1B is also reported.
28	RXPKTDFAIL	0h	R	Receive Packet Data Fail The packet header was saved correctly but data were not. One or more of the following bits are also reported. RXSR.CRCERR, RXSR.WCERR, RXSR.MAXRPSZERR, RXSR.UEXPKTERR, RXSR.RXOVFERR and RXSR.IBERR. When both of this bit and RXRESP are asserted, it means a redundant packet was received in BTA.
27	RXFAIL	0h	R	Receive Fail Expected reception did not proceed. One or more of the following bits are also reported. RXSR.PRESPTOERR, RXSR.ECCERR, RXSR.MLFERR and RXSR.NORETERR.
26	RXFATALERR	0h	R	Fatal Error A fatal timeout occurred during BTA. One or more of the following bits are also reported. FERRSR.TATO and FERRSR.LRXHTO.
25	RXSUC	0h	R	Receive Success When this value is 1b, the Response Packet or ACK trigger is received. One or more of the following bits are also reported. RXSR.RXRESP and RXSR.RXACK.
24	FMT	0h	R	Packet format of Rx Packet Header 0b: Short Packet 1b: Long Packet This field is valid when RXPKTDFAIL = 1 or (RXSUC = 1 and DT != 0).
23,22	VC[1:0]	0h	R	Identifier of Virtual Channel of Rx Packet Header This field is valid when (RXSUC = 1 and DT != 0).
21 to 16	DT[5:0]	0h	R	Data Type of Rx Packet Header 00h: ACK trigger received 01h to 3Fh: Data Type This field is valid when RXSUC = 1.
15 to 8	DATA1[7:0]	0h	R	Data1 of Rx Packet Header When the Rx packet format is Long Packet, this value is the upper 8 bits of the word count. This field is valid when (RXSUC = 1 and DT != 0).
7 to 0	DATA0[7:0]	0h	R	Data0 of Rx Packet Header When Rx packet format is Long Packet, this value is the lower 8 bits of the word count. This field is valid when (RXSUC = 1 and DT != 0).

9.5.2.2.29 Rx Packet Payload Data 0 Register (DSI_LINK_RXPPD0R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 02C0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	R	Rx Packet Payload Data 0 When a Long Packet is received with SQCHxDSCyBR.DTSEL = 0h, the received packet payload is saved. [7: 0]: Data 0 [15: 8]: Data 1 [23: 16]: Data 2 [31: 24]: Data 3

9.5.2.2.30 Rx Packet Payload Data 1 Register (DSI_LINK_RXPPD1R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 02C4h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	R	Rx Packet Payload Data 1 When a Long Packet is received with SQCHxDSCyBR.DTSEL = 0h, the received packet payload is saved. [7: 0]: Data 4 [15: 8]: Data 5 [23: 16]: Data 6 [31: 24]: Data 7

9.5.2.2.31 Rx Packet Payload Data 2 Register (DSI_LINK_RXPPD2R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 02C8h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	R	Rx Packet Payload Data 2 When a Long Packet is received with SQCHxDSCyBR.DTSEL = 0h, the received packet payload is saved. [7: 0]: Data 8 [15: 8]: Data 9 [23: 16]: Data 10 [31: 24]: Data 11

9.5.2.2.32 Rx Packet Payload Data 3 Register (DSI_LINK_RXPPD3R)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 02CCh														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DATA[31:0]	0h	R	Rx Packet Payload Data 3 When a Long Packet is received with SQCHxDSCyBR.DTSEL = 0h, the received packet payload is saved. [7: 0]: Data 12 [15: 8]: Data 13 [23: 16]: Data 14 [31: 24]: Data 15

9.5.2.2.33 HSTX Timeout Set Register (DSI_LINK_HSTXTOSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 02E0h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HTXTO[31:16]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTXTO[15:0]																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name	Initial Value	R/W	Description												
31 to 0	HTXTO[31:0]	0h	RW	HS TX Timeout Count (HTX_TO) The limit time for timeout is calculated by the formula of HTXTO * (the period of high-speed serial UI * 32). When FERRSR.HTXTO is 1b, a timeout is detected. Note that no timeout is detected when HTXTO = 0.												

9.5.2.2.34 LRX-H Timeout Set Register (DSI_LINK_LRXHTOSETR)

Access Size :		8, 16, 32 bits																														
Offset Address :		<DSI_LINK_base> + 02E4h																														
Initial Value :		0000_0000h																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
LRXHTO[31:16]																																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
LRXHTO[15:0]																																
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW																
Bit	Bit Name															Initial Value	R/W	Description														
31 to 0	LRXHTO[31:0]															0h	RW	LP-RX Host Processor Timeout (LRX-H_TO) The limit time for timeout is calculated by the formula of LRXHTO * (the period of lpclk). When FERRSR.LRXHTO is 1b, a timeout is detected. Note that no timeout is detected when LRXHTO = 0.														

9.5.2.2.35 TA Timeout Set Register (DSI_LINK_TATOSETR)

Access Size :		8, 16, 32 bits																
Offset Address :		<DSI_LINK_base> + 02E8h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
TATO[31:16]																		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
TATO[15:0]																		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	Bit Name															Initial Value	R/W	Description
31 to 0	TATO[31:0]															0h	RW	Turnaround Acknowledge Timeout (TA_TO) The limit time for timeout is calculated by the formula of TATO * (the period of lpclk). When FERRSR.TATO is 1b, a timeout is detected. Note that no timeout is detected when TATO = 0.

9.5.2.2.36 Fatal Error Status Register (DSI_LINK_FERRSR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0300h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	ERRCL P1S	ERRCL POS	-	-	-	-	-	-	ERRCL P1	ERRCL P0	ERRCTL RL	ERRSY NESC	ERRESC
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	TATO	LRXHTO	HTXTO
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
28	ERRCLP1S	0h	R	Status of Contention Error of LP1 (ErrContentionLP1)
27	ERRCLPOS	0h	R	Status of Contention Error of LP0 (ErrContentionLP0)
26 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	ERRCLP1	0h	R	Contention Error of LP1 (ErrContentionLP1) When this value is 1b, the Contention Error of LP1 (ErrContentionLP1) is detected on PPI Data Lane 0. Write 1b to the FERRSCR.ERRCLP1 register to clear this field.
19	ERRCLP0	0h	R	Contention Error of LP0 (ErrContentionLP0) When this value is 1b, the Contention Error of LP0 (ErrContentionLP0) is detected on PPI Data Lane 0. Write 1b to the FERRSCR.ERRCLP0 register to clear this field.
18	ERRCTRL	0h	R	Control Error (ErrControl) When this value is 1b, the Control Error (ErrControl) is detected on PPI Data Lane 0. Write 1b to the FERRSCR.ERRCTRL register to clear this field.
17	ERRSYNESC	0h	R	Sync Error of LPDT (ErrSyncEsc) When this value is 1b, the Sync Error of LPDT (ErrSyncEsc) is detected on PPI Data Lane 0. Write 1b to the FERRSCR.ERSYNESC register to clear this field.
16	ERRESC	0h	R	Escape mode Entry Error (ErrEsc) When this value is 1b, the Escape mode Entry Error (ErrEsc) is detected on PPI Data Lane 0. Write 1b to the FERRSCR.ERRESC register to clear this field.
15 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	R	Turnaround Acknowledge Timeout (TA_TO) When this value is 1b, the Turnaround Acknowledge Timeout (TA_TO) is detected. Write 1b to the FERRSCR.TATO register to clear this field.
1	LRXHTO	0h	R	LP-RX Host Processor Timeout (LRX-H_TO) When this value is 1b, the LP-RX Host Processor Timeout (LRX-H_TO) is detected. Write 1b to the FERRSCR.LRXHTO register to clear this field.
0	HTXTO	0h	R	HS TX Timeout (HTX_TO) When this value is 1b, the HS TX Timeout (HTX_TO) is detected. Write 1b to the FERRSCR.HTXTO register to clear this field.

9.5.2.2.37 Fatal Error Status Clear Register (DSI_LINK_FERRSCR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0304h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	ERRCL P1	ERRCL P0	ERRCT RL	ERRSY NESC	ERRES C
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R0W	R0W	R0W	R0W	R0W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	TATO	LRXHT O	HTXTO
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R0W	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	ERRCLP1	0h	R0W	The read value is always 0b. FERRSR.ERRCLP1 Clear Set to 1b to clear FERRSR.ERRCLP1. Writing 0b has no effect.
19	ERRCLP0	0h	R0W	The read value is always 0. FERRSR.ERRCLP0 Clear Set to 1b to clear FERRSR.ERRCLP0. Writing 0b has no effect.
18	ERRCTRL	0h	R0W	The read value is always 0. FERRSR.ERRCTRL Clear Set to 1b to clear FERRSR.ERRCTRL. Writing 0b has no effect.
17	ERRSYNESC	0h	R0W	The read value is always 0. FERRSR.ERRSYNESC Clear Set to 1b to clear FERRSR.ERRSYNESC. Writing 0b has no effect.
16	ERRESC	0h	R0W	The read value is always 0. FERRSR.ERRESC Clear Set to 1b to clear FERRSR.ERRESC. Writing 0b has no effect.
15 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	R0W	The read value is always 0b. FERRSR.TATO Clear Set to 1b to clear FERRSR.TATO. Writing 0b has no effect.
1	LRXHTO	0h	R0W	The read value is always 0b. FERRSR.LRXHTO Clear Set to 1b to clear FERRSR.LRXHTO. Writing 0b has no effect.
0	HTXTO	0h	R0W	The read value is always 0. FERRSR.HTXTO Clear Set to 1b to clear FERRSR.HTXTO. Writing 0b has no effect.

9.5.2.2.38 Fatal Error Interrupt Enable Register (DSI_LINK_FERRIER)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0308h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	ERRCL P1	ERRCL P0	ERRCT RL	ERRSY NESC	ERRES C
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	TATO	LRXHT O	HTXTO
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	ERRCLP1	0h	RW	Interrupt enable for the Contention Error of LP1 on PPI Data Lane 0 (ErrContentionLP1) 0b: Disables interrupt dsi_int_ferr when FERRSR.ERRCLP1 = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.ERRCLP1 = 1
19	ERRCLP0	0h	RW	Interrupt enable for the Contention Error of LP0 on PPI Data Lane 0 (ErrContentionLP0) 0b: Disables interrupt dsi_int_ferr when FERRSR.ERRCLP0 = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.ERRCLP0 = 1
18	ERRCTRL	0h	RW	Interrupt enable for the Control Error on PPI Data Lane 0 (ErrControl) 0b: Disables interrupt dsi_int_ferr when FERRSR.ERRCTRL = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.ERRCTRL = 1
17	ERRSYNESC	0h	RW	Interrupt enable for the LPDT Sync Error on PPI Data Lane 0 (ErrSyncEsc) 0b: Disables interrupt dsi_int_ferr when FERRSR.ERRSYNESC = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.ERRSYNESC = 1
16	ERRESC	0h	RW	Interrupt enable for the Escape mode Error on PPI Data Lane 0 (ErrEsc) 0b: Disables interrupt dsi_int_ferr when FERRSR.ERRESC = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.ERRESC = 1
15 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	TATO	0h	RW	Interrupt enable for Turnaround Acknowledge Timeout (TA_TO) 0b: Disables interrupt dsi_int_ferr when FERRSR.TATO = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.TATO = 1
1	LRXHTO	0h	RW	Interrupt enable for LP-RX Host Processor Timeout (LRX_H_TO) 0b: Disables interrupt dsi_int_ferr when FERRSR.LRXHTO = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.LRXHTO = 1
0	HTXTO	0h	RW	Interrupt enable for HS TX Timeout (HTX_TO) 0b: Disables interrupt dsi_int_ferr when FERRSR.HTXTO = 1 1b: Enables interrupt dsi_int_ferr when FERRSR.HTXTO = 1

9.5.2.2.39 Clock Lane Stop Time Set Register (DSI_LINK_CLSTPTSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0314h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLKKPT[7:0]								CLKBFHT[7:0]							
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	CLKSTPT[9:0]										-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R
Bit	Bit Name	Initial Value	R/W	Description												
31 to 24	CLKKPT[7:0]	0h	RW	Clock Keep Time The time period over which the HS Clock Lane keeps active after HS packet transfer. The time is calculated by the formula of CLKKPT * (the period of high-speed serial UI * 32). This time corresponds to the D-PHY specification's (T_{HS-TRAIL} + T_{CLK-POST}). The value must be greater than the one calculated. For information about each parameter, see Table 9.5-4. Set an appropriate value (other than 0) before HSCLKSETR.HSCLKRUN is set to "1".												
23 to 16	CLKBFHT[7:0]	0h	RW	Clock Beforehand Time The time period over which Clock Lane is changed back to HS before HS packet transfer. If HS transfers are scheduled in LP, Clock Lane will start before HS transfer by this setting. The time is calculated by the formula of CLKBFHT * (the period of high-speed serial UI * 32). This time corresponds to the D-PHY specification's (T_{LPX}(For Clock Lane HS Entry) + T_{CLK-PREPARE} + T_{CLK-ZERO} + T_{CLK-PRE}). The value must be greater than the one calculated. For information about each parameter, see Table 9.5-4. Set an appropriate value (other than 0) before HSCLKSETR.HSCLKRUN is set to "1". The "T_{LPX}(For Clock Lane HS Entry)" means the period of LP-01 in HS entry sequence of Clock Lane. If this period is not same as normal T_{LPX} value of host device (in the range of 67% to 150% of the T_{LPX} value of peripheral device), use the actual period for this calculation.												
15 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												
11 to 2	CLKSTPT[9:0]	0h	RW	Clock Stop Time The time period over which the HS Clock Lane is changed to LP and back to HS. If Data Lanes exit HS and the time of CLKKPT expires, and the scheduled time till the next HS transfer is more than this setting, Clock Lane will be changed to LP. This setting has no meaning if HSCLKSETR.HSCLKMODE is 1b or if HSCLKSETR.HSCLKRUN is 0b. The time is calculated by the formula of CLKSTPT * (the period of high-speed serial UI * 32). This time corresponds to the D-PHY specification's (T_{HS-TRAIL} + T_{CLK-POST} + T_{CLK-TRAIL} + T_{HS-EXIT} + T_{LPX} * 2 + T_{LPX}(For Clock Lane HS Entry) + T_{CLK-PREPARE} + T_{CLK-ZERO} + T_{CLK-PRE} + T_{LPX}(For Data Lane HS Entry) + T_{HS-PREPARE} + T_{HS-ZERO} + T_{HS-SYNC}). For information about each parameter, see Figure 9.5-2 and Table 9.5-4. Of these parameters, it is necessary to set the time actually set in CLKKPT for (T_{HS-TRAIL} + T_{CLK-POST}) and that actually set in CLKBFHT for (T_{LPX}(For Clock Lane HS Entry) + T_{CLK-PREPARE} + T_{CLK-ZERO} + T_{CLK-PRE}). Set the rounded-up value plus at least 3. Setting this field to "0" is prohibited. The "T_{LPX}(For Clock Lane HS Entry)" means the period of LP-01 in HS entry sequence of Clock Lane. And the "T_{LPX}(For Data Lane HS Entry)" means the period of LP-01 in HS entry sequence of Data Lane. If these periods are not same as normal T_{LPX} value of host device (in the range of 67% to 150% of the T_{LPX} value of peripheral device), use the actual periods for this calculation.												
1, 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.												

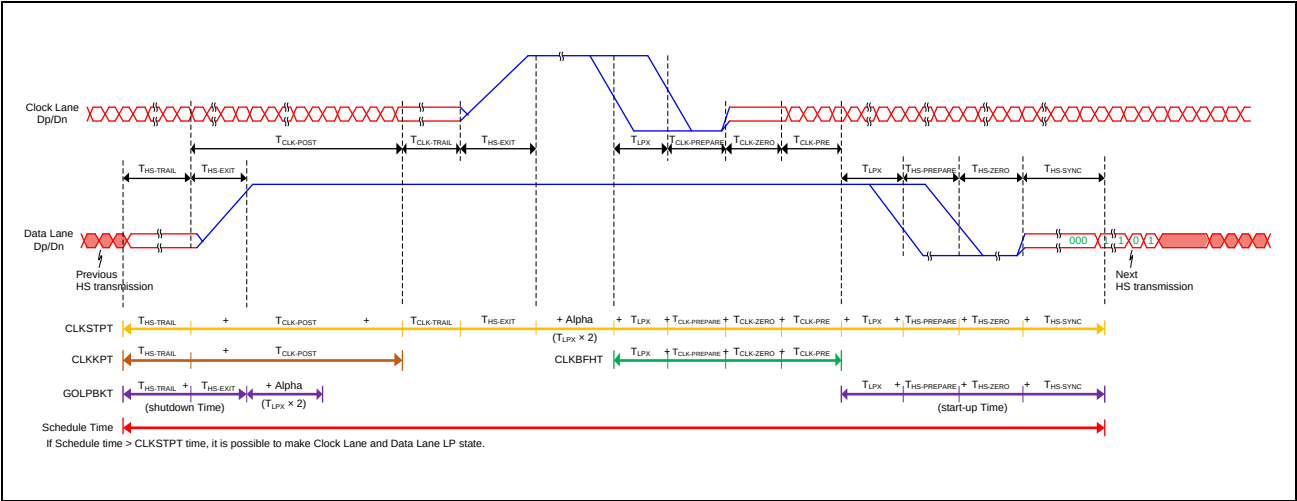


Figure 9.5-2 Switching the Clock Lane between Clock Transmission and Low-Power Mode

9.5.2.2.40 LP Transition Time Set Register (DSI_LINK_LPTRNSTSETR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0318h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	GOLPBKT[9:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
9 to 0	GOLPBKT[9:0]	0h	RW	Go LP and Back Time The time period over which the HS Data Lanes are changed to LP and back to HS. If the scheduled time between HS transfers is more than this setting, Data Lanes will be changed to LP. The time is calculated by the formula of GOLPBKT * (the period of high-speed serial 8 * UI). This time corresponds to D-PHY specification's ($T_{HS-TRAIL} + T_{HS-EXIT} + T_{LPX} * 2 + T_{LPX}$ (For Data Lane HS Entry) + $T_{HS-PREPARE} + T_{HS-ZERO} + T_{HS-SYNC}$). For information about each parameter, see Figure 9.5-3 and Table 9.5-4 . The unit of this setting is 8 UIs, so if the result of this formula contains a fraction of less than 8 UIs, round it up. Set the rounded-up value plus at least 3. The " T_{LPX} (For Data Lane HS Entry)" means the period of LP-01 in HS entry sequence of Data Lane. If this period is not same as normal T_{LPX} value of host device (in the range of 67% to 150% of the T_{LPX} value of peripheral device), use the actual period for this calculation.

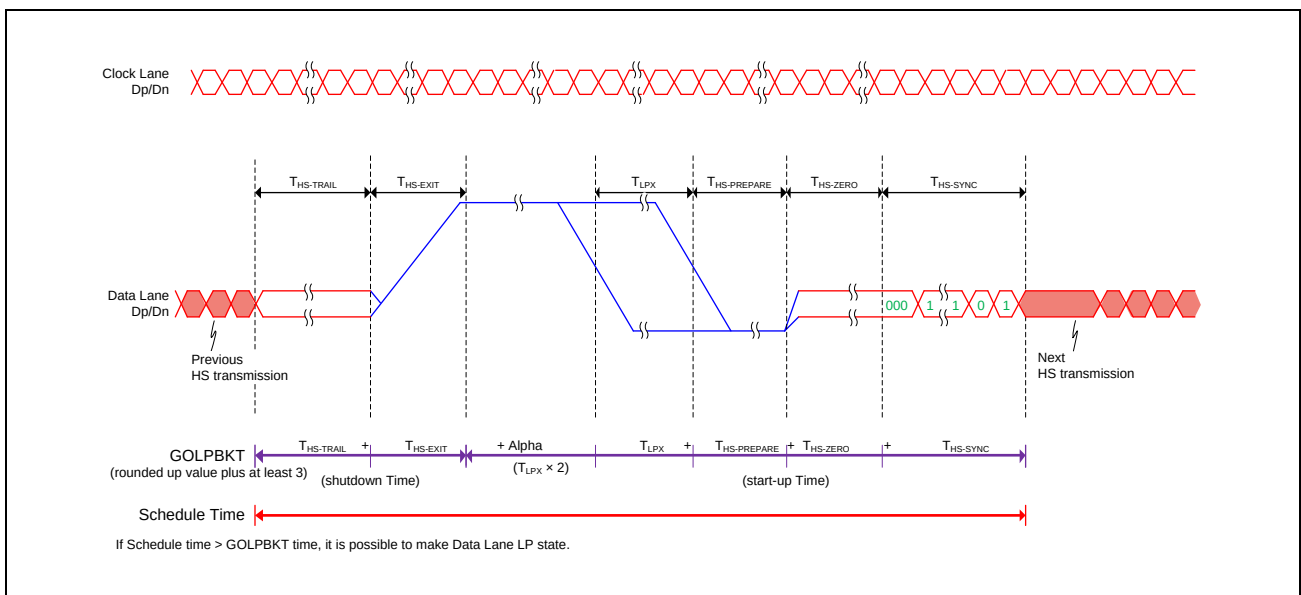


Figure 9.5-3 Switching the Data Lane between High-Speed Mode and Low-Power Mode

9.5.2.2.41 Physical Lane Status Register (DSI_LINK_PLSR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0320h

Initial Value : 0000_0FF3h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	DLFROMULPS	DLTOULPS	CLHS2LP	CLLP2HS	CLFROMULPS	CLTOULPS	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DL0DIR	-	DL0TX2RX	DL0RX2TX	DLSTPST[3:0]			DLULPSACTN[3:0]					DL0RXULPSESC	DL0RXLPDTESC	CLSTPST	CLULPSACTN
Initial Value	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29	DLFROMULPS	0h	R	From ULPS (Data Lane) When this value is 1b, all PPI Data Lanes enabled by TXSETR.NUMLANEUSE were changed from ULPS to Stop State by exiting ULPS. Write 1b to the PLSCR.DLFROMULPS register to clear this field.
28	DLTOULPS	0h	R	To ULPS (Data Lane) When this value is 1b, all PPI Data Lanes enabled by TXSETR.NUMLANEUSE were changed from Stop State to ULPS by entering ULPS. Write 1b to the PLSCR.DLTOULPS register to clear this field.
27	CLHS2LP	0h	R	HS to LP (Clock Lane) When this value is 1b, a transition of Clock Lane from HS to LP is detected. Checking this bit during non-continuous clock mode operation has no meaning. Therefore, ignore this bit when HSCLKSETR.HSCLKMODE = 0b. Write 1b to the PLSCR.CLHS2LP register to clear this field.
26	CLLP2HS	0h	R	LP to HS (Clock Lane) When this value is 1b, a transition of Clock Lane from LP to HS is detected. Checking this bit during non-continuous clock mode operation has no meaning. Therefore, ignore this bit when HSCLKSETR.HSCLKMODE = 0b. Write 1b to the PLSCR.CLLP2HS register to clear this field.
25	CLFROMULPS	0h	R	From ULPS (Clock Lane) When this value is 1b, Clock Lane exiting ULPS is detected. Write 1b to the PLSCR.CLFROMULPS register to clear this field.
24	CLTOULPS	0h	R	To ULPS (Clock Lane) When this value is 1b, Clock Lane entering ULPS is detected. Write 1b to the PLSCR.CLTOULPS register to clear this field.
23 to 16	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
15	DL0DIR	0h	R	Direction (Tx or Rx) of Data Lane 0 0b: PHY is in Tx Mode 1b: PHY is in Rx Mode This field indicates Direction of PPI Data Lane 0 after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
14	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13	DL0TX2RX	0h	R	Tx to Rx on Data Lane 0 When this value is 1b, a transition of Direction from 0 (Tx) to 1 (Rx) is detected on Data Lane 0. Write 1b to the PLSCR.DL0TX2RX register to clear this field.
12	DL0RX2TX	0h	R	Rx to Tx on Data Lane 0 When this value is 1b, a transition of Direction from 1 (Rx) to 0 (Tx) is detected on Data Lane 0. Write 1b to the PLSCR.DL0RX2TX register to clear this field.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	DLSTPST[3:0]	Fh	R	Stop State of Data Lane bit 3: Stop State of Data Lane 3 bit 2: Stop State of Data Lane 2 bit 1: Stop State of Data Lane 1 bit 0: Stop State of Data Lane 0 This field indicates Stop State of a PPI Data Lane after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
7 to 4	DLULPSACTN [3:0]	Fh	R	UlpsActiveNot of Data Lane bit 3: UlpsActiveNot of Data Lane 3 bit 2: UlpsActiveNot of Data Lane 2 bit 1: UlpsActiveNot of Data Lane 1 bit 0: UlpsActiveNot of Data Lane 0 This field indicates UlpsActiveNot of a PPI Data Lane after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
3	DL0RXULPSES C	0h	R	RxUlpsEsc of Data Lane 0 This field indicates RxUlpsEsc of PPI Data Lane 0 after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
2	DL0RXLPDTES C	0h	R	RxLpdtEsc of Data Lane 0 This field indicates RxLpdtEsc of PPI Data Lane 0 after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
1	CLSTPST	1h	R	Stop State of PPI Clock Lane This field indicates Stop State of the PPI Clock Lane after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.
0	CLULPSACTN	1h	R	UlpsActiveNot of PPI Clock Lane This field indicates UlpsActiveNot of the PPI Clock Lane after synchronization to lpclk. This field does not indicate the current status while lpclk is not running.

9.5.2.2.42 Physical Lane Status Clear Register (DSI_LINK_PLSCR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0324h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	DLFROMULPS	DLTOULPS	CLHS2LP	CLLP2HS	CLFROMULPS	CLTOULPS	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R0W	R0W	R0W	R0W	R0W	R0W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DL0TX2RX	DL0RX2TX	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R0W	R0W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29	DLFROMULPS	0h	R0W	The read value is always 0b. PLSR.DLFROMULPS Clear Set to 1b to clear PLSR.DLFROMULPS. Writing 0b has no effect.
28	DLTOULPS	0h	R0W	The read value is always 0b. PLSR.DLTOULPS Clear Set to 1b to clear PLSR.DLTOULPS. Writing 0b has no effect.
27	CLHS2LP	0h	R0W	The read value is always 0b. PLSR.CLHS2LP Clear Set to 1b to clear PLSR.CLHS2LP. Writing 0b has no effect.
26	CLLP2HS	0h	R0W	The read value is always 0b. PLSR.CLLP2HS Clear Set to 1b to clear PLSR.CLLP2HS. Writing 0b has no effect.
25	CLFROMULPS	0h	R0W	The read value is always 0b. PLSR.CLFROMULPS Clear Set to 1b to clear PLSR.CLFROMULPS. Writing 0b has no effect.
24	CLTOULPS	0h	R0W	The read value is always 0b. PLSR.CLTOULPS Clear Set to 1b to clear PLSR.CLTOULPS. Writing 0b has no effect.
23 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13	DL0TX2RX	0h	R0W	The read value is always 0b. PLSR.DL0TX2RX Clear Set to 1b to clear PLSR.DL0TX2RX. Writing 0b has no effect.
12	DL0RX2TX	0h	R0W	The read value is always 0b. PLSR.DL0RX2TX Clear Set to 1b to clear PLSR.DL0RX2TX. Writing 0b has no effect.
11 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.43 Physical Lane Interrupt Enable Register (DSI_LINK_PLIER)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0328h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	DLFROMULPS	DLTOULPS	CLHS2LP	CLLP2HS	CLFROMULPS	CLTOULPS	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DL0TX2RX	DL0RX2TX	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29	DLFROMULPS	0h	RW	Interrupt enable for all Data Lane transition from ULPS 0b: Disables interrupt dsi_int_ppi when PLSR.DLFROMULPS = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.DLFROMULPS = 1.
28	DLTOULPS	0h	RW	Interrupt enable for all Data Lane transition to ULPS 0b: Disables interrupt dsi_int_ppi when PLSR.DLTOULPS = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.DLTOULPS = 1.
27	CLHS2LP	0h	RW	Interrupt enable for Clock Lane transition to LP from HS 0b: Disables interrupt dsi_int_ppi when PLSR.CLHS2LP = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.CLHS2LP = 1.
26	CLLP2HS	0h	RW	Interrupt enable for Clock Lane transition to HS from LP 0b: Disables interrupt dsi_int_ppi when PLSR.CLLP2HS = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.CLLP2HS = 1.
25	CLFROMULPS	0h	RW	Interrupt enable for Clock Lane transition from ULPS 0b: Disables interrupt dsi_int_ppi when PLSR.CLFROMULPS = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.CLFROMULPS = 1.
24	CLTOULPS	0h	RW	Interrupt enable for Clock Lane transition to ULPS 0b: Disables interrupt dsi_int_ppi when PLSR.CLTOULPS = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.CLTOULPS = 1.
23 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13	DL0TX2RX	0h	RW	Interrupt enable for Data Lane 0 direction change from Tx to Rx 0b: Disables interrupt dsi_int_ppi when PLSR.DL0TX2RX = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.DL0TX2RX = 1.
12	DL0RX2TX	0h	RW	Interrupt enable for Data Lane 0 direction change from Rx to Tx 0b: Disables interrupt dsi_int_ppi when PLSR.DL0RX2TX = 1. 1b: Enables interrupt dsi_int_ppi when PLSR.DL0RX2TX = 1.
11 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.44 Video-Input Channel 1 Set 0 Register (DSI_LINK_VICH1SET0R)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0400h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VSEN	-	HFPNO LP	HBPNO LP	HSAN OLP	-	-	-	-	-	-	VSTPA FT	VSTAR T
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R0W	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
12	VSEN	0h	RW	Video Skew Calibration Enable 0b: No skew calibration is issued by this channel. 1b: Periodic skew calibration is issued in the first line of VBP. When periodic skew calibration is enabled, set HSCLKSETR.HSCLKMODE = 1.
11	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
10	HFPNOLP	0h	RW	HFP period No LP Specifies whether suppress to transition to LP during the HFP period. 0b: Does not suppress transition to LP during the period 1b: Suppresses transition to LP during the period and keeps HS. HS is kept by sending a blanking packet.
9	HBPNOLP	0h	RW	HBP period No LP Specifies whether suppress transition to LP during the HBP period. 0b: Does not suppress transition to LP during the period 1b: Suppresses transition to LP during the period and keeps HS. HS is kept by sending a blanking packet.
8	HSANOLP	0h	RW	HSA period No LP Specifies whether suppress transition to LP during the HSA period. When TXESYNC=0, this field has no meaning. 0b: Does not suppress transition to LP during the period 1b: Suppresses transition to LP during the period and keeps HS. HS is kept by sending a blanking packet.
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	-	0h	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
1	VSTPAFT	0h	R0W	The read value is always 0b. Video-Input signals stop after Video-Input Operation stop Stop Video-Input Operation. Video-Input Operation will end at the end of video-frame. After Video-Input Operation end, stop Video-Input signals. If Video-Input Operation is stopped, VICH1SR.RUNNING bit will change to 0b. Set 1b to both VSTPAFT and VSTART is prohibited. Set 1b during VICH1SR.RUNNING = 0 is prohibited. Writing 0b has no effect.

Bit	Bit Name	Initial Value	R/W	Description
0	VSTART	0h	R0W	The read value is always 0b. Video-Input Operation Start Write 1b to start Video-Input Operation. Setting both VSTART and VSTPAFT to 1b is prohibited. Setting this bit to 1b while VICH1SR.RUNNING = 1 is prohibited. Writing 0b has no effect.

9.5.2.2.45 Video-Input Channel 1 Set 1 Register (DSI_LINK_VICH1SET1R)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0404h

Initial Value : 1093_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSPC	-														
Initial Value	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	CSPC	0h	R	Color Space of Input Video Pixel This field reflects the color space of Data Type selected by VICH1PPSETR.DT. 0h: RGB 1h: Reserved
30	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 24	BPP[5:0]	10h	R	Bit Per Pixel of Input Video Pixel This field reflects bits per pixel of Data Type selected by VICH1PPSETR.DT. 10h: 16 bpp 12h: 18 bpp 18h: 24 bpp, 18 bpp (Loosely Packed) (RGB)
23 to 20	CHBUFSZ [3:0]	9h	R	Channel Buffer Size 9h: 64 kB Buffer size = 128 x 2 CHBUFSZ
19, 18	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
17, 16	-	All 1	R	Reserved Whenever it is read, 1 is read. The written value will be ignored.
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 2	DLY[11:0]	0h	RW	Delay Value The delay time is calculated by the formula of DLY * (the period of high-speed serial UI * 32). Setting to this field to 0b is prohibited. If VICH1SET0R.HFPNOLP = 1, set this field to a value greater than the HFP period. If VICH1SET0R.HBPNOLP = 1, set this field to a value greater than the HBP period. If VICH1SET0R.HSANOLP = 1, set this field to a value greater than the HSA period.
1, 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.46 Video-Input Channel 1 Status Register (DSI_LINK_VICH1SR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0410h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VBUFO VF	VBUFU DF	-	TIMER R	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	VIRDY	RUNNI NG	STOP	START
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23	VBUFOVF	0h	R	Video-Input Buffer Overflow Error A data overflow has occurred in Video-Input Buffer. Write 1b to the VICH1SCR.VBUFOVF register to clear this field.
22	VBUFUDF	0h	R	Video-Input Buffer Underflow Error A data underflow has occurred in Video-Input Buffer. Write 1b to the VICH1SCR.VBUFUDF register to clear this field.
21	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
20	TIMERR	0h	R	Timing Error A timing error has occurred during Video-Input Operation. Write 1b to the VICH1SCR.TIMERR register to clear this field.
19 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	VIRDY	0h	R	Video-Input Signal Accept Ready The value will be set to 1b when the input signal can be accepted by Video-Input Operation channel 1 ready.
2	RUNNING	0h	R	Video-Input Operation channel 1 Transfer Running The value will be set to 1b while Video-Input Operation channel 1 transfer is in progress. Changing the values of the VICH1SET0R, VICH1SET1R, VICH1PPSETR, VICH1VSSETR, VICH1VPSETR, VICH1HSSETR and VICH1HPSETR is prohibited while the transfer is in progress (VICH1SR.RUNNING = 1).
1	STOP	0h	R	Video Input Operation channel 1 Transfer Stop Write 1b to the VICH1SCR.STOP register to clear this field.
0	START	0h	R	Video Input Operation channel 1 Transfer Start. Write 1b to the VICH1SCR.START register to clear this field.

9.5.2.2.47 Video-Input Channel 1 Status Clear Register (DSI_LINK_VICH1SCR)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0414h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VBUFOVF	VBUFUDF	-	TIMERR	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R0W	R0W	R0W	R0W	R0W	R	R	R0W	R0W	R0W	R0W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	VIRDY	-	STOP	START
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R0W	R	R	R	R	R	R	R	R	R	R	R0W	R	R0W	R0W

Bit	Bit Name	Initial Value	R/W	Description
31	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 26	-	All 0	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
25 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23	VBUFOVF	0h	R0W	The read value is always 0b. VICH1SR.VBUFOVF Clear Set to 1b to clear VICH1SR.VBUFOVF. Writing 0b has no effect.
22	VBUFUDF	0h	R0W	The read value is always 0b. VICH1SR.VBUFUDF Clear Set to 1b to clear VICH1SR.VBUFUDF. Writing 0b has no effect.
21	-	0h	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
20	TIMERR	0h	R0W	The read value is always 0b. VICH1SR.TIMERR Clear Set to 1b to clear VICH1SR.TIMERR. Writing 0b has no effect.
19 to 15	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14	-	0h	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
13 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	VIRDY	0h	R0W	The read value is always 0b. VICH1SR.VIRDY Clear Set to 1b to clear VICH1SR.VIRDY. Writing 0b has no effect.
2	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	STOP	0h	R0W	The read value is always 0b. VICH1SR.STOP Clear Set to 1b to clear VICH1SR.STOP. Writing 0b has no effect.
0	START	0h	R0W	The read value is always 0b. VICH1SR.START Clear Set to 1b to clear VICH1SR.START. Writing 0b has no effect.

9.5.2.2.48 Video-Input Channel 1 Interrupt Enable Register (DSI_LINK_VICH1IER)

Access Size : 8, 16, 32 bits

Offset Address : <DSI_LINK_base> + 0418h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	VBUFO VF	VBUFU DF	-	TIMER R	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	R	R	R	R

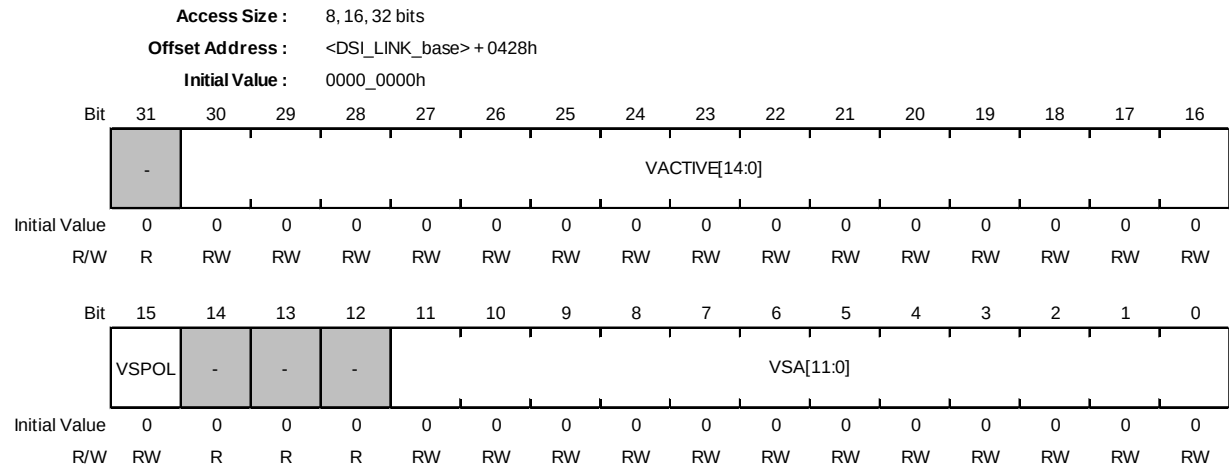
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	VIRDY	-	STOP	START
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	R	R	R	R	R	R	R	R	R	R	RW	R	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 26	-	All 0	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
25, 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
23	VBUFOVF	0h	RW	Interrupt enable for the Video-Input Buffer Overflow Error 0b: Disables interrupt dsi_int_vin1 when VICH1SR.VBUFOVF = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.VBUFOVF = 1.
22	VBUFUDF	0h	RW	Interrupt enable for the Video-Input Buffer Underflow Error 0b: Disables interrupt dsi_int_vin1 when VICH1SR.VBUFUDF = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.VBUFUDF = 1.
21	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
20	TIMERR	0h	RW	Interrupt enable for the Timing Error 0b: Disables interrupt dsi_int_vin1 when VICH1SR.TIMERR = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.TIMERR = 1.
19 to 15	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
14	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
13 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
3	VIRDY	0h	RW	Interrupt enable for the Video Input Accept Ready 0b: Disables interrupt dsi_int_vin1 when VICH1SR.VIRDY = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.VIRDY = 1.
2	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	STOP	0h	RW	Interrupt enable for the Video Input Operation channel 1 Transfer Stop 0b: Disables interrupt dsi_int_vin1 when VICH1SR.STOP = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.STOP = 1.
0	START	0h	RW	Interrupt enable for the Video Input Operation channel 1 Transfer Start 0b: Disables interrupt dsi_int_vin1 when VICH1SR.START = 1. 1b: Enables interrupt dsi_int_vin1 when VICH1SR.START = 1.

9.5.2.2.49 Video-Input Channel 1 Pixel Packet Set Register (DSI_LINK_VICH1PPSETR)

Access Size :		8, 16, 32 bits															
Offset Address :		<DSI_LINK_base> + 0420h															
Initial Value :		000E_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	VC[1:0]		DT[5:0]						
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TXESY NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	RW	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	
Bit	Bit Name	Initial Value	R/W	Description													
31 to 24	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													
23, 22	VC[1:0]	0h	RW	Video-Input Operation Channel 1 Pixel Stream Packet Header Virtual Channel Identifier													
21 to 16	DT[5:0]	Eh	RW	Video-Input Operation Channel 1 Pixel Stream Packet Header Data Type The following values can be set. 0Eh: Packed Pixel Stream, 16-bit RGB 1Eh: Packed Pixel Stream, 18-bit RGB 2Eh: Loosely Packed Pixel Stream, 18-bit RGB 3Eh: Packed Pixel Stream, 24-bit RGB Setting any other values is prohibited.													
15	TXESYNC	0h	RW	Transmit End of Sync Pulse 0b: HSE and VSE are NOT transmitted 1b: HSE and VSE are transmitted The setting 0b is used for “Burst Mode” sequence or “Non-Burst Mode with Sync Events” sequence. The setting 1b is used for “Non-Burst Mode with Sync Pulse” sequence.													
14 to 4	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.													
3 to 0	-	All 0	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.													

9.5.2.2.50 Video-Input Channel 1 Vertical Size Set Register (DSI_LINK_VICH1VSSETR)



Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 16	VACTIVE[14:0]	0h	RW	Sets the Line count of VACTIVE (Vertical Active lines)
15	VSPOL	0h	RW	Polarity of VSYNC Sets the polarity (active state) of the VSYNC signal. 0b: Active high 1b: Active low
14 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	VSA[11:0]	0h	RW	VSA Count Sets the Line count of VSA (Vertical Sync Active)

9.5.2.2.51 Video-Input Channel 1 Vertical Porch Set Register (DSI_LINK_VICH1VPSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 042Ch														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	VFP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	VBP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	VFP[12:0]		0h	RW	VFP Count Sets the Line count of VFP (Vertical Front Porch)											
15 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	VBP[12:0]		0h	RW	VBP Count Sets the Line count of VBP (Vertical Back Porch)											

9.5.2.2.52 Video-Input Channel 1 Horizontal Size Set Register (DSI_LINK_VICH1HSSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0430h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	HACTIVE[14:0]														
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSPOL	-	-	-	HSA[11:0]											
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30 to 16	HACTIVE [14:0]	0h	RW	HACTIVE Count Sets the pixel count of HACTIVE (Horizontal Active pixels)
15	HSPOL	0h	RW	Polarity of HSYNC Sets the polarity (active state) of the HSYNC signal. 0b: Active high 1b: Active low
14 to 12	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
11 to 0	HSA[11:0]	0h	RW	HSA Count Sets the pixel count of HSA (Horizontal Sync Active)

9.5.2.2.53 Video-Input Channel 1 Horizontal Porch Set Register (DSI_LINK_VICH1HPSETR)

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 0434h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	HFP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	HBP[12:0]												
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 29	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
28 to 16	HFP[12:0]		0h	RW	HFP Count Sets the pixel count of HFP (Horizontal Front Porch)											
15 to 13	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
12 to 0	HBP[12:0]		0h	RW	HBP Count Sets the pixel count of HBP (Horizontal Back Porch)											

9.5.2.2.54 Sequence Channel x Set 0 Register (DSI_LINK_SQCHxSET0R)

“x” indicates the channel number.

Note: “x” is 0 or 1.

Access Size :		8, 16, 32 bits															
Offset Address :		<DSI_LINK_base> + 05C0h (x = 0) <DSI_LINK_base> + 0600h (x = 1)															
Initial Value :		0080_0000h															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Initial Value	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
R/W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	R0W	RW	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	START	
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	R	R	R0W	R0W	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	-	All 0	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
23	-	1h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
22 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8 to 4	-	All 0	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
3, 2	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
1	-	0h	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
0	START	0h	R0W	The read value is always 0b. Sequence Operation Start Write 1b to start running from Descriptor #0. Setting it to 1b while RUNNING = 1 is prohibited. Writing 0b has no effect.

9.5.2.2.55 Sequence Channel x Set 1 Register (DSI_LINK_SQCHxSET1R)

“x” indicates the channel number.

Note: “x” is 0 or 1.

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 05C4h (x = 0) <DSI_LINK_base> + 0604h (x = 1)														
Initial Value :		0800_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAXDESNUM[7:0]								CHBUFSZ[3:0]				-	-	-	-
Initial Value	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	MAXDESNUM [7:0]	8h	RW	Max Descriptor Number This field indicates the maximum number of descriptors, with the descriptor numbers starting from zero (e.g. the setting 8 means #0 to #7 can be used as descriptors). Writing a value other than the initial value is prohibited.
23 to 20	CHBUFSZ [3:0]	0h	R	Sequence Channel x Buffer Size 0h: 128 B 1h: 256 B 2h: 512 B 3h: 1 kB 4h: 2 kB 5h: 4 kB 6h: 8 kB 7h: 16 kB 8h: 32 kB 9h: 64 kB Buffer size = 128 x 2 CHBUFSZ Setting a transfer packet size larger than this field is prohibited.
19 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.56 Sequence Channel x Status Register (DSI_LINK_SQCHxSR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 05D0h (x = 0) <DSI_LINK_base> + 0610h (x = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	RXCO RERR	RXPKT DFAIL	RXFAIL	RXFAT ALERR	-	TXIBER R	-	-	-	-	PKTBI GERR	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADESFI N	-	-	-	AACFTFI N	-	RUNNI NG	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	R	Receive Acknowledge and Error Report Packet RXSR.RXAKE is also reported. Write 1b to the SQCHxSCR.RXAKE register to clear this field.
29	RXCORERR	0h	R	Receive Correctable Error The received packet has a correctable error. RXSR.ECCERR1B is also reported. Write 1b to the SQCHxSCR.RXCORERR register to clear this field.
28	RXPKTDFAIL	0h	R	Receive Packet Data Fail The packet Header was saved correctly but data were not. One or more of the following bits are also reported. RXSR.CRCERR, RXSR.WCERR, RXSR.UEXPKTERR, RXSR.MAXRPSZERR, RXSR.RXOVFERR and RXSR.IBERR. Write 1b to the SQCHxSCR.RXPKTDFAIL register to clear this field.
27	RXFAIL	0h	R	Receive Fail Expected reception did not proceed. One or more of the following bits are also reported. RXSR.PRESPTOERR, RXSR.ECCERR, RXSR.MLFERR and RXSR.NORETERR. Write 1b to the SQCHxSCR.RXFAIL register to clear this field.
26	RXFATALERR	0h	R	Receive Fatal Error A fatal timeout occurred during BTA. One or more of the following bits are also reported. FERRSR.TATO and FERRSR.LRXHTO. Write 1b to the SQCHxSCR.RXFATALERR register to clear this field.
25	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	TXIBERR	0h	R	Tx Internal Bus Error When this value is 1b, the Internal Bus Read had failed to respond. Write 1b to the SQCHxSCR.TXIBERR register to clear this field.
23 to 20	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	PKTBIGERR	0h	R	Packet is too big Error Sequence Operation channel 0: Hard wired to 0b. Sequence Operation channel 1: This bit is set when the length of a sequence packet is bigger than Video-Input channel's BLLP. Write 1b to the SQCHxSCR.PKTBIGERR register to clear this field.
18 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
8	ADESFIN	0h	R	All Descriptor Finished by Setting The descriptor is finished by setting SQCHxDSCyAR.NXACT = 0h for the descriptor. Write 1b to the SQCHxSCR.ADESFIN register to clear this field.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	AACTFIN	0h	R	Descriptor's All Action Finished This bit is set if this descriptor's all actions are finished with SQCHxDSCyCR.FINACT[0] = 1b. Write 1b to the SQCHxSCR.AACTFIN register to clear this field.
3	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2	RUNNING	0h	R	Sequence Operation Running The value will be set to "1" during Sequence Operation. Do not change descriptor values while this bit is 1b. Do not change the values in SQCHxSET0R and SQCHxSET1R.
1, 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.57 Sequence Channel x Status Clear Register (DSI_LINK_SQCHxSCR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 05D4h (x = 0) <DSI_LINK_base> + 0614h (x = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	RXCORERR	RXPKTDFAIL	RXFAIL	RXFATALERR	-	TXIBERR	-	-	-	-	PKTBIGERR	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R0W	R0W	R0W	R0W	R0W	R	R0W	R	R0W	R	R	R0W	R	R	R0W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADESFIN	-	-	-	AACTFIN	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R0W	R	R	R	R	R	R	R0W	R	R	R	R0W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	R0W	The read value is always 0b. SQCHxSR.RXAKE Clear Set to 1b to clear SQCHxSR.RXAKE. Writing 0b has no effect.
29	RXCORERR	0h	R0W	The read value is always 0b. SQCHxSR.RXCORERR Clear Set to 1b to clear SQCHxSR.RXCORERR. Writing 0b has no effect.
28	RXPKTDFAIL	0h	R0W	The read value is always 0b. SQCHxSR.RXPKTDFAIL Clear Set to 1b to clear SQCHxSR.RXPKTDFAIL. Writing 0b has no effect.
27	RXFAIL	0h	R0W	The read value is always 0b. SQCHxSR.RXFAIL Clear Set to 1b to clear SQCHxSR.RXFAIL. Writing 0b has no effect.
26	RXFATALERR	0h	R0W	The read value is always 0b. SQCHxSR.RXFATALERR Clear Set to 1b to clear SQCHxSR.RXFATALERR. Writing 0b has no effect.
25	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	TXIBERR	0h	R0W	The read value is always 0b. SQCHxSR.TXIBERR Clear Set to 1b to clear SQCHxSR.TXIBERR. Writing 0b has no effect.
23	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	-	0h	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
21, 20		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	PKTBIGERR	0h	R0W	The read value is always 0b. SQCHxSR.PKTBIGERR Clear Set to 1b to clear SQCHxSR.PKTBIGERR. Writing 0b has no effect.
18, 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

Bit	Bit Name	Initial Value	R/W	Description
16, 15	-	All 0	R0W	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
14 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	ADESFIN	0h	R0W	The read value is always 0b. SQCHxSR.ADESFIN Clear Set to 1b to clear SQCHxSR.ADESFIN. Writing 0b has no effect.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	AACTFIN	0h	R0W	The read value is always 0b. SQCHxSR.AACTFIN Clear Set to 1b to clear SQCHxSR.AACTFIN. Writing 0b has no effect.
3 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.58 Sequence Channel x Interrupt Enable Register (DSI_LINK_SQCHxIER)

“x” indicates the channel number.

Note: “x” is 0 or 1.

Access Size :		8, 16, 32 bits														
Offset Address :		<DSI_LINK_base> + 05D8h (x = 0) <DSI_LINK_base> + 0618h (x = 1)														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	RXAKE	RXCO RERR	RXPKT DFAIL	RXFAIL	RXFAT ALERR	-	TXIBER R	-	-	-	-	PKTBI GERR	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	RW	RW	RW	RW	RW	R	RW	R	RW	R	R	RW	R	R	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADESFI N	-	-	-	AACTFI N	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	R	R	R	R	R	R	RW	R	R	R	RW	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
30	RXAKE	0h	RW	Interrupt enable for the Receive Acknowledge and Error Report Packet 0b: Disables interrupt dsi_int_sqx when SQCHxSR.RXAKE = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.RXAKE = 1.
29	RXCORERR	0h	RW	Interrupt enable for the Receive Correctable Error 0b: Disables interrupt dsi_int_sqx when SQCHxSR.RXCORERR = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.RXCORERR = 1.
28	RXPKTDFAIL	0h	RW	Interrupt enable for the Receive Packet Data Fail 0b: Disables interrupt dsi_int_sqx when SQCHxSR.RXPKTDFAIL = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.RXPKTDFAIL = 1.
27	RXFAIL	0h	RW	Interrupt enable for the Receive Fail 0b: Disables interrupt dsi_int_sqx when SQCHxSR.RXFAIL = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.RXFAIL = 1.
26	RXFATALERR	0h	RW	Interrupt enable for the Receive Fatal Error 0b: Disables interrupt dsi_int_sqx when SQCHxSR.RXFATALERR = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.RXFATALERR = 1.
25	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
24	TXIBERR	0h	RW	Interrupt enable for the Tx Internal Bus Error 0b: Disables interrupt dsi_int_sqx when SQCHxSR.TXIBERR = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.TXIBERR = 1.
23	-	0h	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22	-	0h	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.
21, 20		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
19	PKTBIGERR	0h	RW	Interrupt enable for the Packet is too big Error Sequence Operation channel 0: This field has no meaning. Sequence Operation channel 1: 0b: Disables interrupt dsi_int_sqx when SQCHxSR.PKTBIGERR = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.PKTBIGERR = 1.
18, 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16, 15	-	All 0	RW	Reserved When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written.

Bit	Bit Name	Initial Value	R/W	Description
14 to 9	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
8	ADESFIN	0h	RW	Interrupt enable for the Descriptor Finished by Setting 0b: Disables interrupt dsi_int_sqx when SQCHxSR.ADESFIN = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.ADESFIN = 1.
7 to 5	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
4	AACTFIN	0h	RW	Interrupt enable for the Descriptor's All Action Finished 0b: Disables interrupt dsi_int_sqx when SQCHxSR.AACTFIN = 1. 1b: Enables interrupt dsi_int_sqx when SQCHxSR.AACTFIN = 1.
3 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.2.59 Sequence Channel x Descriptor y-A Register (DSI_LINK_SQCHxDSCyAR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

“y” indicates the descriptor number.

Note: “y” is 00 to 07.

Access Size :		32 bits															
Offset Address :		<DSI_LINK_base> + 0780h + 0010h × (y) (x = 0) <DSI_LINK_base> + 0800h + 0010h × (y) (x = 1)															
Initial Value :		xxxx_xxxxh															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	-	-	NXACT[1:0]		BTA[1:0]		SPD	FMT	VC[1:0]		DT[5:0]						
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	DATA1[7:0]								DATA0[7:0]								
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	xh* ²	RW	Reserved When read, the last written value is read. When writing, be sure to write 0b. Operation is not guaranteed if a value other than 0b is written.
29, 28	NXACT[1:0]	xh* ²	RW	Next Action 0h: Terminates operation after this descriptor is finished. 1h: Operates the next descriptor after this descriptor is finished. 2h: Reserved 3h: Reserved
27, 26	BTA[1:0]	xh* ²	RW	Bus Turn Around Sets Tx Request without either Bus Turn-Around (BTA) or no-operation. Or else, sets Tx Request with BTA. 0h: Tx Request without BTA or no-operation. 1h: Tx non-Read Request with BTA. 2h: Tx Read Request with BTA. 3h: BTA only
25	SPD	xh* ²	RW	Speed Sets the speed type in Tx Request. 0b: High Speed 1b: Low Power Setting this bit to “1” during Video-Input Operation is prohibited.
24	FMT	xh* ²	RW	Format Packet format of Tx Packet Header. This field should be set by Data Type. 0b: Short Packet 1b: Long Packet
23, 22	VC[1:0]	xh* ²	RW	Virtual Channel Identifier of Virtual Channel of Tx Packet Header.
21 to 16	DT[5:0]	xh* ²	RW	Data Type Data Type of Tx Packet Header.
15 to 8	DATA1[7:0]	xh* ²	RW	Data 1 Data 1 of Tx Packet Header* ¹ . When Tx Packet format is Long Packet, this value is the upper 8 bits of the word count. The maximum Long Packet size in Command Transfer mode which uses registers is 16 bytes. (SQCHxDSCyBR.DTSEL = 0h and SQCHxDSCyAR.FMT = 1) The maximum Long Packet size in Command Transfer mode which uses the internal bus memory is the number of bytes indicated by SQCHxSET1R.CHBUFSZ. (SQCHxDSCyBR.DTSEL = 1h and SQCHxDSCyAR.FMT = 1)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DATA0[7:0]	xh* ²	RW	Data 0 Data 0 of Tx Packet Header* ¹ . When Tx Packet format is Long Packet, this value is the lower 8 bits of the word count. The maximum Long Packet size in Command Transfer mode which uses registers is 16 bytes. (SQCHxDSCyBR.DTSEL = 0h and SQCHxDSCyAR.FMT = 1) The maximum Long Packet size in Command Transfer mode which uses the internal bus memory is the number of bytes indicated by SQCHxSET1R.CHBUFSZ. (SQCHxDSCyBR.DTSEL = 1h and SQCHxDSCyAR.FMT = 1)

x = Undefined

- Note 1. The maximum size for Command Transfer mode Long Packet by using the register is 16 bytes. (SQCHxDSCyBR.DTSEL = 0h and SQCHxDSCyAR.FMT = 1h)
 The maximum size for Command Transfer mode Long Packet by using the internal bus memory is SQCHxSET1R.CHBUFSZ bytes. (SQCHxDSCyBR.DTSEL = 1h and SQCHxDSCyAR.FMT = 1h)
- Note 2. The initial values are determined by the initial value of Descriptor RAM.

9.5.2.2.60 Sequence Channel x Descriptor y-B Register (DSI_LINK_SQCHxDSCyBR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

“y” indicates the descriptor number.

Note: “y” is 00 to 07.

Access Size : 32 bits

Offset Address : <DSI_LINK_base> + 0784h + 0010h × (y) (x = 0)

<DSI_LINK_base> + 0804h + 0010h × (y) (x = 1)

Initial Value : xxxx_xxxxh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	DTSEL[1:0]		-	-	-	-	-	-	-	-
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	xh* ¹	RW	Reserved When read, the last written value is read. When writing, be sure to write 0b. Operation is not guaranteed if a value other than 0b is written.
25, 24	DTSEL[1:0]	xh* ¹	RW	Data Select 0h: Packet Payload Data register 1h: Long Packet data use memory space. In the case of writing, internal read causes reading from the SQCHxDSCyDR address. In the case of reading, internal write causes writing to the SQCHxDSCyDR address. Note: Read Header is always saved to the RXRSSxR register. 2h, 3h: Reserved If a write request with Short packet, then set 0h. When DTSEL = 0h, Long packet write request data size shall be equal to or smaller than 16 bytes. If read request data to save exceeds 16 bytes, this is reported as an error.
23 to 0	-	xh* ¹	RW	Reserved When read, the last written value is read. When writing, be sure to write 0b. Operation is not guaranteed if a value other than 0b is written.

x = Undefined

Note 1. The initial values are determined by the initial value of Descriptor RAM.

9.5.2.2.61 Sequence Channel x Descriptor y-C Register (DSI_LINK_SQCHxDSCyCR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

“y” indicates the descriptor number.

Note: “y” is 00 to 07.

Access Size :		32 bits															
Offset Address :		<DSI_LINK_base> + 0788h + 0010h × (y) (x = 0) <DSI_LINK_base> + 0808h + 0010h × (y) (x = 1)															
Initial Value :		xxxx_xxxxh															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	ACTCODE[7:0]								-	AUXOP	-	-	-	-	-	-	
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	-	-	-	-	FINACT[2:0]			
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ACTCODE[7:0]	xh* ¹	RW	Action Code Indicates the rx result same slot number for BTA action. Setting a value greater than or equal to 4 is prohibited. Setting the same rx result save slot number to a different descriptor is prohibited. When AUXOP is 1. 00h: Sends Reset-Trigger. 04h: Sends initial skew calibration Prohibited to use in channel 0 (LP only). 05h: Sends periodic skew calibration Prohibited to use in channel 0 (LP only). 08h: No-operation Other: Reserved Set 00h in other cases.
23	-	xh* ¹	RW	Reserved When read, the last written value is read. When writing, be sure to write 0b. Operation is not guaranteed if a value other than 0b is written.
22	AUXOP	xh* ¹	RW	Auxiliary Operation Used for auxiliary operation execution. When this bit is set to 1b, SQCHxDSCyBR.BTA should be set to 0h. Set this bit to 1b with BTA = 0h. Setting this bit to 1b during Video-Input Operation is prohibited.
21 to 3	-	xh* ¹	RW	Reserved When read, the last written value is read. When writing, be sure to write 0b. Operation is not guaranteed if a value other than 0b is written.
2 to 0	FINACT[2:0]	xh* ¹	RW	Finish Action Bit 2: Set to 0b. Bit 1: Set to 0b. Bit 0: Sets SQCHxSR.AACTFIN if this descriptor's all actions have finished. (0b: Disabled, 1b: Enabled). Since each descriptor's operation is not the same, descriptors may finish out of order. For example, Descriptor#1 may finish before Descriptor#0.

x = Undefined

Note 1. The initial values are determined by the initial value of Descriptor RAM.

9.5.2.2.62 Sequence Channel x Descriptor y-D Register (DSI_LINK_SQCHxDSCyDR)

“x” indicates the channel number.

Note: “x” is 0 or 1.

“y” indicates the descriptor number.

Note: “y” is 00 to 07.

Access Size :		32 bits														
Offset Address :		<DSI_LINK_base> + 078Ch + 0010h × (y) (x = 0)														
		<DSI_LINK_base> + 080Ch + 0010h × (y) (x = 1)														
Initial Value :		xxxx_xxxxh														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LADDR[31:16]															
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LADDR[15:0]															
Initial Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LADDR[31:0]	xh* ¹	RW	Lower Address Sets the lower 32-bit address for Long Packet Payload Data in Sequence Operation channel x Descriptor y. When Sequence Operation Data Select is “use memory space” (SQCHxDSCyBR.DTSEL =1h) and Sequence Operation Format is Long packet (SQCHxDSCyAR.FMT=1b), the DSI-Tx module accesses the address by internal bus Master Read in Write action and internal bus Write in Read action. Set the lower three bits to “000b”.

x = Undefined

Note 1. The initial values are determined by the initial value of Descriptor RAM.

9.5.2.3 Resisters in D-PHY

9.5.2.3.1 DSI PLL Enable Register (DSI_DPHY_PPLENR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0000h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PLEN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W		Description									
31 to 1	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.									
0	PLEN		0h		RW		Initialize PLL during LOW state and enable when it is HIGH state.									

9.5.2.3.2 DSI PHY Reset Register (DSI_DPHY_PHYRSTR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0004h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PHYM RSTN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Bit	Bit Name		Initial Value		R/W	Description										
31 to 1	-		All 0		R	Reserved Whenever it is read, 0b is read. The written value will be ignored.										
0	PHYMRSTN		0h		RW	PHY Master Clock and Data Lane Reset										

9.5.2.3.3 DSI PLL Clock Setting 0 Register (DSI_DPHY_PLLCLKSET0R)

Access Size : 8, 16, 32 bits

Address : <DSI_DPHY_base> + 0010h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	PLL_M[9:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PLL_P[5:0]					-	-	-	-	-	PLL_S[2:0]			
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	RW	RW	RW	R	R	R	R	R	RW	RW	RW

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
25 to 16	PLL_M[9:0]	0h	RW	Value of the main-divider in PLL Can be set in the range of 64 to 1023. Values outside this range are prohibited.
15, 14	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
13 to 8	PLL_P[5:0]	0h	RW	Value of the pre-divider in PLL Can be set in the range of 1 to 4. Values outside this range are prohibited.
7 to 3	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
2 to 0	PLL_S[2:0]	0h	RW	Value of Scaler in PLL Can be set in the range of 0 to 5. Values outside this range are prohibited.

The PLL clock frequency (F_{FOUT}) is expressed by the following equation.

$$F_{\text{FREF}} = 24\text{MHz}/\text{PLL\_P}, 6\text{MHz} \leq F_{\text{FREF}} \leq 24\text{MHz}$$

$$F_{\text{FVCO}} = (\text{PLL\_M} + (\text{PLL\_K}/65536)) \times F_{\text{FREF}}, 1050\text{MHz} \leq F_{\text{FVCO}} \leq 2100\text{MHz}$$

$$F_{\text{FOUT}} = F_{\text{FVCO}}/2^{\text{PLL\_S}}, 80\text{MHz} \leq F_{\text{FOUT}} \leq 1500\text{MHz}$$

$$1 \leq \text{PLL\_P} \leq 4, 64 \leq \text{PLL\_M} \leq 1023, 0 \leq \text{PLL\_S} \leq 5, -32768 \leq \text{PLL\_K} \leq 32767$$

9.5.2.3.4 DSI PLL Clock Setting 1 Register (DSI_DPHY_PLLCLKSET1R)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0014h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PLL_K[15:0]															
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 16	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
15 to 0	PLL_K[15:0]		0h	RW	Value of DSM in PLL PLL_K is a two's complement, and the range is -32768 to 32767.											

9.5.2.3.5 DSI PHY TCLK Setting Register (DSI_DPHY_PHYTCLKSETR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0020h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCLKPRPRCTL[7:0]								TCLKZEROCTL[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCLKPOSTCTL[7:0]								TCLKTRAILCTL[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 24	TCLKPRPRCTL [7:0]		0h	RW	Control TCLK-PREPARE of the Master Clock Lane.											
23 to 16	TCLKZEROCTL [7:0]		0h	RW	Control TCLK-ZERO of the Master Clock Lane.											
15 to 8	TCLKPOSTCTL [7:0]		0h	RW	Control TCLK-POST of the Master Clock Lane.											
7 to 0	TCLKTRAILCTL [7:0]		0h	RW	Control TCLK-TRAIL of the Master Clock Lane.											

9.5.2.3.6 DSI PHY THS Setting Register (DSI_DPHY_PHYTHSSETR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0024h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THSPRPRCTL[7:0]								THSZEROCTL[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THSTRAILCTL[7:0]								THSEXITCTL[7:0]								
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 24	THSPRPRCTL [7:0]		0h	RW	Control THS-PREPARE of the Master Data Lanes.											
23 to 16	THSZEROCTL [7:0]		0h	RW	Control THS-ZERO of the Master Data Lanes.											
15 to 8	THSTRAILCTL [7:0]		0h	RW	Control THS-TRAIL of the Master Data Lanes.											
7 to 0	THSEXITCTL [7:0]		0h	RW	Control THS-EXIT of the Master Clock and Data Lanes.											

9.5.2.3.7 DSI PHY TLPX Setting Register (DSI_DPHY_PHYTLPXSETR)

Access Size :		8, 16, 32 bits																
Address :		<DSI_DPHY_base> + 0028h																
Initial Value :		0000_0000h																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	-	-	-	-	-	-	-	-	TLPXCTL[7:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW		
Bit	Bit Name		Initial Value		R/W		Description											
31 to 8	-		All 0		R		Reserved Whenever it is read, 0b is read. The written value will be ignored.											
7 to 0	TLPXCTL[7:0]		0h		RW		Control TLPX of the Master Clock and Data Lanes.											

9.5.2.3.8 DSI PHY Control Register (DSI_DPHY_PHYCR)

Access Size :		8, 16, 32 bits														
Address :		<DSI_DPHY_base> + 0030h														
Initial Value :		0000_0000h														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	ULPSEXIT[9:0]									
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Bit	Bit Name		Initial Value	R/W	Description											
31 to 10	-		All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.											
9 to 0	ULPSEXIT[9:0]		0h	RW	ULPS EXIT Counter Value Control Should be set during the initialization or power-up sequence. Set 100 [μsec] x lpcclk / 4.											

9.5.2.3.9 DSI PHY Control 1 Register (DSI_DPHY_PHYC1R)

Access Size : 8, 16, 32 bits

Address : <DSI_DPHY_base> + 0034h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	BIASREFVCNT [1:0]		-	-	-	-	-	-	-	-	-	-	-	CLLTX SLWD WN
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	R	R	R	R	R	R	R	R	R	R	R	RW

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLLTX SLWD WN	CLLTX SLWU P	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29, 28	BIASREFVCNT [1:0]	0h	RW	Bias Reference Voltage 710 m Control 00b: 715 mV (default) 01b: 724 mV 10b: 733 mV 11b: 706 mV
27 to 17	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
16, 15	CLLTXSLWDW N[1:0]	0h	RW	Master Clock Lane's LP-TX Driver Slew Rate Down Control 00b: No change (default) 01b: Decrease the slew rate by about 15% 10b: Decrease the slew rate by about 15% 11b: Decrease the slew rate by about 30%
14	CLLTXSLWUP	0h	RW	Master Clock Lane's LP-TX Driver Slew Rate Up Control 0b: No change (default) 1b: Slew rate up
13 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.3.10 DSI PHY Control 2 Register (DSI_DPHY_PHYC2R)

Access Size : 8, 16, 32 bits

Address : <DSI_DPHY_base> + 0038h

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DL3LTXSLEWD WN[1:0]		DL3LT XSLEW UP	DL2LTXSLEWD WN[1:0]		DL2LT XSLEW UP	DL1LTXSLEWD WN[1:0]		DL1LT XSLEW UP	DL0LTXSLEWD WN[1:0]		DL0LT XSLEW UP	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	DL3LTXSLEWD WN[1:0]	0h	RW	Master Data 3 Lane's LP-TX Driver Slew Rate Down Control 00b: No change (default) 01b: Decrease the slew rate by about 15% 10b: Decrease the slew rate by about 15% 11b: Decrease the slew rate by about 30%
29	DL3LTXSLEWU P	0h	RW	Master Data 3 Lane's LP-TX Driver Slew Rate Up Control 0b: No change (default) 1b: Slew rate up
28, 27	DL2LTXSLEWD WN[1:0]	0h	RW	Master Data 2 Lane's LP-TX Driver Slew Rate Down Control 00b: No change (default) 01b: Decrease the slew rate by about 15% 10b: Decrease the slew rate by about 15% 11b: Decrease the slew rate by about 30%
26	DL2LTXSLEWU P	0h	RW	Master Data 2 Lane's LP-TX Driver Slew Rate Up Control 0b: No change (default) 1b: Slew rate up
25, 24	DL1LTXSLEWD WN[1:0]	0h	RW	Master Data 1 Lane's LP-TX Driver Slew Rate Down Control 00b: No change (default) 01b: Decrease the slew rate by about 15% 10b: Decrease the slew rate by about 15% 11b: Decrease the slew rate by about 30%
23	DL1LTXSLEWU P	0h	RW	Master Data 1 Lane's LP-TX Driver Slew Rate Up Control 0b: No change (default) 1b: Slew rate up
22, 21	DL0LTXSLEWD WN[1:0]	0h	RW	Master Data 0 Lane's LP-TX Driver Slew Rate Down Control 00b: No change (default) 01b: Decrease the slew rate by about 15% 10b: Decrease the slew rate by about 15% 11b: Decrease the slew rate by about 30%
20	DL0LTXSLEWU P	0h	RW	Master Data 0 Lane's LP-TX Driver Slew Rate Up Control 0b: No change (default) 1b: Slew rate up
19 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.2.3.11 DSI PHY Control 3 Register (DSI_DPHY_PHYC3R)

Access Size : 8, 16, 32 bits

Address : <DSI_DPHY_base> + 003Ch

Initial Value : 0000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	HSTXRFTIM[2:0]			-	-	-	-	HSTXTMIMPD[2:0]			HSTXTMIMPU[2:0]			-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	RW	RW	RW	R	R	R	R	RW	RW	RW	RW	RW	RW	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
29 to 27	HSTXRFTIM [2:0]	0h	RW	HS-TX Rise & Fall Time Control 000b: 135 ps (default) 001b: 130 ps 010b: 125 ps 011b: 120 ps 100b: 230 ps 101b: 225 ps 110b: 220 ps 111b: 215 ps This is common to both Clock Lane and Data Lane.
26 to 23	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.
22 to 20	HSTXTMIMPD [2:0]	0h	RW	HS-TX Driver Termination Impedance Control (Down) 000b: 50 ohm (default) 001b: 52 ohm 010b: 54 ohm 011b: 56 ohm 100b: 44 ohm 101b: 46 ohm 110b: 47 ohm 111b: 48 ohm Note: Set the same value to HSTXMIMPU and HSTXMIMPD. Note: These bits are common to both Clock Lane and Data Lane.
19 to 17	HSTXTMIMPU [2:0]	0h	RW	HS-TX Driver Termination Impedance Control (Up) 000b: 50 ohm (default) 001b: 52 ohm 010b: 54 ohm 011b: 56 ohm 100b: 44 ohm 101b: 46 ohm 110b: 47 ohm 111b: 48 ohm Note: Set the same value to HSTXMIMPU and HSTXMIMPD. Note: These bits are common to both Clock Lane and Data Lane.
16 to 0	-	All 0	R	Reserved Whenever it is read, 0b is read. The written value will be ignored.

9.5.3 Operation

This section shows various operation procedures for using the DSI-Tx Module. As for the interrupt operation, Interrupt Enable bit of each interrupt source is enabled (set to 1b) before use, and after confirming the interrupt, the interrupt source is cleared. And more, previous action's interrupt source must be cleared before setting the interrupt enable. Flows in this section indicate only interrupt bit. The DSI-Tx Module does not care whether firmware use the bits by interrupt action or flag polling action. These descriptions are omitted in each flow in this section unless otherwise specified. Error checks are also omitted in normal flow.

9.5.3.1 Power on Reset and Initial Settings for All Operations

The power on/off sequence of the DSI-Tx Module is shown **Figure 9.5-4**.

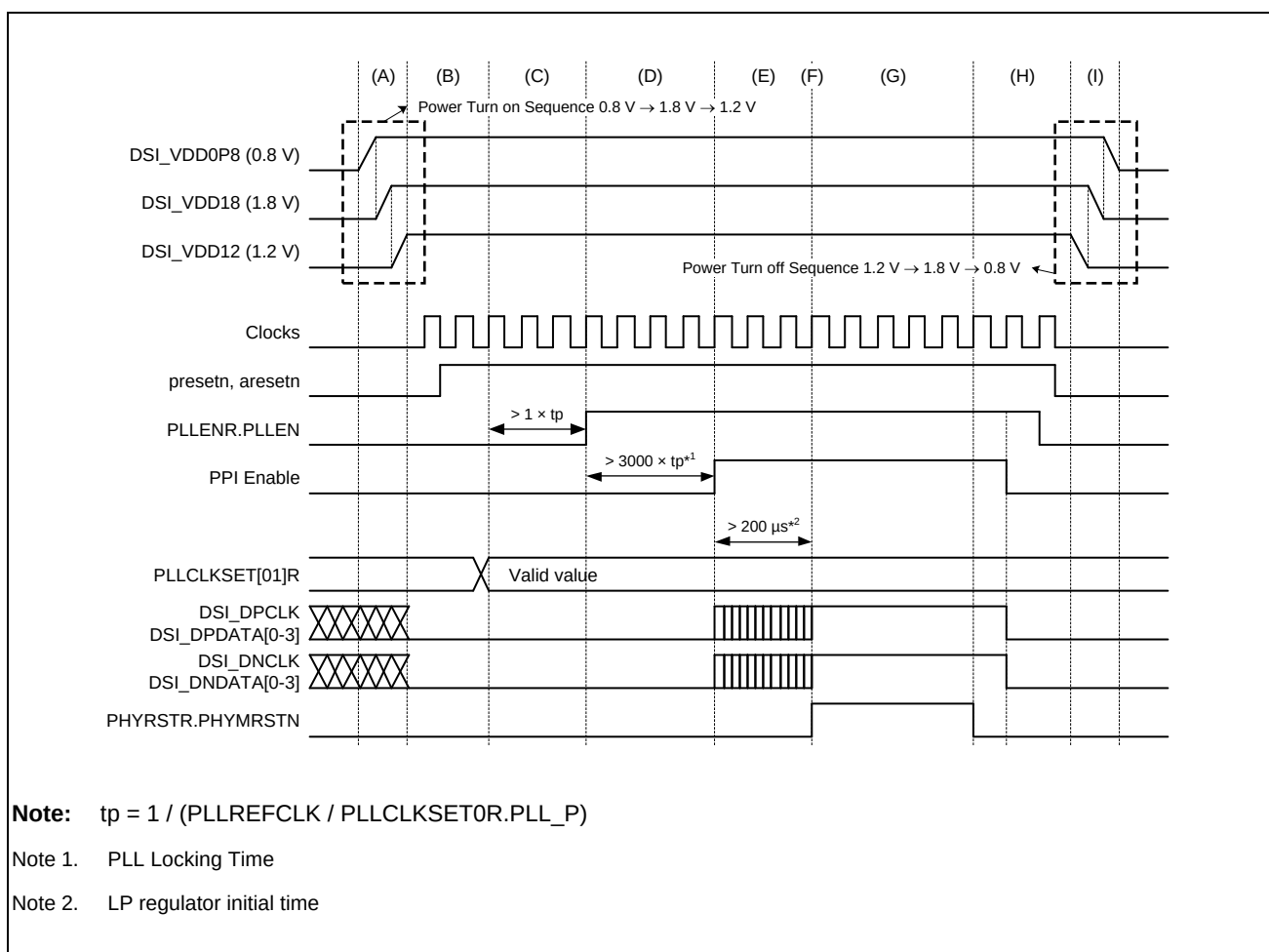


Figure 9.5-4 Power On/Off Sequence

- (A) Bring up DSI_VDD0P8, DSI_VDD18, and DSI_VDD12 in that order.
- (B) De-assert presetn and aresetn after clocks are stable.
- (C) Write appropriate values to D-PHY registers except PLLENR and PHYRSTR, then wait for more than $1 \times tp$.
- (D) Write 1b to PLLENR.PLLEN, then wait for more than $3000 \times tp$.
- (E) Write TXSETR, then wait for more than 200 μsec.

(F) Write 1b to PHYRSTR.PHYMRSTN and write LINK registers.

(G) The DSI-Tx Module is ready.

At power-on reset, all registers of the DSI-Tx Module are initialized.

Perform the initial settings of the register at Step (C) and (F) in the power-on sequence. It is not necessary to set the register that is used with the default value.

The value to be written to the D-PHY register in Step (C) is as follows.

- Set PLLCLKSET0R and PLLCLKSET1R so that the PLL clock frequency (F_{FOUT}) equals the data rate.
- Follow **Table 9.5-4** to set PHYTCLKSETR, PHYTHSSETR, and PHYTLPXSETR.
- Follow **Table 9.5-5** to set PHYCR.
- Follow register descriptions to set PHYC1R, PHYC2R, and PHYC3R.

Table 9.5-4 Recommend Setting for PHYTCLKSETR, PHYTHSSETR, and PHYTLPXSETR (1/4)

Data Rate [Mbps]		TCLKPRPRCTL[7:0]	TCLKZEROCTL[7:0]	TCLKPOSTCTL[7:0]	TCLKTRAILCTL[7:0]	THSPRPRCTL[7:0]	THSZEROCTL[7:0]	THSTRAILCTL[7:0]	TLPXCTL[7:0]	THSEXITCTL[7:0]
		Lower (>)	Upper (<=)							
1490	1500	13	65	17	13	14	24	16	11	18
1480	1490	13	65	17	13	14	24	16	11	18
1470	1480	13	64	17	13	14	24	16	11	18
1460	1470	13	64	17	13	14	24	16	11	18
1450	1460	13	63	17	13	13	24	16	10	18
1440	1450	13	63	17	13	13	23	16	10	18
1430	1440	13	63	17	13	13	23	16	10	18
1420	1430	12	62	17	13	13	23	16	10	17
1410	1420	12	62	17	13	13	23	16	10	17
1400	1410	12	61	16	13	13	23	16	10	17
1390	1400	12	61	16	13	13	23	16	10	17
1380	1390	12	60	16	12	13	22	15	10	17
1370	1380	12	60	16	12	13	22	15	10	17
1360	1370	12	59	16	12	13	22	15	10	17
1350	1360	12	59	16	12	13	22	15	10	17
1340	1350	12	59	16	12	12	22	15	10	16
1330	1340	12	58	16	12	12	21	15	10	16
1320	1330	11	58	16	12	12	21	15	9	16
1310	1320	11	57	16	12	12	21	15	9	16
1300	1310	11	57	16	12	12	21	15	9	16
1290	1300	11	56	16	12	12	21	15	9	16
1280	1290	11	56	16	12	12	21	15	9	16
1270	1280	11	56	15	11	12	20	14	9	16
1260	1270	11	55	15	11	12	20	14	9	15
1250	1260	11	55	15	11	12	20	14	9	15

Table 9.5-4 Recommend Setting for PHYCLKSETR, PHYTHSSETR, and PHYTLPXSETR (2/4)

Data Rate [Mbps]		TCLKPRPRCTL[7:0]	TCLKZEROCTL[7:0]	TCLKPOSTCTL[7:0]	TCLKTRAILCTL[7:0]	THSPRPRCTL[7:0]	THSZEROCTL[7:0]	THSTRAILCTL[7:0]	TLPXCTL[7:0]	THSEXITCTL[7:0]
1240	1250	11	54	15	11	11	20	14	9	15
1230	1240	11	54	15	11	11	20	14	9	15
1220	1230	11	53	15	11	11	19	14	9	15
1210	1220	10	53	15	11	11	19	14	9	15
1200	1210	10	52	15	11	11	19	14	9	15
1190	1200	10	52	15	11	11	19	14	9	15
1180	1190	10	52	15	11	11	19	14	8	14
1170	1180	10	51	15	11	11	19	13	8	14
1160	1170	10	51	15	10	11	18	13	8	14
1150	1160	10	50	15	10	11	18	13	8	14
1140	1150	10	50	15	10	11	18	13	8	14
1130	1140	10	49	14	10	10	18	13	8	14
1120	1130	10	49	14	10	10	18	13	8	14
1110	1120	10	49	14	10	10	17	13	8	14
1100	1110	9	48	14	10	10	17	13	8	13
1090	1100	9	48	14	10	10	17	13	8	13
1080	1090	9	47	14	10	10	17	13	8	13
1070	1080	9	47	14	10	10	17	13	8	13
1060	1070	9	46	14	10	10	17	12	8	13
1050	1060	9	46	14	10	10	16	12	7	13
1040	1050	9	45	14	9	10	16	12	7	13
1030	1040	9	45	14	9	10	16	12	7	13
1020	1030	9	45	14	9	9	16	12	7	12
1010	1020	9	44	14	9	9	16	12	7	12
1000	1010	8	44	13	9	9	15	12	7	12
990	1000	8	43	13	9	9	15	12	7	12
980	990	8	43	13	9	9	15	12	7	12
970	980	8	42	13	9	9	15	12	7	12
960	970	8	42	13	9	9	15	12	7	12
950	960	8	42	13	9	9	15	11	7	12
940	950	8	41	13	9	9	14	11	7	11
930	940	8	41	13	8	9	14	11	7	11
920	930	8	40	13	8	8	14	11	6	11
910	920	8	40	13	8	8	14	11	6	11
900	910	8	39	13	8	8	14	11	6	11
890	900	7	39	13	8	8	13	11	6	11
880	890	7	38	13	8	8	13	11	6	11
870	880	7	38	12	8	8	13	11	6	11
860	870	7	38	12	8	8	13	11	6	10
850	860	7	37	12	8	8	13	11	6	10
840	850	7	37	12	8	8	13	10	6	10

Table 9.5-4 Recommend Setting for PHYCLKSETR, PHYTHSSETR, and PHYTLPXSETR (3/4)

Data Rate [Mbps]		TCLKPRPRCTL[7:0]	TCLKZEROCTL[7:0]	TCLKPOSTCTL[7:0]	TCLKTRAILCTL[7:0]	THSPRPRCTL[7:0]	THSZEROCTL[7:0]	THSTRAILCTL[7:0]	TLPXCTL[7:0]	THSEXITCTL[7:0]
830	840	7	36	12	8	8	12	10	6	10
820	830	7	36	12	8	8	12	10	6	10
810	820	7	35	12	7	7	12	10	6	10
800	810	7	35	12	7	7	12	10	6	10
790	800	7	35	12	7	7	12	10	6	10
780	790	6	34	12	7	7	11	10	5	9
770	780	6	34	12	7	7	11	10	5	9
760	770	6	33	12	7	7	11	10	5	9
750	760	6	33	12	7	7	11	10	5	9
740	750	6	32	12	7	7	11	9	5	9
730	740	6	32	11	7	7	11	9	5	9
720	730	6	31	11	7	7	10	9	5	9
710	720	6	31	11	7	6	10	9	5	9
700	710	6	31	11	6	6	10	9	5	8
690	700	6	30	11	6	6	10	9	5	8
680	690	5	30	11	6	6	10	9	5	8
670	680	5	29	11	6	6	9	9	5	8
660	670	5	29	11	6	6	9	9	5	8
650	660	5	28	11	6	6	9	9	4	8
640	650	5	28	11	6	6	9	9	4	8
630	640	5	28	11	6	6	9	8	4	8
620	630	5	27	11	6	6	9	8	4	7
610	620	5	27	11	6	6	8	8	4	7
600	610	5	26	10	6	5	8	8	4	7
590	600	5	26	10	6	5	8	8	4	7
580	590	5	25	10	5	5	8	8	4	7
570	580	4	25	10	5	5	8	8	4	7
560	570	4	24	10	5	5	7	8	4	7
550	560	4	24	10	5	5	7	8	4	7
540	550	4	24	10	5	5	7	8	4	6
530	540	4	23	10	5	5	7	8	4	6
520	530	4	23	10	5	5	7	7	3	6
510	520	4	22	10	5	5	7	7	3	6
500	510	4	22	10	5	5	6	7	3	6
490	500	4	21	10	5	4	6	7	3	6
480	490	4	21	10	5	4	6	7	3	6
470	480	4	21	9	4	4	6	7	3	6
460	470	3	20	9	4	4	6	7	3	5
450	460	3	20	9	4	4	5	7	3	5
440	450	3	19	9	4	4	5	7	3	5
430	440	3	19	9	4	4	5	7	3	5

Table 9.5-4 Recommend Setting for PHYTCLKSETR, PHYTHSSETR, and PHYTLPXSETR (4/4)

Data Rate [Mbps]		TCLKPRPRCTL[7:0]	TCLKZEROCTL[7:0]	TCLKPOSTCTL[7:0]	TCLKTRAILCTL[7:0]	THSPRPRCTL[7:0]	THSZEROCTL[7:0]	THSTRAILCTL[7:0]	TLPXCTL[7:0]	THSEXITCTL[7:0]
420	430	3	18	9	4	4	5	7	3	5
410	420	3	18	9	4	4	5	6	3	5
400	410	3	17	9	4	4	5	6	3	5
390	400	3	17	9	4	3	4	6	3	5
380	390	3	17	9	4	3	4	6	2	4
370	380	3	16	9	4	3	4	6	2	4
360	370	2	16	9	3	3	4	6	2	4
350	360	2	15	9	3	3	4	6	2	4
340	350	2	15	9	3	3	3	6	2	4
330	340	2	14	8	3	3	3	6	2	4
320	330	2	14	8	3	3	3	6	2	4
310	320	2	14	8	3	3	3	5	2	4
300	310	2	13	8	3	3	3	5	2	3
290	300	2	13	8	3	3	3	5	2	3
280	290	2	12	8	3	2	2	5	2	3
270	280	2	12	8	3	2	2	5	2	3
260	270	2	11	8	3	2	2	5	2	3
250	260	1	11	8	3	2	2	5	1	3
240	250	1	10	8	2	2	2	5	1	3
230	240	1	9	8	2	2	1	5	1	3
220	230	1	8	8	2	2	1	5	1	2
210	220	1	8	8	2	2	1	5	1	2
200	210	1	7	7	2	2	1	4	1	2
190	200	1	7	7	2	2	1	4	1	2
180	190	1	7	7	2	1	1	4	1	2
170	180	1	6	7	2	1	0	4	1	2
160	170	1	6	7	2	1	0	4	1	2
150	160	1	6	7	2	1	0	4	1	2
140	150	0	5	7	2	1	0	4	1	1
130	140	0	5	7	1	1	0	4	1	1
120	130	0	4	7	1	1	0	4	0	1
110	120	0	4	7	1	1	0	4	0	1
100	110	0	3	7	1	0	0	4	0	1
90	100	0	3	7	1	0	0	3	0	1
80	90	0	2	7	1	0	0	3	0	1
—	80	0	2	6	1	0	0	3	0	1

Table 9.5-5 Recommend Setting for PHYCR

lpclk [MHz]	ULPSEXIT[9:0]
15.625	391
7.8125	195
3.90625	98
1.953125	49

The LINK registers that need to be initialized in Step (F) are as follows.

- Common Settings
 - TXSETR
 - ULPSSETR
 - DSISETR
 - CLSTPTSETR
 - LPTRNSTSETR
- Timeout Settings
 - PRESPTOBTASETR
 - PRESPTOLPSETR
 - PRESPTOHSSETR
 - HSTXTOSETR
 - LRXHTOSETR
 - TATOSETR

These registers and D-PHY registers can only be changed during the initial setup after a power-on reset and during a software reset in **9.5.3.2.1 Software Reset**. It is prohibited to change at any other time.

9.5.3.2 Reset

9.5.3.2.1 Software Reset

Figure 9.5-5 shows the operation during software reset. Make sure that APB clock (pclk), Internal Bus clock (aclk), Video clock (vclk1), PLL Multiplied clock (PLLREFCLK), and Escape mode Transmit clock (lpclk) are running and PHYRSTR.PHYMRSTN is high before starting software reset.

Writing 1b to RSTCR.SWRST will issue a software reset. After issuing the software reset, check the start of the software reset process with RSTSR.SWRST_{xx} = 1b. Then write 1b to RSTCR.FCETXSTP to transition the valid data lane of the D-PHY to the Tx Stop State. By checking RSTSR.DL0DIR = 0b and RSTSR.DLSTPST [x] = 1b of the valid data lane, it is possible to confirm that the valid data lane of D-PHY has transitioned to Tx Stop State. After confirming that the valid data lane of the D-PHY has transitioned to the Tx Stop State, write 0b to RSTCR.FCETXSTP. Then write 0b to RSTCR.SWRST and check the completion of the software reset process with RSTSR.SWRST_x = 0b. Since all registers are synchronously reset except for some registers*¹, the software reset process is not completed even if a software reset is issued while the clock is stopped.

Note 1. Registers that require initial settings as described in **9.5.3.1 Power on Reset and Initial Settings for All Operations**, GPO0R register, GPO1R register, RSTCR register, RSTSR register, and SQCHxDSCyAR – SQCHxDSCyDR register are not initialized by software reset.

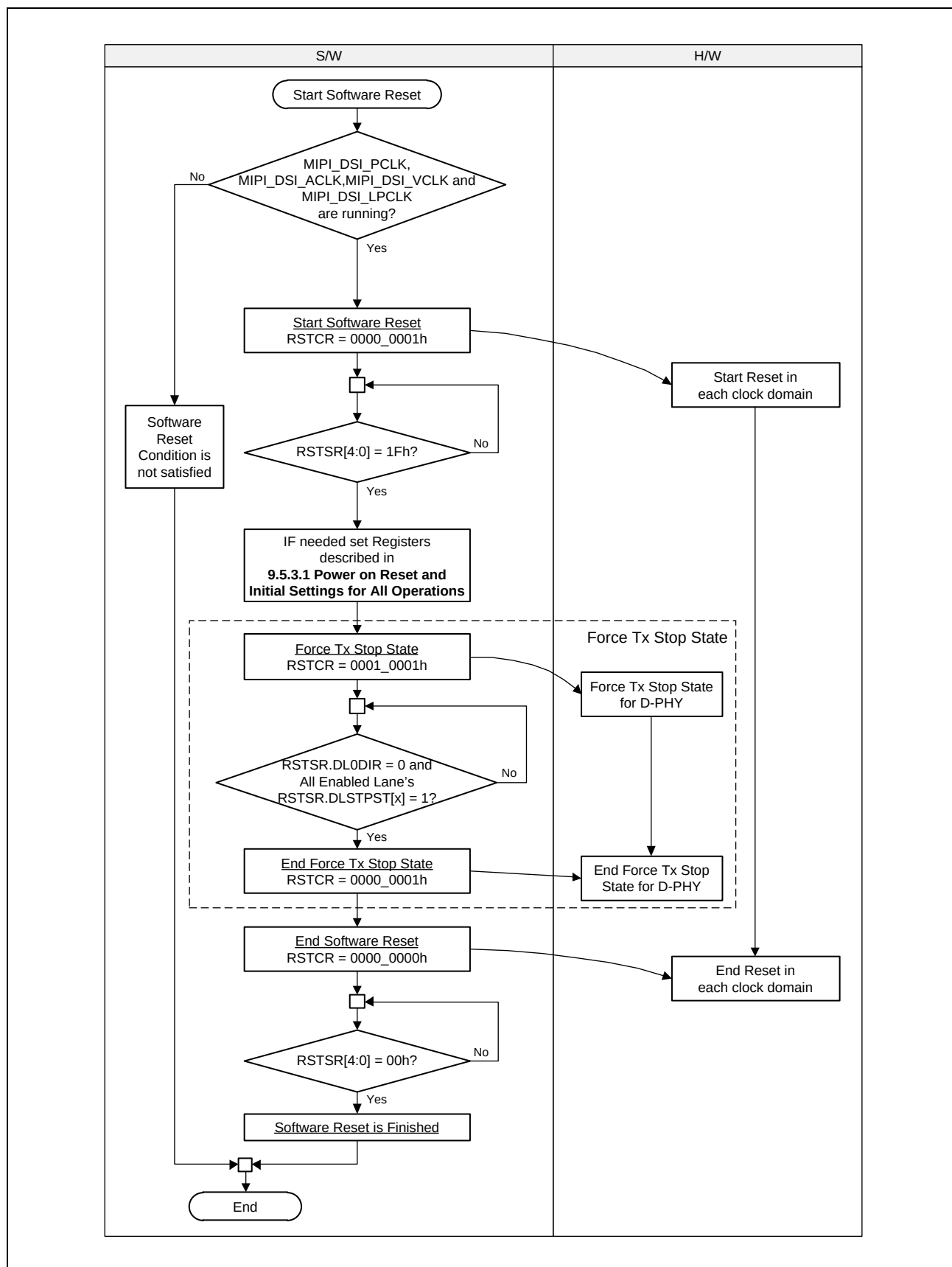


Figure 9.5-5 Software Reset

9.5.3.3 Start/Stop of HS Clock

The following describes the HS Clock Start / Stop sequence. HS Clock Start / Stop is controlled by software.

Figure 9.5-6 shows the operation at HS Clock Start.

Write 1b to TXSETR.CLLEN to enable DPHY clock lanes before starting Video-Input Operation or Sequence Operation channel 1 operation.

After the clock lane is enabled and the source clock of the clock lane is stable, write 1b to HSCLKSETR.HSCLKRUN to set the clock lane to HS. It is forbidden to change HSCLKRUN while Sequence Channel is in operation.

At the same time as writing 1b to HSCLKSETR.HSCLKRUN, set continuous clock mode and non-continuous clock mode with HSCLKSETR.HSCLKMODE.

In continuous clock mode, always set the clock lane to HS. PLSR.CLLP2HS is set to 1b when the clock lane transitions from LP to HS. After checking, perform the following operations.

In non-continuous clock mode, the DSI-Tx Module automatically transitions the clock lane from LP to HS depending on the presence or absence of HS Transmission. PLSR.CLLP2HS and PLSR.CLHS2LP are set for each transition, but confirmation is not required.

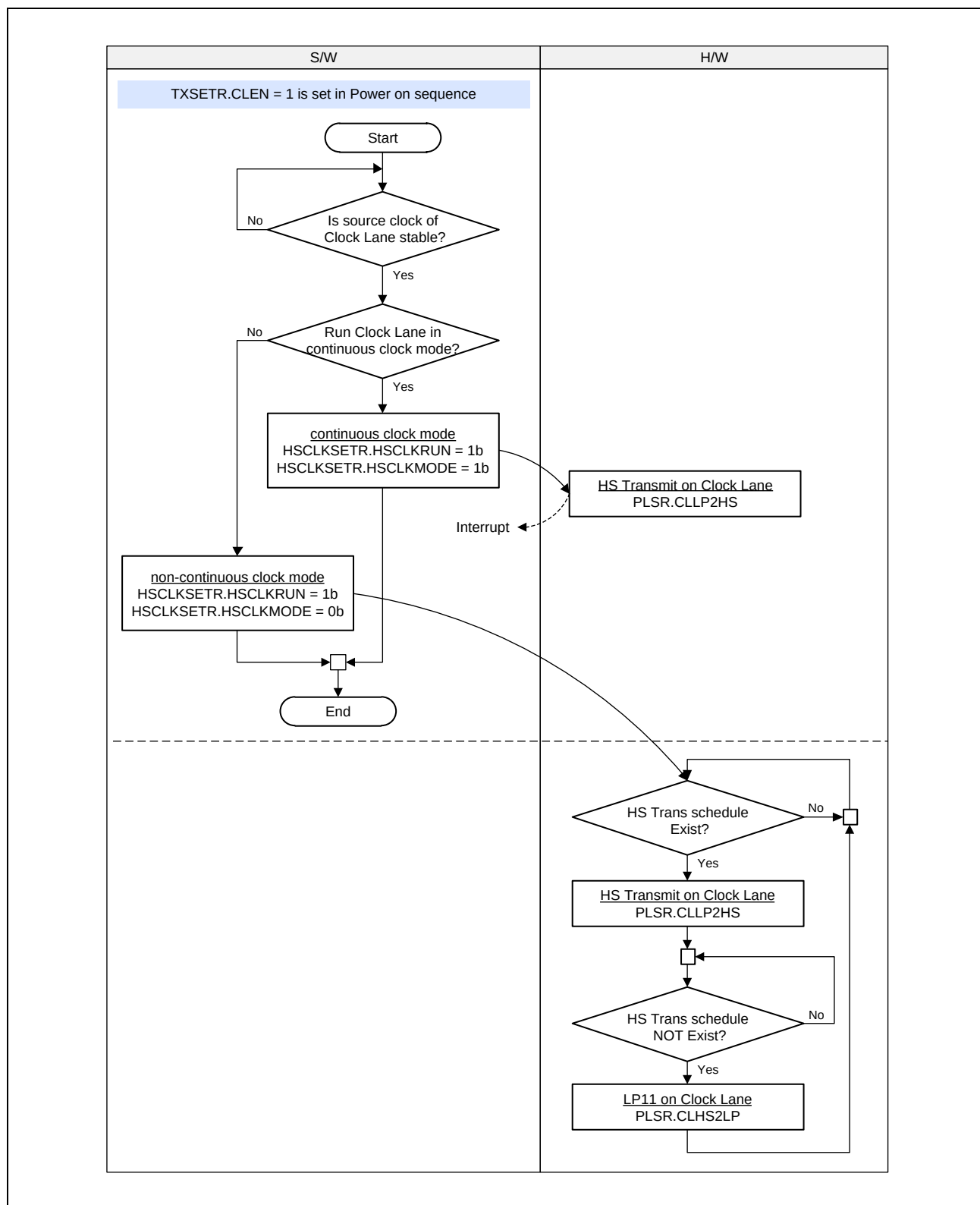


Figure 9.5-6 Start of HS Clock

Figure 9.5-7 shows the operation when HS Clock Stop. After confirming that Video-Input Operation and Sequence Operation channel 1 are not running, write 0b to HCLKSETR.HSCLKRUN to make the clock lane LP. PLSR.CLHS2LP is set to 1b when the clock lane transitions from HS to LP. In non-continuous clock mode, you do not need to check PLSR.CLHS2LP because the clock lane is likely to be LP already when you write 0b to HCLKSETR.HSCLKRUN. It is possible to check the clock lane status by reading PLSR.CLSTPST.

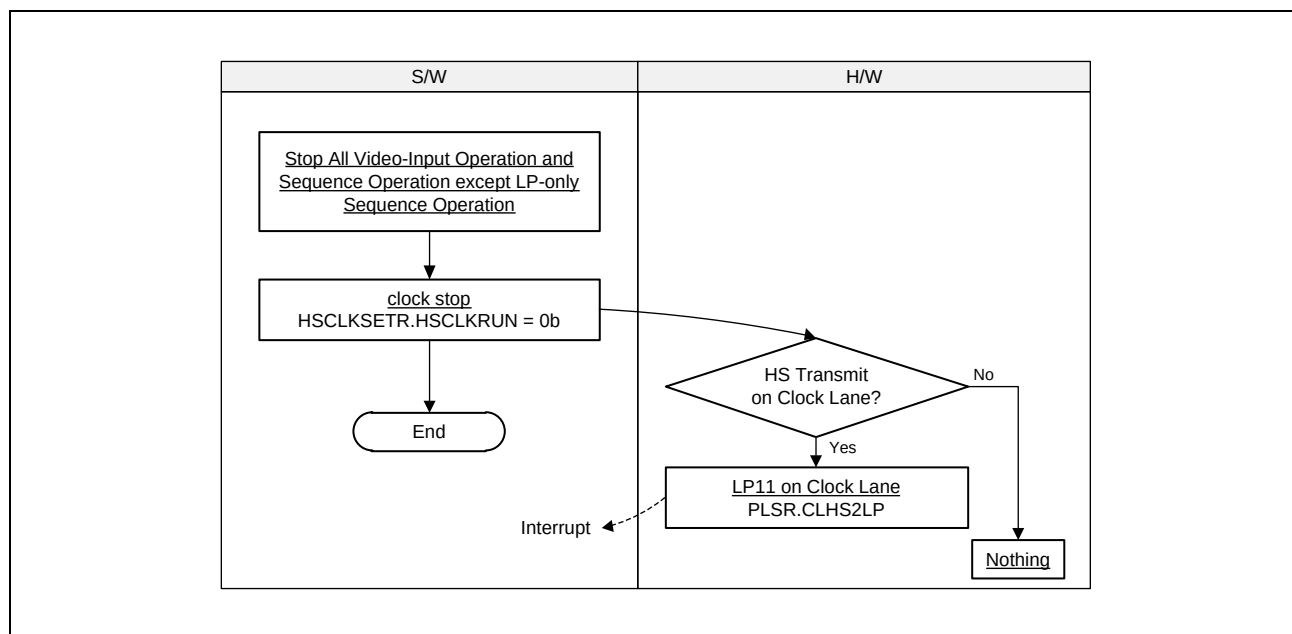


Figure 9.5-7 Stop of HS Clock

9.5.3.4 Sequence Operation

9.5.3.4.1 Basic Running Sequence

Sequence Operation requires initial settings before the operation starts.

- Channel Register Setting
 - Set SQCHxSET1R, SQCHxIER
- Prepare Write Data
 - Prepare Write Data to Memory Area
 - Prepare Write Data to TXPPDxR Registers
- Allocate Read Data Area
 - Clear RXRSSR Register if previous valid flag remains.
 - Allocate each read action's Memory area.
- Write all Descriptors
- Confirm running HSCLK if Sequence channel 1 is targeted to run.

When “1b” is written to SQCHxSET0R.START, Sequence Operation channel x starts the processing of Descriptor, and when the processing of Descriptor of SQCHxDSCyAR.NXACT = “00b” is completed, the processing of Descriptor ends.

Since Descriptor consists of SRAM, the initial value is indefinite. When the user sets the Descriptor, set all the bits.

After writing 1b to SQCHxSET0R.START on Sequence Operation channel x and SQCHxSR.RUNNING becomes 1b, It is prohibited to change the settings of SQCHxSET1R, SQCHxDSCyAR, SQCHxDSCyBR, SQCHxDSCyCR, and SQCHxDSCyDR until SQCHxSR.RUNNING becomes 0b.

HS-related operations are prohibited on Sequence Operation channel 0.

9.5.3.4.2 Single Packet Transmission

The following describes sending non-read packets using the Sequence Operation channel.

For non-read packet transmission, set the following registers before packet transmission.

- SQCHxDSCyAR.NXACT: “00b”
- SQCHxDSCyAR.FMT: “0b” Short Packet or “1b” Long Packet
- SQCHxDSCyAR.SPD: “0b” High Speed or “1b” Low Speed
- SQCHxDSCyAR.BTA: “00b” without BTA or “01b” with BTA
- SQCHxDSCyAR.VC, DT, DATA0, DATA1
- SQCHxDSCyBR.DTSEL: Specify payload data storage location for non-read packets
 - “00b” TXPPD0R – TXPPD3R. Payloads size is limited up to 16 Bytes.
 - “01b” Memory space. Payloads size is limited up to SQCHxSET1R.CHBUFSZ Bytes.
- SQCHxDSCyCR.AUXOP “0b”

When setting a Long Packet with SQCHxDSCyAR.FMT, prepare the payload data in the location specified by SQCHxDSCyBR.DTSEL in the form of little endian.

No payload data is required when configuring Short Packets with SQCHxDSCyAR.FMT.

After the above settings, write “1b” to SQCHxSET0R.START to send non-read packets.

The user can know the completion of Single Packet Transmission by asserting SQCHxSR.ADESFIN.

If “01b” is set in SQCHxDSCyAR.BTA, BTA is performed after sending a non-read packet.

An ACK trigger or Acknowledge and Error Report Packet may be received when the bus rights are transferred to the peripheral.

The user can get an overview of the reception result by reading the RXRSSxR register with the number specified by SQCHxDSCyCR.ACTCODE. Also, by reading the RXSR register, the user can know the details of the reception result.

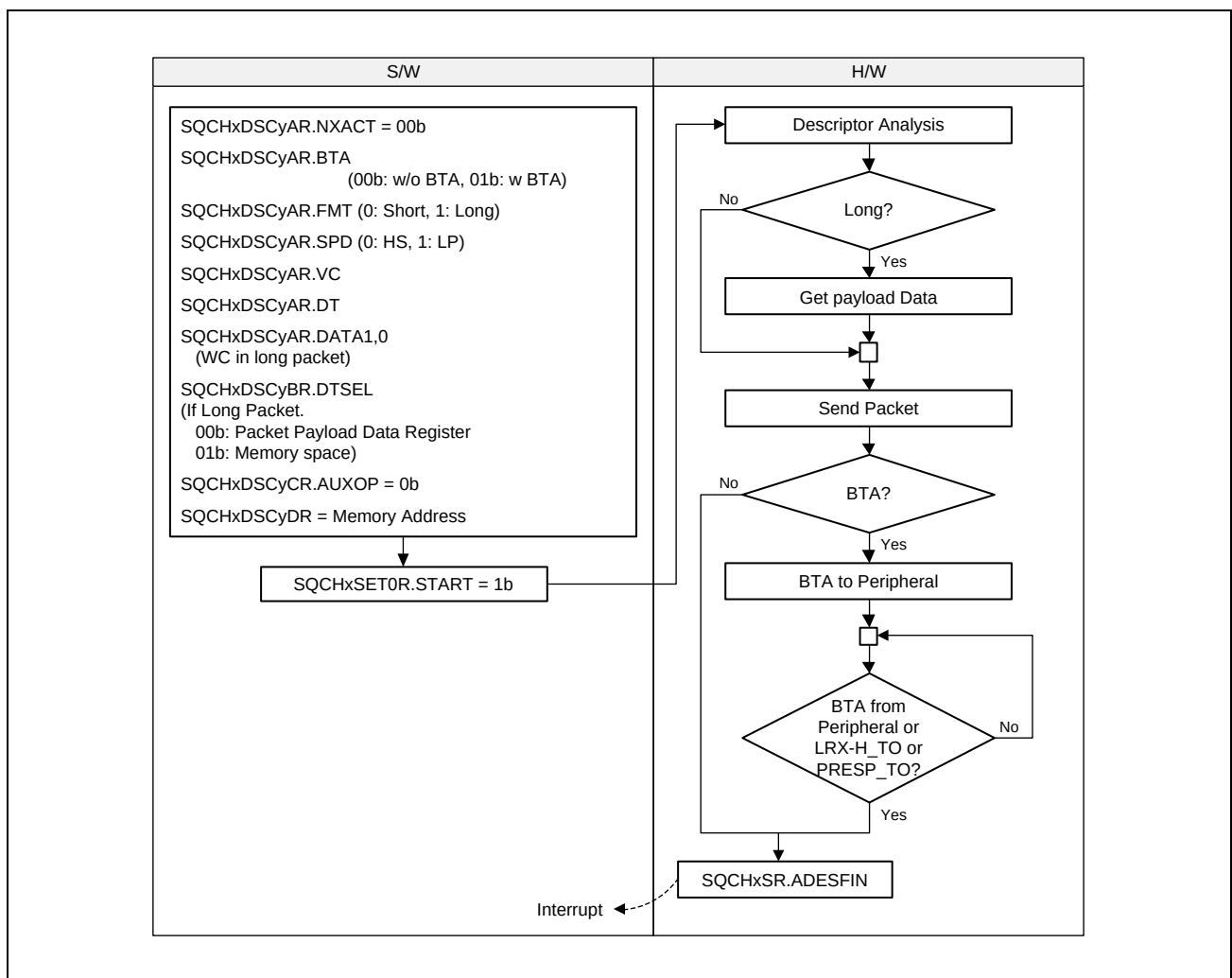


Figure 9.5-8 Single Packet Transfer

9.5.3.4.3 Single Packet Reception

This section describes sending read packets and receiving response packets using the Sequence Operation channel.

Since the read packet is a short packet, set the following registers before sending the packet.

- SQCHxDSCyAR.NXACT: “00b”
- SQCHxDSCyAR.FMT: “0b” Short Packet
- SQCHxDSCyAR.SPD: “0b” High Speed or “1b” Low Speed
- SQCHxDSCyAR.BTA: “10b” Read Request with BTA
- SQCHxDSCyAR.VC, DT, DATA0, DATA1
- SQCHxDSCyBR.DTSEL: Specify the payload data storage location of the response packet
 - “00b” RXPPD0R – RXPPD3R. Payloads size is limited up to 16 Bytes.
 - “01b” Memory space. Payloads size is limited up to RXBUFSZR.RXBUFSZ Bytes.
- SQCHxDSCyCR.AUXOP: “0b”
- SQCHxDSCyCR.ACTCODE: Rx Result Save Slot number

When SQCHxDSC00BR.DTSEL = “00b” is set, receiving payload data of 17 bytes or more is not supported. Therefore, if there is a possibility of receiving payload data of 17 bytes or more, set SQCHxDSC00BR.DTSEL = “01b”.

After the above settings, write “1b” to SQCHxSET0R.START to send the read packet.

BTA is performed after sending the read packet. When the bus right is transferred to the peripheral, the response packet is received. At this time, the Acknowledge and Error Report Packet may also be received at the same time.

The user can know the completion of Single Packet Reception by asserting SQCHxSR.ADESFIN.

The user can get an overview of the reception result by reading the RXRSSxR register with the number specified by SQCHxDSCyCR.ACTCODE, and can know the details of the previous reception result by reading the RXSR register.

If the received response packet is a Short Packet or a Long Packet with WC = 0b, the setting value of SQCHxDSC00BR.DTSEL is meaningless.

Figure 9.5-9 shows the flow of Single Packet Reception.

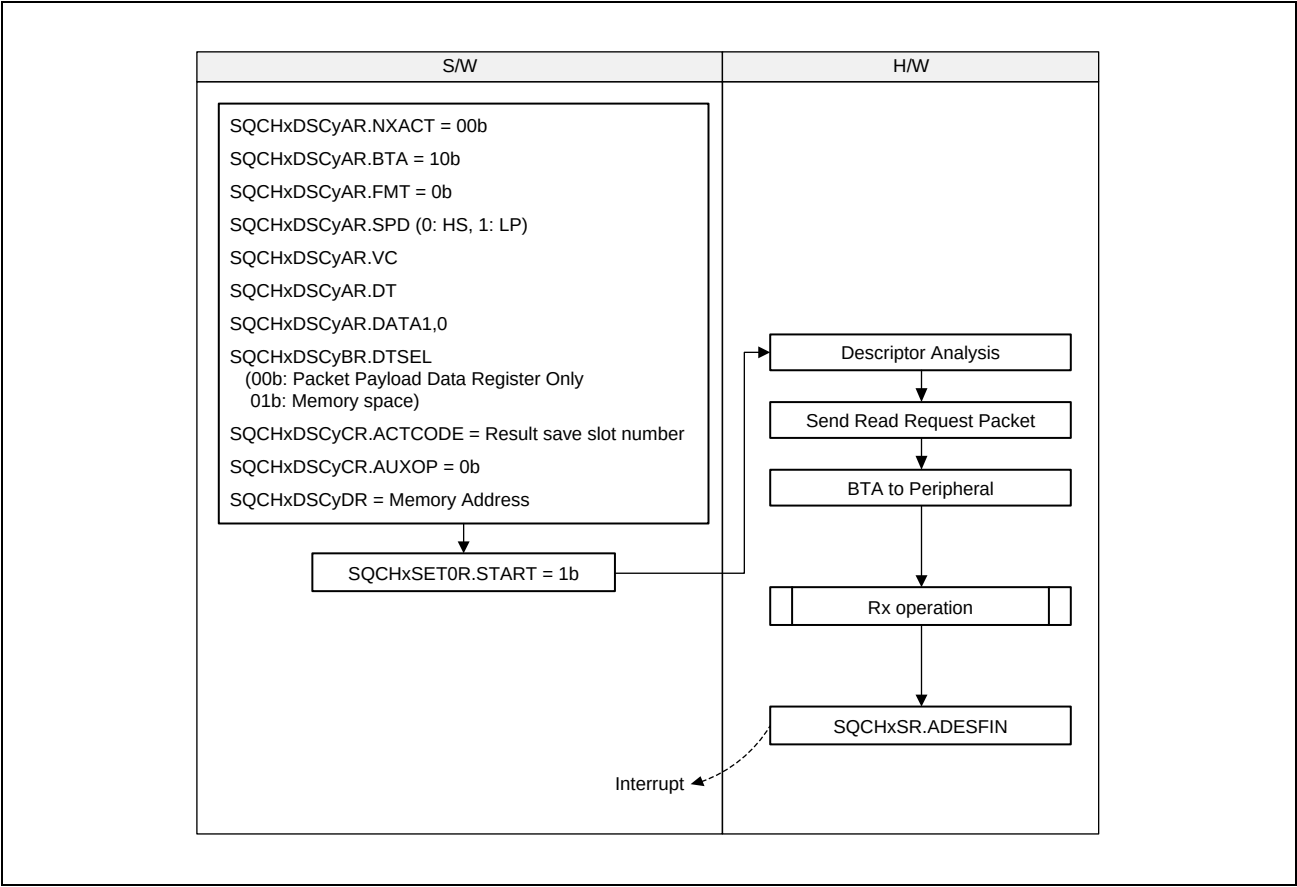


Figure 9.5-9 Single Packet Receive

9.5.3.4.4 Reset Trigger Transmission

The DSI-Tx Module can send triggers using the Sequence Operation channel.

Since the DSI-Tx Module is a host, the only trigger that can be sent is a reset trigger.

After setting `SQCHxDSCyAR.NXACT = "00b"`, `SQCHxDSCyCR.AUXOP = "1b"`, and `SQCHxDSCyCR.ACTCODE = "00h"`, write "1b" to `SQCHxSET0R.START` to send a reset trigger.

The user can know the completion of reset trigger transmission by asserting `SQCHxSR.ADESFIN`.

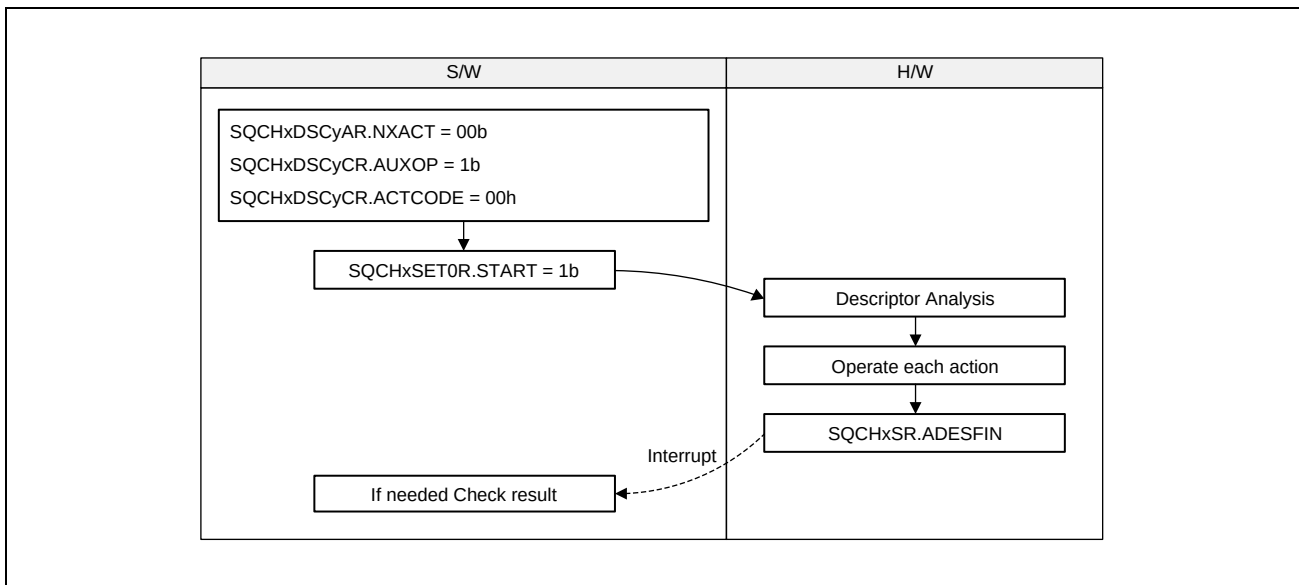


Figure 9.5-10 Trigger Transmission

9.5.3.4.5 Execute Operation in Sequence

The following describes the descriptor in Sequence Operation.

- By writing "1b" to `SQCHxSET0R.START`, Sequence Operation channel x will be enabled and processing will be performed from descriptor # 0.
- When the processing of the corresponding descriptor is completed, the next processing follows `SQCHxDSCyAR.NXACT`.
 - `SQCHxDSCyAR.NXACT = "00b"`:
Sequence Operation channel x stop after the processing of the corresponding descriptor is completed
 - `SQCHxDSCyAR.NXACT = "01b"`:
After the processing of the corresponding descriptor is completed, the next descriptor is executed. The next descriptor after descriptor # N is descriptor # (N + 1).

9.5.3.4.6 Receiving EoTp

The DSI-Tx Module can receive End of Transmission Packets (EoTp).

The DSI-Tx Module notifies the reception of EoTp by setting RXSR.RXEOTP to “1b”.

9.5.3.4.7 Acknowledge and Error Reporting Mechanism

The user can know the reliability of the serial bus by using “Acknowledge and Error Reporting Mechanism”.

If Peripheral detects an error on the serial bus during the period from the previous Peripheral to Host communication to this Peripheral to Host communication, Peripheral will send an “Acknowledge and Error Report” packet.

When the DSI-Tx Module receives the “Acknowledge and Error Report” packet, it sets “1b” to RXRSSxR.RXAKE, RXSR.RXAKE, and SQCHxSR.RXAKE. The Virtual Channel Identifier and Error Report of the latest “Acknowledge and Error Report” packet received are stored in the AKEPLATIR register, and the integrated value is displayed in the AKEPACMSR register.

9.5.3.5 Video-Input Operation

9.5.3.5.1 Video-Input Timing

Video-Input Timing and Video-Input Parameters are described in **Figure 9.5-11** and **Table 9.5-6**.

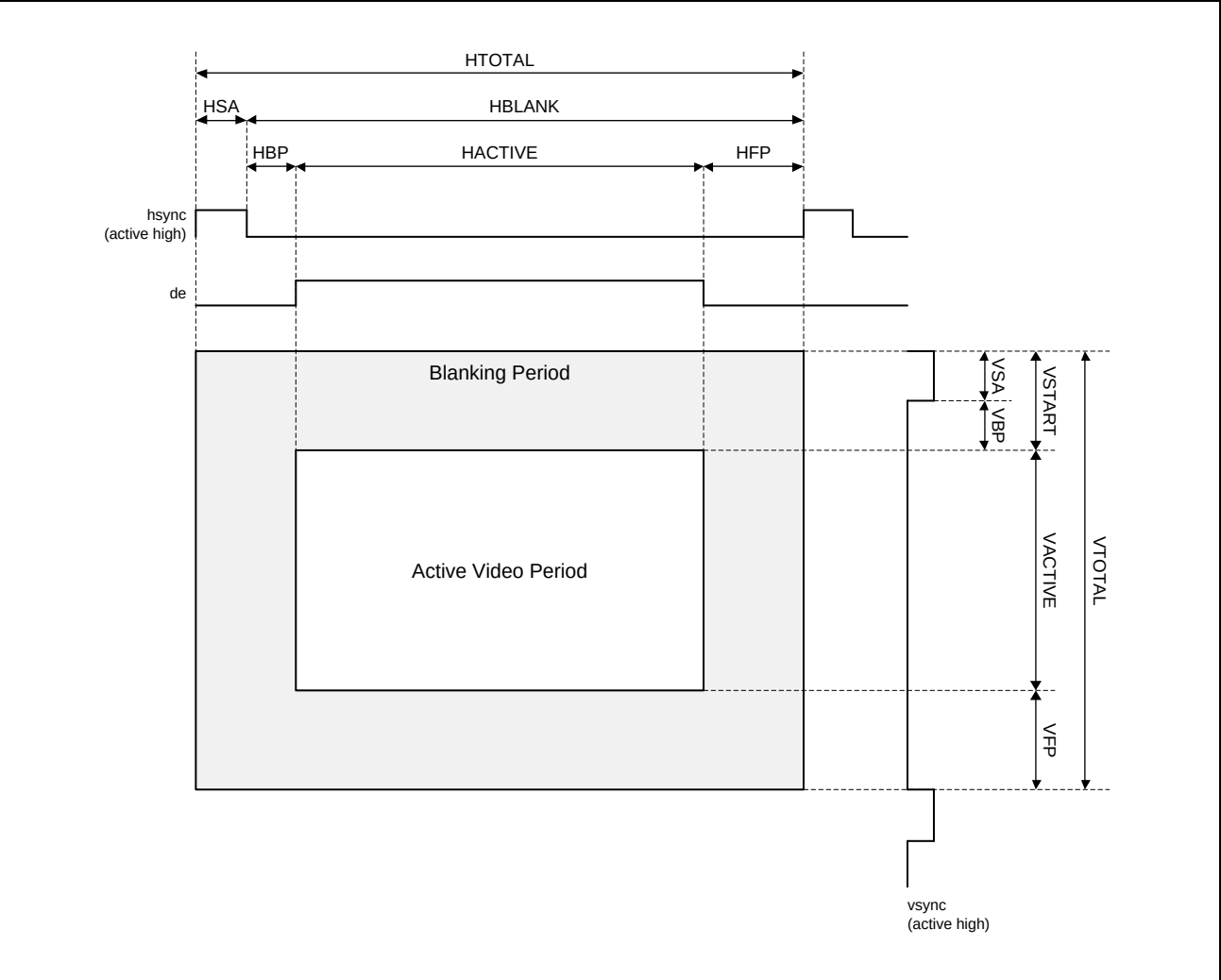


Figure 9.5-11 Video Input Timing

Table 9.5-6 Configurations for Video Input Interface (1/2)

- Common Rules for all mode

Parameter	Description								
All	All parameters related to Input Video should be set to the value of not less than 1b. Example for "All parameters": HSA, HBP, HACTIVE, HFP, VSA, VBP, VACTIVE, VFP								
HACTIVE	<table> <tr> <th>VICH1PPSETR.DT</th><th>VICH1HSSETR.HACTIVE</th></tr> <tr> <td>1Eh: Packed Pixel Stream, 18-bit RGB</td><td>This value should be the multiple of 4.</td></tr> <tr> <td>2Eh: Loosely Packed Pixel Stream, 18-bit RGB</td><td>No limitation</td></tr> <tr> <td>3Eh: Packed Pixel Stream, 24-bit RGB</td><td></td></tr> </table>	VICH1PPSETR.DT	VICH1HSSETR.HACTIVE	1Eh: Packed Pixel Stream, 18-bit RGB	This value should be the multiple of 4.	2Eh: Loosely Packed Pixel Stream, 18-bit RGB	No limitation	3Eh: Packed Pixel Stream, 24-bit RGB	
VICH1PPSETR.DT	VICH1HSSETR.HACTIVE								
1Eh: Packed Pixel Stream, 18-bit RGB	This value should be the multiple of 4.								
2Eh: Loosely Packed Pixel Stream, 18-bit RGB	No limitation								
3Eh: Packed Pixel Stream, 24-bit RGB									
HFP	Round down ($HFP \times BPP / 8, 0$) ≥ 12 , (12 = means Pixel Packet Header & footer 6 + Blanking Packet Header & footer 6)								
HTOTAL	<table> <tr> <th>VICH1PPSETR.DT</th><th>VICH1HSSETR.HSA + VICH1HPSETR.HBP + VICH1HSSETR.HACTIVE + VICH1HPSETR.HFP</th></tr> <tr> <td>1Eh: Packed Pixel Stream, 18-bit RGB</td><td>This value should be the multiple of 4.</td></tr> <tr> <td>2Eh: Loosely Packed Pixel Stream, 18-bit RGB</td><td>No limitation</td></tr> <tr> <td>3Eh: Packed Pixel Stream, 24-bit RGB</td><td></td></tr> </table>	VICH1PPSETR.DT	VICH1HSSETR.HSA + VICH1HPSETR.HBP + VICH1HSSETR.HACTIVE + VICH1HPSETR.HFP	1Eh: Packed Pixel Stream, 18-bit RGB	This value should be the multiple of 4.	2Eh: Loosely Packed Pixel Stream, 18-bit RGB	No limitation	3Eh: Packed Pixel Stream, 24-bit RGB	
VICH1PPSETR.DT	VICH1HSSETR.HSA + VICH1HPSETR.HBP + VICH1HSSETR.HACTIVE + VICH1HPSETR.HFP								
1Eh: Packed Pixel Stream, 18-bit RGB	This value should be the multiple of 4.								
2Eh: Loosely Packed Pixel Stream, 18-bit RGB	No limitation								
3Eh: Packed Pixel Stream, 24-bit RGB									

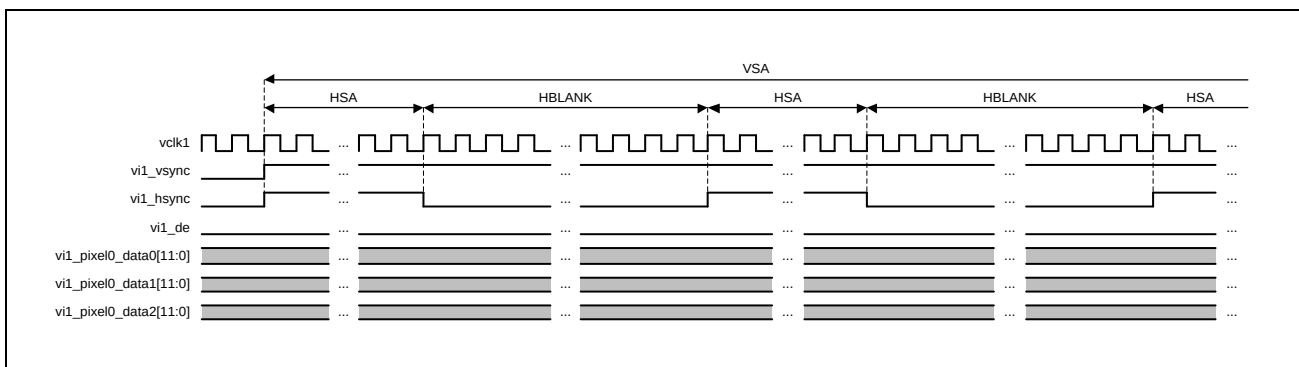
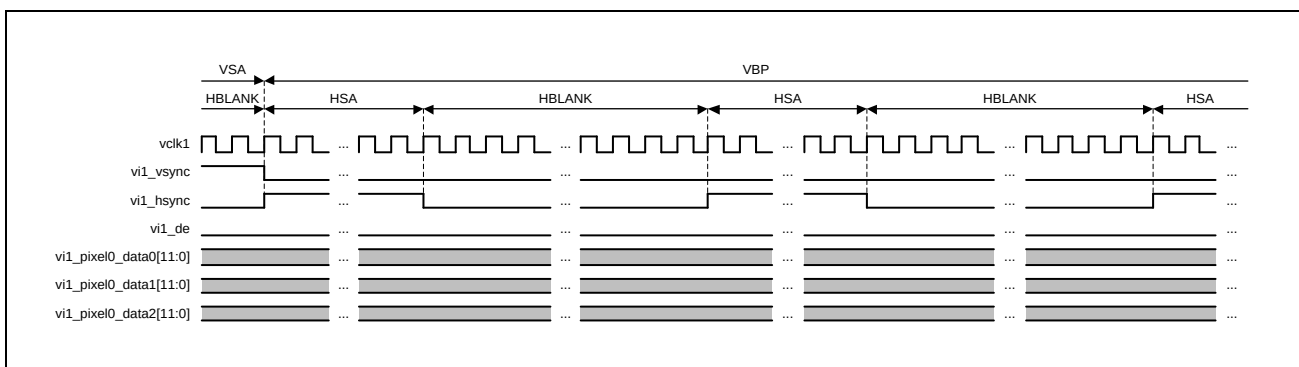
- Additional Rules for non-burst Sync Pulse mode

Parameter	Description
Sequence Operation Packet	When a packet by sequence operation is transmitted during Video Operation (VICH1SR.RUNNING=1b), the period of (HBP + HACTIVE + HFP) should be set the value longer than sum of below periods. - Period for the Tx Sync Event packet (4 bytes) - Period for the packet by sequence operation - Period for the Header and CRC of a Blanking Packet (6 bytes)
HBP+HACTIVE+HFP	Transition of HS-LP-HS is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of (HBP+HACTIVE+HFP) should be set to the value within the range described below. $(HBP + HACTIVE + HFP) > (\text{period for the Tx Sync Event Packet}) + (\text{period for transition time as HS-LP-HS})$
HSA	Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HSA should be set to the value within the range described below. $(HSA \times BPP) / 8 \geq 10b$ Notice, HSA should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet).
HBP	Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HPB should be set to the value within the range described below. $(HBP \times BPP) / 8 \geq 10b$ Notice, HPB should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet).

Table 9.5-6 Configurations for Video Input Interface (2/2)

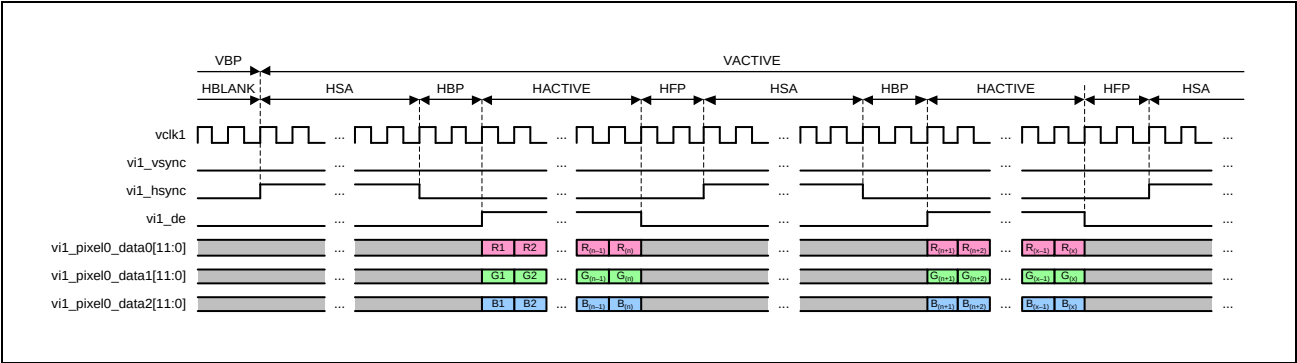
- Additional Rules for non-burst Sync-Event mode and Burst mode

Parameter	Description
Sequence Operation Packet	When a packet by sequence operation is transmitted during Video Operation (VICH1SR.RUNNING=1), the period of HTOTAL should be set the value longer than sum of below periods. • Period for the Tx Sync Event packet (4 bytes) • Period for the packet by sequence operation • Period for the Header and CRC of a Blanking Packet (6 bytes)
HTOTAL	Transition of HS-LP-HS is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HTOTAL should be set to the value within the range described below. $HTOTAL > (\text{period for the Tx Sync Event Packet}) + (\text{period for transition time as HS-LPHS})$
HSA+HBP	Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of (HSA+HBP) should be set to the value within the range described below. $((HAS + HBP) \times BPP) / 8 \geq 10b$ Notice, HAS + HBP should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet).

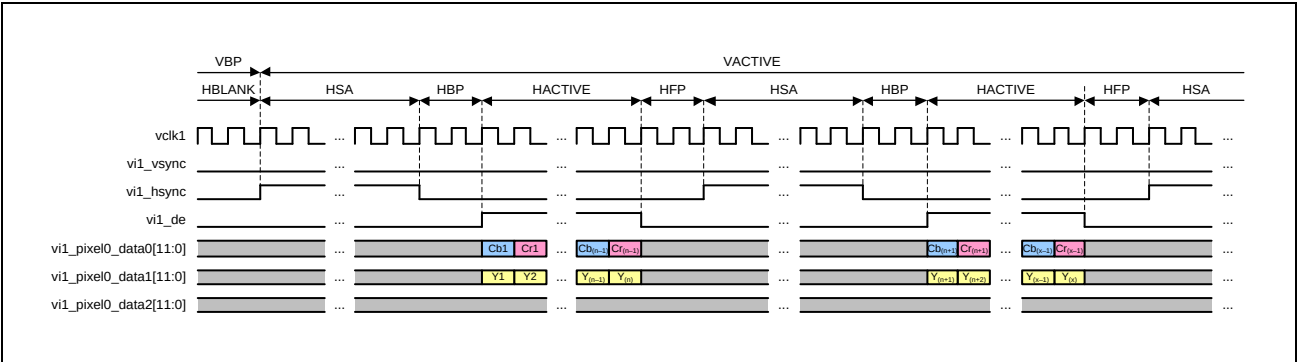
(1) VSA Start (and VFP End)**(2) VSA End and VBP Start**

(3) VBP End and VACTIVE Start

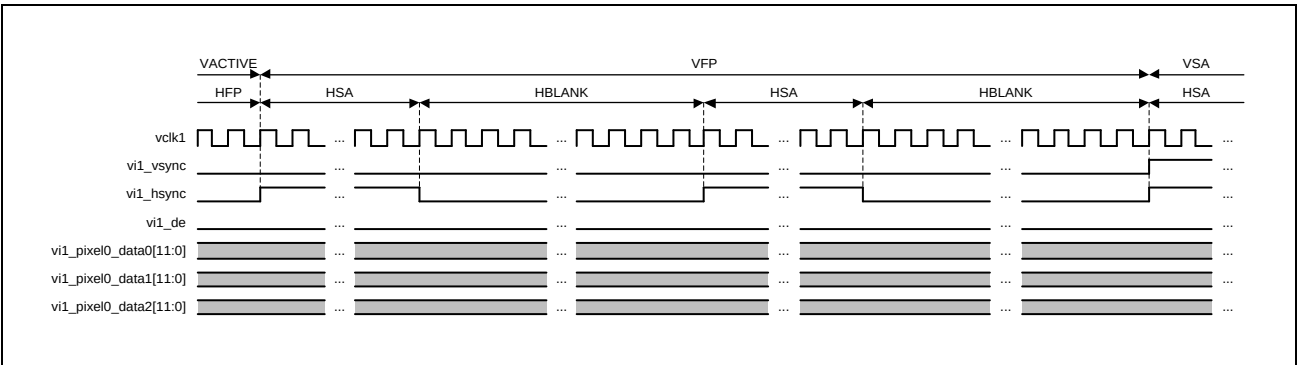
(a) RGB



(b) YCbCr 4: 2: 2



(4) VACTIVE End and VFP Start



9.5.3.5.2 Video-Input Start, End, Change Setting

1) Video Input Start

Video input must be started after Video-Input Operation Start sequence. It must be start from V-Sync start edge.

2) Video Input End

End of Video Input must follow sequence of **9.5.3.5.4 End of Video-Input Operation**.

3) Change of Video Input Setting

Changing of Video-Input setting must be done during Video-Input operation is not running.

9.5.3.5.3 Start of Video-Input Operation

The procedure for Start of Video-Input Operation is shown in **Figure 9.5-12**.

1. Initialize common settings. See **9.5.3.1 Power on Reset and Initial Settings for All Operations**.
2. Some peripheral setting may be done by Sequence operation. Since it is differed from peripheral devices, it is out of this document.
3. Confirm Video-Input clock and hclk stable and set HS clock. See **9.5.3.3 Start/Stop of HS Clock**.
4. Set Video-Input Operation channel parameters (VICH1PPSETR, VICH1VSSETR, VICH1VPSETR, VICH1HSSETR, VICH1HPSETR)
5. Calculate VICH1SET1R.DLY value and set. See **9.5.3.5.9 Delay about Video-Input Operation**.
6. Set "1b" to VICH1SET0R.VSTART with other VICH1SET0R setting values.
7. Wait until VICH1SR.VIRDY = 1b.
8. Start Video signal input

VICH1SET0R, VICH1SET1R, VICH1PPSETR, VICH1VSSETR, VICH1VPSETR, VICH1HSSETR and VICH1HPSETR are prohibited to change value between VICH1SET0R.VSTART = 1b and VICH1SR.RUNNING=0b after VICH1SET0R.STOP = 1b.

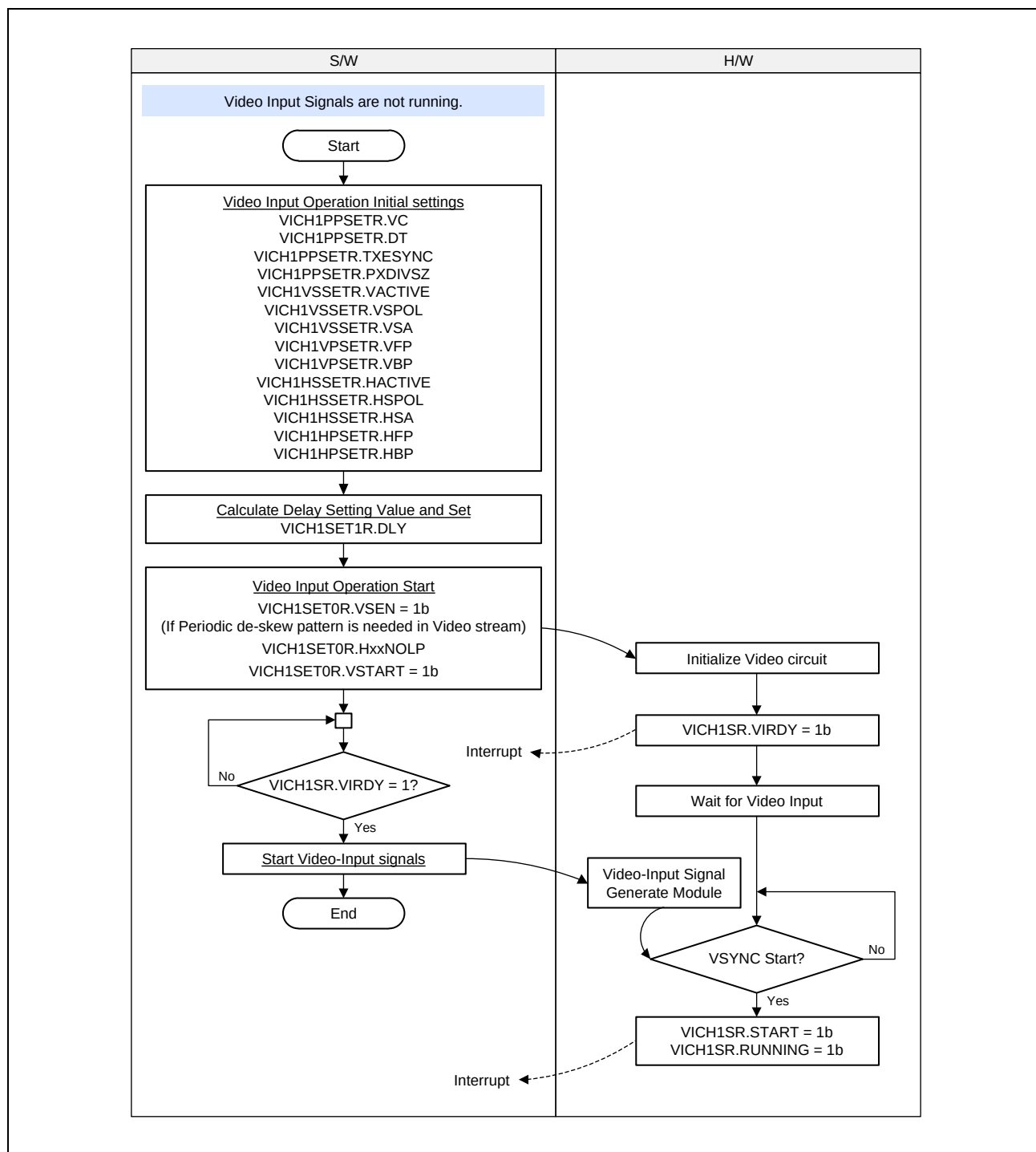


Figure 9.5-12 Video-Input Channel Start

9.5.3.5.4 End of Video-Input Operation

The video-input operation stop flow is shown in **Figure 9.5-13**. Execute the flow when VICH1SR.RUNNING = 1b.

1) VICH1SET0R.VSTPAFT (Figure 9.5-13)

Write 1b to VICH1SET0R.VSTPAFT to request the Video-Input Operation to stop. When the DSI-Tx Module detects the start of one frame (assertion of VSYNC), it stops sending video mode packets and sets VICH1SR.STOP to 1b. After checking VICH1SR.STOP, stop the video input. The video input can be stopped at any time after VICH1SR.STOP becomes 1b.

After stopping the video input, wait for LINKSR.HSBUSY to become 0b.

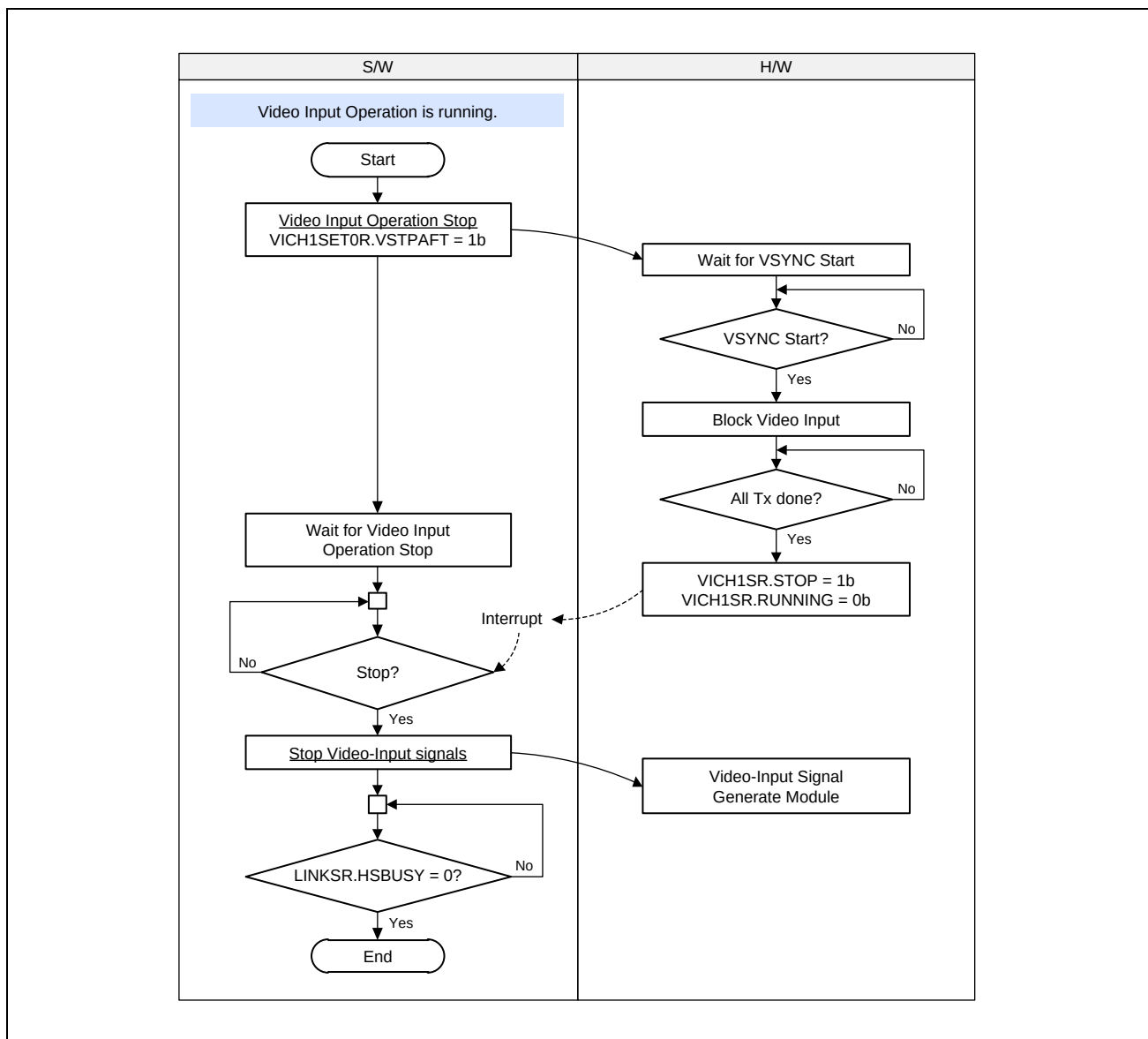


Figure 9.5-13 End of Video-Input Operation (VICH1SET0R.VSTPAFT)

9.5.3.5.5 Burst Mode

The DSI-Tx Module supports Burst Mode.

Burst Mode is a mode that time-compresses pixel data.

Make the D-PHY bandwidth wider than the Video-Input bandwidth for time compression of pixel data. Video transmissions with a 1-line pixel data size ($VICH1HSSETR.HACTIVE \times VICH1SET1R.BPP / 8$) greater than 65,535 Bytes are not supported.

$$\begin{aligned} & \text{Video clock Frequency [Hz]} \times \text{Video Pixel Bit Depth (BPP) [bit]} \\ & \leq \text{DSI HS Byte clock Frequency[Hz]} \times 8[\text{bit}] \times \text{Number of DSI HS Data Lane} \end{aligned}$$

In the case of Video-Input bandwidth equals DSI output bandwidth, it will be indistinguishable from non-burst sync-event mode.

9.5.3.5.6 Non-Burst Mode

The DSI-Tx Module supports Non-Burst Mode.

Make the D-PHY bandwidth the same as the Video-Input bandwidth. Video transmissions with a 1-line pixel data size ($VICH1HSSETR.HACTIVE \times VICH1SET1R.BPP / 8$) greater than 65,535 bytes are not supported.

$$\begin{aligned} & \text{Video clock Frequency [Hz]} \times \text{Video Pixel Bit Depth (BPP) [bit]} \\ & = \text{DSI HS Byte clock Frequency[Hz]} \times 8[\text{bit}] \times \text{Number of DSI HS Data Lane} \end{aligned}$$

9.5.3.5.7 LP Transition and Blanking

The DSI-Tx Module automatically transitions the Data Lane from HS to LP if both of the following two conditions are met:

If it does not transition from HS to LP, it sends a Blanking Packet with the same VC as the VC in the previous packet to maintain the HS.

- There is time to make an HS => LP => HS transition before the next HS transfer starts.
- The period is not prohibited by $VICH1SET0R.xxxNOLP$ to transition to LP

9.5.3.5.8 Non video packet action during Video-Input Operation Running

1) Packet Transmission/Reception by Sequence operation

The DSI-Tx Module can send and receive packets (see **9.5.3.4.2 Single Packet Transmission** and **9.5.3.4.3 Single Packet Reception**) using Sequence Operation while Video-Input Operation is operating*¹.

Make sure that all Sequence Operations are finished before starting the Video-Input Operation.

While Video-Input Operation is operating*¹, packet transmission / reception using Sequence Operation is performed only immediately after the HSE packet ($VICH1PPSETR.TXESYNC = 1b$) or VSS packet ($VICH1PPSETR.TXESYNC = 0b$) on the first horizontal line. Also, sending packets using Sequence Operation is possible only with HS ($SQCHxDSCyAR.SPD = 0b$). Sending with LP ($SQCHxDSCyAR.SPD = 1b$) is not supported. Send and receive packets that are completed within the BLLP period of the first horizontal line.

Execution of Sequence Operation processing is prohibited except for sending and receiving HS packets.

Note 1. “While Video-Input Operation is operating” here is different from VICH1SR.RUNNING. It means the period from setting VICH1SET0R.VSTART to 0b in the flow of **Figure 9.5-12** until VICH1SR.RUNNING becomes 0 in the flow of **Figure 9.5-13**. Also, Note that there is some latency from executing VICH1SET0R.VSTART to being transmitted to the inside of the module for clock transfer.

9.5.3.5.9 Delay about Video-Input Operation

Video-Input Operation run in some latency which call in the DSI-Tx Module “Delay”. “Delay” value must set to VICH1SET1R.DLY before operation.

“Delay” value is defined by calculation using below parameters.

- Clock frequencies
 - Video channel 1 clock (MIPI_DSI_VCLK) frequency
 - DSI High-Speed Transmit Word Clock (TxWordClkHS, hscclk) frequency
- Video parameters
 - HSA period, set by VICH1HSSETR.HSA
 - HACTIVE period, set by VICH1HSSETR.HACTIVE
 - HBP period, set by VICH1HPSETR.HBP
 - HFP period, set by VICH1HPSETR.HFP
 - BPP value, shown in VICH1SET1R.BPP and derived from VICH1PPSETR.DT
- Video transmit settings
 - The number of active lanes, set by TXSETR.NUMLANEUSE
 - Transmit mode, set by VICH1PPSETR.TXESYNC
 - No LP, set by VICH1SET0R.HSANOLP, HBPNOLP, HFPNOLP
 - Clock mode, set by HSCLKSETR.HSCLKMODE
 - Clock stop time, set by CLSTPTSETR.CLKSTPT
 - Go LP and back time, set by LPTRNSTSETR.GOLPBKT
- Other parameters
 - Video-Input channel buffer size, shown in VICH1SET1R.CHBUFSZ

9.5.3.6 Enter/Exit of ULPS

The following describes the ULPS transition / return sequence.

The clock lane and data lane (lanes 0 to 3) can be independently transitioned / restored to ULPS.

The clock lane can make ULPS transitions when TXSETR.CLEN = 1b. It is forbidden to change TXSETR.CLEN to 0b while the clock lane is ULPS.

The data lane can make ULPS transitions when TXSETR.DLEN = 1b. It is forbidden to change TXSETR.DLEN to 0b while the data lane is ULPS.

Data lanes enabled in TXSETR.NUMLANEUSE can be ULPS transitioned / restored. ULPS transition / return of data lanes is done simultaneously for all valid lanes. Only certain data lanes cannot be ULPS transitioned / restored.

When the data lane transitions to ULPS, another operation*¹ using the data lane cannot be performed until ULPS is restored.

When the clock lane transitions to ULPS, another operation*² using the clock lane cannot be performed until ULPS is restored.

Note 1. Sequence Operation, Video-Input Operation.

Note 2. HS Sequence Operation, Video-Input Operation.

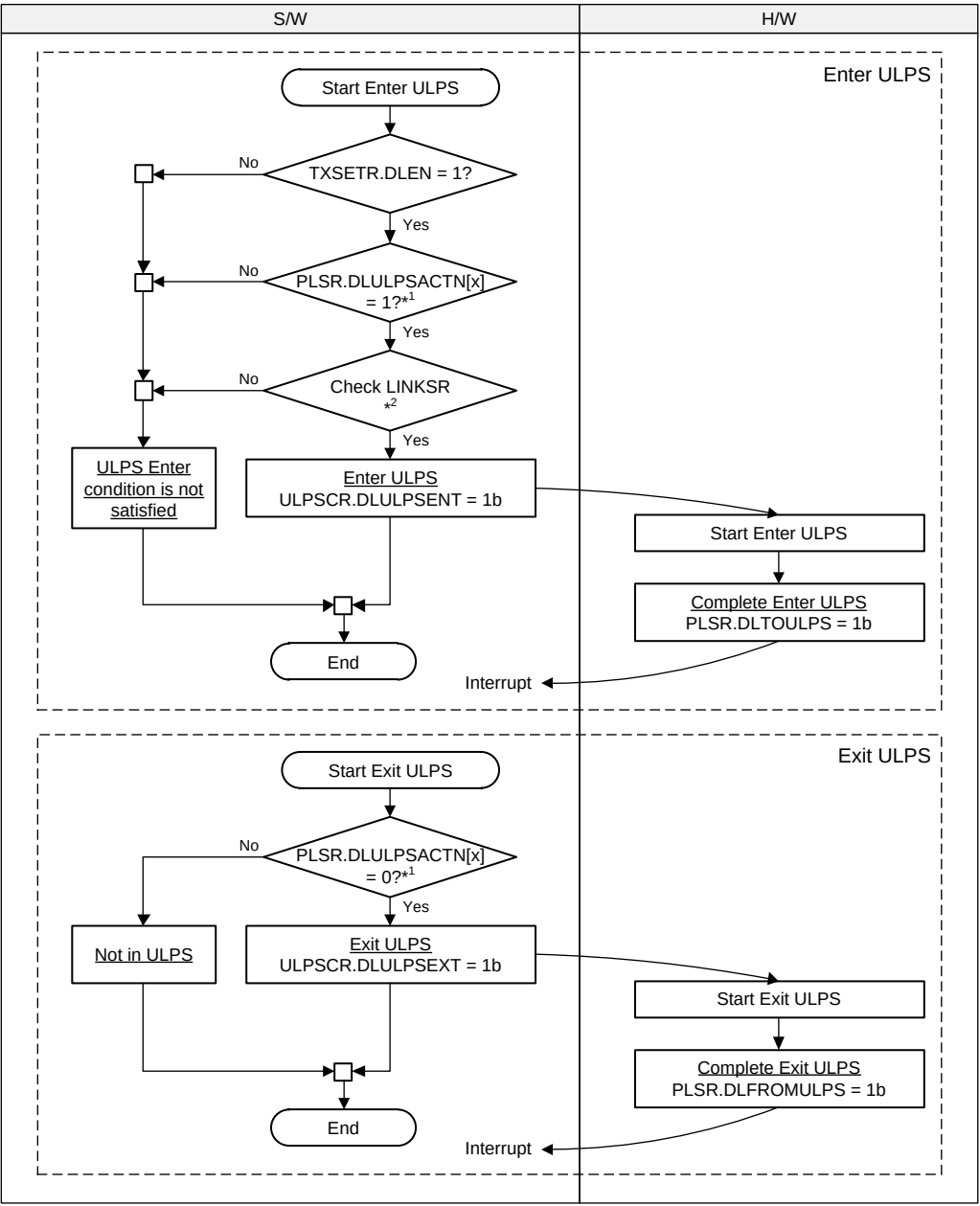


Figure 9.5-14 Enter/Exit of ULPS (Data Lane)

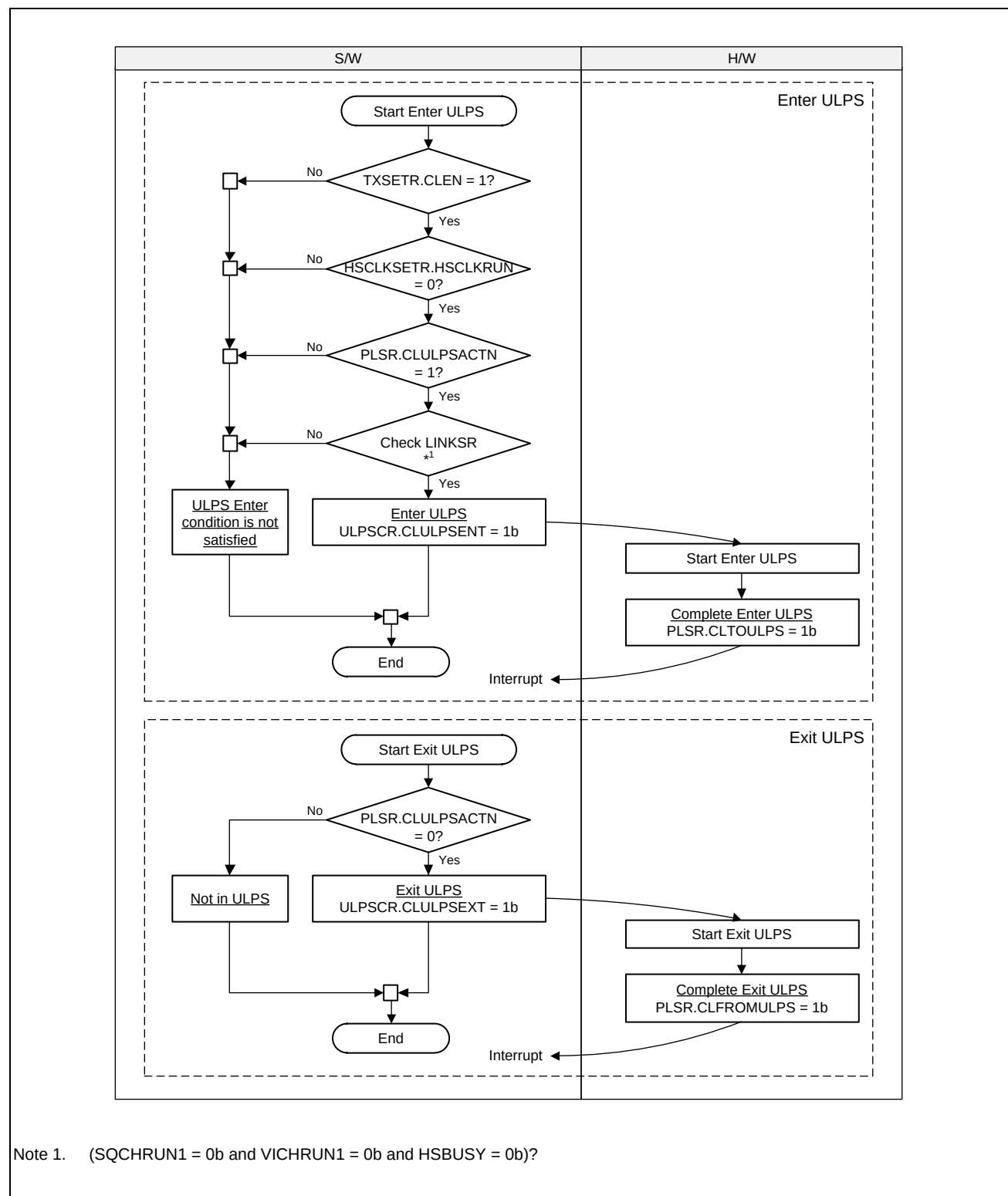
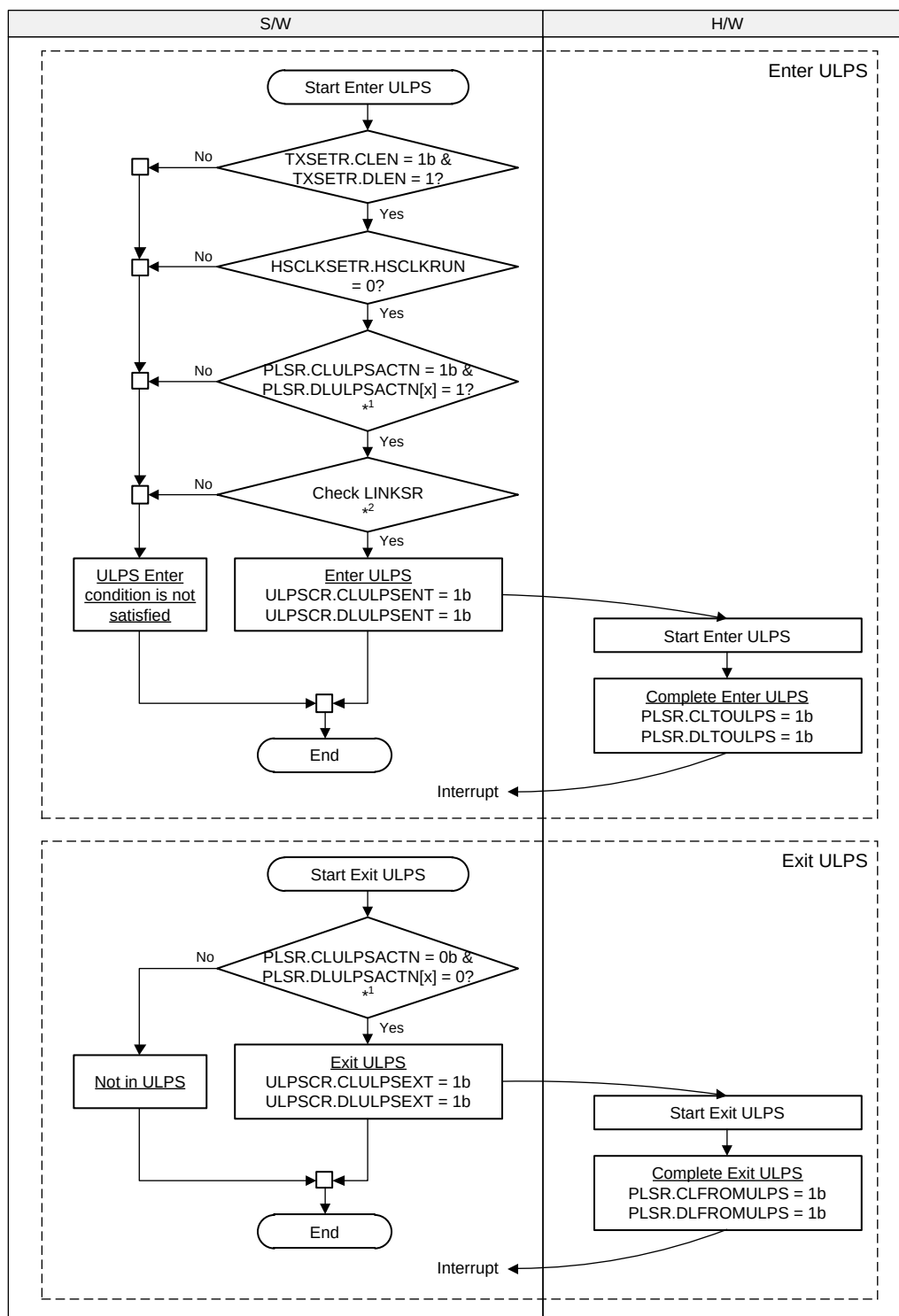


Figure 9.5-15 Enter/Exit of ULPS (Clock Lane)



Note 1. Check all enabled data lanes selected by TXSETR.NUMLANEUSE

Note 2. (SQCHRUN0 = 0b and SQCHRUN1 = 0b and VICHRUN1 = 0b and HSBUSY = 0b and LPBUSY = 0b)?

Figure 9.5-16 Enter/Exit of ULPS (Clock and Data Lane)

9.5.3.7 Error Handling

9.5.3.7.1 Error Detection by Peripheral

Peripheral will notify you of the detected error by “Acknowledge and Error Report” packet.

When the DSI-Tx Module receives the “Acknowledge and Error Report” packet, it displays the latest result in the AKEPLATIR register and the accumulated result in the AKEPACMSR register.

When the user receives the “Acknowledge and Error Report” packet, review and change various settings as necessary.

9.5.3.7.2 Error Detection in D-PHY

If the D-PHY detects an error, the DSI-Tx Module sets the corresponding register flag.

The errors supported by the DSI-Tx Module are as follows. For details on the error, refer to Annex A Logical PHY-Protocol Interface Description of MIPI Alliance Specification for D-PHY Version 1.2.

- ErrEsc
- ErrSyncEsc
- ErrControl
- ErrContentionLP0
- ErrContentionLP1

If the user detects this error, perform Software Reset (**9.5.3.2.1 Software Reset**) and Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

When an ErrContentionLP0 / ErrContentionLP1 error occurs, be sure to execute Software Reset (**9.5.3.2.1 Software Reset**). At that time, write 1b to RSTCR.FCETXSTP, wait for the time when Contention is resolved (equivalent to LRX-H_TO), and then write 0b to RSTCR.FCETXSTP. Also, if necessary, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**).

For more information on returning from Contention, see Section 7.2, Contention Detection and Recovery in the MIPI Alliance Specification for Display Serial Interface 1.3.1.

9.5.3.7.3 Timeout Error

The DSI-Tx Module has the timers defined in Section 7.2.2, Contention Recovery Using Timers of the DSI standard and the DSI-Tx Module original timers.

- Timers defined in DSI Specification
 - HS TX Timeout Error (HTX_TO)
 - LP-RX Host Processor Timeout Error (LRX-H_TO)
 - Turnaround Acknowledge Timeout Error (TA_TO)
- Original Timers
 - Rx Response Timeout Error

(1) HS TX Timeout (HTX_TO) Error

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**). In addition, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

If this error is detected, the setting value of HSTXTOSETR.HTXTO may be too small. When multiple channels are operating at the same time, multiple HS packets may be concatenated and transmitted continuously by HS. Set HSTXTOSETR.HTXTO to a value larger than the HS transmission period.

(2) LP-RX Host Processor Timeout (LRX-H_TO) Error

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**). In addition, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

If this error is detected, the setting value of LRXHTOSETR.LRXHTO may be too small.

There is a possibility of receiving multiple packets such as response packet and “Acknowledge and Error Report” packet in one reception.

Measure the time from the completion of the Bus Turnaround sequence from Host to Peripheral (LP-11 drive detection by Peripheral) to the completion of Bus Turnaround sequence from Peripheral to Host (LP-11 drive start by Host).

Check the Peripheral specifications, and set the value of DSI_LPCLK cycle count (rounded up) + X or more corresponding to the above time in the LRXHTOSETR.LRXHTO register.

(3) Turnaround Acknowledge Timeout (TA_TO) Error

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**) and Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

If this error is detected, the setting value of TATOSETR.TATO may be too small.

Measure the time from the start of the Peripheral Bus Turnaround sequence (LP-10 drive by the Host) to the completion (LP-11 drive detection by the Peripheral) from the Host.

Check the Peripheral specifications and set the value of DSI_LPCLK cycle count (rounded up) + X or more corresponding to the above time in the TATOSETR.TATO register. This error also occurs if the Peripheral does not support bidirectional communication.

(4) Rx Response Timeout Error

Refer to the Peripheral Response Timeout Error ((a) **Peripheral Response Timeout Error**) for more information.

9.5.3.7.4 Rx Related Error in Sequence Operation

(1) Rx Packet Related Error in Sequence Operation

The following describes the received packet error detected by the DSI-Tx Module.

Detects up to one error per packet. (However, ECC 1bit Error may occur at the same time as other errors.)

Errors are determined in the following order:

Peripheral Response Timeout Error, Malformed Packet Error, ECC Error, Unexpected Packet Error, WC Error, CRC Error, Nothing Return Error

(a) Peripheral Response Timeout Error

RXSR.PRESPTOERR is set to 1b if the BTA does not start receiving packets or triggers within the time set in the register after the bus right transitions from Host to Peripheral.

If this error is detected, packets and triggers will not be received until the bus right transitions from Peripheral to Host.

[Timeout time setting register]

- Peripheral Response Timeout value for BTA-only is set by PRESPTOBTASETR.
- Peripheral Response Timeout value for LP Read with BTA is set by PRESPTOLPSETR.PRESPTOLPR.
- Peripheral Response Timeout value for LP Write with BTA is set by PRESPTOLPSETR.PRESPTOLPW.
- Peripheral Response Timeout value for HS Read with BTA is set by PRESPTOHSSETR.PRESPTOHSR.
- Peripheral Response Timeout value for HS Write with BTA is set by PRESPTOHSSETR.PRESPTOHSW.

If this error is detected, the register setting value may be too small, Measure the time from the completion of the Bus Turnaround sequence of Peripheral from the Host (LP-11 drive detection by Peripheral) to the completion of reception of the first byte of the packet or the completion of trigger reception. After checking the Peripheral specifications, set the value of DSI_LPCLK cycle number (rounded up) + X or more corresponding to the above time.

If this error occurs, perform the necessary processing such as resending the previous command.

(b) Malformed Packet Error

If a packet less than 4 bytes is received, RXSR.MLFERR is set to 1b. Since there is an error in the received data, perform necessary processing such as resending the previous command.

(c) ECC Error

ECC generation is required for DSI v1.3.1 standard compliant Peripherals. However, since ECC generation is an option for “earlier revision of DSI Peripherals”, the DSI-Tx Module can select whether to enable or disable ECC checks using DSISETR.ECCEN.

When connecting to a Peripheral that does not support ECC generation, disable ECC checking by setting DSISETR.ECCEN to 0b.

(d) ECC 1-bit Error

If an ECC 1bit error is detected, RXSR.ECCERR1B is set to 1b and 1bit error correction is performed.

The DSI-Tx Module automatically corrects the error and restores the normal packet header, so the subsequent packet reception processing continues.

(e) ECC Multi-bit Error

If an ECC multi-bit error is detected, RXSR.ECCERR is set to 1b and the received packet is discarded. If this error occurs, perform the necessary processing such as resending the previous command.

(f) Unexpected Packet Error

If an Unexpected Packet is received, RXSR.UEXPKTERR is set to 1b.

When this error occurs, the packet header is stored in the RXRSSxR register, and the subsequent reception processing is stopped until the bus right transitions from Peripheral to Host.

The DSI-Tx Module determines the following packets as Unexpected Packets.

- Received packets in which Data Type field of the packet header is “Reserved” as described in Table 23 Data Types for Peripheral-Sourced Packets of the DSI standard
- Received packets other than “Acknowledge and Error Report” packet and “EoTp” in BTA after sending non-read packet
- Received packets other than “Acknowledge and Error Report” packet and “EoTp” when the BTA is BTA only
- BTA after sending read packet receives response packet and then receives response packet again

(g) WC Error

RXSR.WCERR is set to 1b if a long packet with payload data shorter than the WC value is received.

When this error occurs, there is an error in the payload data length, but the payload data is not discarded due to this error, and the packet header is stored in the RXRSSxR register.

Since there is an error in the received data, perform necessary processing such as resending the previous command.

(h) CRC Error

To connect to Peripherals that do not support CRC generation, the DSI-Tx Module can enable or disable CRC checking for each Virtual Channel using DSISETR.CRCEN[3:0]. When connecting to a Peripheral that does not support CRC generation, disable the CRC check by setting the corresponding bit in DSISETR.CRCEN to 0b.

When this error occurs, there is an error in the payload data, but the payload data is not discarded due to this error, and the packet header is stored in the RXRSSxR register.

If this error occurs, perform the necessary processing such as resending the previous command.

(i) Nothing Return Error

RXSR.NORETERR is set to 1b if the BTA transitions the bus right from Host to Peripheral and then does not receive packets or triggers and the bus right transitions from Peripheral to Host.

If this error occurs, perform the necessary processing such as resending the command. If the error recurs, perform Software Reset (**9.5.3.2.1 Software Reset**) and Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) as necessary.

(2) Rx Long Packet Data Payload Related Error in Sequence Operation**(a) Maximum Return Packet Size Error**

RXSR.MAXRPSZERR is set to 1b if the WC value of the received long packet is greater than the DSISETR.MRPSZ setting.

When this error occurs, the packet header is stored in the RXRSSxR register and the payload data is discarded.

Although the payload data is discarded, errors related to payload data reception (WC Error, CRC Error) are detected.

If this error occurs, set DSISETR.MRPSZ to a value greater than or equal to the value set by the Set Max Return Packet Size command.

If you still get this error despite the correct settings, consider setting DSISETR.MRPSZ to a larger value because Peripheral is returning a large packet in violation of the DSI standard.

(b) Internal Bus Error

If Internal Bus Write detects an error, RXSR.IBERR is set to 1b.

Review the settings of SQCHxDSCyDR.LADDR and DSISETR.MRPSZ.

(c) Rx Buffer Overflow Error

If the following Rx Buffer Overflow Error is detected, RXSR.RXOVFERR is set to 1b.

In the case of SQCHxDSCyBR.DTSEL = 00b Packet Payload Data register Mode:

- The Rx Payload Data is over 16 Byte length

In the case of SQCHxDSCyBR.DTSEL = 01b Long Packet data use memory space Mode:

- RAM Overflow for the LP Rx is detected

If this error occurs, execute the following.

In the case of Data register Mode:

- Change to use memory space Mode
- Change peripheral device's Maximum Return Packet Size as not over 16 bytes

In the case of memory space Mode:

- Change peripheral device's Maximum Return Packet Size smaller
- Make Internal Bus faster

9.5.3.7.5 Tx Packet Related Error in Sequence Operation

(1) Internal Bus Error

If Internal Bus Read detects an error, SQCHxSR.TXIBERR is set to 1b. Review the settings of SQCHxDSCyAR.DATA1 / 0b and SQCHxDSCyDR.LADDR.

(2) Packet too Big Error

SQCH1SR.PKTBIGERR is set to 1b if the Sequence Operation packet is of a size that cannot be sent within the BLLP period of the first horizontal line of the Video-Input Operation.

If this error occurs, the Sequence Operation packet will not be sent until the Video-Input Operation is completed.

9.5.3.7.6 Error in Video-Input Operation

(1) Video-Input Buffer Overflow Error

VICH1SR.VBUFOVF is set to 1b if Overflow of Video-Input Buffer is detected.

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**). Also, take necessary measures such as increasing the frequency of TxWordClkHS (hsclk) and decreasing the VICH1SET1R.DLY value. In addition, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

(2) Video-Input Buffer Underflow Error

If the Video-Input Buffer Underflow is detected, VICH1SR.VBUFUDF is set to 1b.

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**). Also, take necessary measures such as lowering the frequency of TxWordClkHS (hsclk) and increasing the VICH1SET1R.DLY value. In addition, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

(3) Timing Error

VICH1SR.TIMERR is set to 1b if no video packets (including Sync Event packets) were sent during the Video-Input Operation at the intended timing.

If this error occurs, perform Software Reset (**9.5.3.2.1 Software Reset**). Also, take necessary measures such as reviewing the setting value of LPTRNSTSETR.GOLPBKT. In addition, perform Reset Trigger Transmission (**9.5.3.4.4 Reset Trigger Transmission**) if necessary.

9.5.4 Interrupts

9.5.4.1 Interrupt List

Interrupts can be divided into the following categories:

Table 9.5-7 Interrupt Signals (1/2)

Interrupt Signal	Category	Interrupt Condition	Interrupt Source Register
dsi_int_sq0	Sequence operation channel 0 interrupt	Receive Acknowledge and Error Report Packet in Sequence Operation BTA	SQCH0SR.RXAKE
		Receive Correctable Error in Sequence Operation BTA	SQCH0SR.RXCORERR
		Receive Packet Data Error in Sequence Operation BTA	SQCH0SR.RXPKTDFAIL
		Receive Fail Error in Sequence Operation BTA	SQCH0SR.RXFAIL
		Receive Fatal Error in Sequence Operation BTA	SQCH0SR.RXFATALERR
		Transmit Internal Bus Error in Sequence Operation	SQCH0SR.TXIBERR
		Descriptor Finished by Setting in Sequence Operation	SQCH0SR.ADESFIN
		Descriptor's All Action Finished in Sequence Operation	SQCH0SR.AACTFIN
dsi_int_sq1	Sequence operation channel 1 interrupt	Receive Acknowledge and Error Report Packet in Sequence Operation BTA	SQCH1SR.RXAKE
		Receive Correctable Error in Sequence Operation BTA	SQCH1SR.RXCORERR
		Receive Packet Data Error in Sequence Operation BTA	SQCH1SR.RXPKTDFAIL
		Receive Fail Error in Sequence Operation BTA	SQCH1SR.RXFAIL
		Receive Fatal Error in Sequence Operation BTA	SQCH1SR.RXFATALERR
		Transmit Internal Bus Error in Sequence Operation	SQCH1SR.TXIBERR
		Transmit Packet is too big Error in Sequence Operation	SQCH1SR.PKTBIGERR
		Descriptor Finished by Setting in Sequence Operation	SQCH1SR.ADESFIN
		Descriptor's All Action Finished in Sequence Operation	SQCH1SR.AACTFIN
dsi_int_vin1	Video-Input operation channel 1 interrupt	Transmit Buffer Overflow in Video-Input Operation	VICH1SR.VBUFOVF
		Transmit Buffer Underflow in Video-Input Operation	VICH1SR.VBUFUDF
		Transmit Timing Error in Video-Input Operation	VICH1SR.TIMERR
		Video-Input Signal Accept Ready	VICH1SR.VIRDY
		Video-Input Operation Stopped	VICH1SR.STOP
		Video-Input Operation Started	VICH1SR.START

Table 9.5-7 Interrupt Signals (2/2)

Interrupt Signal	Category	Interrupt Condition	Interrupt Source Register
dsi_int_rcv	DSI Packet Receive interrupt	Receive Acknowledge and Error Report Packet	RXSR.RXAKE
		Single-bit ECC Error	RXSR.ECCERR1B
		Maximum Return Packet Size Error	RXSR.MAXRPSZERR
		No Return Error	RXSR.NORETERR
		Peripheral Response timeout	RXSR.PRESPTOERR
		Receive Buffer overflow error	RXSR.RXOVFERR
		Internal Bus Error detect on data save	RXSR.IBERR
		CRC Error	RXSR.CRCERR
		Word Count Error	RXSR.WCERR
		Unexpected Packet Error	RXSR.UEXPKTERR
		Multi-bit ECC Error	RXSR.ECCERR
		Malformed Error	RXSR.MLFERR
		External Tearing Effect	RXSR.EXTTEDET
		Receive ACK trigger	RXSR.RXACK
		Receive Tearing Effect trigger	RXSR.RXTE
		Receive Reset trigger	RXSR.RXRTRG
		Receive Unknown-5 trigger	RXSR.RXUK5TRG
		Receive EoTp	RXSR.RXEOTP
		Receive Response packet	RXSR.RXRESP
		Turnaround Acknowledge Timeout	RXSR.TATO
		LP-RX Host Processor Timeout	RXSR.LRXHTO
		BTA request End	RXSR.BTAREQEND
dsi_int_ferr	DSI Fatal Error interrupt	Contention Error LP1	FERRSR.ERRCLP1
		Contention Error LP0	FERRSR.ERRCLP0
		Control Error	FERRSR.ERRCTRL
		Sync Escape Error	FERRSR.ERRSYNESEC
		Error Escape	FERRSR.ERRESC
		Turnaround Acknowledge Timeout	FERRSR.TATO
		LP-RX Host Processor Timeout	FERRSR.LRXHTO
		HS TX Timeout	FERRSR.HTXTO
dsi_int_ppi	DSI D-PHY PPI interrupt	Data Lanes return from ULPS	PLSR.DLFROMULPS
		Data Lanes go to ULPS	PLSR.DLTOULPS
		Clock Lane change HS to LP	PLSR.CLHS2LP
		Clock Lane change LP to HS	PLSR.CLLP2HS
		Clock Lanes return from ULPS	PLSR.CLFROMULPS
		Clock Lanes go to ULPS	PLSR.CLTOULPS
		Data Lane 0 change TX to RX	PLSR.DL0TX2RX
		Data Lane 0 change RX to TX	PLSR.DL0RX2TX

The interrupt signal consists of combination logic. The interrupt signal from a Status Register is enabled by the signal related Interrupt Enable Register. These interrupt signals are bound by OR logic.

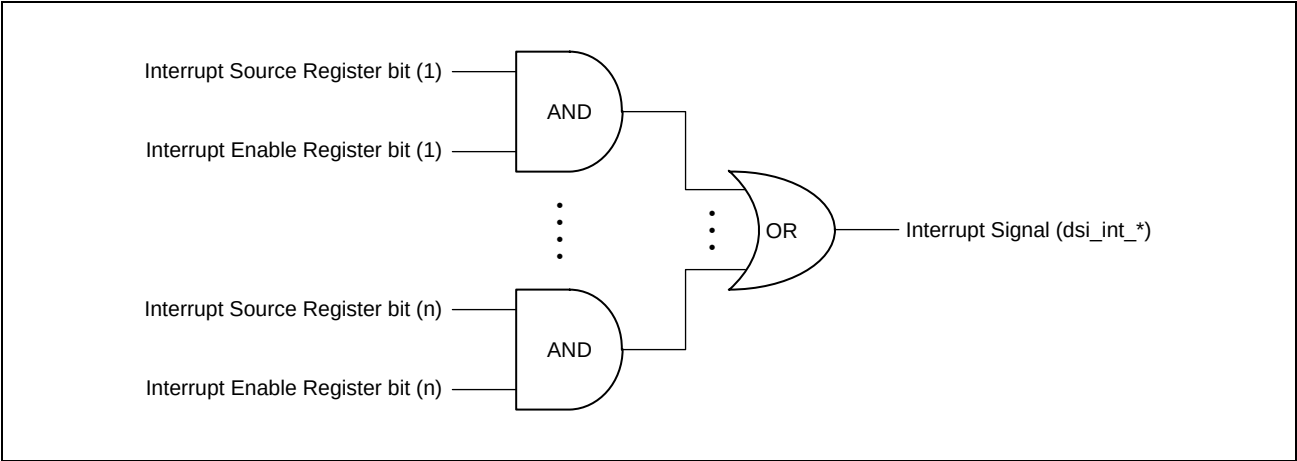


Figure 9.5-17 Interrupt Signal Output

9.5.5 Usage Notes

9.5.5.1 Restriction and Notes

In this section, the restriction and notes on the DSI-Tx Module are described.

Table 9.5-8 Note and Restriction List

Items	Restriction and Notes
TxWordClkHS	Each Lanes TxWordClkHS must be synchronized. The DSI-Tx Module uses Data Lane0 TxWordClkHS as hscclk.
Unused input signals	All unused input signals must be input 0, except following signals. Following signals must be input 1, when they are not used. • ppi_dl1_ulpsactivenot • ppi_dl2_ulpsactivenot • ppi_dl3_ulpsactivenot
TxReadyHS assert	D-PHY must be asserted TxReadyHS within ($2^{\text{RELDISI2TX\_LNK\_RAM\_WORD\_DEPTH}} - 2$) hscclk cycles after TxRequestHS asserted.

SECTION 9 IMAGE

9.6 H.265/H.264 Multi Codec (VCD)

This section describes the functions of the H.265/H.264 multi codec (VCD).

9.6.1 Functional Overview

The VCD is a video codec that supports H.264 and H.265. It has the features shown below.

- H.265/HEVC MP (main profile) supported
 - H.265/HEVC main profile at level 5
- H.264/AVC constrained baseline/main/high profile supported
 - High profile at level 4.2,
 - Main profile at level 4.2,
 - Baseline profile at level 4.2
- H.265 encoding and decoding performance
 - 3840×2160 encoding, 3840×2160 decoding
- H.264 encoding and decoding performance
 - 1920×1080 p encoding, 1920×1080 p decoding
- Equipped with lossless compression and reference image buffer to reduce memory bandwidth

9.6.2 Connection Configuration

Figure 9.6-1 shows the connection configuration for the H.265/H.264 multi codec (VCD).

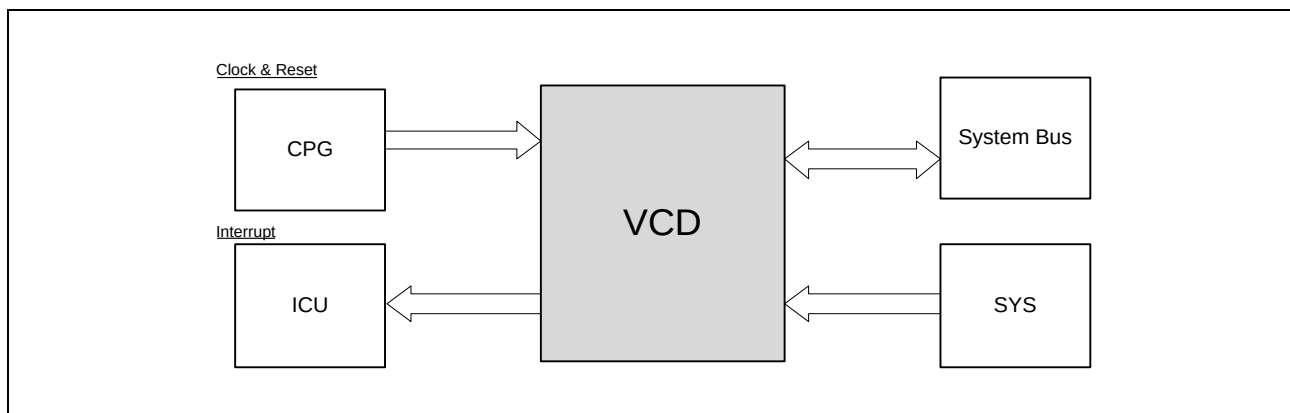


Figure 9.6-1 Configuration Diagram of VCD Connection

9.6.3 Operation Procedure

Use the driver provided by Renesas to operate this unit.

SECTION 9 IMAGE

9.7 3D Graphics Engine (GE3D)

9.7.1 Overview

This LSI includes the Arm Mali-G31 GPU as a 3D Graphics Engine (GE3D). The Arm Mali-G31 GPU is a graphics acceleration platform that is based on open standards. It supports 2D graphics, 3D graphics, and General Purpose computing on GPU (GPGPU).

9.7.1.1 Features

The GE3D has the following features.

- One single-pixel shader core
- 8 Kbytes L2 cache
- A programmable architecture.
- An API feature set with support for shader-based and fixed-function graphics APIs.
- Anti-aliasing capabilities.
- An effective core for General Purpose computing on GPU (GPGPU) applications.
- High memory bandwidth and low power consumption for 3D graphics content.
- Latency tolerance.
- Compressed texture formats.
- Coherency aware interconnects for system memory and resource sharing.

The GE3D and software support compute and graphics API standards.

— The GE3D supports these compute API standards:

OpenCL 2.0 Full Profile.

— The GE3D supports these graphics API standards:

OpenGL ES 1.1, 2.0, 3.0, 3.1 and 3.2.

9.7.1.2 Block Diagram

The GE3D contains the shader processor core and associated blocks, including the tiler, the memory-management unit, and the level 2 cache memory subsystem.

Figure 9.7-1 shows a block diagram.

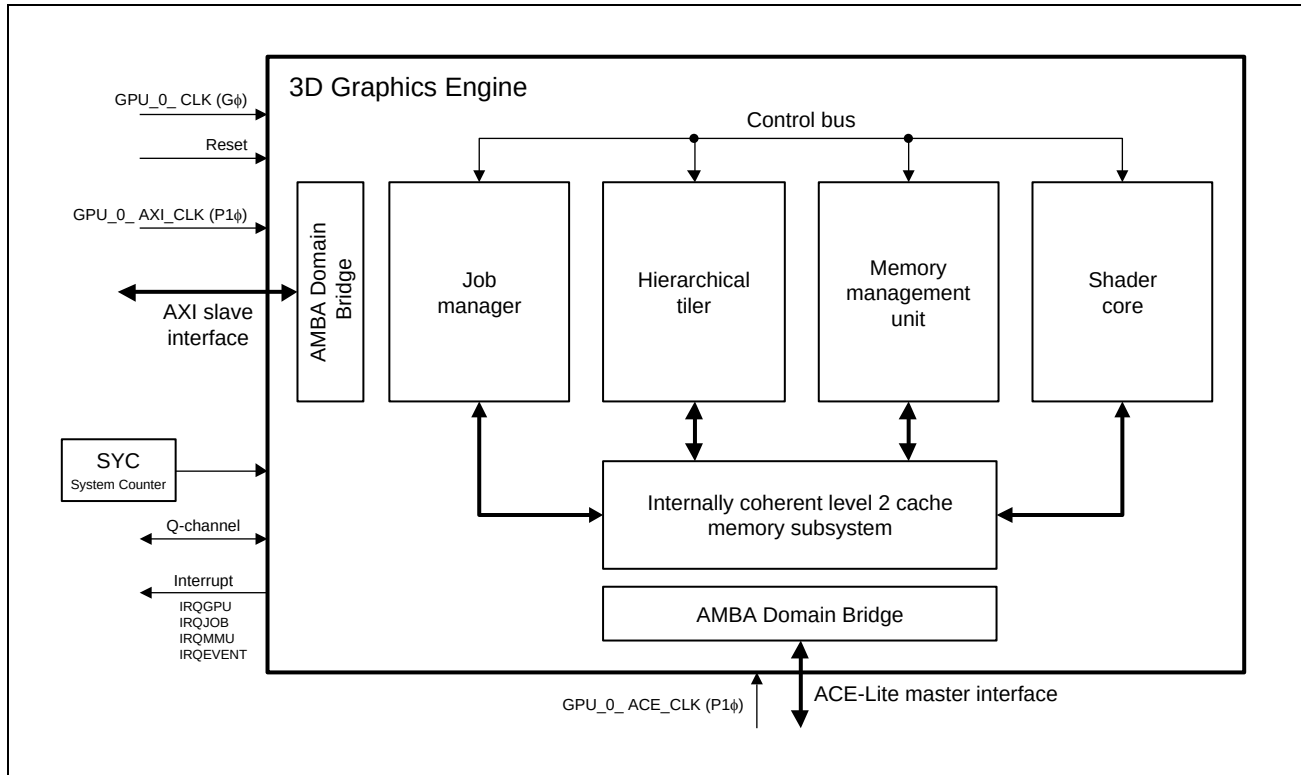


Figure 9.7-1 Block Diagram

The GE3D uses a centralized hardware resource manager, called the job manager, that controls the internal GPU functions.

The application software runs on the application processor. When it requires graphics to be displayed on screen, it uses the graphics device driver to send a graphics job to the GE3D. In the GE3D, the job manager receives the job, interprets it, and then sends a series of graphics tasks to the internal functional units.

The job manager issues and sends commands to specific functional blocks in the GE3D.

- The shader core is the main processing engines of the GE3D. The shader core carries out all the rendering and computation operations.
- The tiler creates lists of all the objects in a scene, so that the shader core processes the objects efficiently.
- The memory management unit performs virtual address to physical address translation of external bus interface accesses.
- The Level 2 cache memory subsystem provides caching for all internal master blocks. It also supports cache coherency for caches in the shader cores. The L2 cache sits between the internal master blocks and the external interconnect. All transactions go through the cache. If the cache cannot service an internal transaction, it performs an external transaction to read or write data.

9.7.1.3 External Pins

In 3D Graphics Engine, there are no external pins.

9.7.1.4 Register Configuration

Base address of registers is allocated to 0_1485_0000h. The address range is 64 KB (0_1485_0000h - 0_1485_FFFFh).

9.7.1.5 Usage Note

Renesas does not provide technical support for users to program GE3D registers directly. Renesas provides users with drivers to program GE3D. GE3D supports OpenGL ES as a graphics API. When drawing graphics, control according to this API.

9.7.1.6 Starting SYC

When using GE3D, it is necessary to supply the clock to the SYC (system counter) and release the reset to keep it running.

9.8.1.2 Image Processing Functions

For more information, refer to the User's Manual Additional Document.

SECTION 10 ELECTRICAL CHARACTERISTICS

10.1 Electrical Characteristics

This section describes the electrical characteristics of this LSI.

10.1.1 Absolute Maximum Ratings

Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Table 10.1-1 Absolute Maximum Ratings (1/3)

Unit Name	Item	Symbol	Min.	Max.	Unit
CA55	VDD09_CA55	CA55_V _{DD09}	-0.4	1.2	V
PD_OTHERS	VDD08_OTHERS	OTHERS_V _{DD08}	-0.4	1.2	V
	VDD33_OTHERS	OTHERS_V _{DD33}	-0.4	3.8	V
	VDD33_PRE18_OTHERS	OTHERS_PRE18V _{DD33}	-0.4	2.5	V
	VDD1833_OTHERS_A	OTHERSA_V _{DD1833}	-0.4	3.8	V
	VDD1833_OTHERS_B	OTHERSB_V _{DD1833}	-0.4	3.8	V
	VDD1833_OTHERS_C	OTHERSC_V _{DD1833}	-0.4	3.8	V
	VDD1833_OTHERS_D	OTHERSD_V _{DD1833}	-0.4	3.8	V
PD_AWO	VDD08_AWO	AWO_V _{DD08}	-0.4	1.2	V
	VDD18_AWO	AWO_V _{DD18}	-0.4	2.5	V
	VDD1833_AWO	AWO_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_AWO	AWO_PRE18V _{DD1833}	-0.4	2.5	V
USB30	USB30_USVPH	USB30_USV _{PH}	-0.4	2.5	V
	USB30_USVPTX	USB30_USV _{PTX}	-0.4	1.2	V
	USB30_USVDD33	USB30_USV _{DD33}	-0.4	3.8	V
	USB30_USVDD18	USB30_USV _{DD18}	-0.4	2.5	V
	USB30_USDVDD	USB30_USDV _{DD}	-0.4	1.2	V
USB20	USB20_USVDD33	USB20_USV _{DD33}	-0.4	3.8	V
	USB20_USVDD18	USB20_USV _{DD18}	-0.4	2.5	V
	USB20_USDVDD	USB20_USDV _{DD}	-0.4	1.2	V
TSU0	TS0AVDD18	TS0AV _{DD18}	-0.4	2.5	V
	TS0DVDD08A	TS0DV _{DD08A}	-0.4	1.2	V
TSU1	TS1AVDD18	TS1AV _{DD18}	-0.4	2.5	V
	TS1DVDD08A	TS1DV _{DD08A}	-0.4	1.2	V
xSPI	VDD1833_XSPI	XSPI_V _{DD1833}	-0.4	3.8	V
SD0	VDD1833_SD0	SD0_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_SD	SD_PRE18V _{DD1833}	-0.4	2.5	V
SD1	VDD1833_SD1	SD1_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_SD	SD_PRE18V _{DD1833}	-0.4	2.5	V
SD2	VDD1833_SD2	SD2_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_SD2	SD2_PRE18V _{DD1833}	-0.4	2.5	V

Table 10.1-1 Absolute Maximum Ratings (2/3)

Unit Name	Item	Symbol	Min.	Max.	Unit
OTP	OTP_VDD18	OTP_V _{DD18}	-0.4	2.5	V
DDR0	VDD08_DDR	DDR_V _{DD08}	-0.4	1.2	V
	DDR0_VDDQ	DDR0_V _{DDQ}	-0.4	1.5	V
	DDR0_VDDQLP	DDR0_V _{DDQLP}	-0.4	1.5	V
	DDR0_VAA	DDR0_V _{AA}	-0.4	2.5	V
GBETH0	VDD1833_ET0	ET0_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_ET0	ET0_PRE18V _{DD1833}	-0.4	2.5	V
GBETH1	VDD1833_ET1	ET1_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_ET1	ET1_PRE18V _{DD1833}	-0.4	2.5	V
CRU0	CSI0_MSVDD18	CSI0_MSV _{DD18}	-0.4	2.5	V
	CSI0_MSVDD0P8	CSI0_MSV _{DD0P8}	-0.4	1.2	V
CRU1	CSI1_MSVDD18	CSI1_MSV _{DD18}	-0.4	2.5	V
	CSI1_MSVDD0P8	CSI1_MSV _{DD0P8}	-0.4	1.2	V
DSI	DSI_VDD0P8	DSI_V _{DD0P8}	-0.4	1.2	V
	DSI_VDD12	DSI_V _{DD12}	-0.4	2.5	V
	DSI_VDD18	DSI_V _{DD18}	-0.4	2.5	V
PCIE	PCIE_VCC18ACMN	PCIE_V _{CC18ACMN}	-0.4	2.5	V
	PCIE_VCC18AL01	PCIE_V _{CC18AL01}	-0.4	2.5	V
	PCIE_VCC08AL01	PCIE_V _{CC08AL01}	-0.4	1.2	V
I3C	VDD1218_I3C	I3C_V _{DD1218}	-0.4	2.5	V
ADC	ADC0_ADAVDD18	ADC0_ADAV _{DD18}	-0.4	2.5	V
	ADC1_ADAVDD18	ADC1_ADAV _{DD18}	-0.4	2.5	V
	ADC2_ADAVDD18	ADC2_ADAV _{DD18}	-0.4	2.5	V
CPG	PLVDD_PLLCM33	PLL _{CM33} _PLV _{DD}	-0.4	2.5	V
	PLVDD_PLLCLN_DTY_DRP	PLL _{CLN_DTY_DRP} _PLV _{DD}	-0.4	2.5	V
	PLVDD_PLLCA55	PLL _{CA55} _PLV _{DD}	-0.4	2.5	V
	PLVDD_PLLVDO_DSI	PLL _{VDO_DSI} _PLV _{DD}	-0.4	2.5	V
	PLVDD_PLDDR0	PLDDR0_PLV _{DD}	-0.4	2.5	V
	PLVDD_PLLETH_GPU	PLLETH_GPU_PLV _{DD}	-0.4	2.5	V
	PLDVDD08_PLLCM33	PLL _{CM33} _PLDV _{DD08}	-0.4	1.2	V
	PLDVDD08_PLLCLN_DTY_DRP	PLL _{CLN_DTY_DRP} _PLDV _{DD08}	-0.4	1.2	V
	PLDVDD09_PLLCA55	PLL _{CA55} _PLDV _{DD09}	-0.4	1.2	V
	PLDVDD08_PLLVDO_DSI	PLL _{VDO_DSI} _PLDV _{DD08}	-0.4	1.2	V
	PLDVDD08_PLDDR0	PLDDR0_PLDV _{DD08}	-0.4	1.2	V
	PLDVDD08_PLLETH_GPU	PLLETH_GPU_PLDV _{DD08}	-0.4	1.2	V
CST	VDD1833_JTAG	JTAG_V _{DD1833}	-0.4	3.8	V
	VDD1833_PRE18_JTAG	JTAG_PRE18V _{DD1833}	-0.4	2.5	V
PWC	VDD18_PWC	PWC_V _{DD18}	-0.4	2.5	V
—	Input voltage (0.6-V I/O)	V _{in06}	-0.4	DDRn_V _{DDQLP} + 0.3* ¹	V
—	Input voltage (1.1-V I/O)	V _{in11}	-0.4	DDRn_V _{DDQ} + 0.3* ¹	V
—	Input voltage (1.2-V I/O)	V _{in12}	-0.4	V ₁₂ + 0.3* ²	V
—	Input voltage (1.8-V I/O)	V _{in18}	-0.4	V ₁₈ + 0.3* ³	V
—	Input voltage (1.8-V I/O (3.3-V tolerant))* ⁴	V _{in18_tol}	-0.4	3.6	V

Table 10.1-1 Absolute Maximum Ratings (3/3)

Unit Name	Item	Symbol	Min.	Max.	Unit
—	Input voltage (3.3-V I/O)	V_{in33}	-0.4	$V_{33} + 0.3^{*5}$	V
—	Analog input voltage (ADC I/O)	V_{ain18}	0	$ADAV_{DD18}$	V
—	Junction temperature	T_j	-40	125	°C
—	Storage temperature	T_{stg}	-40	150	°C

Note 1. n = 0, 1. The voltage to be applied must be within the absolute maximum rating (1.5 V).

Note 2. The voltage to be applied must be within the absolute maximum rating (2.5 V). V_{12} indicates the power supply voltage for 1.2-V I/O pins.

Note 3. The voltage to be applied must be within the absolute maximum rating (2.5 V). V_{18} indicates the power supply voltage for 1.8-V I/O pins. When 1.8 V is used for the 3.3/1.8-V switching I/O, this specification is applied.

Note 4. Pxx pins (with the exceptions of P2x, P90, P91, P92, and PBx)

Note 5. The voltage to be applied must be within the absolute maximum rating (3.8 V). V_{33} indicates the power supply voltage for 3.3-V I/O pins. When 3.3 V is used for the 3.3/1.8-V switching I/O, this specification is applied.

10.1.2 Recommended Operating Range

Table 10.1-2 Recommended Operating Range (1/2)

Unit Name	Item	Symbol	Min.	Typ.	Max.	Unit	Note
CA55	VDD09_CA55	CA55_V _{DD09}	0.86	0.9	0.94	V	0.9 V: OD*1
			0.76	0.8	0.84	V	0.8 V: ND*1
PD_OTHERS	VDD08_OTHERS	OTHERS_V _{DD08}	0.76	0.8	0.84	V	*2
	VDD33_OTHERS	OTHERS_V _{DD33}	3.135	3.3	3.465	V	
	VDD33_PRE18_OTHERS	OTHERS_PRE18V _{DD33}	1.71	1.8	1.89	V	
	VDD1833_OTHERS_A	OTHERSA_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_OTHERS_B	OTHERSB_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_OTHERS_C	OTHERSC_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
PD_AWO	VDD1833_OTHERS_D	OTHERSD_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD08_AWO	AWO_V _{DD08}	0.76	0.8	0.84	V	
	VDD18_AWO	AWO_V _{DD18}	1.71	1.8	1.89	V	
USB30	VDD1833_AWO	AWO_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_AWO	AWO_PRE18V _{DD1833}	1.71	1.8	1.89	V	
	VDD1833_AWO	AWO_V _{DD1833}	3.14	3.3	3.46	V	
USB30	USB30_USVPH	USB30_USV _{PH}	1.71	1.8	1.89	V	
	USB30_USVPTX	USB30_USV _{PTX}	0.76	0.8	0.84	V	
	USB30_USVDD33	USB30_USV _{DD33}	3.14	3.3	3.46	V	
	USB30_USVDD18	USB30_USV _{DD18}	1.71	1.8	1.89	V	
	USB30_USDVDD	USB30_USDV _{DD}	0.76	0.8	0.84	V	
USB20	USB20_USVDD33	USB20_USV _{DD33}	3.14	3.3	3.46	V	
	USB20_USVDD18	USB20_USV _{DD18}	1.71	1.8	1.89	V	
	USB20_USDVDD	USB20_USDV _{DD}	0.76	0.8	0.84	V	
TSU0	TS0AVDD18	TS0AV _{DD18}	1.71	1.8	1.89	V	
	TS0DVDD08A	TS0DV _{DD08A}	0.76	0.8	0.84	V	
TSU1	TS1AVDD18	TS1AV _{DD18}	1.71	1.8	1.89	V	
	TS1DVDD08A	TS1DV _{DD08A}	0.76	0.8	0.84	V	
xSPI	VDD1833_XSPI	XSPI_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
SD0	VDD1833_SD0	SD0_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_SD	SD_PRE18V _{DD1833}	1.71	1.8	1.89	V	
SD1	VDD1833_SD1	SD1_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_SD	SD_PRE18V _{DD1833}	1.71	1.8	1.89	V	
SD2	VDD1833_SD2	SD2_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_SD2	SD2_PRE18V _{DD1833}	1.71	1.8	1.89	V	
OTP	OTPVDD18	OTP_V _{DD18}	1.71	1.8	1.89	V	

Table 10.1-2 Recommended Operating Range (2/2)

Unit Name	Item	Symbol	Min.	Typ.	Max.	Unit	Note
DDR0	VDD08_DDR	DDR_V _{DD08}	0.76	0.8	0.84	V	*2
	DDR0_VDDQ	DDR0_V _{DDQ}	1.06	1.1	1.17	V	
	DDR0_VDDQLP	DDR0_V _{DDQLP}	0.57	0.6	0.65	V	0.6 V: LPDDR4X
			1.06	1.1	1.17	V	1.1 V: LPDDR4
	DDR0_VAA	DDR0_V _{AA}	1.71	1.8	1.89	V	
GBETH0	VDD1833_ET0	ET0_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_ET0	ET0_PRE18V _{DD1833}	1.71	1.8	1.89	V	
GBETH1	VDD1833_ET1	ET1_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_ET1	ET1_PRE18V _{DD1833}	1.71	1.8	1.89	V	
CRU0	CSI0_MSVDD18	CSI0_MSV _{DD18}	1.71	1.8	1.89	V	
	CSI0_MSVDD0P8	CSI0_MSV _{DD0P8}	0.76	0.8	0.84	V	
CRU1	CSI1_MSVDD18	CSI1_MSV _{DD18}	1.71	1.8	1.89	V	
	CSI1_MSVDD0P8	CSI1_MSV _{DD0P8}	0.76	0.8	0.84	V	
DSI	DSI_VDD0P8	DSI_V _{DD0P8}	0.76	0.8	0.84	V	
	DSI_VDD12	DSI_V _{DD12}	1.14	1.2	1.26	V	
	DSI_VDD18	DSI_V _{DD18}	1.71	1.8	1.89	V	
PCIE	PCIE_VCC18ACMN	PCIE_V _{CC18ACMN}	1.71	1.8	1.89	V	
	PCIE_VCC18AL01	PCIE_V _{CC18AL01}	1.71	1.8	1.89	V	
	PCIE_VCC08AL01	PCIE_V _{CC08AL01}	0.76	0.8	0.84	V	
I3C	VDD1218_I3C	I3C_V _{DD1218}	1.71	1.8	1.89	V	
			1.14	1.2	1.26	V	
ADC	ADC0_ADAVDD18	ADC0_ADAV _{DD18}	1.71	1.8	1.89	V	
	ADC1_ADAVDD18	ADC1_ADAV _{DD18}	1.71	1.8	1.89	V	
	ADC2_ADAVDD18	ADC2_ADAV _{DD18}	1.71	1.8	1.89	V	
CPG	PLVDD_PLLCM33	PLL _{CM33} _PLV _{DD}	1.71	1.8	1.89	V	
	PLVDD_PLLCLN_DTY_DRP	PLL _{CLN_DTY_DRP} _PLV _{DD}	1.71	1.8	1.89	V	
	PLVDD_PLLCA55	PLL _{CA55} _PLV _{DD}	1.71	1.8	1.89	V	
	PLVDD_PLLVDO_DSI	PLL _{VDO_DSI} _PLV _{DD}	1.71	1.8	1.89	V	
	PLVDD_PLDDR0	PLL _{DDR0} _PLV _{DD}	1.71	1.8	1.89	V	
	PLVDD_PLLETH_GPU	PLLETH_GPU_PLV _{DD}	1.71	1.8	1.89	V	
	PLDVDD08_PLLCM33	PLL _{CM33} _PLDV _{DD08}	0.76	0.8	0.84	V	
	PLDVDD08_PLLCLN_DTY_DRP	PLL _{CLN_DTY_DRP} _PLDV _{DD08}	0.76	0.8	0.84	V	
	PLDVDD09_PLLCA55	PLL _{CA55} _PLDV _{DD09}	0.86	0.9	0.94	V	0.9 V: OD*1
			0.76	0.8	0.84	V	0.8 V: ND*1
	PLDVDD08_PLLVDO_DSI	PLL _{VDO_DSI} _PLDV _{DD08}	0.76	0.8	0.84	V	
	PLDVDD08_PLDDR0	PLL _{DDR0} _PLDV _{DD08}	0.76	0.8	0.84	V	
	PLDVDD08_PLLETH_GPU	PLLETH_GPU_PLDV _{DD08}	0.76	0.8	0.84	V	
CST	VDD1833_JTAG	JTAG_V _{DD1833}	3.14	3.3	3.46	V	
			1.71	1.8	1.89	V	
	VDD1833_PRE18_JTAG	JTAG_PRE18V _{DD1833}	1.71	1.8	1.89	V	
PWC	VDD18_PWC	PWC_V _{DD18}	1.71	1.8	1.89	V	

Note 1. OD: Over drive (up to 1.8-GHz operation frequency)
 ND: Normal drive (up to 1.1-GHz operation frequency)

10.1.3 Power-On/Off Sequence

10.1.3.1 CM33 Boot Mode (PWC Enabled)

The state diagram of CM33 cold boot is shown in **Figure 10.1-1**. The boot mode states (1) to (4) refer to the sequence of (1) to (4) in **Figure 10.1-2**.

For the details of boot mode, refer to **4.4.7 Control of Booting**.

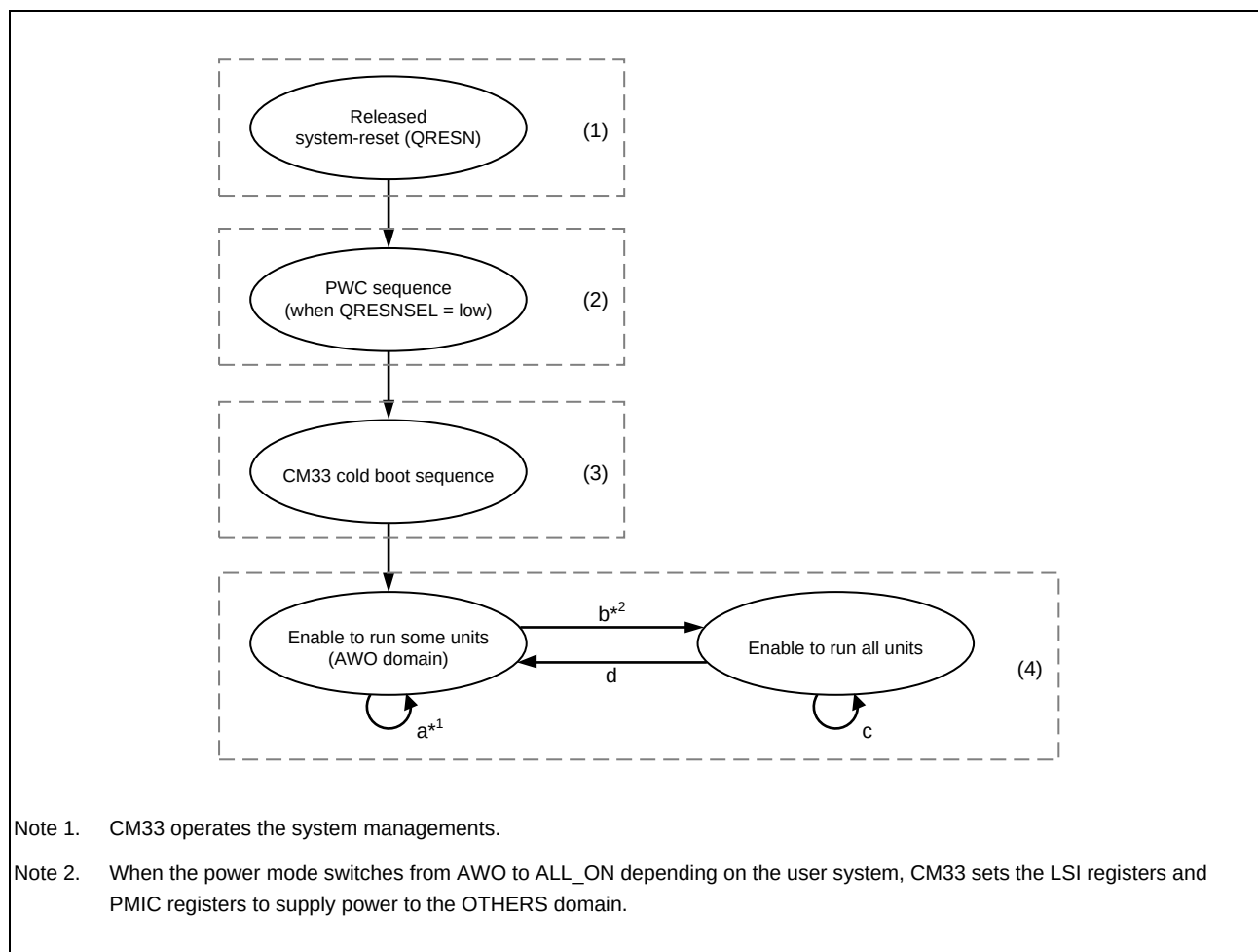
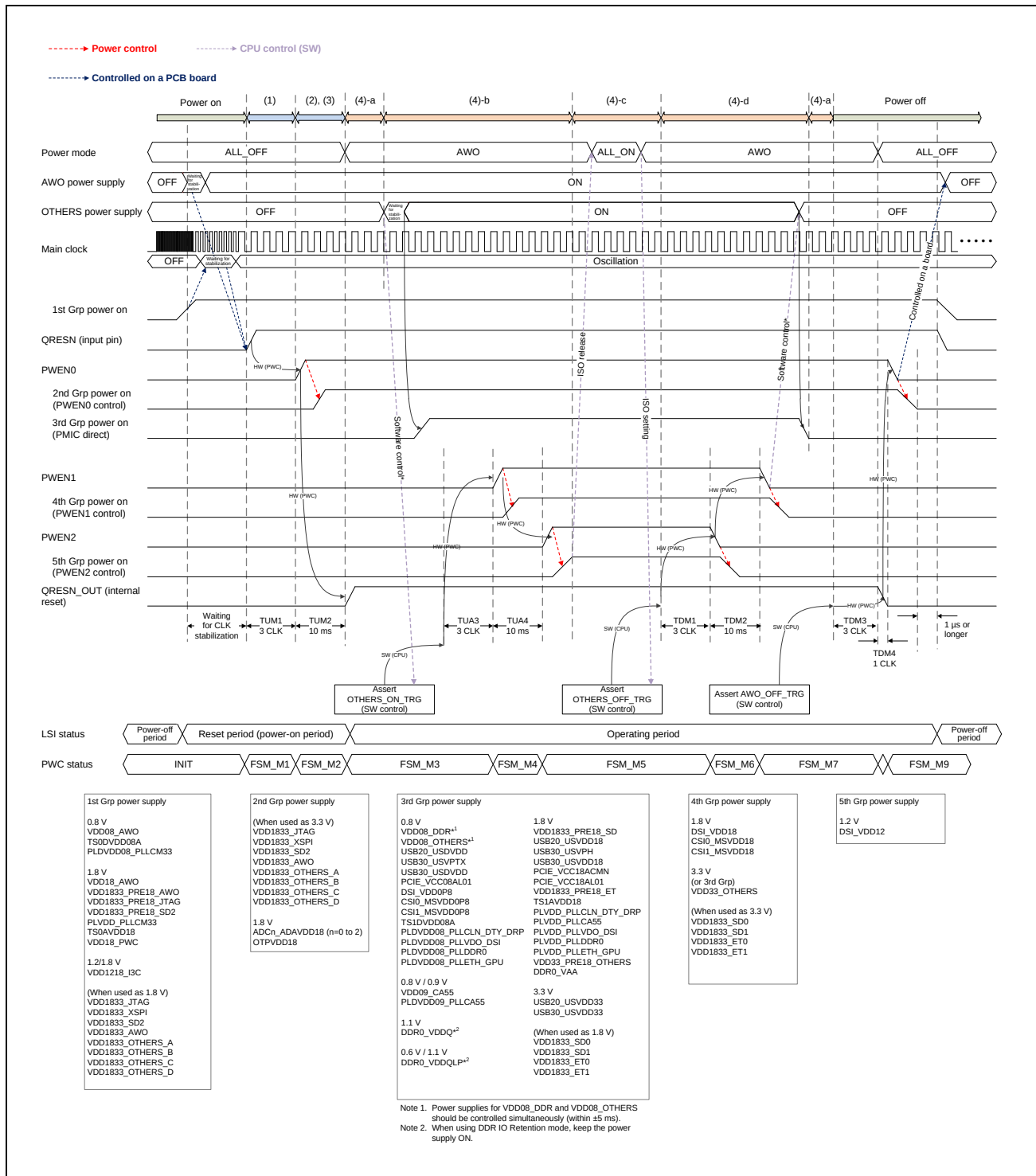


Figure 10.1-1 CM33 Boot State Diagram



(Continued on next page)

(Continued from previous page)

Software control: RIIC or PFC (GPIO) control

For details of the procedure, refer to **4.5.3 Power Domains**.

Note: The clock stabilization time depends on the board design. Make the setting according to the results of evaluation.

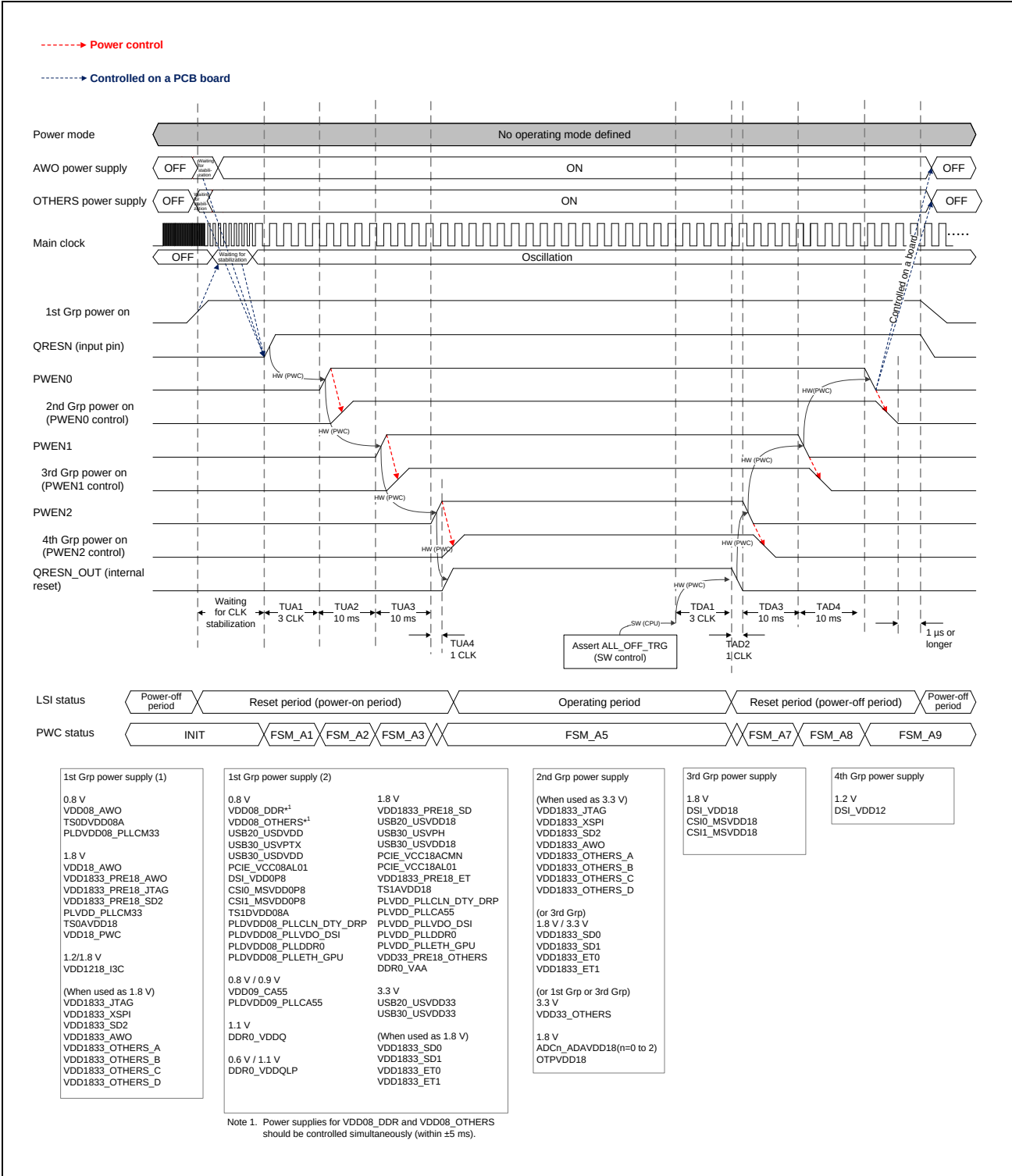
Note: Refer to the notes in **10.1.3.3** and **10.1.3.4** for details on the restrictions on the rise time and fall time of each power supply.

Note: When the QRESN pin becomes low, the PWEN0 to PWEN2 pins become low simultaneously.

Note: When using the PWC, set QRESN to the low level at the same time as or after the 1st Grp power off.

Figure 10.1-2 Power-On/Power-Off Sequence (CM33 Boot)

10.1.3.2 CA55 Boot Mode (PWC Enabled)



(Continued on next page)

(Continued from previous page)

Note: The clock stabilization time depends on the board design. Make the setting according to the results of evaluation.

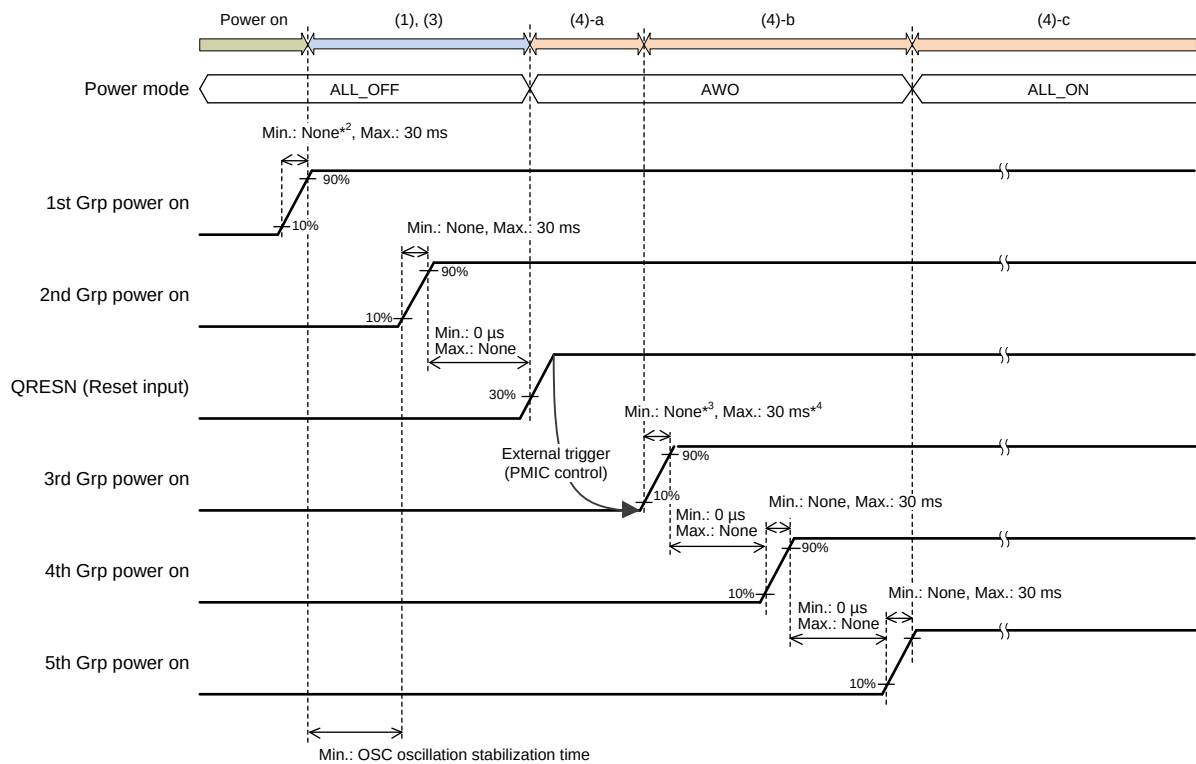
Note: Refer to the notes in **10.1.3.5** and **10.1.3.6** for details on the restrictions on the rise time and fall time of each power supply.

Note: When the QRESN pin becomes low, the PWEN0 to PWEN2 pins become low simultaneously.

Note: When using the PWC, set QRESN to the low level at the same time as or after the 1st Grp power off.

Figure 10.1-3 Power-On/Power-Off Sequence (CA55 Boot)

10.1.3.3 Power-On Sequence – CM33 Boot Mode (PWC Disabled)



1st Grp power supply

0.8 V
VDD08_AWO
TS0DVDD08A
PLDVDD08_PLLCM33

1.8 V
VDD18_AWO
VDD1833_PRE18_AWO
VDD1833_PRE18_JTAG
VDD1833_PRE18_SD2
PLVDD_PLLCM33
TS0AVDD18
VDD18_PWC

1.2/1.8 V
VDD1218_I3C

(When used as 1.8 V)
VDD1833_JTAG
VDD1833_XSPI
VDD1833_SD2
VDD1833_AWO
VDD1833_OTHERS_A
VDD1833_OTHERS_B
VDD1833_OTHERS_C
VDD1833_OTHERS_D

2nd Grp power supply

(When used as 3.3 V)
VDD1833_JTAG
VDD1833_XSPI
VDD1833_SD2
VDD1833_AWO
VDD1833_OTHERS_A
VDD1833_OTHERS_B
VDD1833_OTHERS_C
VDD1833_OTHERS_D

1.8 V
ADCN_ADAVDD18(n=0 to 2)
OTPVDD18

3rd Grp power supply

0.8 V
VDD08_DDR⁺¹
VDD08_OTHERS⁺¹
USB20_USDVDD
USB30_USVPTX
USB30_USDVDD
PCIE_VCC08AL01
DSI_VDD0P8
CSI0_MSVDD0P8
CSI1_MSVDD0P8
TS1DVDD08A
PLDVDD08_PLLCLN_DTY_DRP
PLDVDD08_PLLVDO_DSI
PLDVDD08_PLDDR0
PLDVDD08_PLLETH_GPU

0.8 V / 0.9 V
VDD09_CA55
PLDVDD09_PLLCA55

1.1 V
DDR0_VDDQ⁺⁵
0.6 V / 1.1 V
DDR0_VDDQLP⁺⁵

1.8 V
VDD1833_PRE18_SD
USB20_USVDD18
USB30_USVPH
USB30_USVDD18
PCIE_VCC18ACMN
PCIE_VCC18AL01
VDD1833_PRE18_ET
TS1AVDD18
PLVDD_PLLCLN_DTY_DRP
PLVDD_PLLCA55
PLVDD_PLLVDO_DSI
PLVDD_PLDDR0
PLVDD_PLLETH_GPU
VDD33_PRE18_OTHERS
DDR0_VAA

3.3 V
USB20_USVDD33
USB30_USVDD33
(When used as 1.8 V)
VDD1833_SD0
VDD1833_SD1
VDD1833_ET0
VDD1833_ET1

4th Grp power supply

1.8 V
DSI_VDD18
CSI0_MSVDD18
CSI1_MSVDD18

3.3 V
(or 3rd Grp)
VDD33_OTHERS

(When used as 3.3 V)
VDD1833_SD0
VDD1833_SD1
VDD1833_ET0
VDD1833_ET1

5th Grp power supply

1.2 V
DSI_VDD12

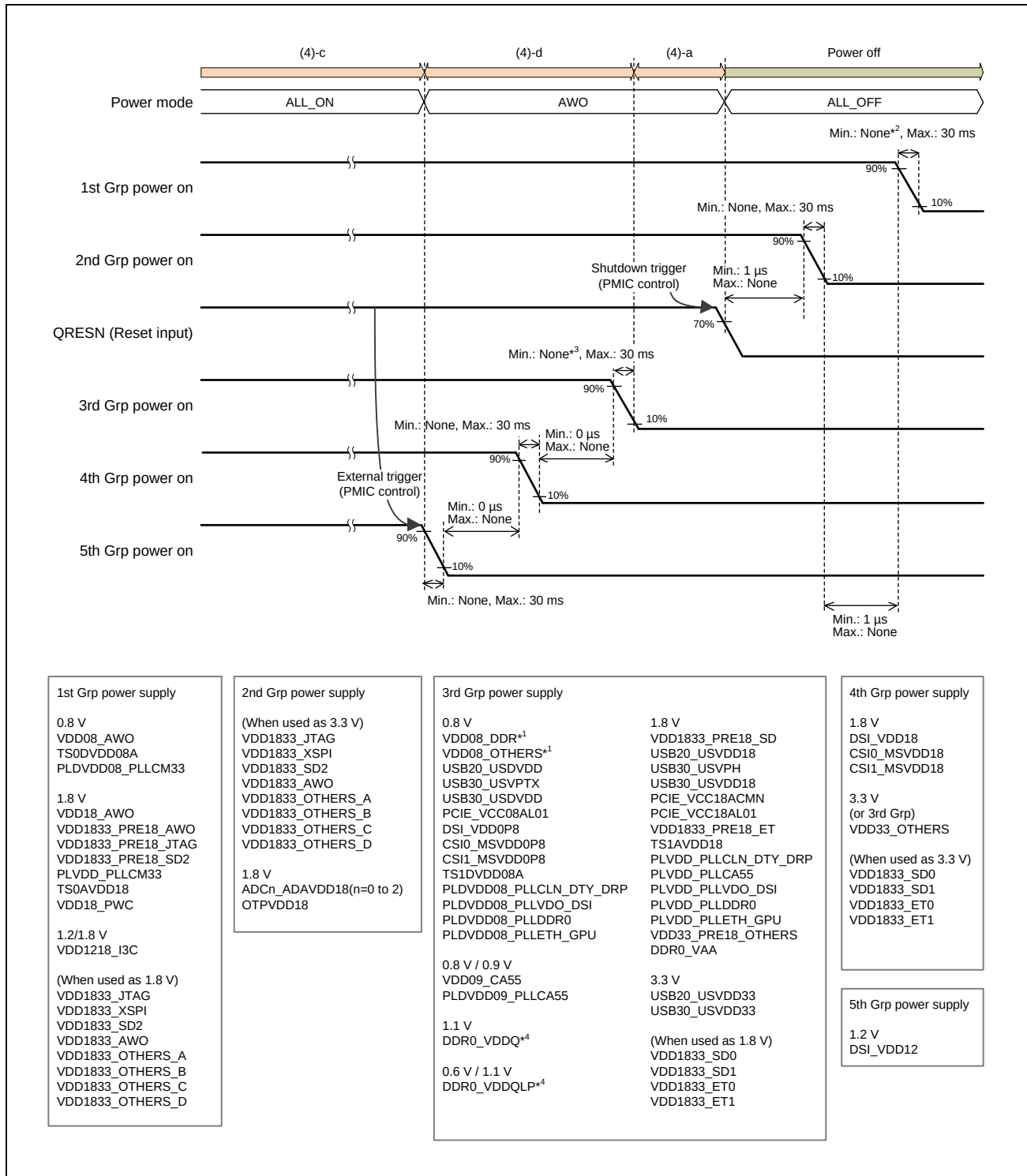
(Continued on next page)

(Continued from previous page)

- Note 1. Power supplies for VDD08_DDR and VDD08_OTHERS should be controlled simultaneously (within ± 5 ms).
- Note 2. TS0DVDD08A, TS0AVDD18: Min. 10 μ s
- Note 3.
- DDR0_VDDQ, DDR0_VDDQLP (1.1 V): Min. 180 μ s
 - DDR0_VDDQLP (0.6 V): Min. 100 μ s
 - DDR0_VAA: Min. 290 μ s
 - VDD08_DDR: Min. 5 μ s
 - USB30_USVPTX, USB30_USVPH: Min. 10 μ s
 - USB20_USDVDD, USB30_USDVDD: Min. 10 μ s
 - USB20_USVDD18, USB30_USVDD18: Min. 20 μ s
 - USB20_USVDD33, USB30_USVDD33: Min. 30 μ s
 - TS1DVDD08A, TS1AVDD18: Min. 10 μ s
- Note 4. USB20_USDVDD, USB20_USVDD18, USB20_USVDD33, USB30_USDVDD, USB30_USVDD18, USB30_USVDD33: Max. 10 ms
- Note 5. When using DDR IO Retention mode, keep the power supply ON.
- Note:** The clock stabilization time depends on the board design. Make the setting according to the results of evaluation.
- Note:** The rise time of each power supply must be compliant with the maximum 30 ms regulation.

Figure 10.1-4 Power-On Sequence (CM33 Boot Mode)

10.1.3.4 Power-Off Sequence – CM33 Boot Mode (PWC Disabled)



(Continued on next page)

(Continued from previous page)

Note 1. Power supplies for VDD08_DDR and VDD08_OTHERS should be controlled simultaneously (within ± 5 ms).

Note 2. TS0DVDD08A, TS0AVDD18: Min. 10 μ s

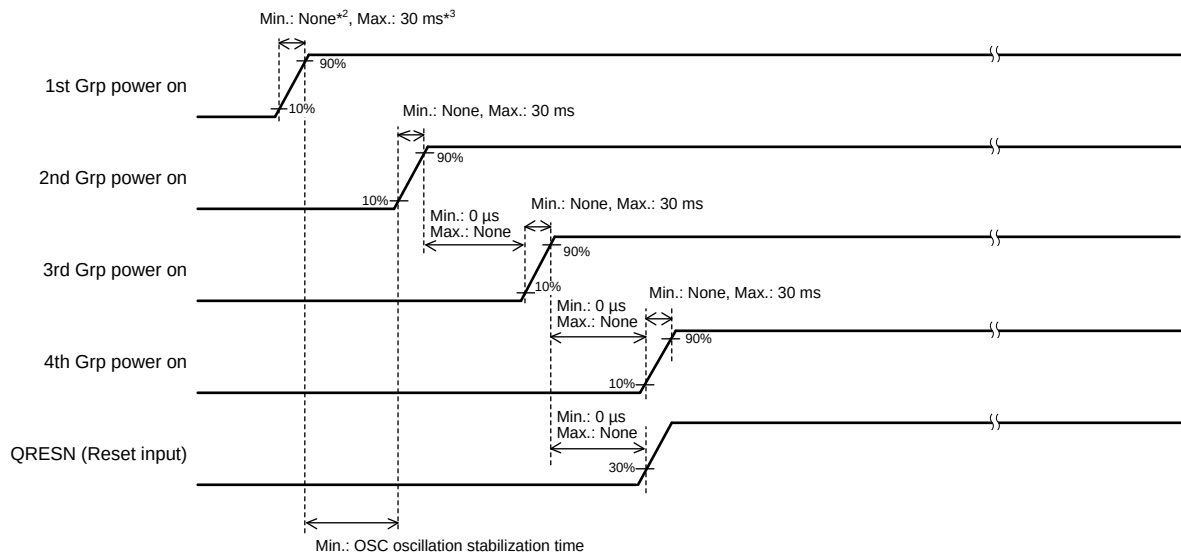
Note 3. TS1DVDD08A, TS1AVDD18: Min. 10 μ s

Note 4. When using DDR IO Retention mode, keep the power supply ON.

Note: The fall time of each power supply must be compliant with the maximum 30 ms regulation.

Figure 10.1-5 Power-Off Sequence (CM33 Boot Mode)

10.1.3.5 Power-On Sequence – CA55 Boot Mode (PWC Disabled)



1st Grp power supply (1) 0.8 V VDD08_AWO TS0DVDD08A PLDVDD08_PLLCM33 1.8 V VDD18_AWO VDD1833_PRE18_AWO VDD1833_PRE18_JTAG VDD1833_PRE18_SD2 PLVDD_PLLCM33 TS0AVDD18 VDD18_PWC 1.2/1.8 V VDD1218_I3C (When used as 1.8 V) VDD1833_JTAG VDD1833_XSPI VDD1833_SD2 VDD1833_AWO VDD1833_OTHERS_A VDD1833_OTHERS_B VDD1833_OTHERS_C VDD1833_OTHERS_D	1st Grp power supply (2) 0.8 V VDD08_DDR⁺¹ VDD08_OTHERS⁺¹ USB20_USDVDD USB30_USVPTX USB30_USDVDD PCIE_VCC08AL01 DSI_VDD0P8 CSI0_MSVDD0P8 CSI1_MSVDD0P8 TS1DVDD08A PLDVDD08_PLLCLN_DTY_DRP PLDVDD08_PLVDDO_DSI PLDVDD08_PLDDR0 PLDVDD08_PLLETH_GPU 0.8 V / 0.9 V VDD09_CA55 PLDVDD09_PLCCA55 1.1 V DDR0_VDDQ 0.6 V / 1.1 V DDR0_VDDQLP	1.8 V VDD1833_PRE18_SD USB20_USVDD18 USB30_USVPH USB30_USVDD18 PCIE_VCC18ACMN PCIE_VCC18AL01 VDD1833_PRE18_ET TS1AVDD18 PLVDD_PLLCLN_DTY_DRP PLVDD_PLCCA55 PLVDD_PLVDDO_DSI PLVDD_PLDDR0 PLVDD_PLLETH_GPU VDD33_PRE18_OTHERS DDR0_VAA 3.3 V USB20_USVDD33 USB30_USVDD33 (When used as 1.8 V) VDD1833_SD0 VDD1833_SD1 VDD1833_ET0 VDD1833_ET1	2nd Grp power supply (When used as 3.3 V) VDD1833_JTAG VDD1833_XSPI VDD1833_SD2 VDD1833_AWO VDD1833_OTHERS_A VDD1833_OTHERS_B VDD1833_OTHERS_C VDD1833_OTHERS_D (or 3rd Grp) 1.8 V / 3.3 V VDD1833_SD0 VDD1833_SD1 VDD1833_ET0 VDD1833_ET1 (or 1st Grp or 3rd Grp) 3.3 V VDD33_OTHERS 1.8 V ADCn_ADAVDD18(n=0 to 2) OTPVDD18	3rd Grp power supply 1.8 V DSI_VDD18 CSI0_MSVDD18 CSI1_MSVDD18	4th Grp power supply 1.2 V DSI_VDD12

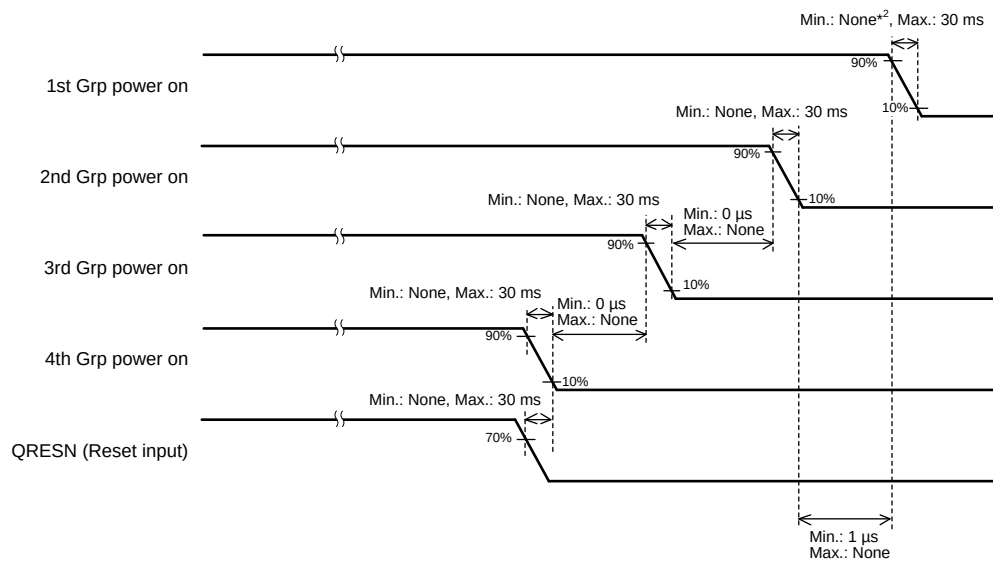
(Continued on next page)

(Continued from previous page)

- Note 1. Power supplies for VDD08_DDR and VDD08_OTHERS should be controlled simultaneously (within ± 5 ms).
- Note 2.
- TS0DVDD08A, TS0AVDD18: Min. 10 μ s
 - DDR0_VDDQ, DDR0_VDDQLP (1.1 V): Min. 180 μ s
 - DDR0_VDDQLP (0.6 V): Min. 100 μ s
 - DDR0_VAA: Min. 290 μ s
 - VDD08_DDR: Min. 5 μ s
 - USB30_USVPTX, USB30_USVPH: Min. 10 μ s
 - USB20_USDVDD, USB30_USDVDD: Min. 10 μ s
 - USB20_USVDD18, USB30_USVDD18: Min. 20 μ s
 - USB20_USVDD33, USB30_USVDD33: Min. 30 μ s
 - TS1DVDD08A, TS1AVDD18: Min. 10 μ s
- Note 3. USB20_USDVDD, USB20_USVDD18, USB20_USVDD33, USB30_USDVDD, USB30_USVDD18, USB30_USVDD33:
Max. 10 ms
- Note:** The clock stabilization time depends on the board design. Make the setting according to the results of evaluation.
- Note:** The rise time of each power supply must be compliant with the maximum 30 ms regulation.

Figure 10.1-6 Power-On Sequence (CA55 Boot Mode)

10.1.3.6 Power-Off Sequence – CA55 Boot Mode (PWC Disabled)



1st Grp power supply (1)	1st Grp power supply (2)	2nd Grp power supply	3rd Grp power supply
0.8 V VDD08_AWO TS0DVDD08A PLVDVDD08_PLLCM33 1.8 V VDD18_AWO VDD1833_PRE18_AWO VDD1833_PRE18_JTAG VDD1833_PRE18_SD2 PLVDD_PLLCM33 TS0AVDD18 VDD18_PWC 1.2/1.8 V VDD1218_I3C (When used as 1.8 V) VDD1833_JTAG VDD1833_XSPI VDD1833_SD2 VDD1833_AWO VDD1833_OTHERS_A VDD1833_OTHERS_B VDD1833_OTHERS_C VDD1833_OTHERS_D	0.8 V VDD08_DDR* ¹ VDD08_OTHERS* ¹ USB20_USDVDD USB30_USVPTX USB30_USDVDD PCIE_VCC08AL01 DSI_VDD0P8 CSI0_MSVDD0P8 CSI1_MSVDD0P8 TS1DVDD08A PLVDVDD08_PLLCLN_DTY_DRP PLVDVDD08_PLLVDO_DSI PLVDVDD08_PLLDDR0 PLVDVDD08_PLETH_GPU 0.8 V / 0.9 V VDD09_CA55 PLVDVDD09_PLLCA55 1.1 V DDR0_VDDQ 0.6 V / 1.1 V DDR0_VDDQLP	1.8 V VDD1833_PRE18_SD USB20_USVDD18 USB30_USVPH USB30_USVDD18 PCIE_VCC18ACMN PCIE_VCC18AL01 VDD1833_PRE18_ET TS1AVDD18 PLVDD_PLLCLN_DTY_DRP PLVDD_PLLCA55 PLVDD_PLLVDO_DSI PLVDD_PLLDDR0 PLVDD_PLETH_GPU VDD33_PRE18_OTHERS DDR0_VAA 3.3 V USB20_USVDD33 USB30_USVDD33 (When used as 1.8 V) VDD1833_SD0 VDD1833_SD1 VDD1833_ET0 VDD1833_ET1	(When used as 3.3 V) VDD1833_JTAG VDD1833_XSPI VDD1833_SD2 VDD1833_AWO VDD1833_OTHERS_A VDD1833_OTHERS_B VDD1833_OTHERS_C VDD1833_OTHERS_D (or 3rd Grp) 1.8 V / 3.3 V VDD1833_SD0 VDD1833_SD1 VDD1833_ET0 VDD1833_ET1 (or 1st Grp or 3rd Grp) 3.3 V VDD33_OTHERS 1.8 V ADCn_ADAVDD18(n=0 to 2) OTPVDD18
			1.8 V DSI_VDD18 CSI0_MSVDD18 CSI1_MSVDD18 4th Grp power supply 1.2 V DSI_VDD12

Note 1. Power supplies for VDD08_DDR and VDD08_OTHERS should be controlled simultaneously (within ± 5 ms).

Note 2. – TS0DVDD08A, TS0AVDD18: Min. 10 μ s
 – TS1DVDD08A, TS1AVDD18: Min. 10 μ s

Note: The fall time of each power supply must be compliant with the maximum 30 ms regulation.

Figure 10.1-7 Power-Off Sequence (CA55 Boot Mode)

10.1.4 DC Characteristics

10.1.4.1 Maximum Supply Current

Conditions for the supply current: Power supply voltage = Max. value, $T_j = -40$ to 125°C

Table 10.1-3 Max. Supply Currents during Operation (1/2)

Unit Name	Item	Symbol	Max.	Unit	Note
CA55	0.8-V (or 0.9-V) power supply current	I_{DD09_CA55}	3031	mA	VDD09_CA55
PD_OTHERS	0.8-V core power supply current	I_{DD08_OTHERS}	7489	mA	VDD08_OTHERS
	3.3-V core power supply current	I_{DD33_OTHERS}	1	mA	VDD33_OTHERS
	Pre-driver power supply current	$I_{DD33PRE18\_OTHERS}$	1	mA	VDD33_PRE18_OTHERS
	Group A I/O power supply current	$I_{DD1833\_OTHERSA}$	25	mA	VDD1833_OTHERS_A
	Group B I/O power supply current	$I_{DD1833\_OTHERSB}$	25	mA	VDD1833_OTHERS_B
	Group C I/O power supply current	$I_{DD1833\_OTHERSC}$	25	mA	VDD1833_OTHERS_C
	Group D I/O power supply current	$I_{DD1833\_OTHERSD}$	13	mA	VDD1833_OTHERS_D
PD_AWO	0.8-V core power supply current	I_{DD08_AWO}	315	mA	VDD08_AWO
	1.8-V core power supply current	I_{DD18_AWO}	3	mA	VDD18_AWO
	I/O power supply current	I_{DD1833_AWO}	8	mA	VDD1833_AWO
	Pre-driver power supply current	$I_{DD1833PRE18\_AWO}$	14	mA	VDD1833_PRE18_AWO
USB30	1.8-V PHY power supply current	$I_{DDUSB30\_USVPH}$	31	mA	USB30_USVPH
	0.8-V PHY power supply current	$I_{DDUSB30\_USVPTX}$	53	mA	USB30_USVPTX
	3.3-V PHY power supply current	$I_{DDUSB30\_USVDD33}$	12	mA	USB30_USVDD33
	1.8-V PHY power supply current	$I_{DDUSB30\_USVDD18}$	56	mA	USB30_USVDD18
	0.8-V PHY power supply current	$I_{DDUSB30\_USDVDD}$	16	mA	USB30_USDVDD
USB20	3.3-V PHY power supply current	$I_{DDUSB20\_USVDD33}$	12	mA	USB20_USVDD33
	1.8-V PHY power supply current	$I_{DDUSB20\_USVDD18}$	56	mA	USB20_USVDD18
	0.8-V PHY power supply current	$I_{DDUSB20\_USDVDD}$	16	mA	USB20_USDVDD
TSU0	1.8-V power supply current	$I_{DDTS0AVDD18}$	1	mA	TS0AVDD18
	0.8-V power supply current	$I_{DDTS0DVDD08A}$	1	mA	TS0DVDD08A
TSU1	1.8-V power supply current	$I_{DDTS1AVDD18}$	1	mA	TS1AVDD18
	0.8-V power supply current	$I_{DDTS1DVDD08A}$	1	mA	TS1DVDD08A
xSPI	I/O power supply current	I_{DD1833_XSPI}	16	mA	VDD1833_XSPI
SD0	I/O power supply current	I_{DD1833_SD0}	16	mA	VDD1833_SD0
	Pre-driver power supply current	$I_{DD1833PRE18\_SD}$	2	mA	VDD1833_PRE18_SD
SD1	I/O power supply current	I_{DD1833_SD1}	9	mA	VDD1833_SD1
SD2	I/O power supply current	I_{DD1833_SD2}	9	mA	VDD1833_SD2
	Pre-driver power supply current	$I_{DD1833PRE18\_SD2}$	2	mA	VDD1833_PRE18_SD2
OTP	1.8-V power supply current	$I_{DDOTPVDD18}$	6	mA	OTPVDD18
DDR0	0.8-V core power supply current	I_{DD08_DDR}	934	mA	VDD08_DDR
	1.1-V PHY power supply current	I_{DDQ_DDR0}	760	mA	DDR0_VDDQ
	PHY power supply current	I_{DDQLP_DDR0}	242	mA	DDR0_VDDQLP
	1.8-V PLL power supply current	I_{DDVAA_DDR0}	5	mA	DDR0_VAA
GBETH0	I/O power supply current	I_{DD1833_ET0}	11	mA	VDD1833_ET0
	Pre-driver power supply current	$I_{DD1833PRE18\_ET0}$	4	mA	VDD1833_PRE18_ET
GBETH1	I/O power supply current	I_{DD1833_ET1}	11	mA	VDD1833_ET1

Table 10.1-3 Max. Supply Currents during Operation (2/2)

Unit Name	Item	Symbol	Max.	Unit	Note
CRU0	1.8-V PHY power supply current	I _{DDMSVDD18_CSI0}	8	mA	CSI0_MSVDD18
	0.8-V core power supply current	I _{DDMSVDD0P8_CSI0}	25	mA	CSI0_MSVDD0P8
CRU1	1.8-V PHY power supply current	I _{DDMSVDD18_CSI1}	8	mA	CSI1_MSVDD18
	0.8-V core power supply current	I _{DDMSVDD0P8_CSI1}	25	mA	CSI1_MSVDD0P8
DSI	0.8-V core power supply current	I _{DD0P8_DSI}	43	mA	DSI_VDD0P8
	1.2-V PHY power supply current	I _{DD12_DSI}	1	mA	DSI_VDD12
	1.8-V PHY power supply current	I _{DD18_DSI}	8	mA	DSI_VDD18
PCIE	1.8-V power supply current	I _{DDPCIEVCC18ACMN}	19	mA	PCIE_VCC18ACMN
	1.8-V PHY power supply current	I _{DDPCIEVCC18AL01}	53	mA	PCIE_VCC18AL01
	0.8-V PHY power supply current	I _{DDPCIEVCC08AL01}	112	mA	PCIE_VCC08AL01
I3C	I/O power supply current	I _{DD1218_I3C}	1	mA	VDD1218_I3C
ADC	1.8-V analog power supply current	I _{DDADC0_ADAVDD18}	1	mA	ADC0_ADAVDD18
	1.8-V analog power supply current	I _{DDADC1_ADAVDD18}	1	mA	ADC1_ADAVDD18
	1.8-V analog power supply current	I _{DDADC2_ADAVDD18}	1	mA	ADC2_ADAVDD18
CPG	PLLCM33 1.8-V power supply current	I _{DDPLVDD_PLLCM33}	2	mA	PLVDD_PLLCM33
	PLLCLN_DTY_DRP 1.8-V power supply current	I _{DDPLVDD_PLLCLNDTYDRP}	6	mA	PLVDD_PLLCLN_DTY_DRP
	PLLCA55 1.8-V power supply current	I _{DDPLVDD_PLLCA55}	2	mA	PLVDD_PLLCA55
	PLLVD0_DSI 1.8-V power supply current	I _{DDPLVDD_PLLVCD0SI}	4	mA	PLVDD_PLLVDO_DSI
	PLLDDR0 1.8-V power supply current	I _{DDPLVDD_PLLDDR0}	2	mA	PLVDD_PLLDDR0
	PLLETH_GPU 1.8-V power supply current	I _{DDPLVDD_PLLETHGPU}	4	mA	PLVDD_PLLETH_GPU
	PLLCM33 0.8-V power supply current	I _{DDPLVDD08_PLLCM33}	3	mA	PLDVDD08_PLLCM33
	PLLCLN_DTY_DRP 0.8-V power supply current	I _{DDPLVDD08_PLLCLNDTYDRP}	8	mA	PLDVDD08_PLLCLN_DTY_DRP
	PLLCA55 0.8-V (or 0.9-V) power supply current	I _{DDPLVDD08_PLLCA55}	3	mA	PLDVDD09_PLLCA55
	PLLVD0_DSI 0.8-V power supply current	I _{DDPLVDD08_PLLVCD0SI}	5	mA	PLDVDD08_PLLVDO_DSI
	PLLDDR0 0.8-V power supply current	I _{DDPLVDD08_PLLDDR0}	3	mA	PLDVDD08_PLLDDR0
	PLLETH_GPU 0.8-V power supply current	I _{DDPLVDD08_PLLETHGPU}	5	mA	PLDVDD08_PLLETH_GPU
CST	I/O power supply current	I _{DD1833_JTAG}	2	mA	VDD1833_JTAG
	Pre-driver power supply current	I _{DD1833PRE18_JTAG}	1	mA	VDD1833_PRE18_JTAG
PWC	1.8-V I/O power supply current	I _{DD18_PWC}	1	mA	VDD18_PWC

10.1.4.2 Standard I/O Characteristics

For the I/O types, refer to the external pin list in **1.2 Pin**.

Table 10.1-4 DC Characteristics

$V_{DD} = 1.11\text{ V to }1.95\text{ V}$ (1.8/1.2-V switching I/O type), $V_{DD} = 1.65\text{ V to }1.95\text{ V}$ (1.8-V I/O type and 1.8-V OSC I/O type), $V_{DD} = 1.65\text{ V to }3.60\text{ V}$ (3.3/1.8-V switching I/O types 1, 2 and 3), $V_{DD} = 3.00\text{ V to }3.60\text{ V}$ (3.3-V I/O type)

Item	I/O Type	Symbol	Min.	Typ.	Max.	Unit	Condition
External voltage tolerance	3.3/1.8-V switching I/O type 2	V_{TOL}	—	—	3.6	V	V_{DD} power-off & on
High-level input voltage	1.8/1.2-V switching I/O type (1.2 V)	V_{IH}	$0.8 \times V_{DD}$	—	$V_{DD} + 0.3$	V	—
	1.8/1.2-V switching I/O type (1.8 V)	V_{IH}	$0.7 \times V_{DD}$	—	$V_{DD} + 0.3$	V	—
	1.8-V I/O type	V_{IH}	$0.7 \times V_{DD}$	—	$V_{DD} + 0.3$	V	—
	1.8-V OSC I/O type						
	3.3/1.8-V switching I/O type 1						
	3.3/1.8-V switching I/O type 2						
	3.3/1.8-V switching I/O type 3						
Low-level input voltage	1.8/1.2-V switching I/O type (1.2 V)	V_{IL}	-0.3	—	$0.2 \times V_{DD}$	V	—
	1.8/1.2-V switching I/O type (1.8 V)	V_{IL}	-0.3	—	$0.3 \times V_{DD}$	V	—
	1.8-V I/O type	V_{IL}	-0.3	—	$0.3 \times V_{DD}$	V	—
	1.8-V OSC I/O type						
	3.3/1.8-V switching I/O type 1						
	3.3/1.8-V switching I/O type 2						
	3.3/1.8-V switching I/O type 3						
Hysteresis voltage	1.8/1.2-V switching I/O type	ΔV	$0.1 \times V_{DD}$	—	—	V	—
	1.8-V I/O type						
	3.3/1.8-V switching I/O type 1* ¹	ΔV	$0.08 \times V_{DD}$	—	—	V	—
	3.3/1.8-V switching I/O type 2* ²						
	3.3/1.8-V switching I/O type 3* ^{1,3}	ΔV	0.1	—	—	V	—

Item	I/O Type	Symbol	Min.	Typ.	Max.	Unit	Condition
Input leakage current	1.8/1.2-V switching I/O type (1.2 V)	I_I	-10	—	10	μA	$V_{in} = V_{SS}$ or V_{DD} max & V_{DD} power-on
			-10	—	10	μA	$V_{in} = V_{SS}$ or V_{DD} max & V_{DD} power-off
	1.8/1.2-V switching I/O type (1.8 V)	I_I	-15	—	15	μA	$V_{in} = V_{DD}$ max & V_{DD} power-on
			-18	—	18	μA	$V_{in} = V_{SS}$ or V_{DD} max & V_{DD} power-off
	1.8-V I/O type	I_I	-12	—	12	μA	$V_{in} = V_{DD}$ max & V_{DD} power-on
	1.8-V OSC I/O type						
	3.3/1.8-V switching I/O type 1						
	3.3/1.8-V switching I/O type 2	I_I	-12	—	12	μA	$V_{in} = V_{DD}$ max & V_{DD} power-on
						μA	$V_{in} = V_{SS}$ or V_{DD} max & V_{DD} power-off
	3.3/1.8-V switching I/O type 3 3.3-V I/O type	I_I	-12	—	12	μA	$V_{in} = V_{DD}$ max & V_{DD} power-on
Input pull-down resistor current	1.8/1.2-V switching I/O type* ³ (1.2 V)	I_{RPU}	10	—	100	μA	$V_{in} = V_{DD}$ max
	1.8/1.2-V switching I/O type* ³ (1.8 V)	I_{RPU}	25	—	130	μA	$V_{in} = V_{DD}$ max
	1.8-V I/O type* ⁵	I_{RPU}	25	—	130	μA	$V_{in} = V_{DD}$ max
	3.3/1.8-V switching I/O type 1* ⁷ 3.3/1.8-V switching I/O type 2* ⁹	I_{RPU}	25	—	200	μA	$V_{in} = V_{DD}$ max
	3.3/1.8-V switching I/O type 3* ¹¹	I_{RPU}	18	—	148	μA	$V_{in} = V_{DD}$ max
Input pull-up resistor current	1.8/1.2-V switching I/O type* ⁴ (1.2 V)	I_{RPD}	-10	—	-100	μA	$V_{in} = V_{SS}$
	1.8/1.2-V switching I/O type* ⁴ (1.8 V)	I_{RPD}	-35	—	-185	μA	$V_{in} = V_{SS}$
	1.8-V I/O type* ⁶	I_{RPD}	-35	—	-185	μA	$V_{in} = V_{SS}$
	3.3/1.8-V switching I/O type 1* ⁸ 3.3/1.8-V switching I/O type 2* ¹⁰	I_{RPD}	-25	—	-200	μA	$V_{in} = V_{SS}$
	3.3/1.8-V switching I/O type 3* ¹²	I_{RPD}	-18	—	-192	μA	$V_{in} = V_{SS}$

Item	I/O Type	Symbol	Min.	Typ.	Max.	Unit	Condition
High-level output voltage	1.8/1.2-V switching I/O type (1.2 V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -1/-2/-4/-6$ mA (drive strength X1/X2/X4/X6)
	1.8/1.2-V switching I/O type (1.8 V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -2/-4/-8/-12$ mA (drive strength X1/X2/X4/X6)
	1.8-V I/O type 1.8-V OSC I/O type	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -2/-4/-8/-12$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 1 (1.8 V) 3.3/1.8-V switching I/O type 2 (1.8V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -1.6/-3.2/-6.4/-9.6$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 1 (3.3 V) 3.3/1.8-V switching I/O type 2 (3.3V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -2/-4/-8/-12$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 3 (1.8 V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -5/-6/-7/-10$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 3 (3.3 V)	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -9/-11/-13/-18$ mA (drive strength X1/X2/X4/X6)
	3.3-V I/O type	V_{OH}	$0.8 \times V_{DD}$	—	V_{DD}	V	$I_{OH} = -2/-4/-8/-12$ mA (drive strength X1/X2/X4/X6)
Low-level output voltage	1.8/1.2-V switching I/O type (1.2 V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 1/2/4/6$ mA (drive strength X1/X2/X4/X6)
	1.8/1.2-V switching I/O type (1.8 V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 2/4/8/12$ mA (drive strength X1/X2/X4/X6)
	1.8-V I/O type 1.8-V OSC I/O type	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 2/4/8/12$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 1 (1.8 V) 3.3/1.8-V switching I/O type 2 (1.8V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 1.6/3.2/6.4/9.6$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 1 (3.3 V) 3.3/1.8-V switching I/O type 2 (3.3V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 2/4/8/12$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 3 (1.8 V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 5/6/7/10$ mA (drive strength X1/X2/X4/X6)
	3.3/1.8-V switching I/O type 3 (3.3 V)	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 9/11/13/18$ mA (drive strength X1/X2/X4/X6)
	3.3-V I/O type	V_{OL}	0	—	$0.2 \times V_{DD}$	V	$I_{OL} = 2/4/8/12$ mA (drive strength X1/X2/X4/X6)

Item	I/O Type	Symbol	Min.	Typ.	Max.	Unit	Condition
Pull-up resistance	1.8/1.2-V switching I/O type* ⁴ (1.2 V)	R _{PU}	15	—	160	kΩ	—
	1.8/1.2-V switching I/O type* ⁴ (1.8 V)	R _{PU}	10	—	50	kΩ	—
	1.8-V I/O type* ⁶	R _{PU}	10	—	50	kΩ	—
	3.3/1.8-V switching I/O type 1* ⁸ (1.8 V)	R _{PU}	10	—	50	kΩ	—
	3.3/1.8-V switching I/O type 2* ¹⁰ (1.8 V)						
	3.3/1.8-V switching I/O type 1* ⁸ (3.3 V)	R _{PU}	10	—	100	kΩ	—
	3.3/1.8-V switching I/O type 2* ¹⁰ (3.3 V)						
	3.3/1.8-V switching I/O type 3* ¹²	R _{PU}	12	—	92	kΩ	—
Pull-down resistance	1.8/1.2-V switching I/O type* ³ (1.2 V)	R _{PD}	15	—	160	kΩ	—
	1.8/1.2-V switching I/O type* ³ (1.8 V)	R _{PD}	15	—	60	kΩ	—
	1.8-V I/O type* ⁵	R _{PD}	15	—	60	kΩ	—
	3.3/1.8-V switching I/O type 1* ⁷ (1.8 V)	R _{PD}	10	—	50	kΩ	—
	3.3/1.8-V switching I/O type 2* ⁹ (1.8 V)						
	3.3/1.8-V switching I/O type 1* ⁷ (3.3 V)	R _{PD}	10	—	100	kΩ	—
	3.3/1.8-V switching I/O type 2* ⁹ (3.3 V)						
	3.3/1.8-V switching I/O type 3* ¹¹	R _{PD}	13	—	92	kΩ	—
Input capacitance	—	C _{in}	—	—	10	pF	—

Note 1. Only for the TRSTN pin

Note 2. When the RIIC function is in use or the value of the PFC_SMT_mn register is 1

Note 3. Only for the P20 and P21 pins (when the value of the PFC_PUPD_mn register is 10b)

Note 4. Only for the P20 and P21 pins (when the value of the PFC_PUPD_mn register is 11b)

Note 5. Only for the QBYPASS, BSCANP, MD_BOOT0, MD_BOOT3, MD_BOOT4, BOOTSELCPU, and BOOTPLLCA_0 pins

Note 6. Only for the MD_BOOT1, MD_BOOT2, BOOTPLLCA_1, and MD_CLKS pins

Note 7. Only for the WDTUDFCA, WDTUDFCM, and SCIF_RXD, SCIF_TXD, XSPI0_CS0N, XSPI0_RESET0N, XSPI0_RST00N, XSPI0_INT0N, and XSPI0_ECS0N pins (when the value of the PFC_PUPD_mn register is 10b)

Note 8. Only for the WDTUDFCA, WDTUDFCM, and SCIF_RXD, SCIF_TXD, XSPI0_CS0N, XSPI0_RESET0N, XSPI0_RST00N, XSPI0_INT0N, and XSPI0_ECS0N pins (when the value of the PFC_PUPD_mn register is 11b)

Note 9. When the value of the PFC_PUPD_mn register is 10b

Note 10. When the value of the PFC_PUPD_mn register is 11b

Note 11. Only for the pins other than SD0CLK, SD0RSTN, SD1CLK, ET0_PHYINTR, and ET1_PHYINTR (when the value of the PFC_PUPD_mn register is 10b)

Note 12. Only for the pins other than SD0CLK, SD0RSTN, SD1CLK, ET0_PHYINTR, and ET1_PHYINTR (when the value of the PFC_PUPD_mn register is 11b)

Note 13. Only for the P90, P91, P92, PB0, PB1, PB2, PB3, PB4, and PB5 pins (when the RIIC function is in use or the value of the PFC_SMT_mn register is 1)

10.1.5 AC Characteristics

Conditions:

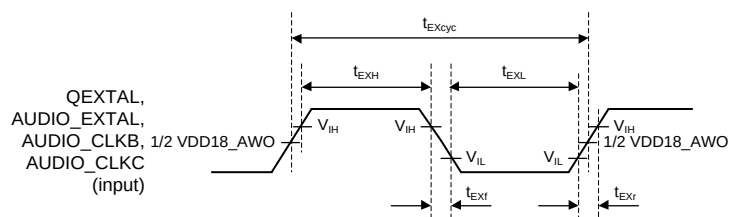
$VDD18 = VDD18\_AWO = VDD1833\_*$ (1.8 V mode)

$VDD33 = VDD1833\_*$ (3.3 V mode)

10.1.5.1 Clock Timing

Table 10.1-5 Clock Timing Table

Item	Symbol	Min.	Max.	Unit	Figures
QEXTAL clock input frequency	f_{EX}	24 -50 ppm	24 +50 ppm	MHz	Figure 10.1-8
QEXTAL clock input cycle time	t_{EXcyc}	41.67	41.67	ns	
AUDIO_EXTAL clock input frequency	f_{EX}	4	48	MHz	
AUDIO_EXTAL clock input cycle time	t_{EXcyc}	20.83	250	ns	
AUDIO_CLKB, AUDIO_CLKC clock input frequency (external clock is input)	f_{EX}	4	50	MHz	
AUDIO_CLKB, AUDIO_CLKC clock input cycle time (external clock is input)	t_{EXcyc}	20	250	ns	
QEXTAL clock input low-level pulse width	t_{EXL}	0.4	0.6	t_{EXcyc}	
QEXTAL clock input high-level pulse width	t_{EXH}	0.4	0.6	t_{EXcyc}	
AUDIO_EXTAL, AUDIO_CLKB, AUDIO_CLKC clock input low-level pulse width	t_{EXL}	0.45	0.55	t_{EXcyc}	
AUDIO_EXTAL, AUDIO_CLKB, AUDIO_CLKC clock input high-level pulse width	t_{EXH}	0.45	0.55	t_{EXcyc}	
QEXTAL, AUDIO_EXTAL, AUDIO_CLKB, AUDIO_CLKC clock input rise time	t_{EXr}	—	4	ns	
QEXTAL, AUDIO_EXTAL, AUDIO_CLKB, AUDIO_CLKC clock input fall time	t_{EXf}	—	4	ns	
Mode hold time	t_{MDH}	—	100	ns	Figure 10.1-9
Mode setup time	t_{MDS}	—	100	ns	



Note: When the clock is input on the QEXTAL, AUDIO_EXTAL, AUDIO_CLKB, or AUDIO_CLKC

Figure 10.1-8 Clock Input Timing

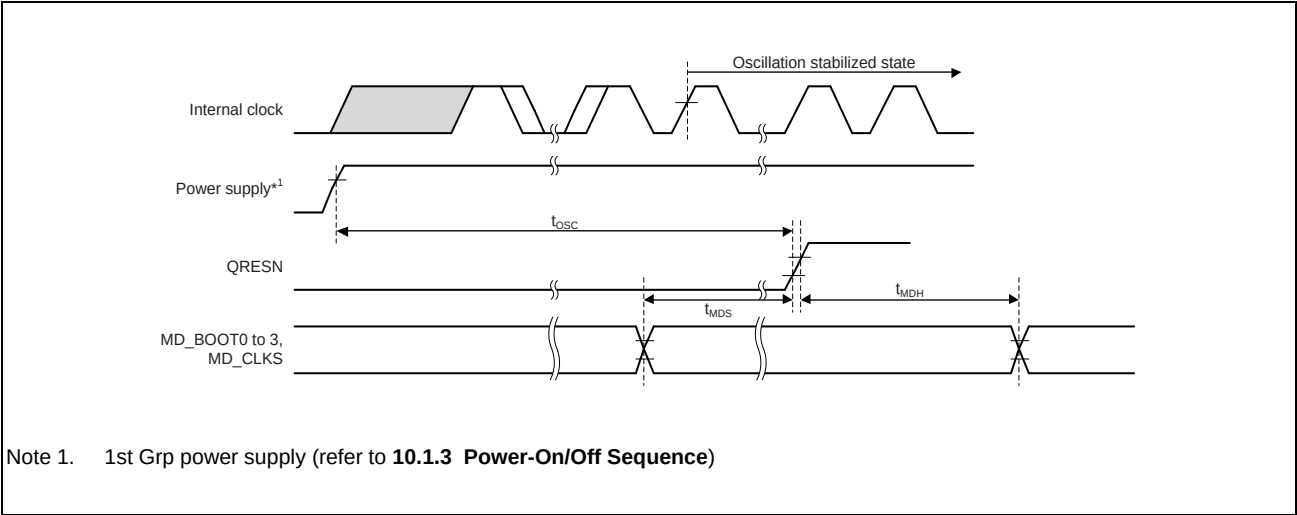


Figure 10.1-9 Power-On Oscillation Settling Time

10.1.5.2 CMTW Timing

Table 10.1-6 CMTW Timing

Parameter		Symbol	Min.	Max.	Unit	Figure
CMTW	Input capture input pulse width	Single-edge setting	t_{CMTWICW}	1.5	—	t_{PLCyc}^{*1} Figure 10.1-10
		Both-edge setting		2.5	—	

Note 1. t_{PLCyc} : PCLKL cycle

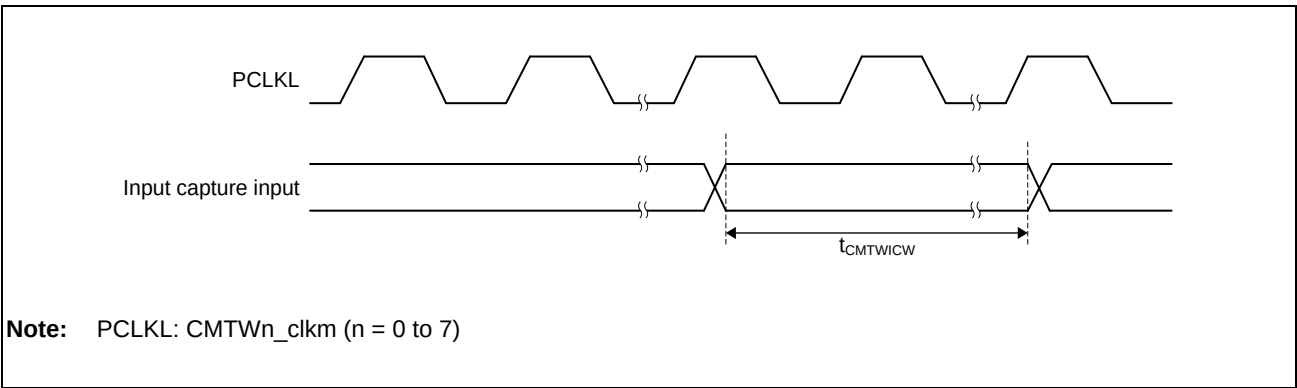


Figure 10.1-10 CMTW Input Capture Input Timing

10.1.5.3 POEG and GPT Trigger Timings

GPT Conditions: High-drive output is selected in the Port Drive Capability bit in the PFC_IOLH_mn register.

Table 10.1-7 POEG and GPT Trigger Timings

Parameter		Symbol	Min.	Max.	Unit	Figure
POEG	POEG input trigger pulse width	t_{POEW}	1.5	—	t_{Pcyc}^{*1}	Figure 10.1-11
GPT	Input capture pulse width	Single edge t_{GTICW}	1.5	—	t_{PDcyc}^{*2}	Figure 10.1-12
		Dual edge	2.5	—		

Note 1. t_{Pcyc} : POEGx_n_PCLK cycle (x = A to D, n = 0, 1)

Note 2. t_{PDcyc} : GPT_n_clks_gpt cycle (n = 0, 1)

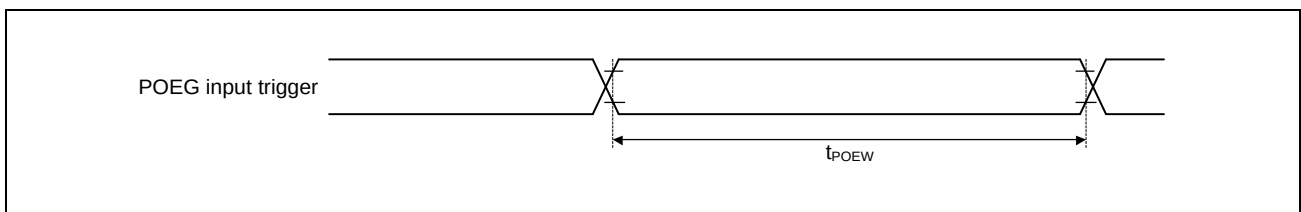


Figure 10.1-11 POEG Input Trigger Timing

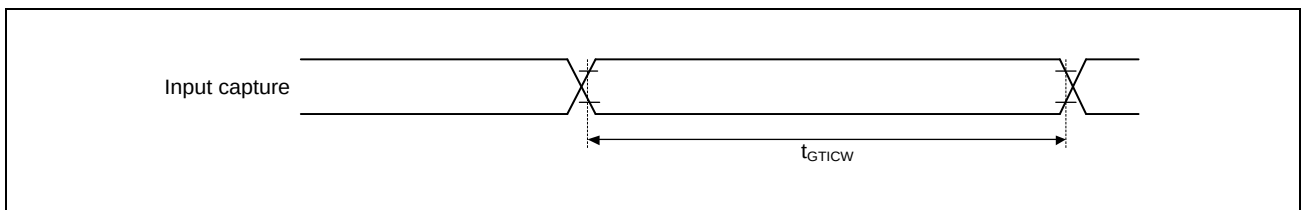


Figure 10.1-12 GPT Input Capture Timing

10.1.5.4 Watchdog Timer Access Timing

Table 10.1-8 Watchdog Timer Timing

Item	Symbol	Min.	Max.	Unit	Figures
WDTUDFCM / WDTUDFCA output time	t_L	64	64	t_{P1cyc}^{*1}	Figure 10.1-13

Note 1. t_{P1cyc} indicates peripheral clock WDT_n_clk_loco (n = 0 to 3).

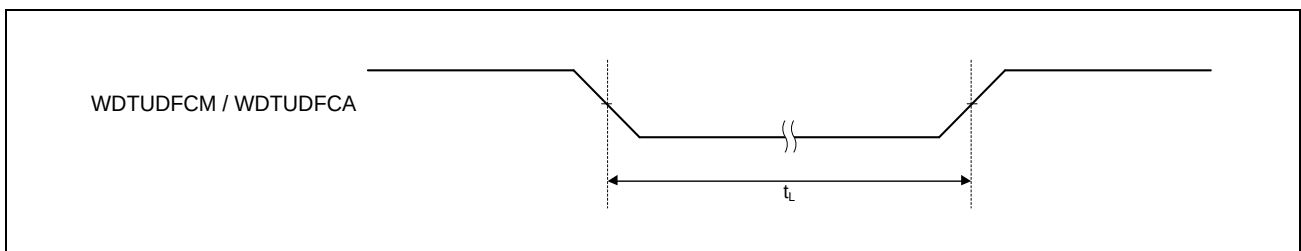


Figure 10.1-13 Watchdog Timer Output Timing

10.1.5.5 DMAC Timing

Table 10.1-9 DMAC Timing

Item	Symbol	Min.	Max.	Unit	Figures
DREQn pulse width	t_{DREQW}	20	—	t_{cyc}^{*1}	Figure 10.1-14
TENDn pulse width	t_{TENDW}	16	16	$t_{PCLKcyc}^{*2}$	Figure 10.1-15

- Note 1. t_{cyc} = 41.666 ns (24 MHz)
Note 2. $t_{PCLKcyc}$ = 10 ns (100 MHz): ICU_0_PCLK_I cycle

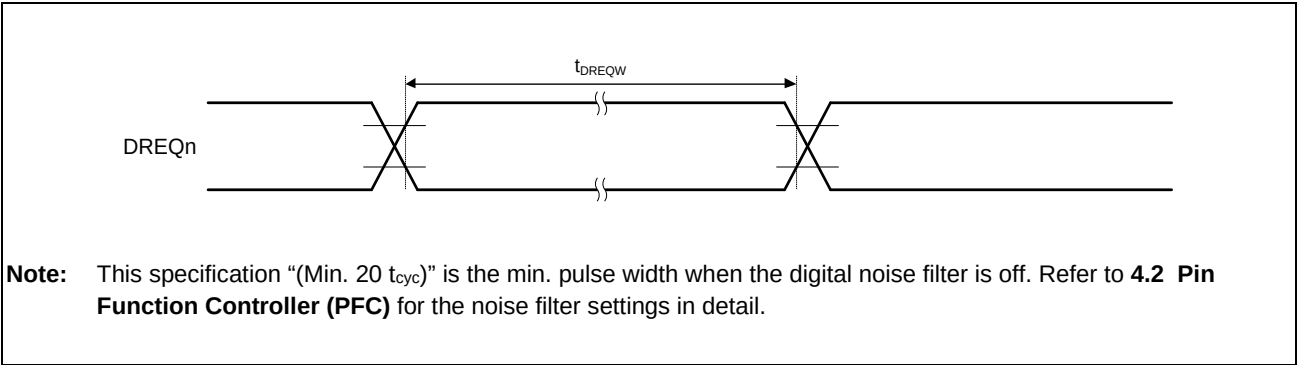


Figure 10.1-14 DMAC DREQn Timing

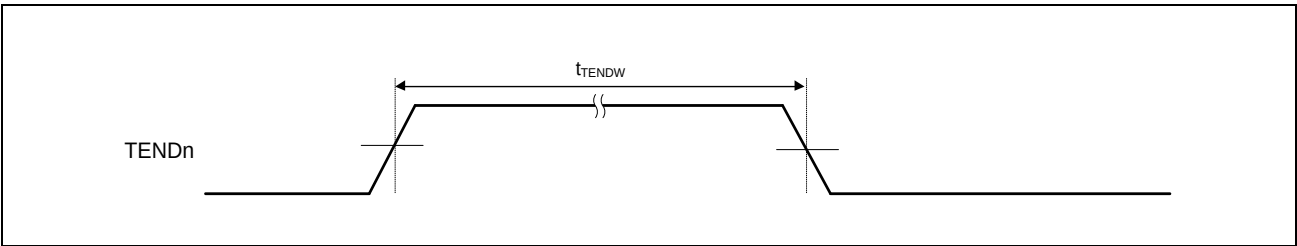


Figure 10.1-15 DMAC TENDn Timing

10.1.5.6 LPDDR4 PHY Characteristics

The LPDDR4 PHY of this LSI is compliant with the JEDEC 209-4D / JEDEC 209-4-1A standard.

10.1.5.7 SD Access Timing

Conditions:

$$V_{OH} = VDD33 \times 0.7$$

$$V_{OL} = VDD33 \times 0.3$$

$$C = 40 \text{ pF (3.3 V)}$$

Drive strength: $\times 6$

10.1.5.7.1 SD Access Timing (SDR 3.3-V)

Table 10.1-10 SD AC Access Timing (SDR at 3.3-V Operation)

Item	Symbol	Default Speed Mode (25 MHz)		High Speed Mode (50 MHz)		Unit	Figures
		Min.	Max.	Min.	Max.		
SDnCLK clock cycle	t_{SDCYC}	40.0	—	20.0	—	ns	Figure 10.1-16
SDnCLK clock high level width	t_{SDWH}	10	—	7	—	ns	
SDnCLK clock low level width	t_{SDWL}	10	—	7	—	ns	
SDnCLK clock rise time	t_{SDLH}	—	10	—	3	ns	
SDnCLK clock fall time	t_{SDHL}	—	10	—	3	ns	
SDnCMD,SDnDATm output delay	t_{SDODLY}	-7.5	2.5	-6.2	2.5	ns	
SDnCMD,SDnDATm input set up time	t_{SDIS}	4.0	—	4.0	—	ns	
SDnCMD,SDnDATm input hold time	t_{SDIH}	2.0	—	2.0	—	ns	
SDnCMD,SDnDATm input data width	t_{SDIDW}	—	—	—	—	ns	

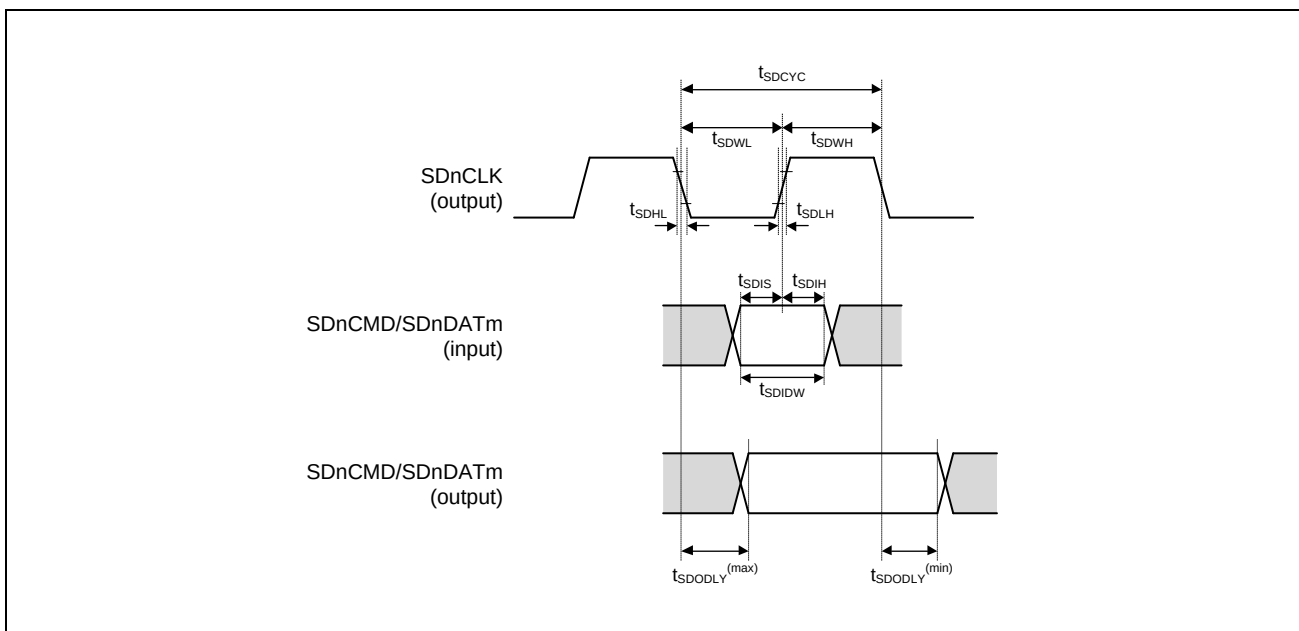


Figure 10.1-16 SDHC Interface Timing (SDR 3.3-V Power Supply)

NOTE

The disclosure of other characteristics of the SD interface needs the conclusion of the following agreement.

- SD Host/Ancillary Product License Agreement (SD HALA)

For details, contact Renesas sales representatives.

10.1.5.8 eMMC Access Timing

Conditions:

$$V_{OH} = V_{DD18} \times 0.7, V_{OL} = V_{DD18} \times 0.3, C = 15 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.7, V_{OL} = V_{DD33} \times 0.3, C = 30 \text{ pF (3.3 V)}$$

Drive strength: ×6

10.1.5.8.1 eMMC host interface timing (default)

Table 10.1-11 eMMC Host Interface Timing (MMC Default 3.3-V Power Supply)

Item	Symbol	Min.	Max.	Unit	Figures
SDnCLK clock cycle	t_{MMCPP}	20.0	—	ns	Figure 10.1-17
SDnCLK clock high level width	t_{MMCWH}	7	—	ns	
SDnCLK clock low level width	t_{MMCWL}	7	—	ns	
SDnCLK clock rise time	t_{MMCLH}	—	3	ns	
SDnCLK clock fall time	t_{MMCHL}	—	3	ns	
SDnCMD/SDnDATm output delay	$t_{MMCODLY}$	−6.2	2.5	ns	
SDnCMD/SDnDATm input setup time	t_{MMCISU}	4.0	—	ns	
SDnCMD/SDnDATm input hold time	t_{MMCIH}	2.0	—	ns	
SDnCMD/SDnDATm input data width	t_{MMCIDW}	—	—	ns	

Table 10.1-12 eMMC Host Interface Timing (MMC Default 1.8-V Power Supply)

Item	Symbol	Min.	Max.	Unit	Figures
SDnCLK clock cycle	t_{MMCPP}	20.0	—	ns	Figure 10.1-17
SDnCLK clock high level width	t_{MMCWH}	7	—	ns	
SDnCLK clock low level width	t_{MMCWL}	7	—	ns	
SDnCLK clock rise time	t_{MMCLH}	—	3	ns	
SDnCLK clock fall time	t_{MMCHL}	—	3	ns	
SDnCMD/SDnDATm output delay	$t_{MMCODLY}$	−4.2	1.6	ns	
SDnCMD/SDnDATm input setup time	t_{MMCISU}	1.3	—	ns	
SDnCMD/SDnDATm input hold time	t_{MMCIH}	1.878	—	ns	
SDnCMD/SDnDATm input data width	t_{MMCIDW}	—	—	ns	

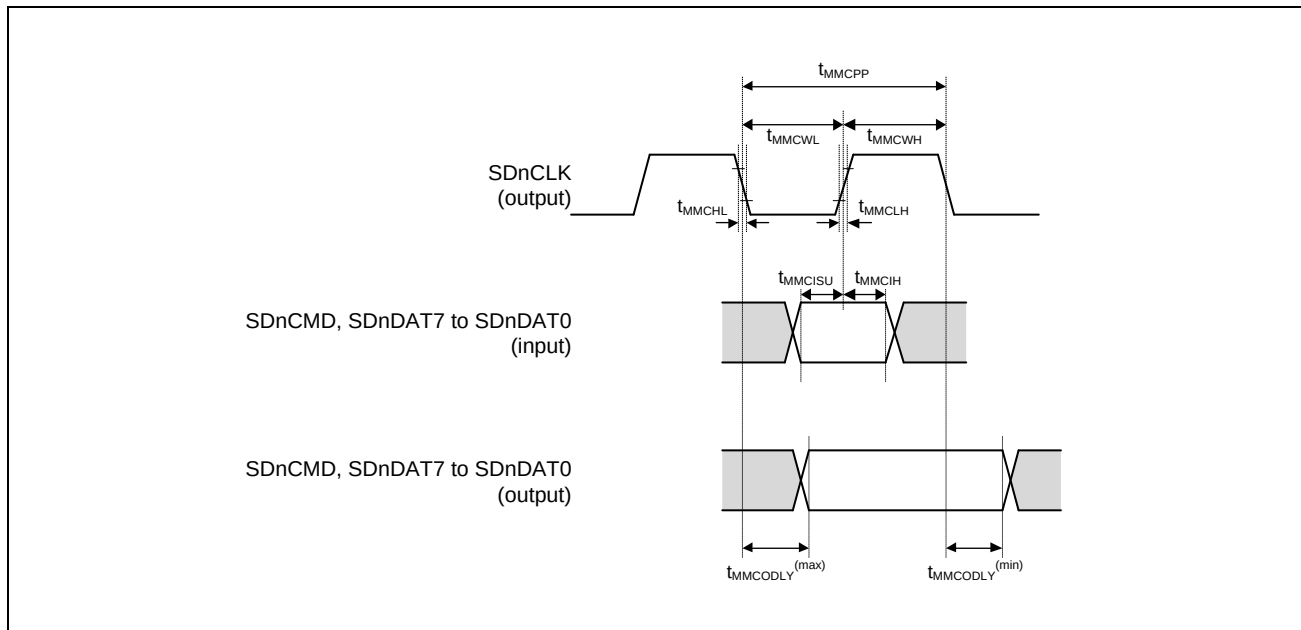


Figure 10.1-17 eMMC Host Interface Timing (MMC Default 1.8-V/3.3-V Power Supply)

10.1.5.8.2 eMMC host interface timing (HS-SDR)

NOTES

1. The spec of eMMC host interface timing (HS-SDR 3.3-V power supply) is the same as **Table 10.1-11 eMMC Host Interface Timing (MMC Default 3.3-V Power Supply)**.
2. The spec of eMMC host interface timing (HS-SDR 1.8V power supply) is the same as **Table 10.1-12 eMMC Host Interface Timing (MMC Default 1.8-V Power Supply)**.

10.1.5.8.3 eMMC host interface timing (HS-DDR)

Table 10.1-13 eMMC Host Interface Timing (HS-DDR 3.3-V Power Supply Operation)

Item	Symbol	High Speed Mode (50 MHz)		Unit	Figures
		Min.	Max.		
SDnCLK clock cycle	t_{SDCYC}	20.0	—	ns	Figure 10.1-18
SDnCLK clock high level width	t_{SDWH}	9.0	11.0	ns	
SDnCLK clock low level width	t_{SDWL}	9.0	11.0	ns	
SDnCLK clock rise time	t_{SDLH}	—	3.0	ns	
SDnCLK clock fall time	t_{SDHL}	—	3.0	ns	
SDnCMD output delay	t_{SDODLY}	-6.0	6.0	ns	
SDnCMD input set up time	t_{SDIS}	4.8	—	ns	
SDnCMD input hold time	t_{SDIH}	2.5	—	ns	
SDnDATm output delay	t_{SDODLY_DDR}	2.5	6.5	ns	
SDnDATm input set up time	t_{SDIS_DDR}	1.768	—	ns	
SDnDATm input hold time	t_{SDIH_DDR}	1.5	—	ns	

Table 10.1-14 eMMC Host Interface Timing (HS-DDR 1.8-V Power Supply Operation)

Item	Symbol	High Speed Mode (50 MHz)		Unit	Figures
		Min.	Max.		
SDnCLK clock cycle	t_{MMCCYC}	20.0	—	ns	Figure 10.1-18
SDnCLK clock high level width	t_{MMCWH}	9.0	11.0	ns	
SDnCLK clock low level width	t_{MMCWL}	9.0	11.0	ns	
SDnCLK clock rise time	t_{MMCLH}	—	3.0	ns	
SDnCLK clock fall time	t_{MMCHL}	—	3.0	ns	
SDnCMD output delay	$t_{MMCODLY}$	−6.0	3.0	ns	
SDnCMD input set up time	t_{MMCIS}	4.8	—	ns	
SDnCMD input hold time	t_{MMCIH}	2.5	—	ns	
SDnDATm output delay	$t_{MMCODLY\_DDR}$	2.5	6.5	ns	
SDnDATm input set up time	t_{MMCIS_DDR}	1.768	—	ns	
SDnDATm input hold time	t_{SMCIIH_DDR}	1.5	—	ns	

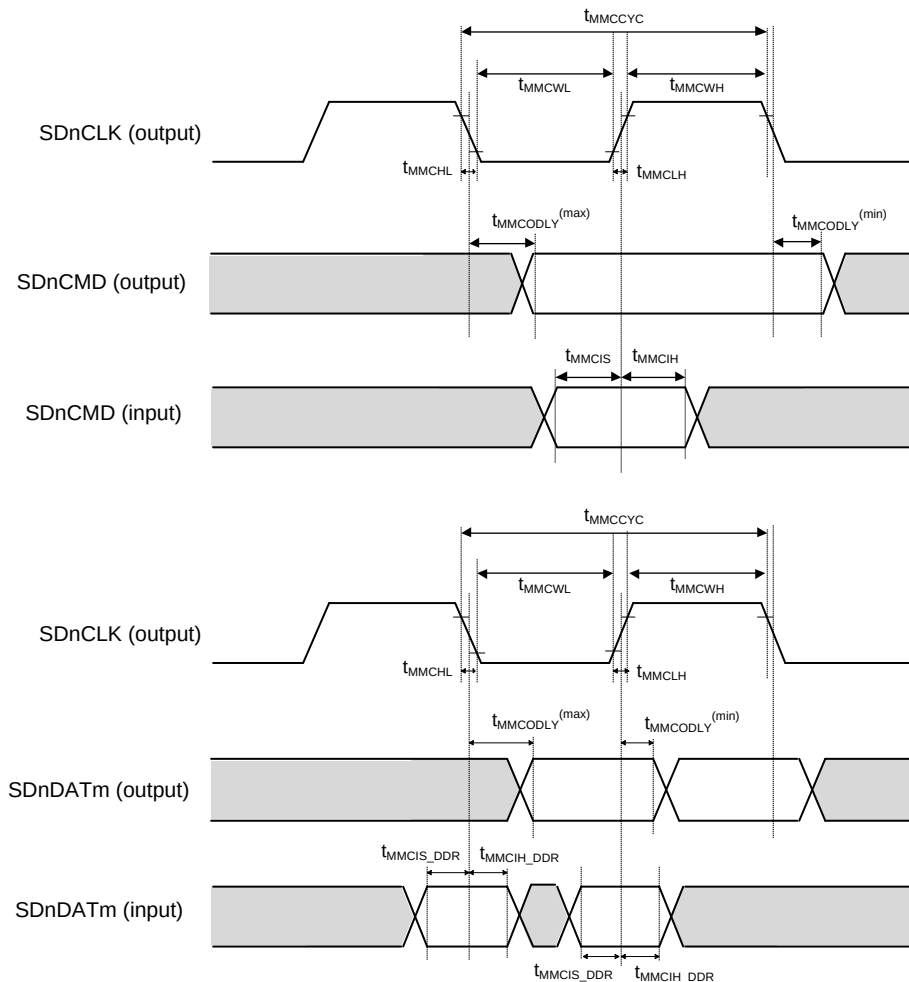


Figure 10.1-18 eMMC Host Interface (MMC Interface HS-DDR Mode 1.8/3.3-V Power Supply Selection)

10.1.5.8.4 eMMC host interface timing (HS200)

Table 10.1-15 eMMC Host Interface Timing (HS200 1.8-V Power Supply Operation, Output Load 15 pF)

Item	Symbol	Min.	Max.	Unit	Figures
SDnCLK clock cycle	$t_{\text{MMC}PP}$	5.0	10.0	ns	Figure 10.1-19
SDnCLK clock high level width	$t_{\text{MMC}WH}$	1.5	—	ns	
SDnCLK clock low level width	$t_{\text{MMC}WL}$	1.5	—	ns	
SDnCLK clock rise time	$t_{\text{MMC}LH}$	—	1.0	ns	
SDnCLK clock fall time	$t_{\text{MMC}HL}$	—	1.0	ns	
SDnCMD/SDnDATm output delay	$t_{\text{MMC}ODLY}$	-1.7	0.9	ns	
SDnCMD/SDnDATm input setup time	$t_{\text{MMC}ISU}$	—	—	ns	
SDnCMD/SDnDATm input hold time	$t_{\text{MMC}IH}$	—	—	ns	
SDnCMD/SDnDATm input data width	$t_{\text{MMC}IDW}$	2.88	—	ns	

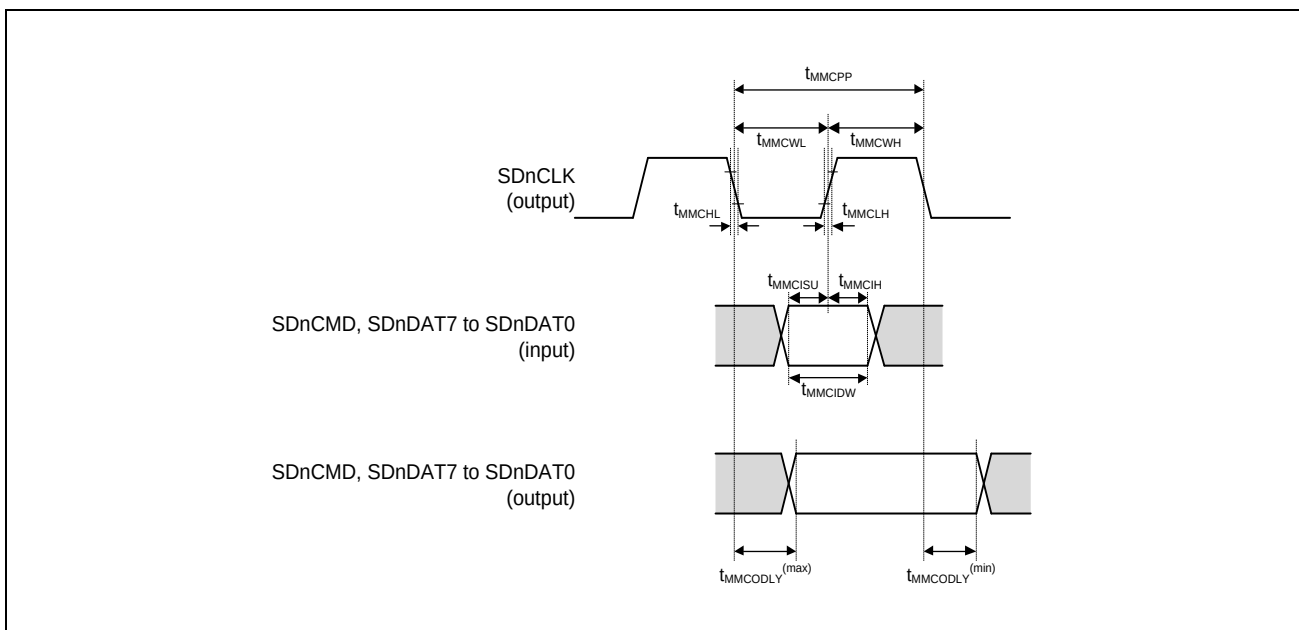


Figure 10.1-19 eMMC Host Interface (MMC Interface HS200 Mode 1.8-V Power Supply Selection)

10.1.5.9 Ethernet Interface Timing

Conditions:

$$V_{OH} = VDD18 \times 0.5, V_{OL} = VDD18 \times 0.5$$

$$V_{OH} = VDD33 \times 0.5, V_{OL} = VDD33 \times 0.5$$

$$C = 15 \text{ pF (RGMII)}, 30 \text{ pF (MII)}$$

Drive strength: $\times 2$, $\times 4$

Table 10.1-16 Ethernet Interface Timing (n = 0, 1)

Parameter		Symbol	Min.	Max.	Unit	Figure	
Ethernet (RGMII)	ETn_TXC_TXCLK, ETn_RXC_RXCLK cycle time duration	1 Gbps	$t_{RGMIIck}$	7.2	8.8	ns	Figure 10.1-20
		100 Mbps		36	44	ns	
		10 Mbps		360	440	ns	
	ETn_TXC_TXCLK, ETn_RXC_RXCLK frequency	1 Gbps	—	125 – 50 ppm	125 + 50 ppm	MHz	
		100 Mbps		25 – 50 ppm	25 + 50 ppm	MHz	
		10 Mbps		2.5 – 50 ppm	2.5 + 50 ppm	MHz	
	ETn_TXC_TXCLK, ETn_RXC_RXCLK duty cycle	1 Gbps	—	45	55	%	
		100 Mbps		40	60	%	
		10 Mbps					
	ETn_TXC_TXCLK, ETn_TXD0 to ETn_TXD3, ETn_TXCTL_TXEN, ETn_RXC_RXCLK, ETn_RXD0 to ETn_RXD3, ETn_RXCTL_RXDV rise/fall time		t_{RGMIIr} , t_{RGMIf}	—	0.75*1	ns	
	ETn_TXD0 to ETn_TXD3, ETn_TXCTL_TXEN, ETn_TXC_TXCLK output skew		$t_{RGMIIos}$	–0.5	0.5	ns	
	ETn_RXD0 to ETn_RXD3, ETn_RXCTL_RXDV setup time		t_{RGMIIs}	1	—	ns	
	ETn_RXD0 to ETn_RXD3, ETn_RXCTL_RXDV hold time		t_{RGMIIh}	1	—	ns	
Ethernet (MII)	ETn_TXC_TXCLK, ETn_RXC_RXCLK cycle time	100 Mbps	t_{MIICK}	40	—	ns	Figure 10.1-21
		10 Mbps		400	—	ns	
	ETn_TXC_TXCLK, ETn_RXC_RXCLK frequency	100 Mbps	—	25 – 50 ppm	25 + 50 ppm	MHz	
		10 Mbps		2.5 – 50 ppm	2.5 + 50 ppm	MHz	
	ETn_TXD0 to ETn_TXD3, ETn_TXCTL_TXEN, ETn_TXER output delay time		t_{MIId}	0	20	ns	
	ETn_RXD0 to ETn_RXD3, ETn_RXCTL_RXDV, ETn_RXER setup time		t_{MIIs}	10	—	ns	
	ETn_RXD0 to ETn_RXD3, ETn_RXCTL_RXDV, ETn_RXER hold time		t_{MIIf}	10	—	ns	

Note 1. The measurement condition of t_{RGMIIr} and t_{RGMIf} is in FIGURE 3 in Reduced Gigabit Media Independent Interface (RGMII) 12/10/2000 Version 1.3.

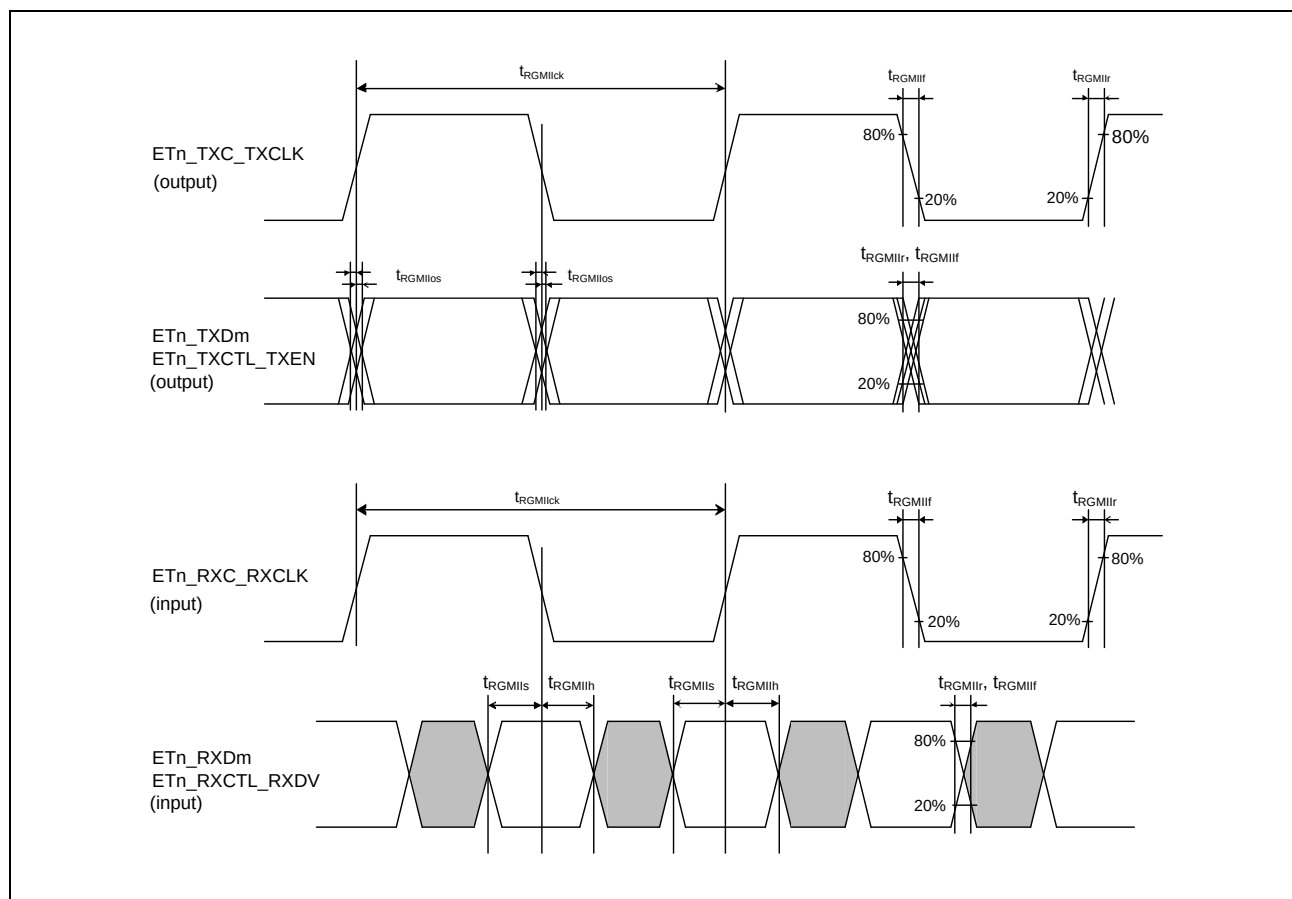


Figure 10.1-20 RGMII Transmission and Reception Timing (n = 0, 1)

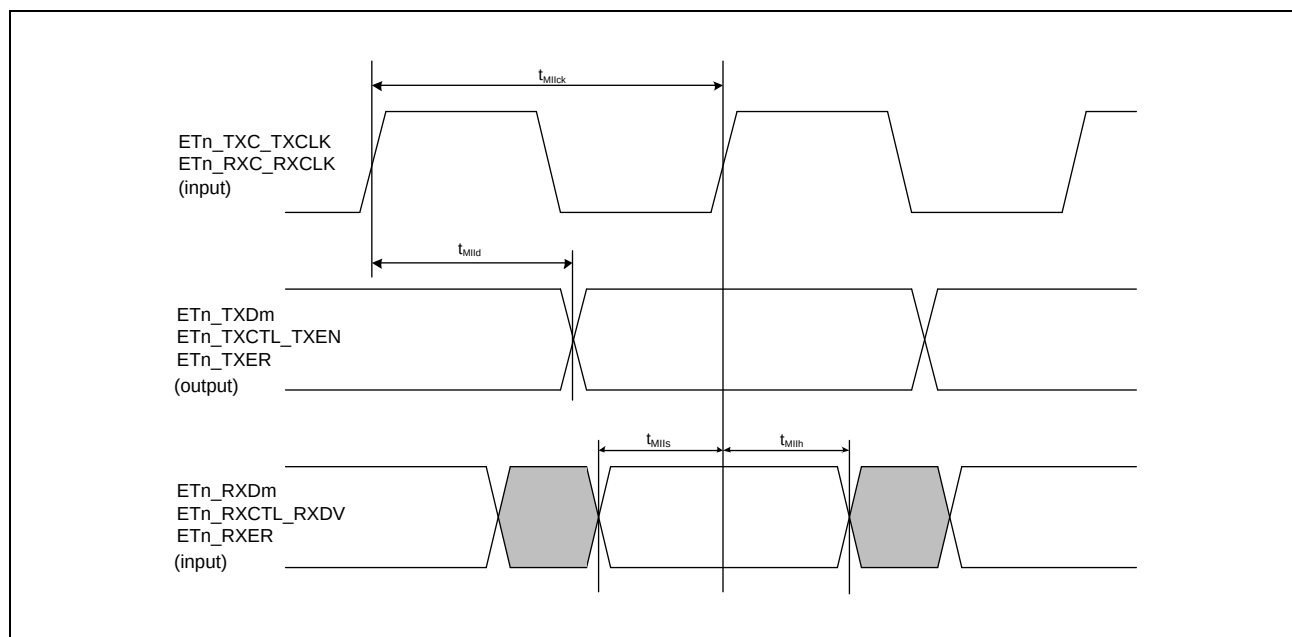


Figure 10.1-21 MII Transmission and Reception Timing (n = 0, 1)

10.1.5.10 USB 3.2 PHY Characteristics

The USB3 PHY of this LSI is compliant with the following USB 3.2 Gen2x1 standard:

Universal Serial Bus 3.2 Specification

10.1.5.11 USB 2.0 PHY Characteristics

The USB2 PHY of this LSI is compliant with the following USB 2.0 standard:

Universal Serial Bus 2.0 Specification

10.1.5.12 PCI Express PHY Characteristics

The PCI Express PHY of this LSI is compliant with the following PCIe standard:

Revision 4.0 of the PCI Express® Base Specification for Gen1/Gen 2/ Gen 3

10.1.5.13 xSPI Timing

Conditions:

- Single-end clock

$$V_{OH} = V_{DD18} \times 0.8, V_{OL} = V_{DD18} \times 0.2, C = 15 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.8, V_{OL} = V_{DD33} \times 0.2, C = 15 \text{ pF (3.3 V)}$$

- Data

$$V_{OH} = V_{DD18} \times 0.8, V_{OL} = V_{DD18} \times 0.2, C = 15 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.8, V_{OL} = V_{DD33} \times 0.2, C = 15 \text{ pF (3.3 V)}$$

Drive strength: ×6

Table 10.1-17 xSPI Timing (1/2)

Parameter		Symbol	1.8V		3.3V		Unit	Figure
			Min.	Max.	Min.	Max.		
Cycle time	SDR	t _{PERIOD}	7.5	—	12.5	—	ns	Figure 10.1-22
	DDR		7.5	—	12.5	—	ns	
Clock output slew rate		t _{SRck}	0.75 / 0.56* ¹	—	1.03	—	V/ns	
Clock duty cycle distortion		t _{CKDCD}	0.0	t _{PERIOD} × 0.05	0.0	t _{PERIOD} × 0.05	ns	
Clock minimum pulse width		t _{CKMPW}	t _{PERIOD} × 0.45	—	t _{PERIOD} × 0.45	—	ns	
Differential clock crossing voltage		V _{OX(AC)}	0.4 × VDD18	0.6 × VDD18	—	—	V	
DS duty cycle distortion		t _{DSDCD}	0.0	t _{PERIOD} × 0.04	0.0	t _{PERIOD} × 0.04	ns	
DS minimum pulse width		t _{DSMPW}	t _{PERIOD} × 0.41	—	t _{PERIOD} × 0.41	—	ns	
Data input/output slew rate		t _{SR}	0.75 / 0.56* ¹	—	1.03	—	V/ns	
Data input setup time (to CK)	SDR	t _{SU}	2.0	—	2.4	—	ns	Figure 10.1-23
Data input hold time (to CK)		t _H	1.0	—	1.0	—	ns	
Data output delay time		t _{OD}	—	1.6* ²	—	1.8* ²	ns	
Data output hold time		t _{OH}	–1.5	—	–2.3	—	ns	
Data output buffer off time		t _{BOFF}	–1.5	—	–2.3	—	ns	
Data input setup time (to DS)	DDR* ²	t _{SU}	–0.6 / –0.8* ¹	—	–0.6 / –0.8* ¹	—	ns	Figure 10.1-24, Figure 10.1-25
Data input hold time (to DS)		t _H	t _{PERIOD} × 0.41 – 0.6 / t _{PERIOD} × 0.41 – 0.8* ¹	—	t _{PERIOD} × 0.41 – 0.6 / t _{PERIOD} × 0.41 – 0.8* ¹	—	ns	
Data output setup time (to CK)		t _{SUO}	0.6 / 1.0* ^{1,*4}	—	1.0	—	ns	
Data output hold time (to CK)		t _{HO}	0.6 / 1.0* ^{1,*4}	—	1.0	—	ns	
CS low to clock high		t _{CSLCKH}	6.0 / 8.0* ^{1,*3}	—	8.0* ³	—	ns	Figure 10.1-23 to Figure 10.1-25
Clock low to CS high		t _{CKLCSH}	6.0 / 8.0* ¹	—	8.0	—	ns	
CS high time		t _{CSTD}	1	16	1	16	t _{PERIOD}	

Table 10.1-17 xSPI Timing (2/2)

Parameter	Symbol	1.8V		3.3V		Unit	Figure
		Min.	Max.	Min.	Max.		
DS low to CS high	t_{DSLCSH}	6.0 / 8.0*1	—	10.6	—	ns	Figure 10.1-26
CS high to DS Tri-state	t_{CSHDST}	0.0	t_{PERIOD}	0.0	t_{PERIOD}	ns	
CS low to DS low*5	t_{CSLDSL}	0.0	12.5*6	0.0	17.4*6	ns	
DS Tri-state to CS low	t_{DSTCSL}	0.0	—	0.0	—	ns	

Note: CK: XSPI0_CKP (XSPI0_CKN)

DS: XSPI0_DS

CS: XSPI0_CS0N, XSPI0_CS1N

Note 1. Specification at 133 MHz / Specification at 100 MHz

Note 2. These are values when the OEN assertion is extended in the Output Enable Asserting extension bit (COMCFG.OEASTEX = 1b).

Note 3. These are the values when the CS assertion is extended in the CS asserting extension bit (LIOCFGCSn.CSASTEX = 1b).

Note 4. The standard value for xSPI266 is 0.8 ns.

Note 5. If the DS is high during the command & modifier phase when using JESD251 Profile 2.0 memory, the time from CS low to DS high must also meet this specification.

Note 6. When using JESD251 Profile 1.0 memory or JESD251 Profile 2.0 memory with LIOCFGCSn.LATEMD set to 0, this constraint does not apply if the internal pull-down resistor of the DS pin is enabled.

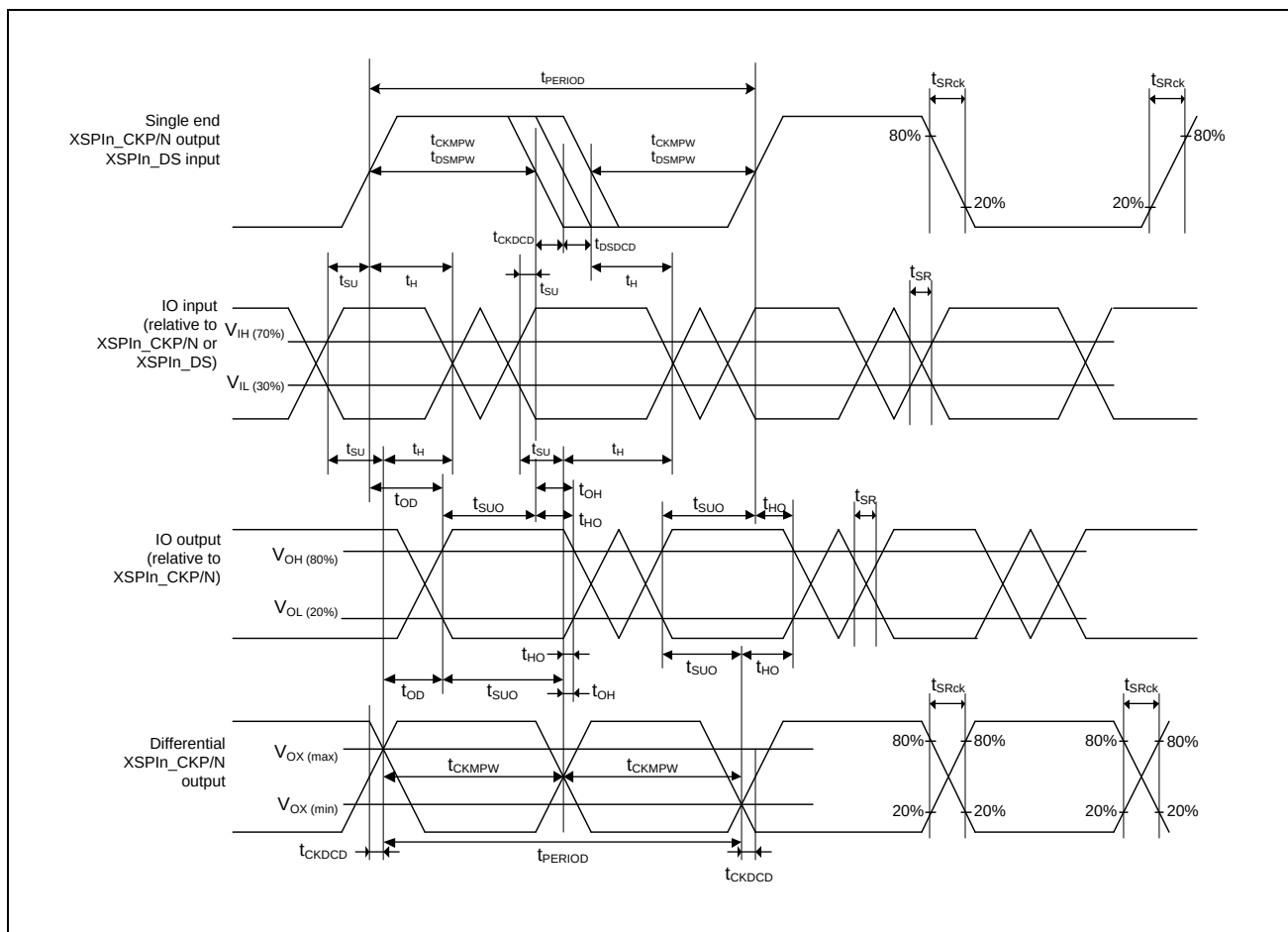


Figure 10.1-22 xSPI Clock / DS Timing

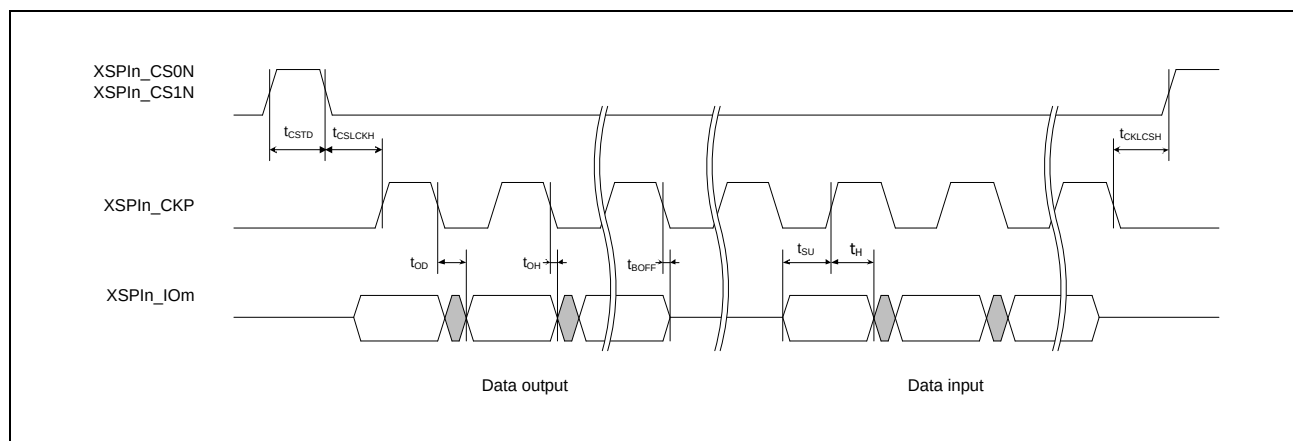


Figure 10.1-23 SDR Transmission and Reception Timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

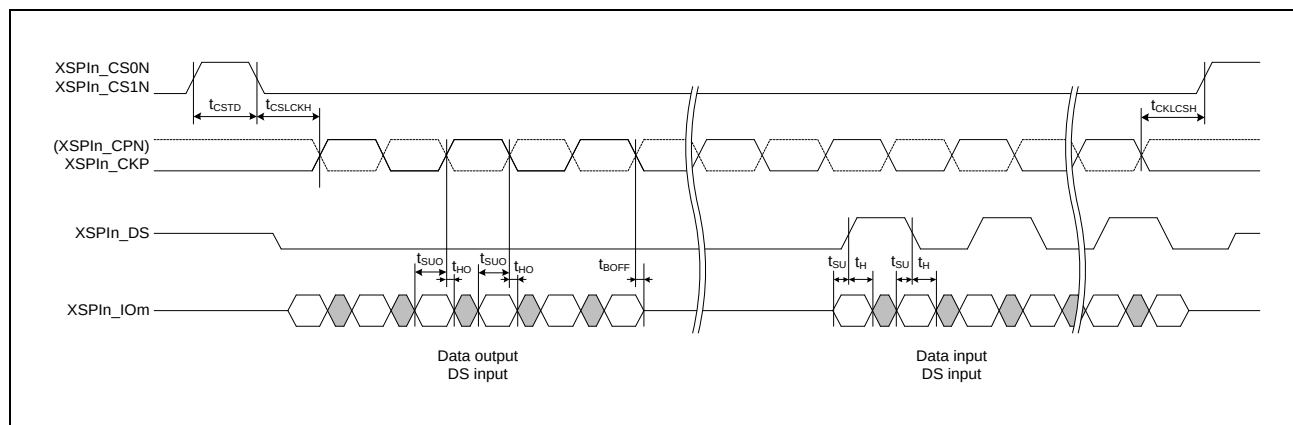


Figure 10.1-24 DDR Transmission and Reception Timing (4S-4D-4D, 8D-8D-8D)

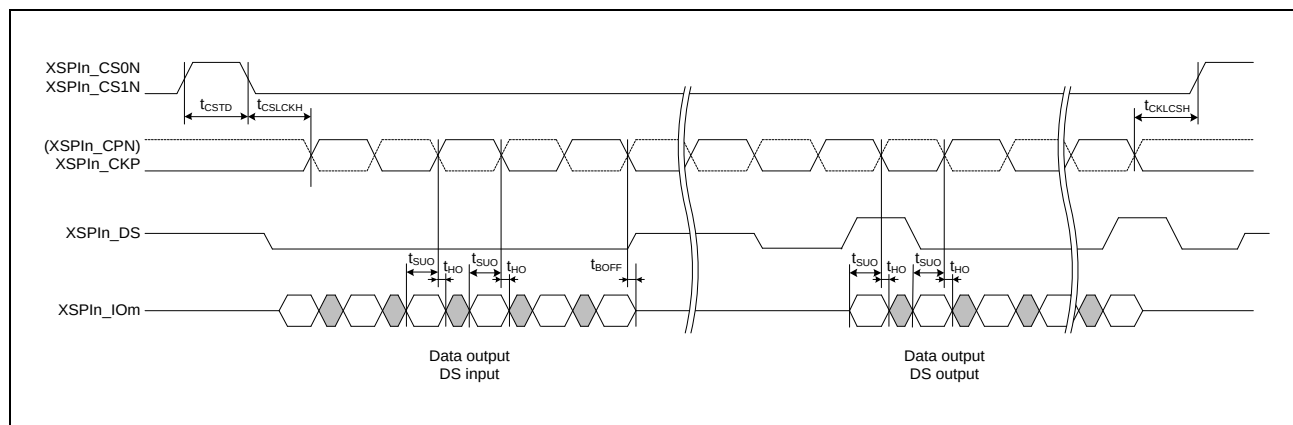


Figure 10.1-25 DDR Transmission and Reception Timing (HyperRAM write)

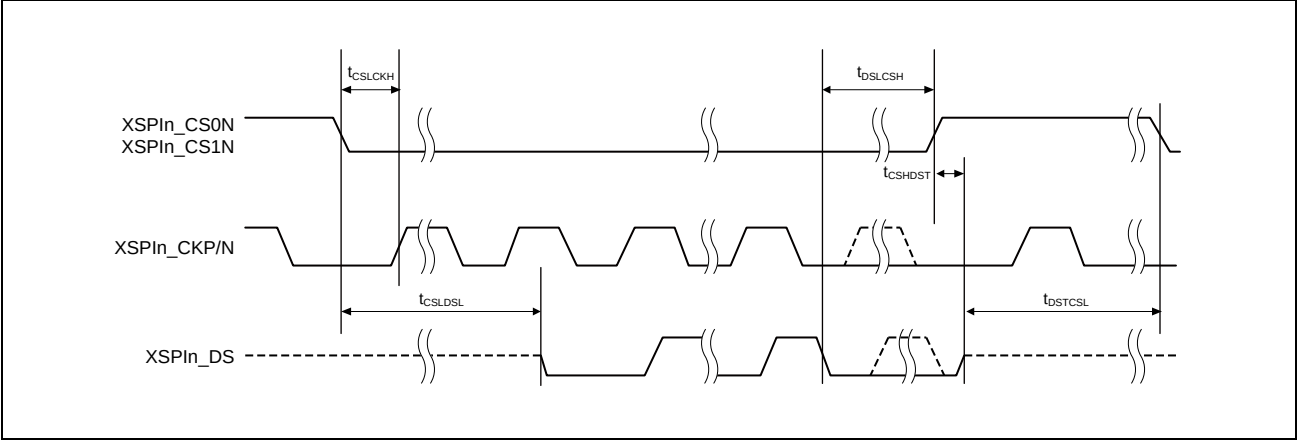


Figure 10.1-26 DS to CS Signal Timing

10.1.5.14 Serial Communications Interface (RSCI) Access Timing

Conditions:

$$V_{OH} = V_{DD18} \times 0.5, V_{OL} = V_{DD18} \times 0.5, C = 30 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.5, V_{OL} = V_{DD33} \times 0.5, C = 30 \text{ pF (3.3 V)}$$

Drive strength: $\times 2$, $\times 4$ (However, $\times 6$ only for SCL (P93) and SDA (P92) of RSCI0 in simple I2C mode)

Table 10.1-18 RSCI Timing (1/2)

Parameter	Symbol	Min.	Max.	Unit	Figure
RSCI (Asynchronous)	Input clock cycle	t_{Scyc}	4	—	$t_{PSCIcyc}$ Figure 10.1-27
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}
	Input clock rise time	t_{SCKr}	—	3	ns
	Input clock fall time	t_{SCKf}	—	3	ns
	Output clock cycle	t_{Scyc}	6	—	$t_{PSCIcyc}$
	Output clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}
	Output clock rise time	$V_{DD1833} = 1.8 \text{ V}$ $V_{DD1833} = 3.3 \text{ V}$ t_{SCKr}	— —	6.18^{*2} 7.9^{*2}	ns ns
	Output clock fall time	$V_{DD1833} = 1.8 \text{ V}$ $V_{DD1833} = 3.3 \text{ V}$ t_{SCKf}	— —	6.18^{*2} 7.9^{*2}	ns ns
RSCI (Simple I2C, Standard mode)	SDA input rise time	t_{Sr}	—	1000	ns Figure 10.1-28
	SDA input fall time	t_{Sf}	—	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	0	$2 \times NF_{cyc}^{*1}$	ns
	Data input setup time	t_{SDAS}	250	—	ns
	Data input hold time	t_{SDAH}	0	—	ns
	SCL, SDA capacitive load	C_b	—	400	pF
RSCI (Simple I2C, Fast mode)	SDA input rise time	t_{Sr}	—	300	ns Figure 10.1-28
	SDA input fall time	t_{Sf}	—	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	0	$2 \times NF_{cyc}^{*1}$	ns
	Data input setup time	t_{SDAS}	100	—	ns
	Data input hold time	t_{SDAH}	0	—	ns
	SCL, SDA capacitive load	C_b	—	400	pF

Table 10.1-19 RSCI Timing (2/2)

Parameter		Symbol	Min.	Max.	Unit	Figure
RSCI (Clock sync, Simple SPI)	SCK output clock cycle (master)	t_{SPcyc}	4	65536	$t_{PSClCyc}$	Figure 10.1-29 to Figure 10.1-34
	SCK input clock cycle (slave)		4	65536	$t_{PSClCyc}$	
	SCK clock high-level pulse width	t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low-level pulse width	t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	Input clock rise time	t_{SPCKR}	—	3	ns	
	Input clock fall time	t_{SPCKF}	—	3	ns	
	Output clock rise time	$V_{DD1833} = 1.8\text{ V}$	t_{SPCKR}	—	6.18^{*2}	ns
		$V_{DD1833} = 3.3\text{ V}$		—	7.9^{*2}	ns
	Output clock fall time	$V_{DD1833} = 1.8\text{ V}$	t_{SPCKF}	—	6.18^{*2}	ns
		$V_{DD1833} = 3.3\text{ V}$		—	7.9^{*2}	ns
	Data input setup time	Internal clock	t_{SU}	7	—	ns
		External clock		3	—	ns
	Data input hold time	Internal clock	t_H	3	—	ns
		External clock		3	—	ns
	Data output delay time	Internal clock	t_{OD}	—	3	ns
		External clock		—	12	ns
	Data output hold time	Internal clock	t_{OH}	−3	—	ns
		External clock		0	—	ns
	Data rise/fall time	$V_{DD1833} = 1.8\text{ V}$	t_{DR}, t_{DF}	—	6.18^{*2}	ns
		$V_{DD1833} = 3.3\text{ V}$		—	7.9^{*2}	ns
	Slave access time	Internal clock	t_{SA}	—	$3 \times t_{PSClCyc} + 12$	ns
		External clock		—	$3 \times t_{PSClCyc} + 12$	ns
	Slave output release time	Internal clock	t_{REL}	—	$3 \times t_{PSClCyc} + 12$	ns
		External clock		—	$3 \times t_{PSClCyc} + 12$	ns
RSCI (Simple SPI)	SS input setup time	t_{LEAD}	1	—	t_{SPcyc}	Figure 10.1-29 to Figure 10.1-34
	SS input hold time	t_{LAG}	1	—	t_{SPcyc}	
	SS input rise/fall time	t_{SSR}, t_{SSF}	—	3	ns	

Note: $t_{PSClCyc}$: RSCI_n_TCLK cycle ($n = 0$ to 9)

Note 1. $NFCyc = 4p \times 2q - 1 \times t_{PSClCyc}$
 p : CCR2.CKS[1:0] ($p = 0, 1, 2, 3$)
 q : CCR1.NFCS[2:0] ($q = 1, 2, 3, 4$)

Note 2. Output transition time from 20% to 80%

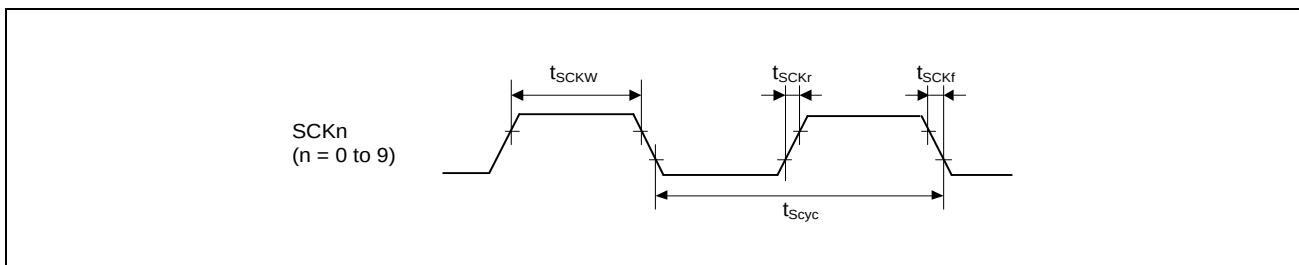


Figure 10.1-27 SCK Clock Input/Output Timing

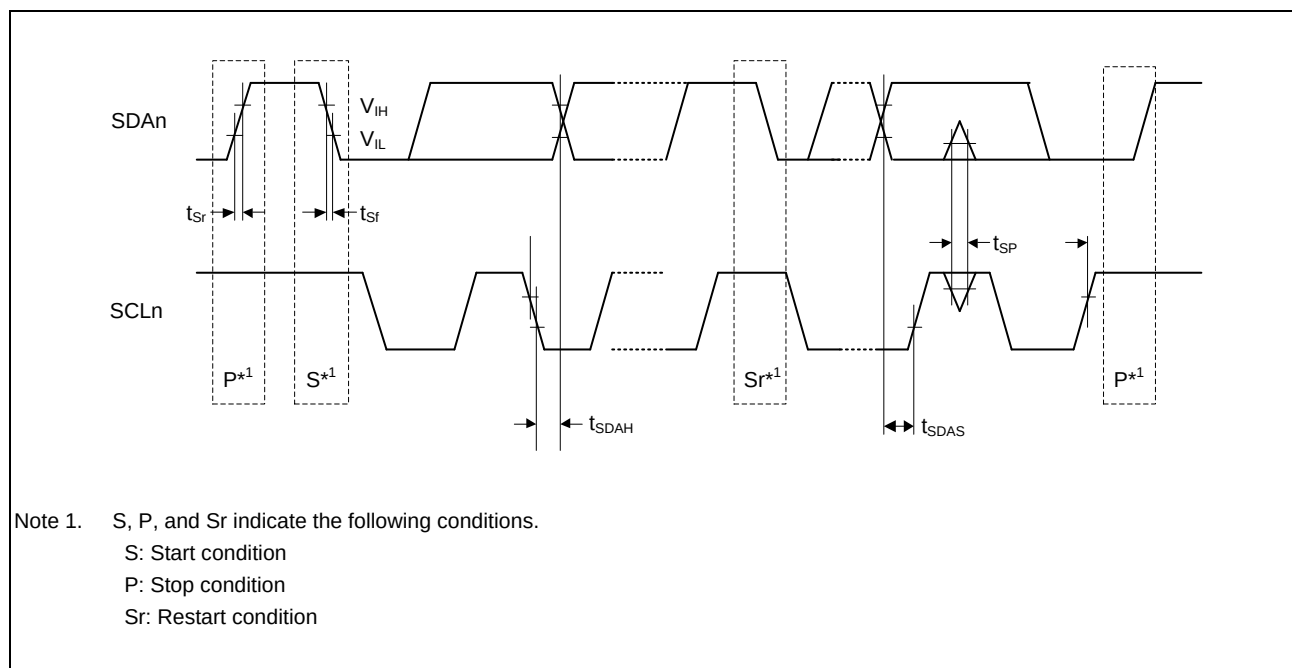


Figure 10.1-28 RSCI Simple I2C Mode Timing

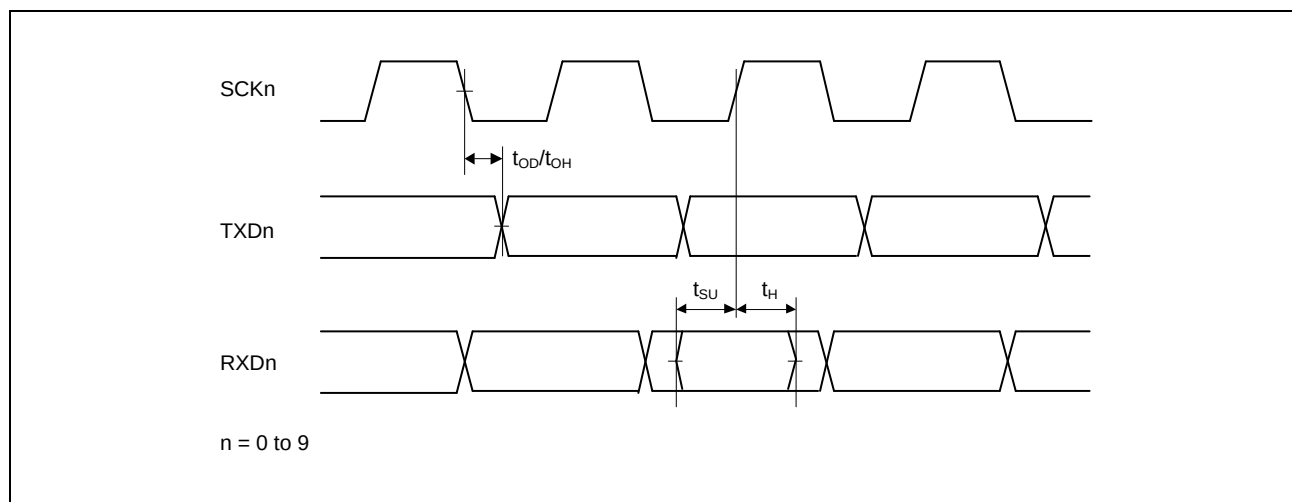


Figure 10.1-29 RSCI Input/Output Timing in Clock Synchronous Mode

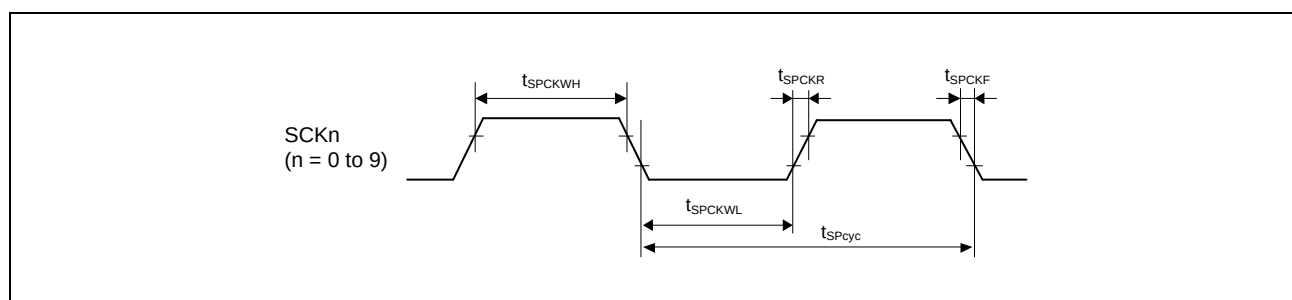


Figure 10.1-30 RSCI Simple SPI Mode Clock Timing

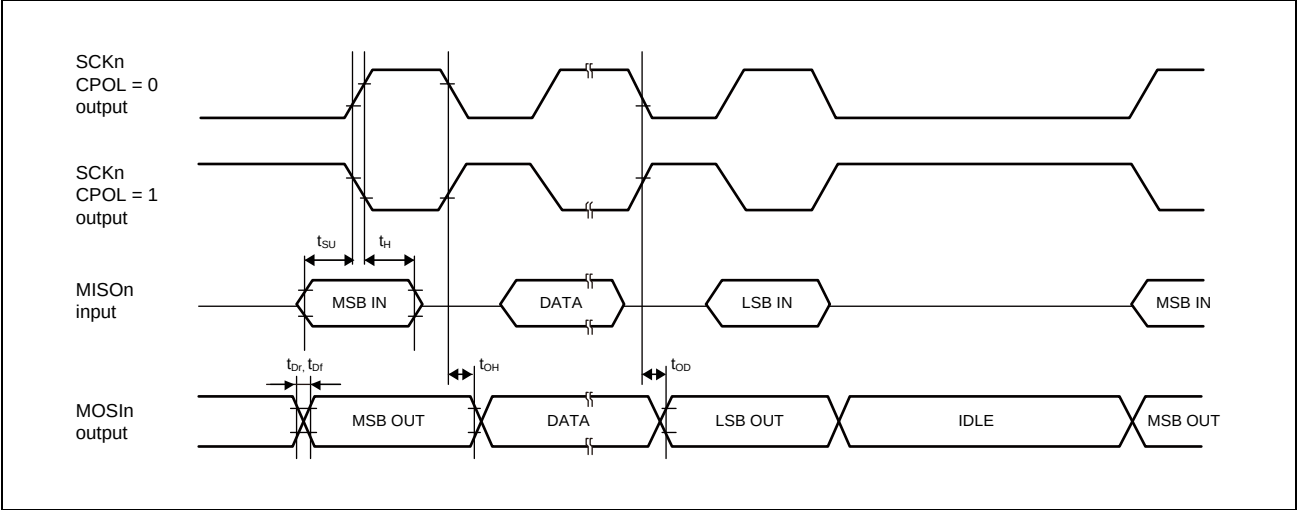


Figure 10.1-31 RSCI Simple SPI Mode Timing for Master when CPHA = 0

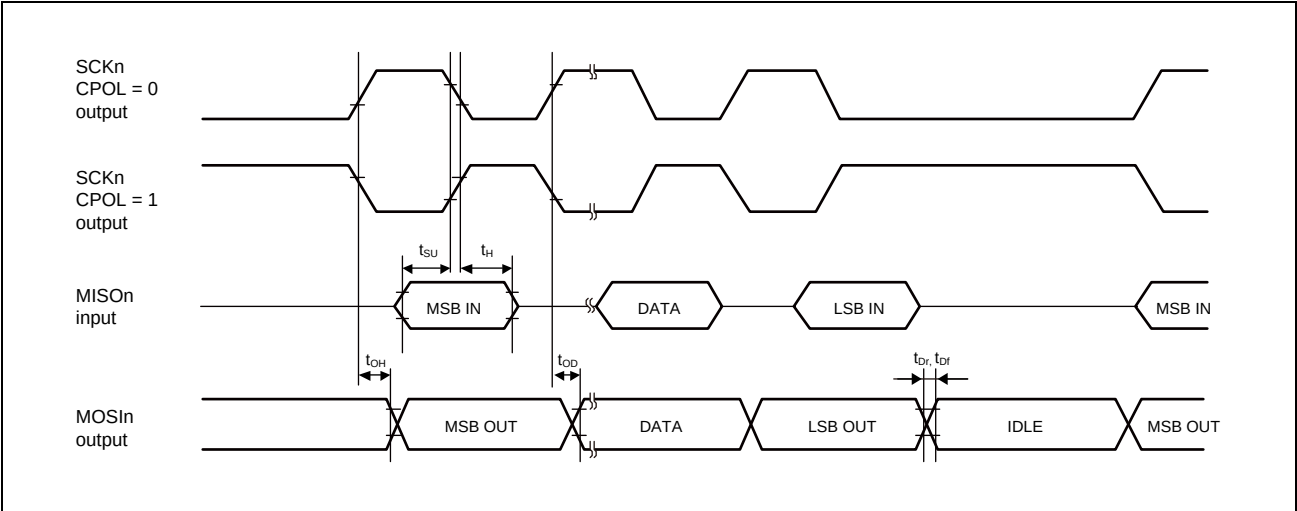


Figure 10.1-32 RSCI Simple SPI Mode Timing for Master when CPHA = 1

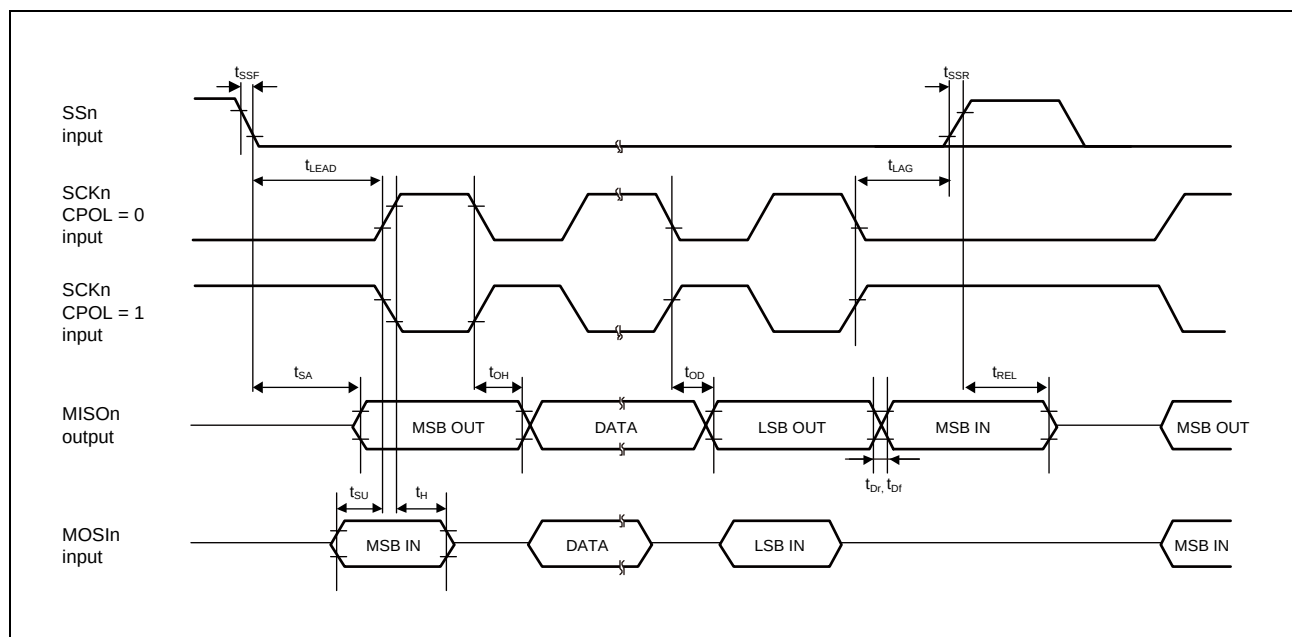


Figure 10.1-33 RSCI Simple SPI Mode Timing for Slave when CPHA = 0

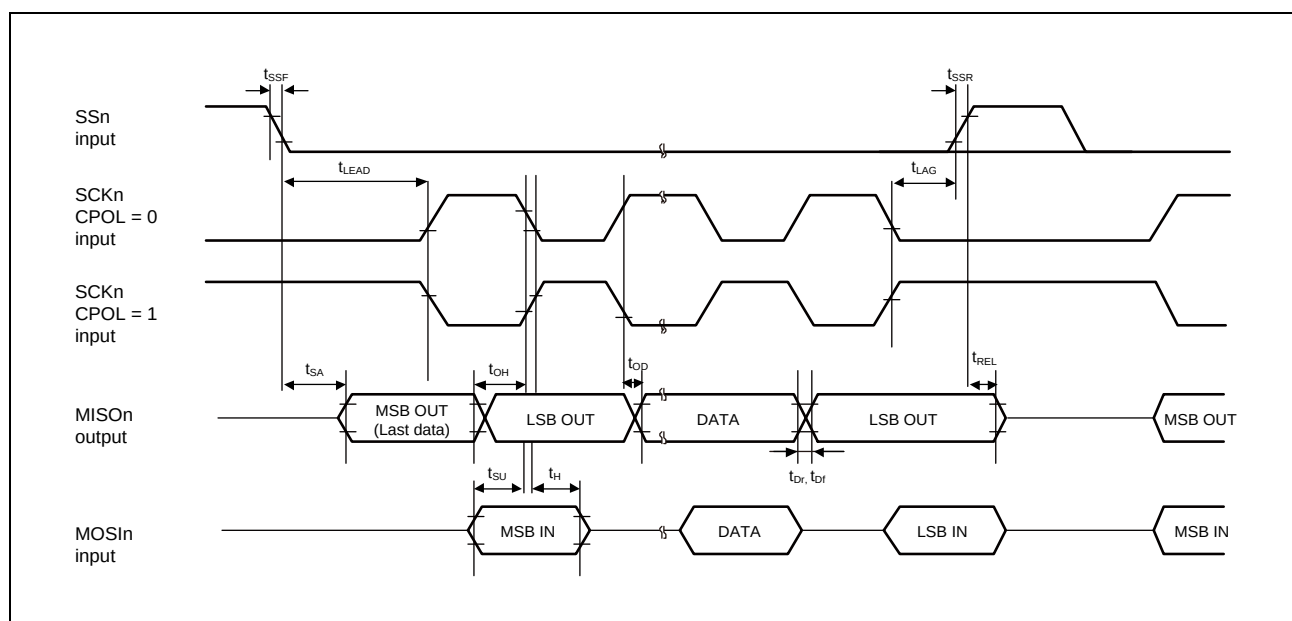


Figure 10.1-34 RSCI Simple SPI Mode Timing for Slave when CPHA = 1

10.1.5.15 Renesas Serial Peripheral Interface (RSPI) Access Timing

Conditions:

$$V_{OH} = V_{DD18} \times 0.5, V_{OL} = V_{DD18} \times 0.5, C = 30 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.5, V_{OL} = V_{DD33} \times 0.5, C = 30 \text{ pF (3.3 V)}$$

Drive strength: ×6

Table 10.1-20 RSPI Timing (1/2)

Parameter	Symbol	Min.*1	Max.*1	Unit	Figure
RSPCK clock cycle	Master t_{SPCyc}	4	4096	t_{SPICyc}	Figure 10.1-35
	Slave	4	4096	t_{SPICyc}	
RSPCK clock high-level pulse width	Master t_{SPCKWH}	$(t_{SPCyc} - t_{SPCKr} - t_{SPCKf})/2 - 2.5$	—	ns	
	Slave	1	—	t_{SPICyc}	
RSPCK clock low-level pulse width	Master t_{SPCKWL}	$(t_{SPCyc} - t_{SPCKr} - t_{SPCKf})/2 - 2.5$	—	ns	
	Slave	1	—	t_{SPICyc}	
RSPCK clock rise/fall time	Output t_{SPCKr}	—	3*5	ns	
	Input t_{SPCKf}	—	3*5	ns	
Data input setup time	Master t_{SU}	5.3	—	ns	Figure 10.1-36 to Figure 10.1-42
	Slave	3	—	ns	
Data input hold time	Master t_{H}	3	—	ns	
	Slave	3	—	ns	
SSL setup time	Master t_{LEAD}	$N \times t_{SPCyc} - 3^{*2}$	$N \times t_{SPCyc} + 3^{*2}$	ns	Figure 10.1-36 to Figure 10.1-39
	Slave	5	—	t_{SPICyc}	
SSL hold time	Master t_{LAG}	$N \times t_{SPCyc} - 3^{*3}$	$N \times t_{SPCyc} + 3^{*3}$	ns	
	Slave	5	—	t_{SPICyc}	
Continuous transmission delay	Master t_{TD}	$t_{SPCyc} + 2 \times t_{SPICyc}$	$8 \times t_{SPCyc} + 2 \times t_{SPICyc}$	ns	
	Slave	$t_{SPCyc} + 5 \times t_{SPICyc}$	—	ns	
TI-SSP SS input setup time	t_{TISS}	3.1	—	ns	Figure 10.1-40 to Figure 10.1-42
TI-SSP SS input hold time	t_{TISH}	3	—	ns	
TI-SSP next access time	t_{TIND}	M^{*4}	—	t_{SPICyc}	
TI-SSP Master SS output delay	t_{TISSOD}	-3	3	ns	
TI-SSP Master OE delay 1	$t_{TIMOED1}$	—	2	ns	
TI-SSP Master OE delay 2	$t_{TIMOED2}$	—	2	ns	
TI-SSP Slave OE delay 1	$t_{TISOED1}$	—	7.5	ns	
TI-SSP Slave OE delay 2	$t_{TISOED2}$	—	7.5	ns	
SSL Activation to Data Output Delay	t_{OD1}	—	3	ns	Figure 10.1-36
Data output delay time	Master t_{OD}	—	3	ns	Figure 10.1-36 to Figure 10.1-42
	Slave	—	7.5	ns	
Data output hold time	Master t_{OH}	-3	—	ns	
	Slave	3	—	ns	
MOSI, MISO rise/fall time	Output t_{Dr}, t_{Df}	—	3*5	ns	
	Input	—	1	μs	
SSL rise/fall time	Output t_{SSLr}, t_{SSLf}	—	3*5	ns	Figure 10.1-36, Figure 10.1-37
	Input	—	1	μs	

Table 10.1-20 RSPI Timing (2/2)

Parameter	Symbol	Min.*1	Max.*1	Unit	Figure
Slave access time	t_{SA}	—	8	ns	Figure 10.1-38,
Slave output release time	t_{REL}	—	8	ns	Figure 10.1-39

Note 1. t_{SPCyc} : RSPI_n_TCLK cycle ($n = 0$ to 2)

Note 2. N: RSPI_m_SPCKD.SCKDL[2:0] set value + 1 (1 to 8)

Note 3. N: RSPI_m_SSLND.SLNDL[2:0] set value + 1 (1 to 8)

Note 4. M: RSPI_m_SSLND.SLNDL[2:0] set value + 2 (2 to 9)

Note 5. Output transition time from 20% to 80%

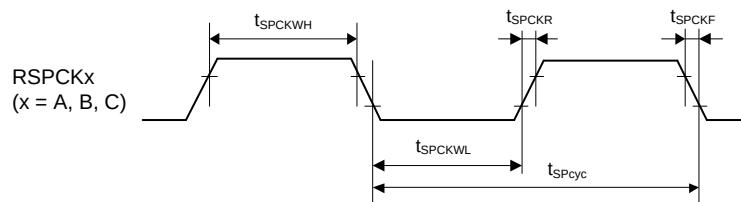


Figure 10.1-35 RSPI Clock Timing

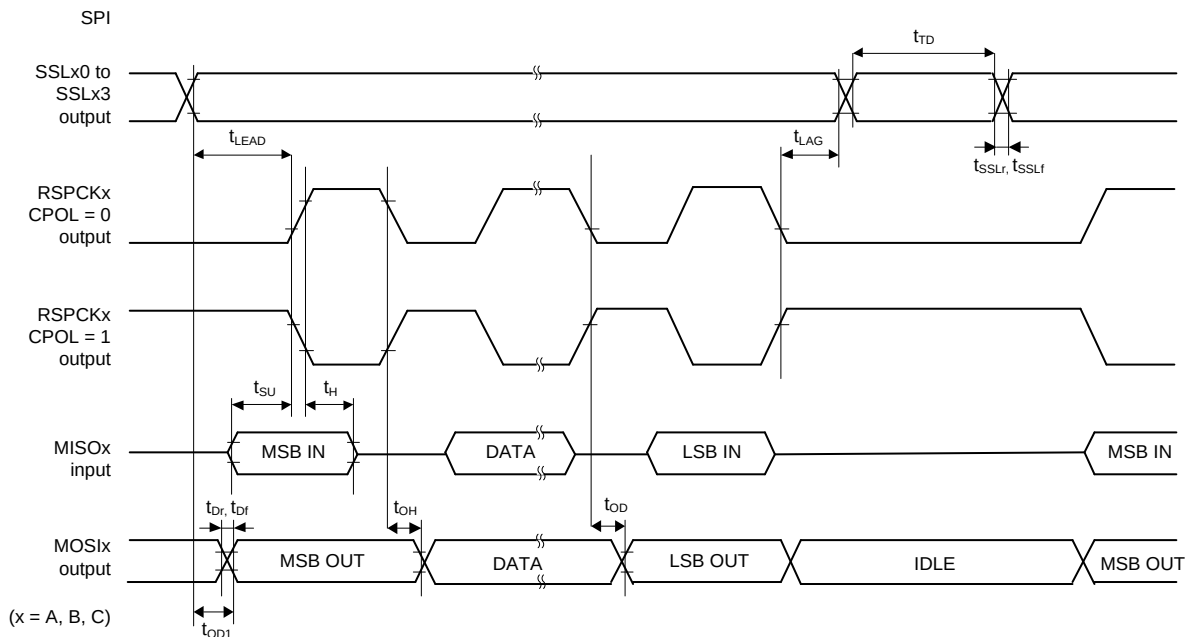


Figure 10.1-36 RSPI Timing (Master, Motorola RSPI, CPHA = 0)

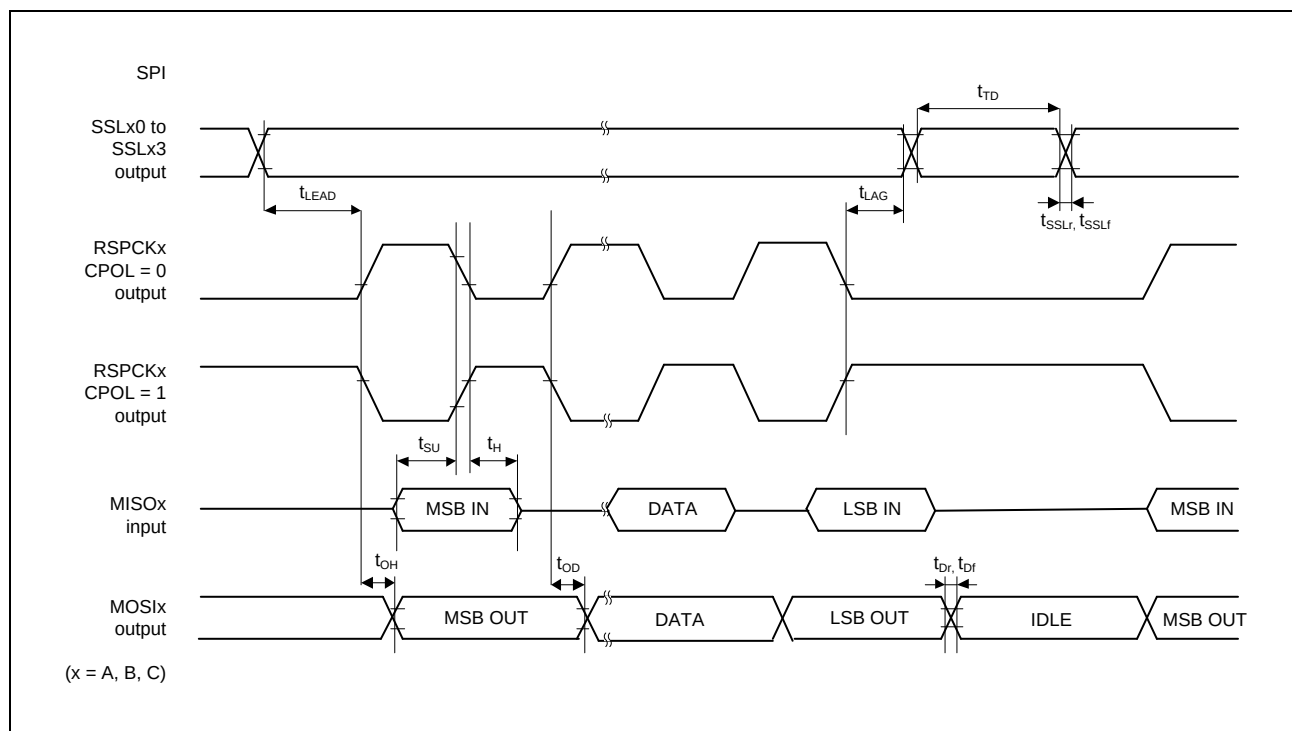


Figure 10.1-37 RSPI Timing (Master, Motorola RSPI, CPHA = 1)

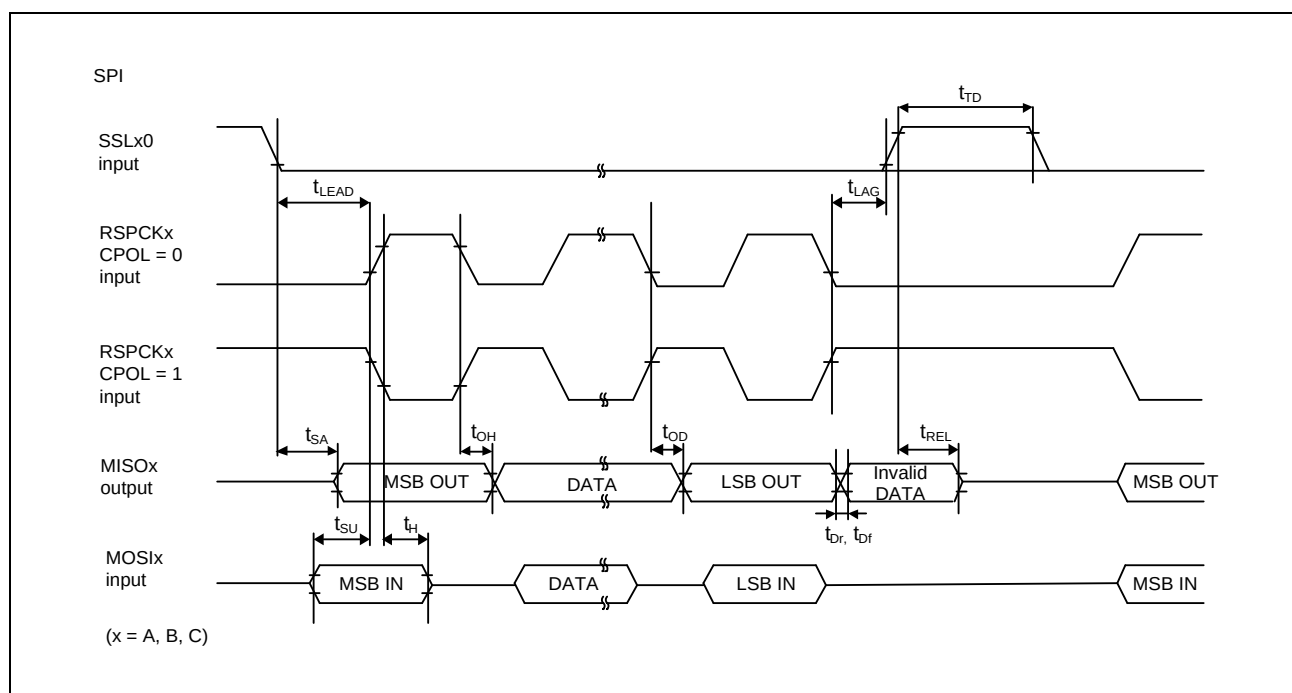


Figure 10.1-38 RSPI Timing (Slave, Motorola RSPI, CPHA = 0)

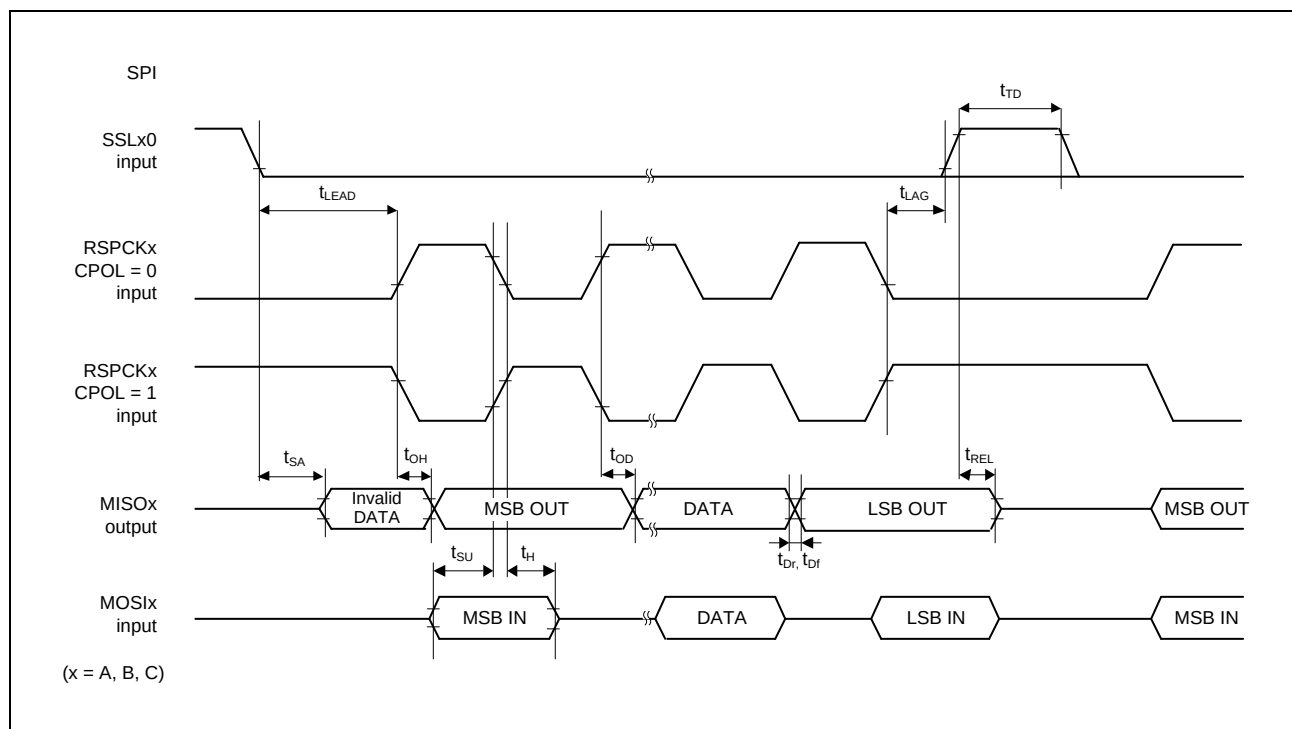


Figure 10.1-39 RSPI Timing (Slave, Motorola RSPI, CPHA = 1)

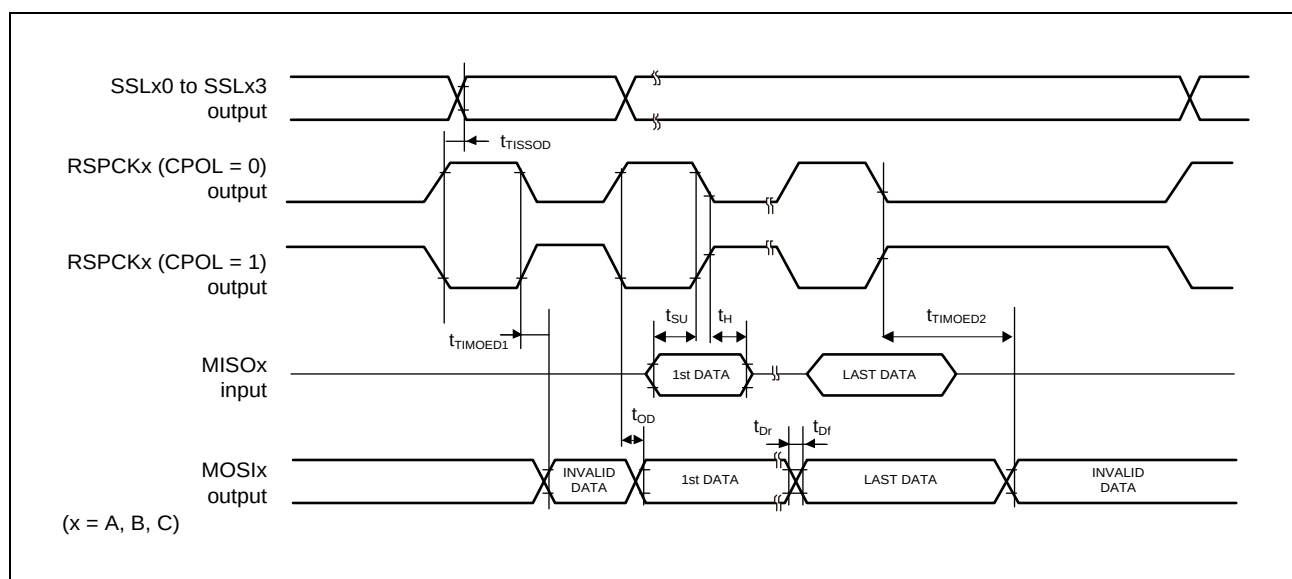


Figure 10.1-40 RSPI Timing (Master, TI SSP)

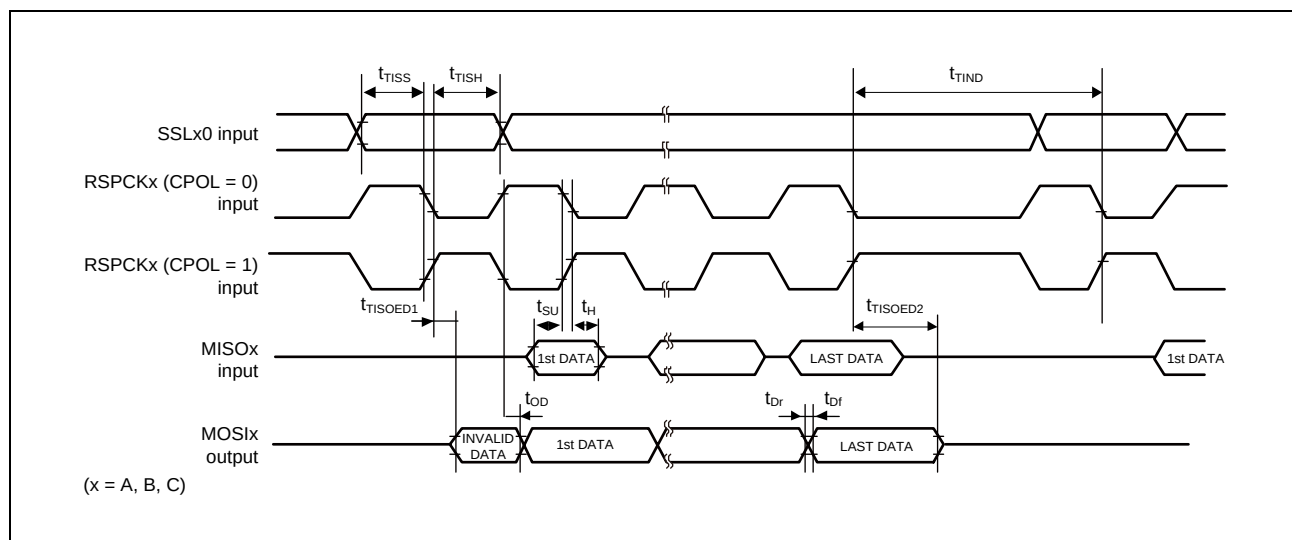


Figure 10.1-41 RSPI Timing (Slave, TI-SSP, with delay in burst transfer)

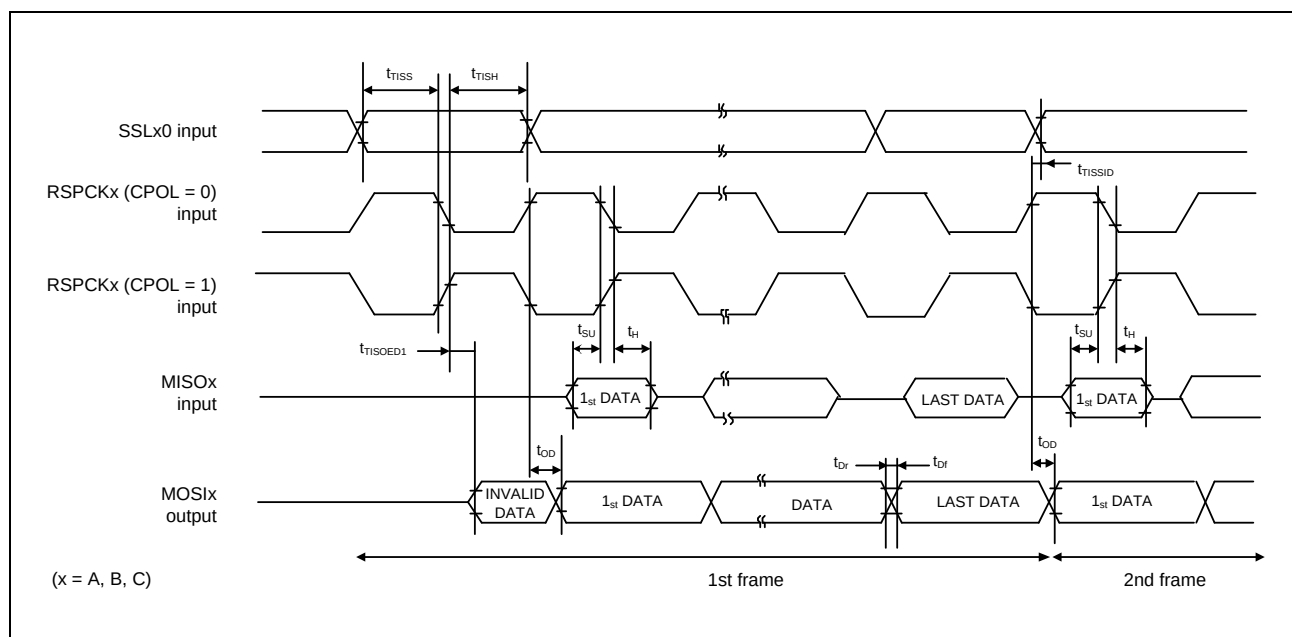


Figure 10.1-42 RSPI Timing (Slave, TI-SSP, without delay in burst transfer)

10.1.5.16 Renesas IIC Bus Interface (RIIC) Access Timing

Conditions: $V_{OL} = 0.4\text{ V}$

Drive strength: $\times 6$

Table 10.1-21 RIIC Timing

Parameter		Symbol	Min. ^{*1,*2}	Max. ^{*1,*2}	Unit	Figure
RIIC (Standard-mode)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 1300	—	ns	Figure 10.1-43
	SCL input high-level pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low-level pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t _{sr}	—	1000	ns	
	SCL, SDA input fall time	t _{sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Restart condition input setup time	t _{STAS}	1000	—	ns	
	Stop condition input setup time	t _{STOS}	1000	—	ns	
	Data input setup time	t _{SDAS}	t _{IICcyc} + 50	—	ns	
	Data input hold time	t _{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C _b	—	400	pF	
RIIC (Fast-mode)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 600	—	ns	Figure 10.1-43
	SCL input high-level pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL input low-level pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t _{sr}	— ^{*4}	300	ns	
	SCL, SDA input fall time	t _{sf}	— ^{*4}	300	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 300	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Restart condition input setup time	t _{STAS}	300	—	ns	
	Stop condition input setup time	t _{STOS}	300	—	ns	
	Data input setup time	t _{SDAS}	t _{IICcyc} + 50	—	ns	
	Data input hold time	t _{SDAH}	0	—	ns	
	SCL, SDA capacitive load ^{*3}	C _b	—	400	pF	
RIIC (Fast-mode Plus)	SCL input cycle time	t _{SCL}	6(12) × t _{IICcyc} + 240	—	ns	Figure 10.1-43
	SCL input high-level pulse width	t _{SCLH}	3(6) × t _{IICcyc} + 120	—	ns	
	SCL input low-level pulse width	t _{SCLL}	3(6) × t _{IICcyc} + 120	—	ns	
	SCL, SDA input rise time	t _{sr}	— ^{*4}	120	ns	
	SCL, SDA input fall time	t _{sf}	— ^{*4}	120	ns	
	SCL, SDA input spike pulse removal time	t _{SP}	0	1(4) × t _{IICcyc}	ns	
	SDA input bus free time	t _{BUF}	3(6) × t _{IICcyc} + 120	—	ns	
	Start condition input hold time	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Restart condition input setup time	t _{STAS}	300	—	ns	
	Stop condition input setup time	t _{STOS}	300	—	ns	
	Data input setup time	t _{SDAS}	t _{IICcyc} + 50	—	ns	
	Data input hold time	t _{SDAH}	0	—	ns	
	SCL, SDA capacitive load ^{*3}	C _b	—	550	pF	

Note 1. t_{IICcyc} : RIIC internal reference clock (IIC ϕ) cycle

- Note 2. The values outside parentheses apply when the value of the ICMR3.NF[1:0] bits is 00b while the digital filter is enabled by setting ICFER.NFE = 1. The values within parentheses apply when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by setting ICFER.NFE = 1.
- Note 3. C_b is the total capacitance of the bus lines.
- Note 4. The minimum values are not specified for t_{sf} and t_{sf} in Fast-mode or Fast-mode Plus.

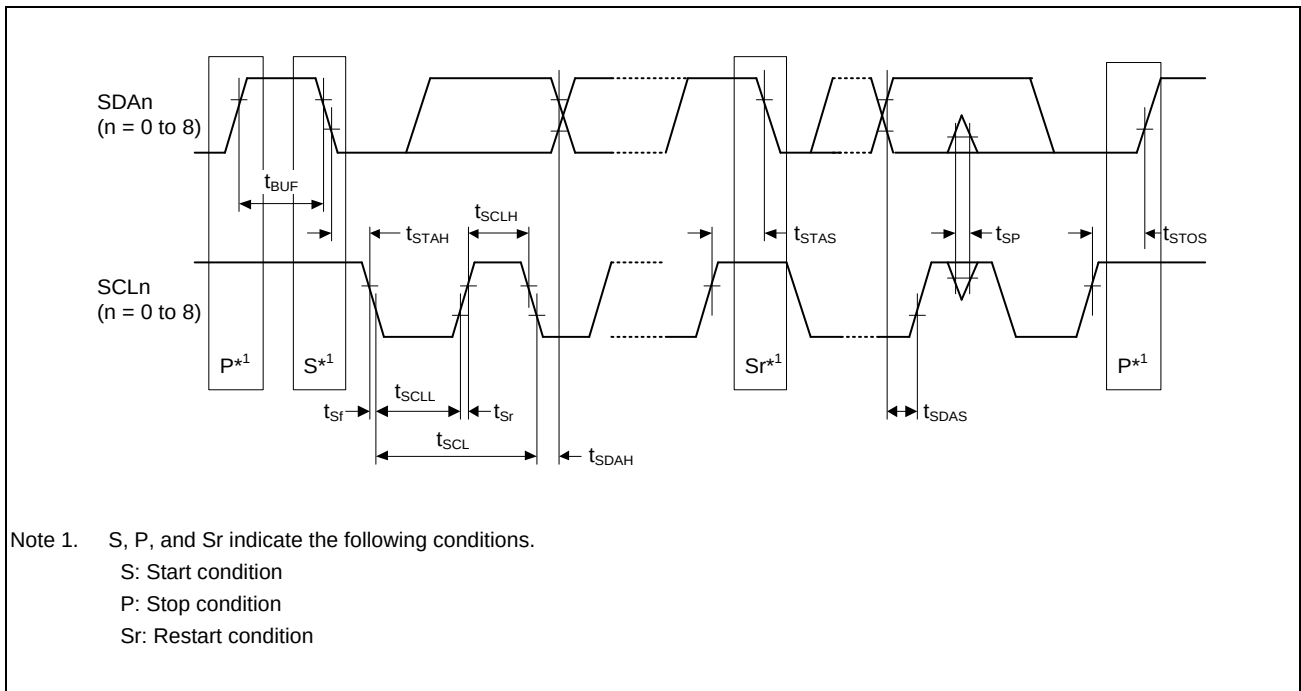


Figure 10.1-43 RIIC Bus Interface Input/Output Timing

10.1.5.17 I3C Timing

Conditions: $V_{OH} = V_{DD1218\_I3C} \times 0.5$, $V_{OL} = V_{DD1218\_I3C} \times 0.5$, $C = 30$ pF (1.2 V or 1.8 V)

Drive strength: $\times 6$

Table 10.1-22 I3C Timing

Parameter	Symbol	Min.* ¹	Max.	Unit	Figure	
IIC (Standard mode, SMBus)	SCL3n cycle time	t _{SCL}	4(36) × t _{IICcyc} + 4 × t _{Pcyc} + 1300	—	ns	Figure 10.1-44
	SCL3n high-level pulse width	t _{SCLH}	2(18) × t _{IICcyc} + 2 × t _{Pcyc} + 300	—	ns	
	SCL3n low-level pulse width	t _{SCLL}	2(18) × t _{IICcyc} + 2 × t _{Pcyc} + 800	—	ns	
	SCL3n, SDA3n rise time	t _{Sr}	—	1000	ns	
	SCL3n, SDA3n fall time	t _{Sf}	—	300	ns	
	SCL3n, SDA3n spike pulse removal time	t _{SP}	0	1(16) × t _{IICcyc}	ns	
	SDA3n bus free time	t _{BUF}	3(20) × t _{IICcyc} + 300	—	ns	
	Hold time for START condition	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Setup time for repeated START condition	t _{STAS}	1000	—	ns	
	Setup time for STOP condition	t _{STOS}	1000	—	ns	
	Data setup time	t _{SDAS}	t _{IICcyc} + 50	—	ns	
	Data hold time	t _{SDAH}	0	—	ns	
	SCL3n, SDA3n capacitive load	C _b	—	400	pF	
IIC (Fast mode)	SCL3n cycle time	t _{SCL}	4(36) × t _{IICcyc} + 4 × t _{Pcyc} + 600	—	ns	Figure 10.1-44
	SCL3n high-level pulse width	t _{SCLH}	2(18) × t _{IICcyc} + 2 × t _{Pcyc} + 300	—	ns	
	SCL3n low-level pulse width	t _{SCLL}	2(18) × t _{IICcyc} + 2 × t _{Pcyc} + 300	—	ns	
	SCL3n, SDA3n rise time	t _{Sr}	—	300	ns	
	SCL3n, SDA3n fall time	t _{Sf}	—	300	ns	
	SCL3n, SDA3n spike pulse removal time	t _{SP}	0	1(16) × t _{IICcyc}	ns	
	SDA3n bus free time	t _{BUF}	3(20) × t _{IICcyc} + 300	—	ns	
	Hold time for START condition	t _{STAH}	t _{IICcyc} + 300	—	ns	
	Setup time for repeated START condition	t _{STAS}	300	—	ns	
	Setup time for STOP condition	t _{STOS}	300	—	ns	
	Data setup time	t _{SDAS}	t _{IICcyc} + 50	—	ns	
	Data hold time	t _{SDAH}	0	—	ns	
	SCL3n, SDA3n capacitive load	C _b	—	400	pF	

Note 1. t_{IICcyc} : I3C internal reference clock (I3C ϕ) cycle, t_{Pcyc} : I3C_0_TCLK cycle
Refer to **7.8.3.2.14 Reference Clock Control Register** for details.
Values in parentheses apply when INCTL.DNFS[3:0] is set to 1111b while the digital filter is enabled with DNFE.DNFE set to 1b.

Table 10.1-23 IIC Timing (Fast-mode+)

Parameter	Symbol	Min.*1	Max.	Unit	Figure
IIC (Fast-mode+)	SCL3n cycle time	t_{SCL}	$4(26) \times t_{IICcyc} + 4 \times t_{Pcyc} + 240$	—	ns Figure 10.1-44
	SCL3n high-level pulse width	t_{SCLH}	$2(18) \times t_{IICcyc} + 2 \times t_{Pcyc} + 120$	—	
	SCL3n low-level pulse width	t_{SCLL}	$2(18) \times t_{IICcyc} + 2 \times t_{Pcyc} + 120$	—	
	SCL3n, SDA3n rise time	t_{Sr}	120	ns	
	SCL3n, SDA3n fall time	t_{Sf}	120	ns	
	SCL3n, SDA3n spike pulse removal time	t_{SP}	$1(16) \times t_{IICcyc}$	ns	
	SDA3n bus free time	t_{BUF}	$3(20) \times t_{IICcyc} + 120$	—	
	Hold time for START condition	t_{STAH}	$t_{IICcyc} + 135$	—	
	Setup time for repeated START condition	t_{STAS}	260	—	
	Setup time for STOP condition	t_{STOS}	260	—	
	Data setup time	t_{SDAS}	50	—	
	Data hold time	t_{SDAH}	0	—	
	SCL3n, SDA3n capacitive load	C_b	550	pF	

Note 1. t_{IICcyc} : I3C internal reference clock (I3C ϕ) cycle, t_{Pcyc} : I3C_0_TCLK cycle.

Refer to **7.8.3.2.14 Reference Clock Control Register** for details.

Values in parentheses apply when INCTL.DNFS[3:0] is set to 1111b while the digital filter is enabled with INCTL.DNFE set to 1b.

Table 10.1-24 IIC Timing (HS mode)

Parameter		Symbol	Cb = 100 pF		Cb = 400 pF		Unit	Figure
			Min.*1	Max.	Min.*1	Max.		
IIC (HS mode)	SCL3n cycle time	t _{SCL}	$3(36) \times t_{IICyc} + 4 \times t_{PCyc} + 240$	—	$3(36) \times t_{IICyc} + 4 \times t_{PCyc} + 240$	—	ns	Figure 10.1-44
	SCL3n high-level pulse width	t _{SCLH}	$2(18) \times t_{IICyc} + 2 \times t_{PCyc} + 120$	—	$2(18) \times t_{IICyc} + 2 \times t_{PCyc} + 120$	—	ns	
	SCL3n low-level pulse width	t _{SCLL}	$2(18) \times t_{IICyc} + 2 \times t_{PCyc} + 120$	—	$2(18) \times t_{IICyc} + 2 \times t_{PCyc} + 120$	—	ns	
	SCL3n rise time	t _{Sr}	—	40	—	80	ns	
	SCL3n rise time after a repeated START condition and after an acknowledge bit	t _{Sr}	—	80	—	160	ns	
	SDA3n rise time	t _{Sr}	—	80	—	160	ns	
	SCL3n fall time	t _{Sf}	—	40	—	80	ns	
	SDA3n fall time	t _{Sf}	—	80	—	160	ns	
	SCL3n, SDA3n spike pulse removal time	t _{SP}	0	$1(16) \times t_{IICyc}$	0	$1(16) \times t_{IICyc}$	ns	
	Hold time for START condition	t _{STAH}	$t_{IICyc} + 135$	—	$t_{IICyc} + 135$	—	ns	
	Setup time for repeated START condition	t _{STAS}	160	—	160	—	ns	
	Setup time for STOP condition	t _{STOS}	160	—	160	—	ns	
	Data setup time	t _{SDAS}	10	—	10	—	ns	
	Data hold time	t _{SDAH}	0	80	0	150	ns	
	SCL3n, SDA3n capacitive load	C _b	—	100	—	400	pF	

Note 1. t_{IICyc}: I3C internal reference clock (I3C ϕ) cycle, t_{PCyc}: I3C_0_TCLK cycle.

Refer to **7.8.3.2.14 Reference Clock Control Register** for details.

Values in parentheses apply when INCTL.DNFS[3:0] is set to 1111b while the digital filter is enabled with INCTL.DNFE set to 1b.

Note 2. The maximum SCL clock frequency is 1.7 MHz.

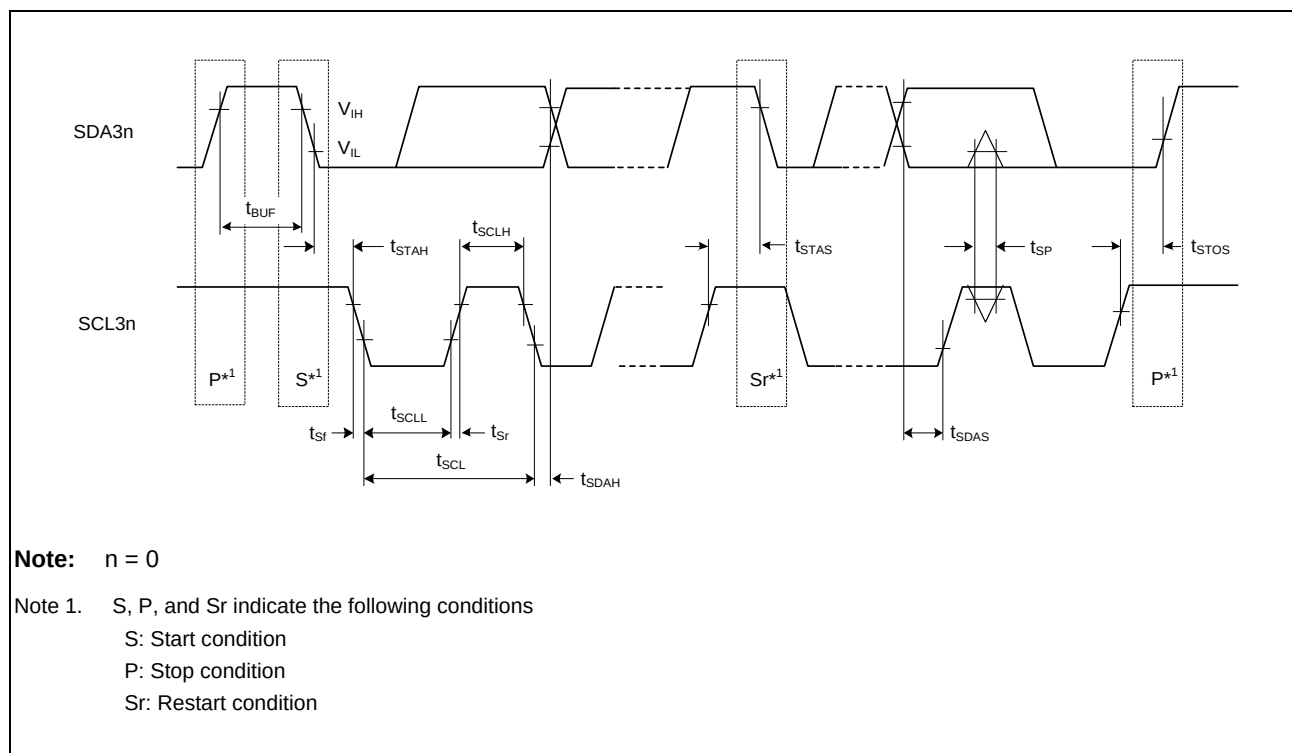


Figure 10.1-44 I3C Bus Interface Input/Output Timing

Table 10.1-25 I3C Timing (Open Drain Timing Parameters)

Parameter	Symbol	Min.*1	Max.	Unit	Figure	Notes
SCL3n clock Low period	t_{LOW_OD}	200	—	ns	Figure 10.1-47	1, 2
	$t_{DIG\_OD\_L}$	$t_{LOW\_ODmin} +$ t_{DA_ODmin}	—		Figure 10.1-47	—
SCL3n clock High period	t_{HIGH}	—	41	ns	Figure 10.1-47	3, 4
	t_{DIG_H}	36 (when 1.8 V) 40 (when 1.2 V)	$t_{HIGH} + t_{CF}$	ns	Figure 10.1-47	—
SDA3n signal fall time	t_{DA_OD}	t_{CF}	33	ns	Figure 10.1-47	—
SDA3n data setup time open drain mode	$V_{DD1218} = 1.8\text{ V}$	t_{SU_OD}	12	ns	Figure 10.1-46, Figure 10.1-47	1
	$V_{DD1218} = 1.2\text{ V}$		13.9	ns	Figure 10.1-46, Figure 10.1-47	
Clock after START (S) condition	t_{CAS}	38.4	For ENTAS0: 1 μ For ENTAS1: 100 μ For ENTAS2: 2 m For ENTAS3: 50 m	seconds	Figure 10.1-47	5, 6
Clock before STOP (P) condition	t_{CBP}	$t_{CASmin}/2$	—	seconds	Figure 10.1-48	—
Current master to secondary master overlap time during handoff	$t_{MMOverlap}$	$t_{DIG\_OD\_Lmin}$	—	ns	Figure 10.1-53	—
Bus available condition	t_{AVAL}	1	—	us	—	7
Bus idle condition	t_{DLE}	1	—	ms	—	—
Time interval where new master not driving SDA3n low	t_{MMLock}	$t_{AVALmin}$	—	us	Figure 10.1-53	—

Note 1. This is approximately equal to $t_{LOWmin} + t_{DS\_ODmin} + t_{DA\_ODtyp} + t_{SU\_ODmin}$.

Note 2. The Master may use a shorter Low period if it knows that this is safe, i.e., that SDA is already above VIH.

Note 3. This is based on t_{SPIKE} , rise and fall times, and interconnect.

Note 4. This maximum High period may be exceeded when the signals can be safely seen by Legacy I²C Devices, and/or in consideration of the interconnect (e.g., a short bus).

Note 5. On a Legacy Bus where I2C Devices need to see Start.

Note 6. Slaves that do not support the optional ENTASx CCCs shall use the t_{CAS} Max value shown for ENTAS3

Note 7. On a Mixed Bus with Fm Legacy I2C Devices, t_{AVAL} is 300 ns shorter than the Fm Bus Free Condition time (t_{BUF})

Table 10.1-26 I3C Timing (Push-Pull Timing Parameters for SDR)

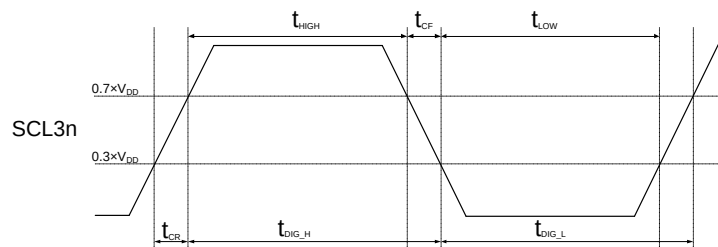
Parameter	Symbol	Min.*1	Max.	Unit	Figure	Notes
SCL3n clock frequency	$V_{DD1218} = 1.8\text{ V}$ $V_{DD1218} = 1.2\text{ V}$	t_{SCL}	0.01	12.5	MHz	1
			0.01	12.39	MHz	—
SCL3n clock Low period		t_{LOW}	24	—	ns	Figure 10.1-45
		t_{DIG_L}	32	—	ns	Figure 10.1-45 2, 4
SCL3n clock High period for Mixed Bus		t_{HIGH}	24	—	ns	Figure 10.1-45
		t_{DIG_H}	32	45	ns	Figure 10.1-45 2, 3
SCL3n clock High period		t_{HIGH}	24	—	ns	Figure 10.1-45
		t_{DIG_H}	32	45	ns	Figure 10.1-45 2
Clock in to data out for a slave	$V_{DD1218} = 1.8\text{ V}$	t_{SCO}	—	12	ns	Figure 10.1-50
	$V_{DD1218} = 1.2\text{ V}$		—	12.7	ns	
SCL3n clock rise time		t_{CR}	—	$150 \times 1/f_{SCL}$ (capped at 60)	ns	Figure 10.1-45
SCL3n clock fall time		t_{CF}	—	$150 \times 1/f_{SCL}$ (capped at 60)	ns	Figure 10.1-45
SDA3n signal data hold in push-pull mode	Master	t_{HD_PP}	$t_{CR} + 3$ and $t_{CF} + 3$	—	—	Figure 10.1-49 4
	Slave		0	—	—	Figure 10.1-51
SDA3n signal data setup in push-pull mode	$V_{DD1218} = 1.8\text{ V}$	t_{SU_PP}	12	N/A	ns	Figure 10.1-49, Figure 10.1-50, Figure 10.1-51
	$V_{DD1218} = 1.2\text{ V}$		13.9	N/A	ns	
Clock after repeated START (Sr)		t_{CASr}	t_{CASmin}	N/A	ns	Figure 10.1-52
Clock before repeated START (Sr)		t_{CBSr}	$t_{CASmin}/2$	N/A	ns	Figure 10.1-52
Capacitive load per bus line (SDA3n / SCL3n)		C_b	—	50	pF	—

Note 1. $F_{SCL} = 1/(t_{DIG\_L} + t_{DIG\_H})$

Note 2. t_{DIG_L} and t_{DIG_H} are the clock Low and High periods as seen at the receiver end of the I3C Bus using VIL and VIH (see Figure 10.1-45)

Note 3. When communicating with an I3C Device on a mixed bus, the $t_{DIG\_H\_MIXED}$ period must be constrained in order to make sure that I²C Devices do not interpret I3C signaling as valid I²C signaling.

Note 4. As both edges are used, the hold time must be satisfied for the respective edges, for example, $t_{CF} + 3$ for falling edge clocks, and $t_{CR} + 3$ for rising edge clocks.



Note: V_{DD} : VDD1218_I3C

Figure 10.1-45 t_{DIG_H} and t_{DIG_L}

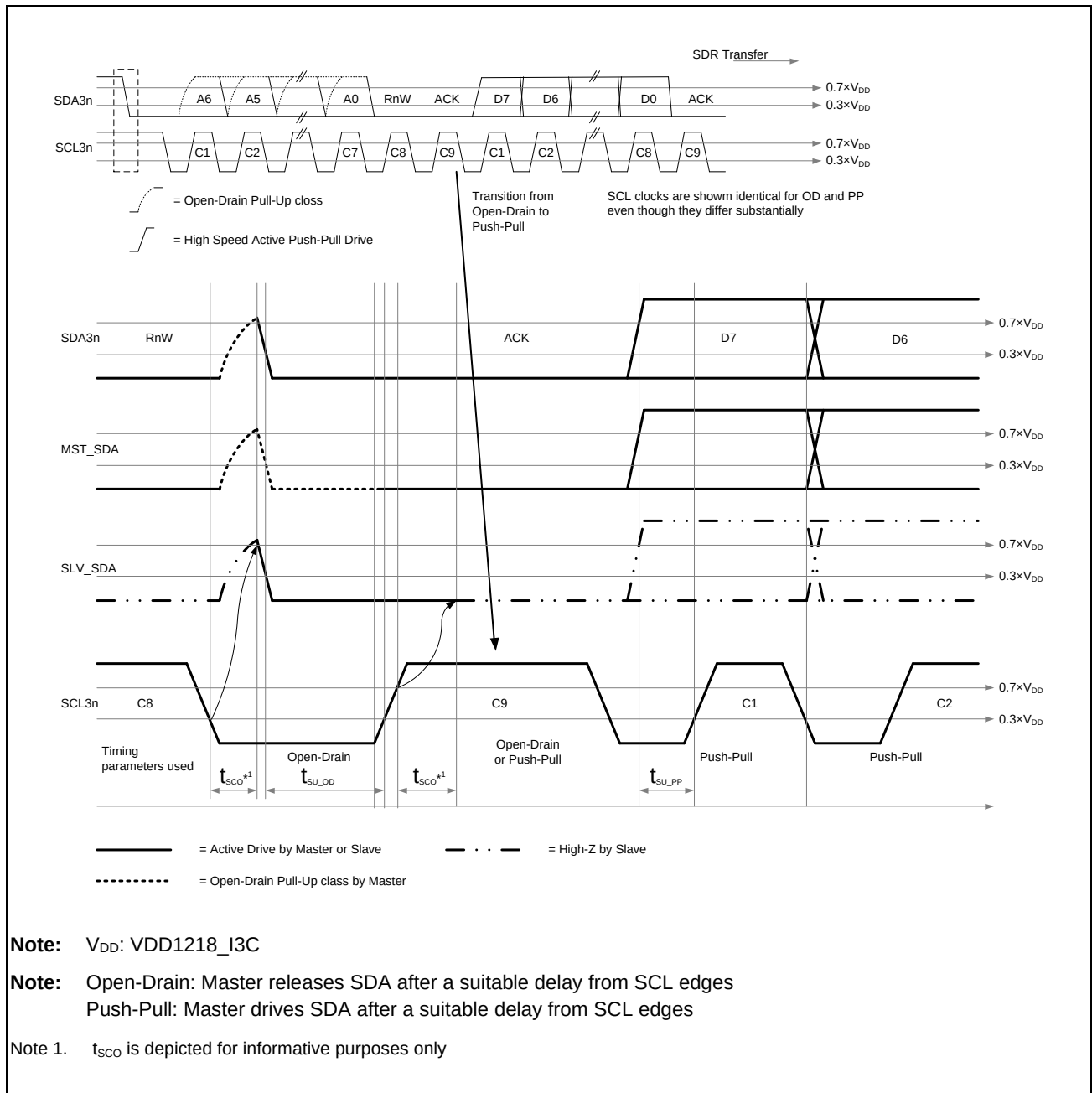


Figure 10.1-46 I3C Data Transfer – ACK by Slave

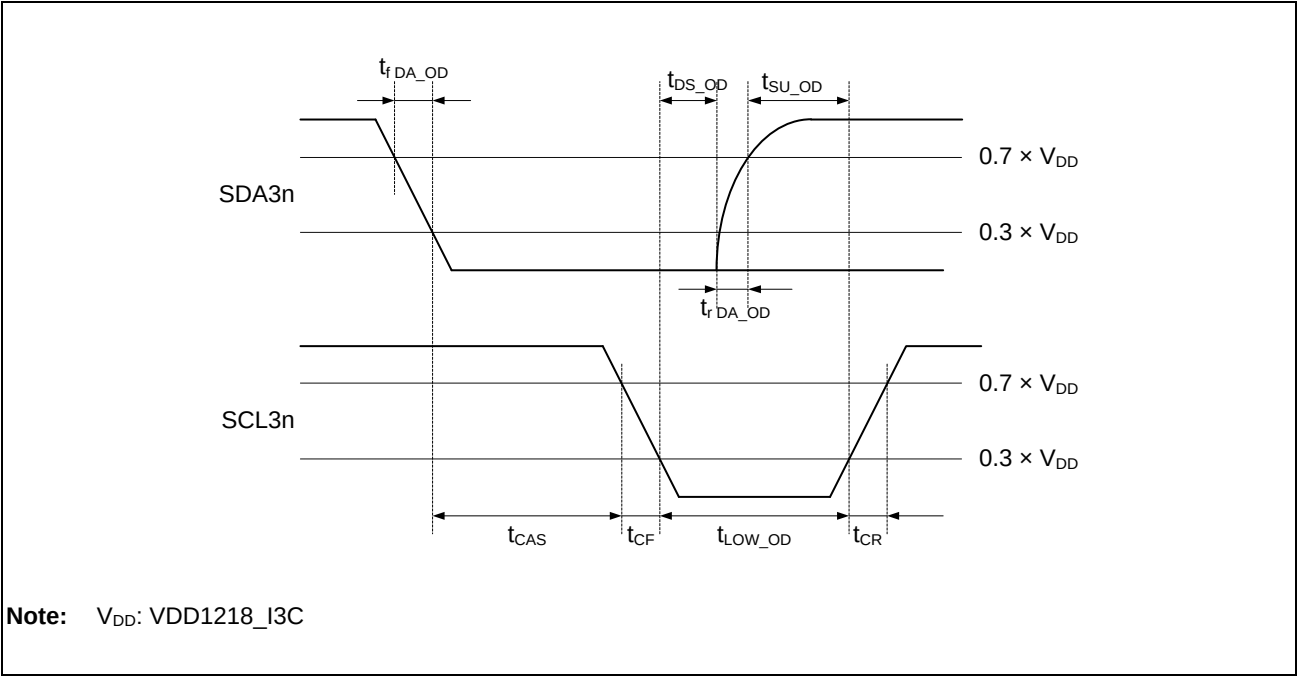


Figure 10.1-47 I3C START Condition Timing

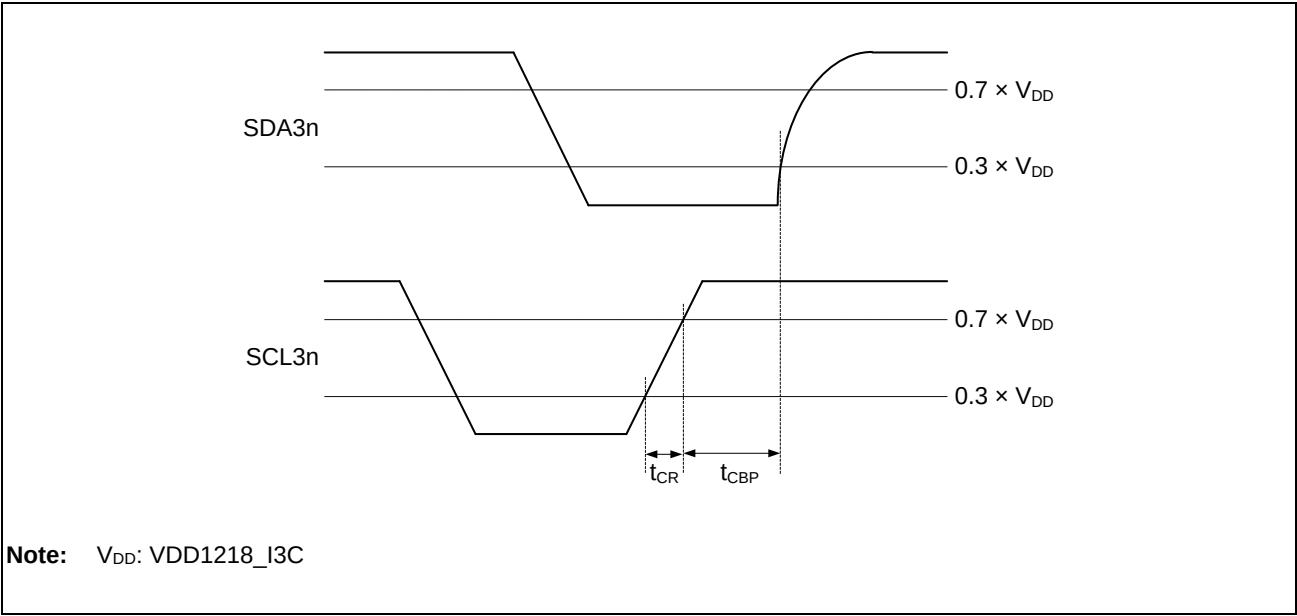


Figure 10.1-48 I3C STOP Condition Timing

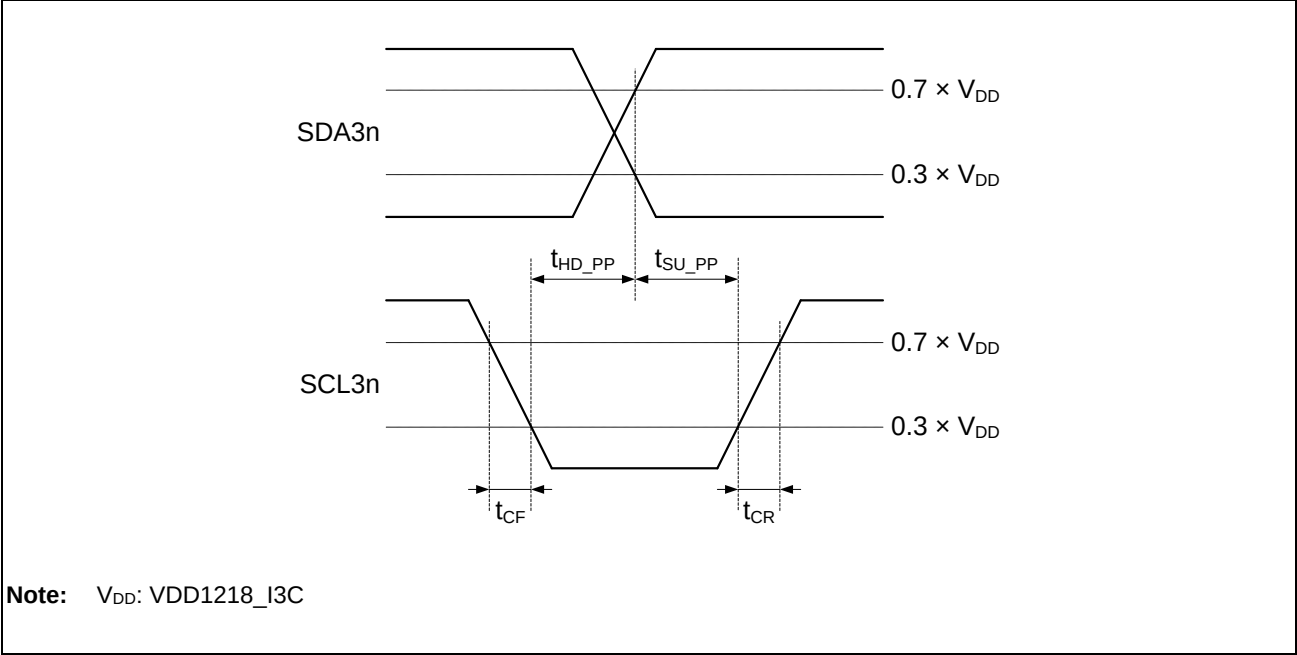


Figure 10.1-49 I3C Master Out Timing

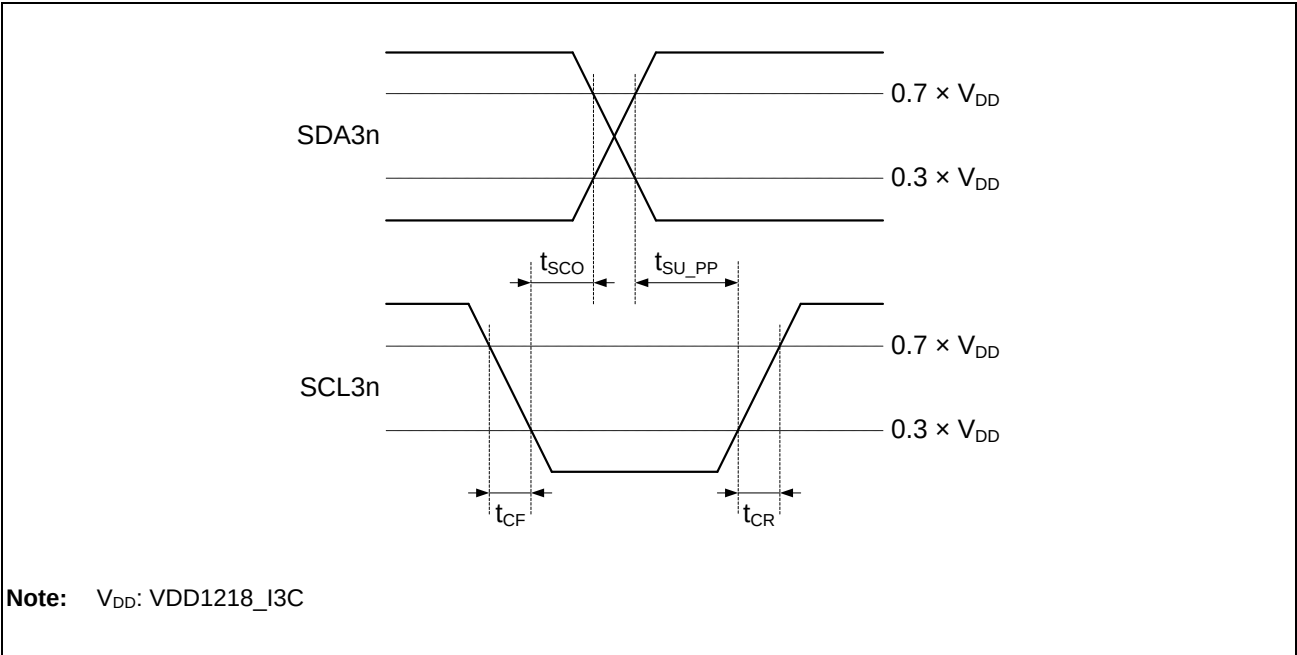


Figure 10.1-50 I3C Slave Out Timing

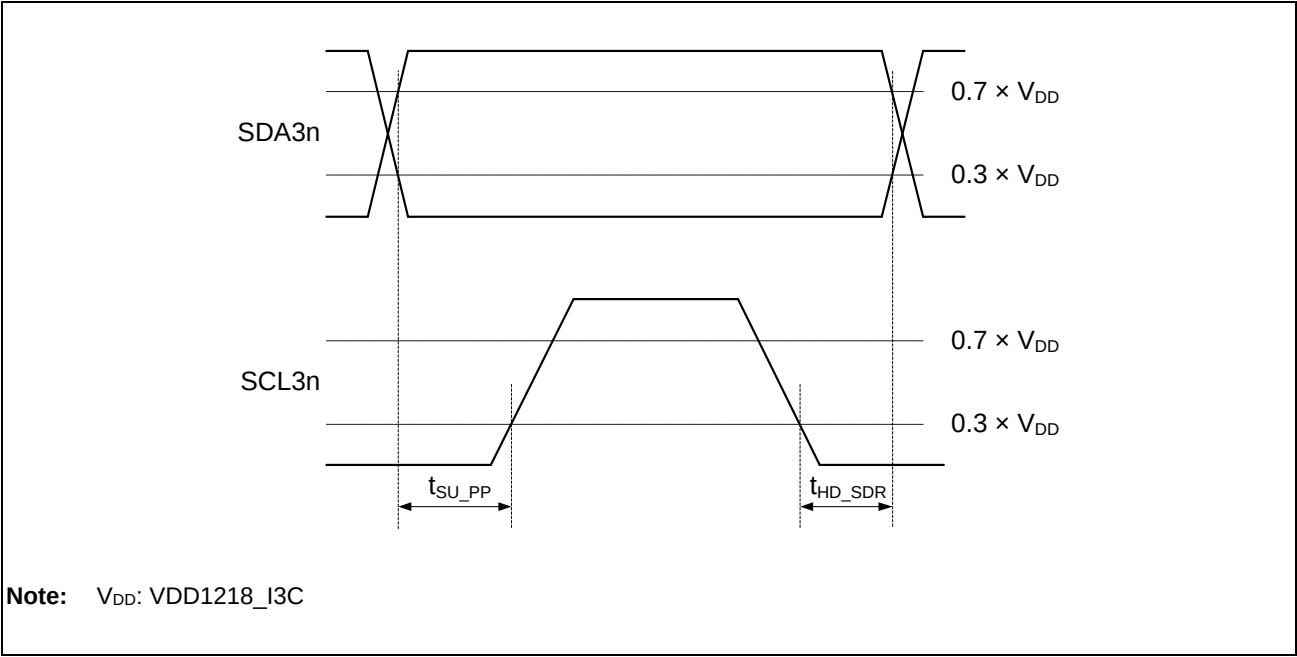


Figure 10.1-51 Master SDR Timing

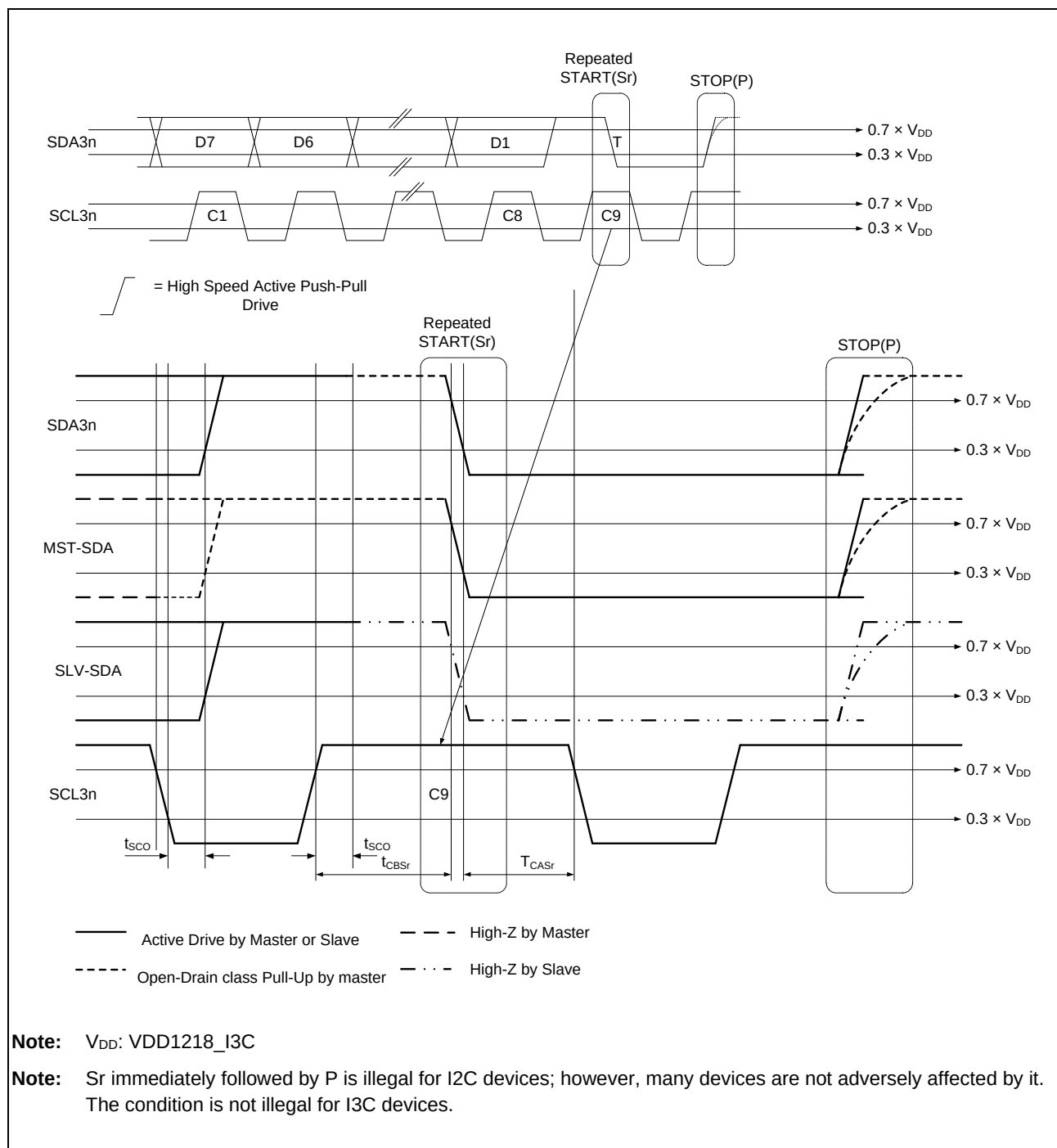


Figure 10.1-52 T-Bit When Master Ends Read with Repeated START and STOP

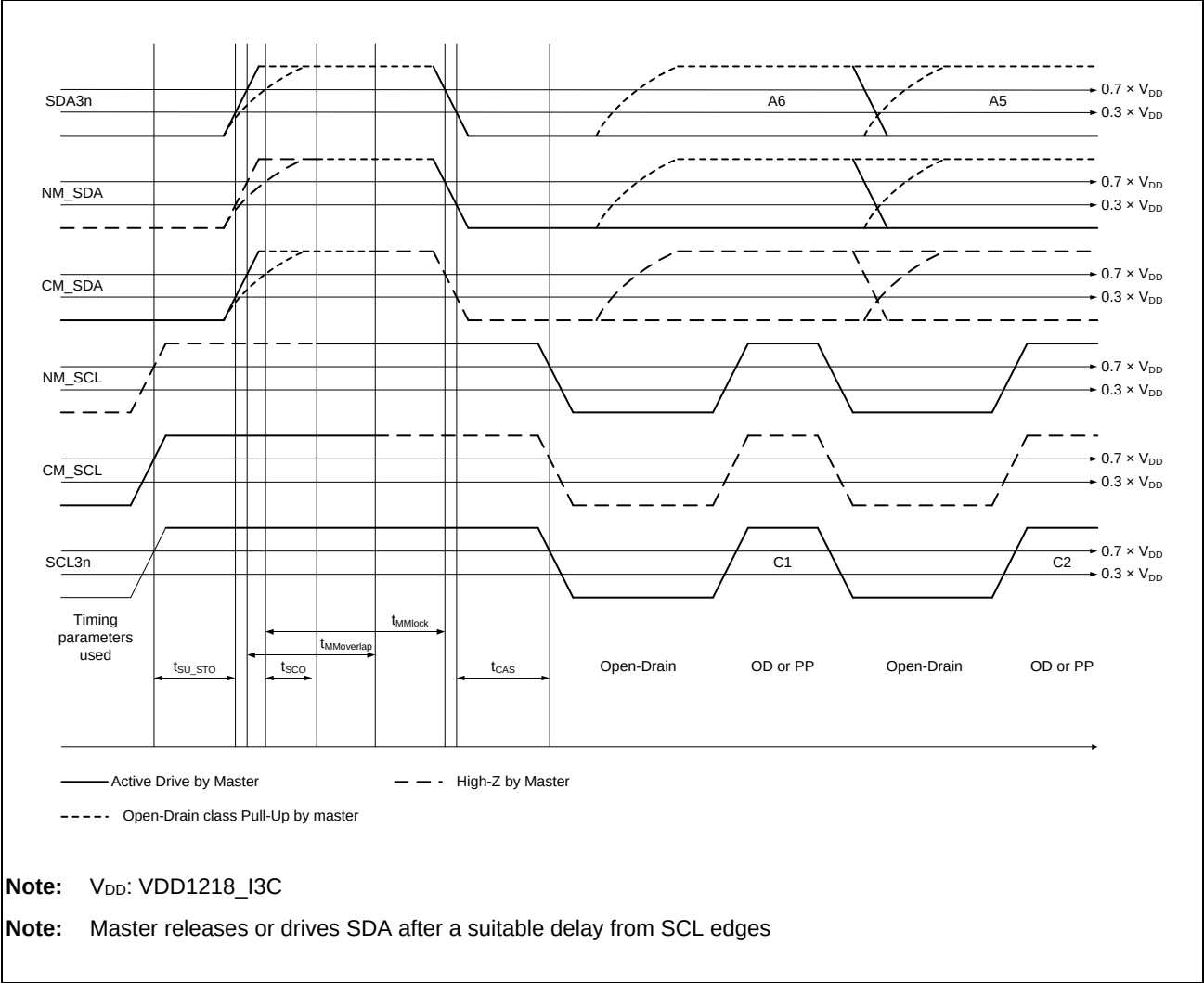


Figure 10.1-53 I3C Timing

10.1.5.18 CANFD Interface Access Timing

Table 10.1-27 CANFD Interface Timing

Parameter		Symbol	CAN		CANFD		Unit	Figures
			Min.	Max.	Min.	Max.		
CANFD	Internal delay time	t_{node}^{*1}	—	100	—	50	ns	Figure 10.1-54
	Transmission rate	—	—	1	—	8	Mbps	

Note 1. Internal delay time (t_{node}) = Internal transmission delay time (t_{output}) + Internal reception delay time (t_{input})

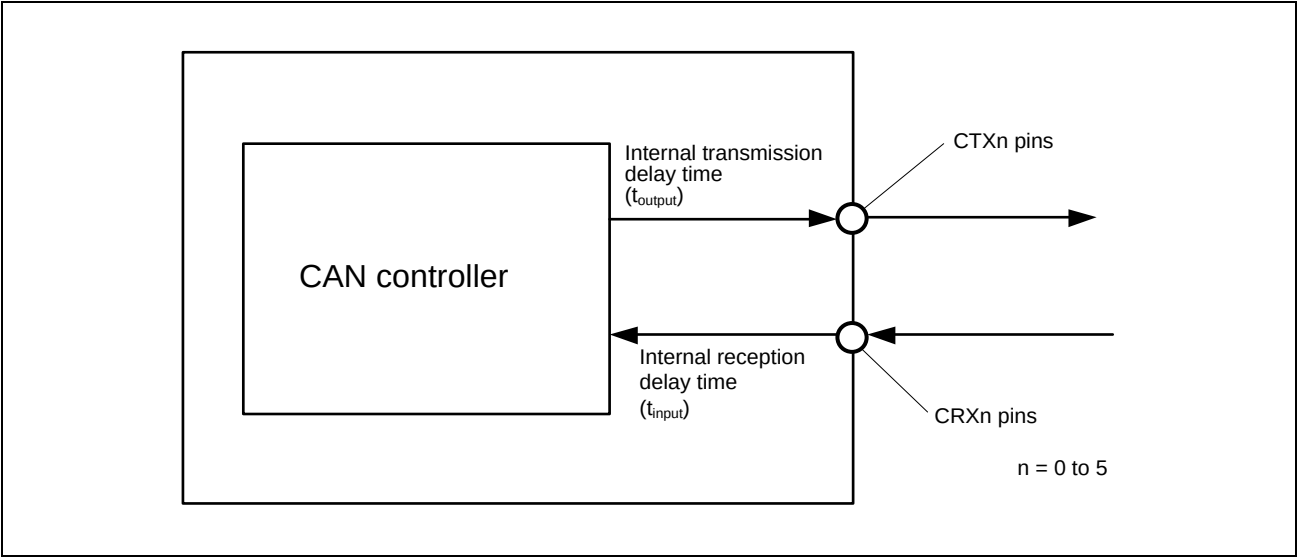


Figure 10.1-54 CANFD Interface Condition

10.1.5.19 A/D Converter Access Timing

Table 10.1-28 A/D Converter Trigger Timing

Parameter			Symbol	Min.	Max.	Unit*1	Figure
A/D converter	A/D converter trigger input pulse width	ADTRG	t _{TRGW}	1.5	—	t _{PADCcyc}	Figure 10.1-55

Note 1. t_{PADCcyc}: ADC_0_PCLK cycle

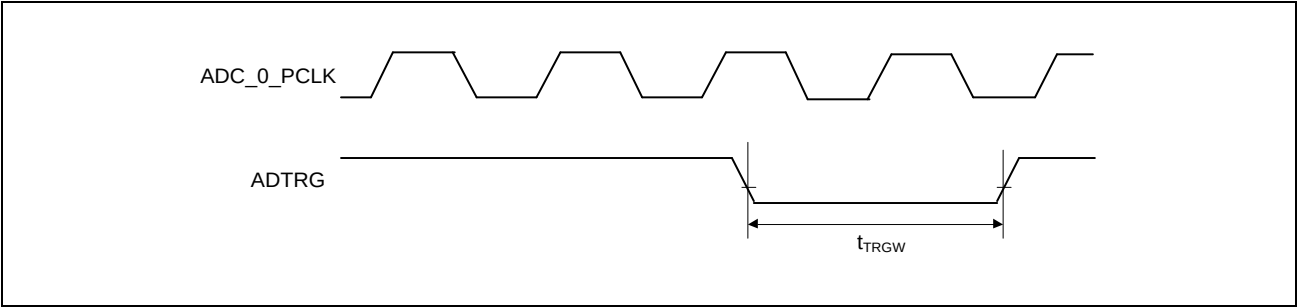


Figure 10.1-55 A/D Converter Trigger Input Timing (ADTRG)

10.1.5.20 SSIU Timing

Conditions:

$$V_{OH} = VDD18 \times 0.5, V_{OL} = VDD18 \times 0.5, C = 30 \text{ pF}^{*1} (1.8 \text{ V})$$

$$V_{OH} = VDD33 \times 0.5, V_{OL} = VDD33 \times 0.5, C = 30 \text{ pF}^{*1} (3.3 \text{ V})$$

Drive strength: $\times 1, \times 2, \times 4, \times 6$

Note 1. Other than t_{RC}: Rise-edge clock timing

Table 10.1-29 SSIU Signal Timing

Parameter	Symbol	Min.	Max.	Unit	Note	Figure
Output clock cycle	t _O	80	15625	ns	—	Figure 10.1-56
Input clock cycle	t _I	80	15625	ns	—	
Output clock high-cycle	t _{HC}	35 ^{*1}	—	ns	—	Figure 10.1-57 to Figure 10.1-60
Output clock low-cycle	t _{LC}	35 ^{*1}	—	ns	—	
Input clock high-cycle	t _{HC}	35	—	ns	—	
Input clock low-cycle	t _{LC}	35	—	ns	—	
Rise-edge clock timing	t _{RC}	—	20 ^{*2}	ns	Output (100 pF)	
Output delay	t _D	-5	19	ns	—	
Setup time	t _S	15	—	ns	—	
Hold time	t _H	5	—	ns	—	

Note 1. The width at high or low level of the clock signal when the input on AUDIO_CLKA, AUDIO_CLKB, or AUDIO_CLKC is output from SCK without frequency division in master mode is min. 30 ns.

The relevant registers are as follows:

ADG: Audio Clock Select Register m (AUDIO_CLK_SELm (m = 0 to 2))

SSIU: Control Register (SSICRn (n = 0 to 9))

Note 2. Output transition time from 20% to 80%

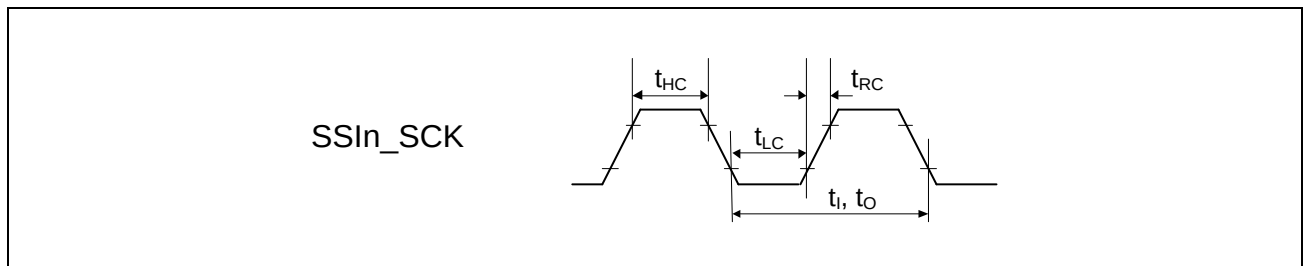


Figure 10.1-56 SCK Clock Input/Output Timing

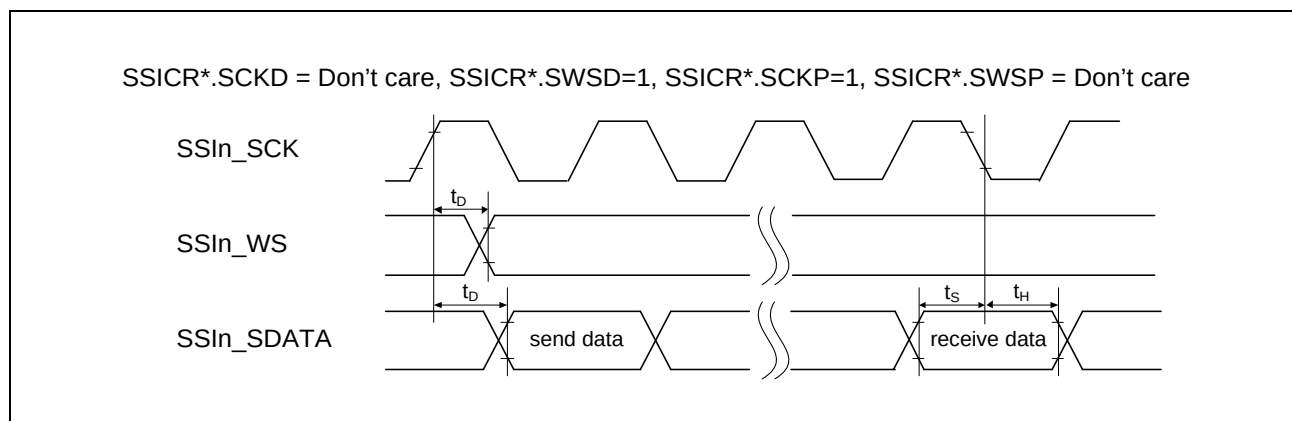


Figure 10.1-57 SSI Timing (1)

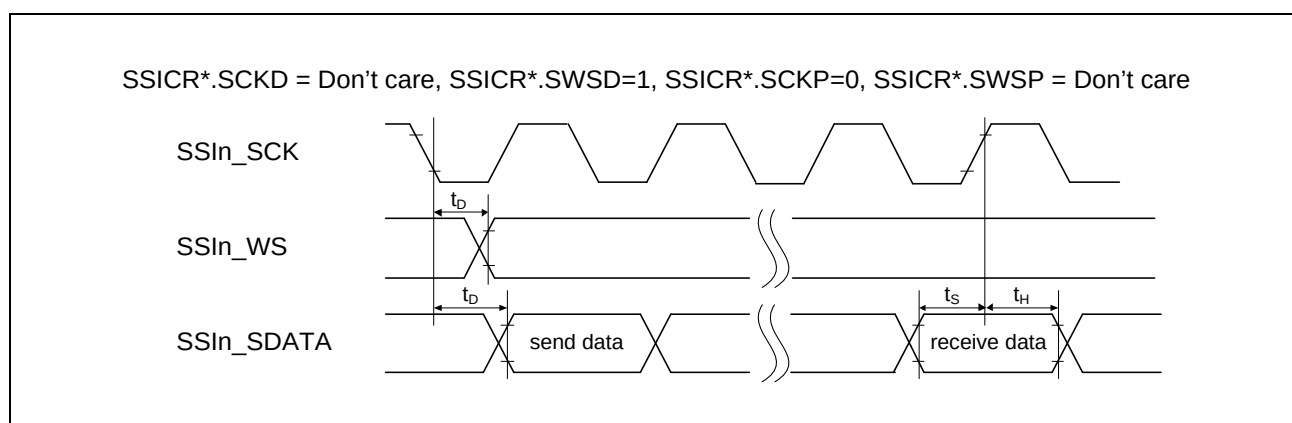


Figure 10.1-58 SSI Timing (2)

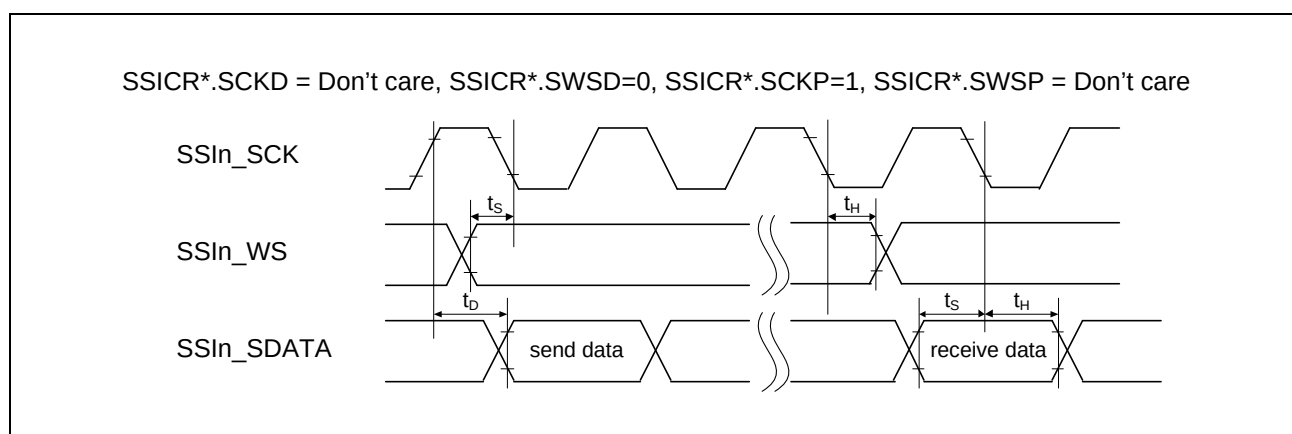


Figure 10.1-59 SSI Timing (3)

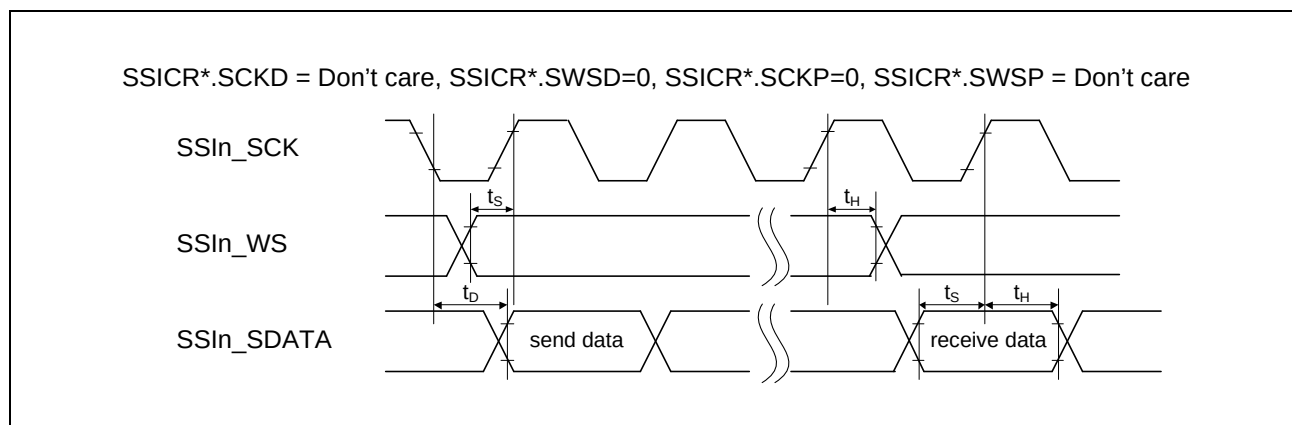


Figure 10.1-60 SSI Timing (4)

10.1.5.21 PDM Timing

Conditions:

$$V_{OH} = V_{DD18} \times 0.5, V_{OL} = V_{DD18} \times 0.5, C = 30 \text{ pF (1.8 V)}$$

$$V_{OH} = V_{DD33} \times 0.5, V_{OL} = V_{DD33} \times 0.5, C = 30 \text{ pF (3.3 V)}$$

Drive strength: $\times 1, \times 2, \times 4, \times 6$

Table 10.1-30 PDM Interface Timing

Parameter	Symbol	Min.	Max.	Unit	Figure
Clock period	t_{PSYNC}	2	32	$t_{CCyc} = 208.33 \text{ ns}$ (4.8 MHz)* ¹	Figure 10.1-61
Clock high-level period	t_{PDCKWH}	$t_{PSYNC} \times 0.45$	$t_{PSYNC} \times 0.55$	ns	
Clock low-level period	t_{PDCKWL}	$t_{PSYNC} \times 0.45$	$t_{PSYNC} \times 0.55$	ns	
Clock rise time	t_{R-EDGE}	—	3×2	ns	
Clock fall time	t_{F-EDGE}	—	3×2	ns	
Setup time	t_{SU}	15	—	ns	Figure 10.1-62,
Hold time	t_H	0	—	ns	Figure 10.1-63

Note 1. t_{CCyc} is the period of PDMCLKnm ($n = 0, 1; m = 0 \text{ to } 2$).

Note 2. Output transition time from 20% to 80%

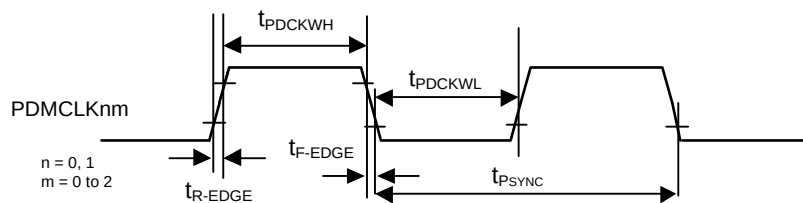


Figure 10.1-61 Timing of Clock Output (PDMCLKnm)

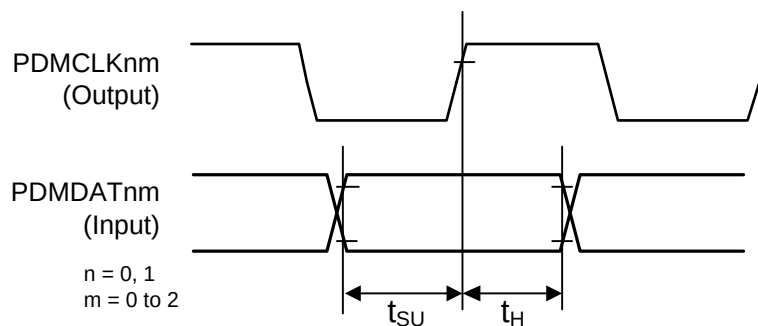


Figure 10.1-62 Timing of Clock Output (Synchronized with the rise of PDMCLKnm)

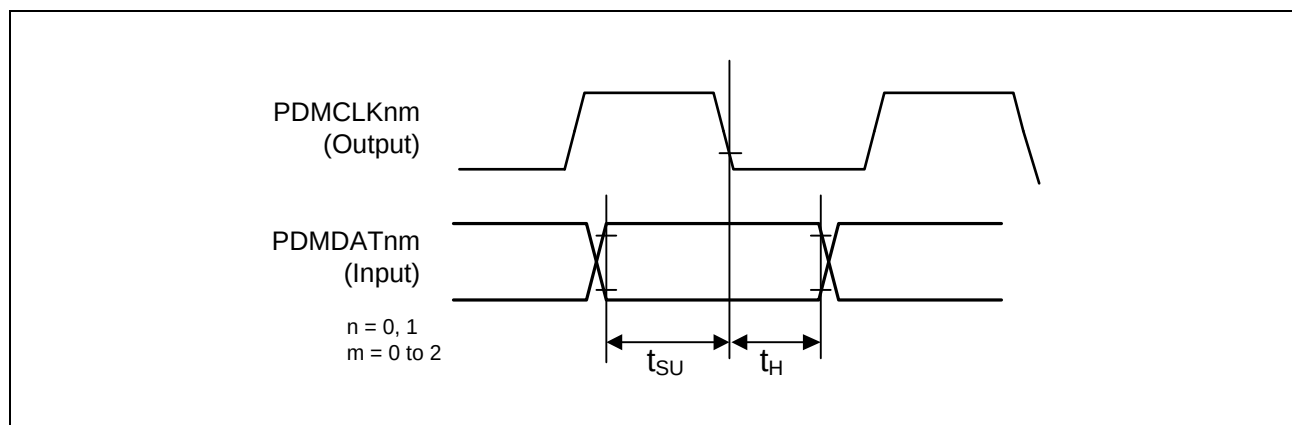


Figure 10.1-63 Timing of Clock Output (Synchronized with the fall of PDMCLKnm)

10.1.5.22 MIPI CSI-2 PHY Characteristics

The MIPI CSI-2 Rx D-PHY of this LSI is equivalent to the MIPI D-PHY Version 1.2.

For details, refer to the MIPI specification.

10.1.5.23 MIPI DSI Tx D-PHY Characteristics

The MIPI DSI Tx D-PHY of this LSI is compliant with the MIPI D-PHY Version 1.2.

For details, refer to the MIPI specification.

10.1.5.24 Control Signal Access Timing

Table 10.1-31 Control Signal Timing

Item	Symbol	Min.	Max.	Unit	Figures
QRESN pulse width	t_{RESW}	1	—	μs	Figure 10.1-64
TRSTN pulse width	t_{TRSW}	1	—	μs	
NMI pulse width	t_{NMIW}	20	—	t_{cyc}^{*1}	Figure 10.1-65
IRQ pulse width	t_{IRQW}	20	—	t_{cyc}^{*1}	
TINT pulse width	t_{TINTW}	20	—	t_{cyc}^{*1}	

Note 1. t_{cyc} = 41.666 ns (24 MHz)

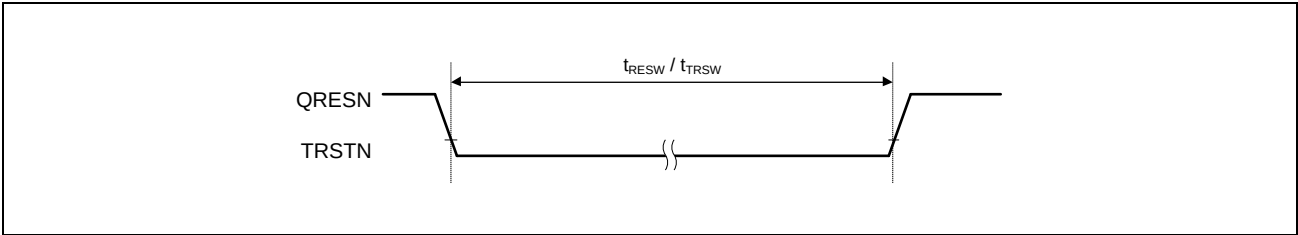


Figure 10.1-64 Reset Input Timing

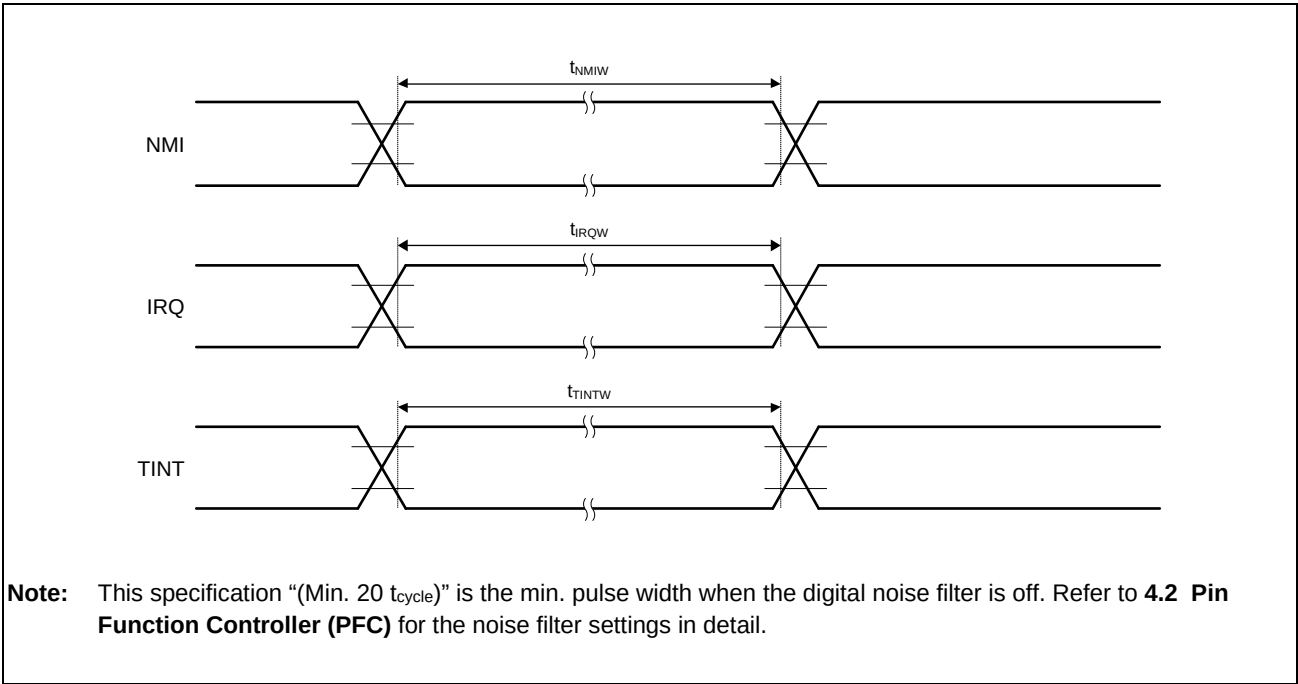


Figure 10.1-65 Interrupt Signal Input Timing

10.1.5.25 JTAG Debugger Interface Access Timing

Table 10.1-32 Debugger IF Timing

Item	Symbol	Min.	Max.	Unit	Figures
TCK_SWCLK cycle time	t_{TCKcyc}	50	—	ns	Figure 10.1-66
TCK_SWCLK high-level pulse width	t_{TCKH}	20	—	ns	Figure 10.1-67
TCK_SWCLK low-level pulse width	t_{TCKL}	20	—	ns	
TDI setup time	t_{TDIS}	15	—	ns	
TDI hold time	t_{TDIH}	15	—	ns	
TMS_SWDIO setup time	t_{TMSS}	15	—	ns	
TMS_SWDIO hold time	t_{TMSh}	15	—	ns	
TMS_SWDIO delay time	t_{SWDO}	—	14	ns	
TDO delay time	t_{TDOD}	—	14	ns	
Capture register setup time	t_{CAPTS}	10	—	ns	Figure 10.1-68
Capture register hold time	t_{CAPTH}	10	—	ns	
Update register delay time	$t_{UPDATED}$	—	20	ns	

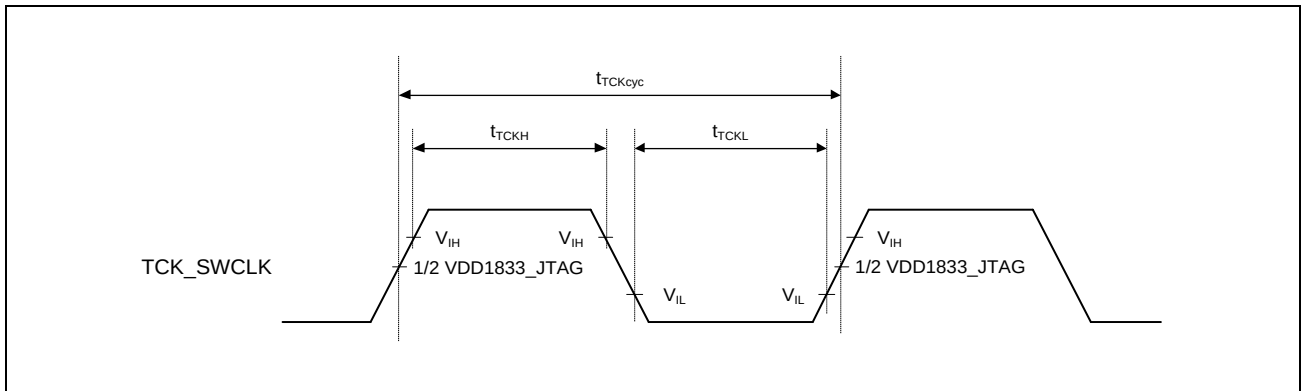


Figure 10.1-66 TCK_SWCLK Input Timing

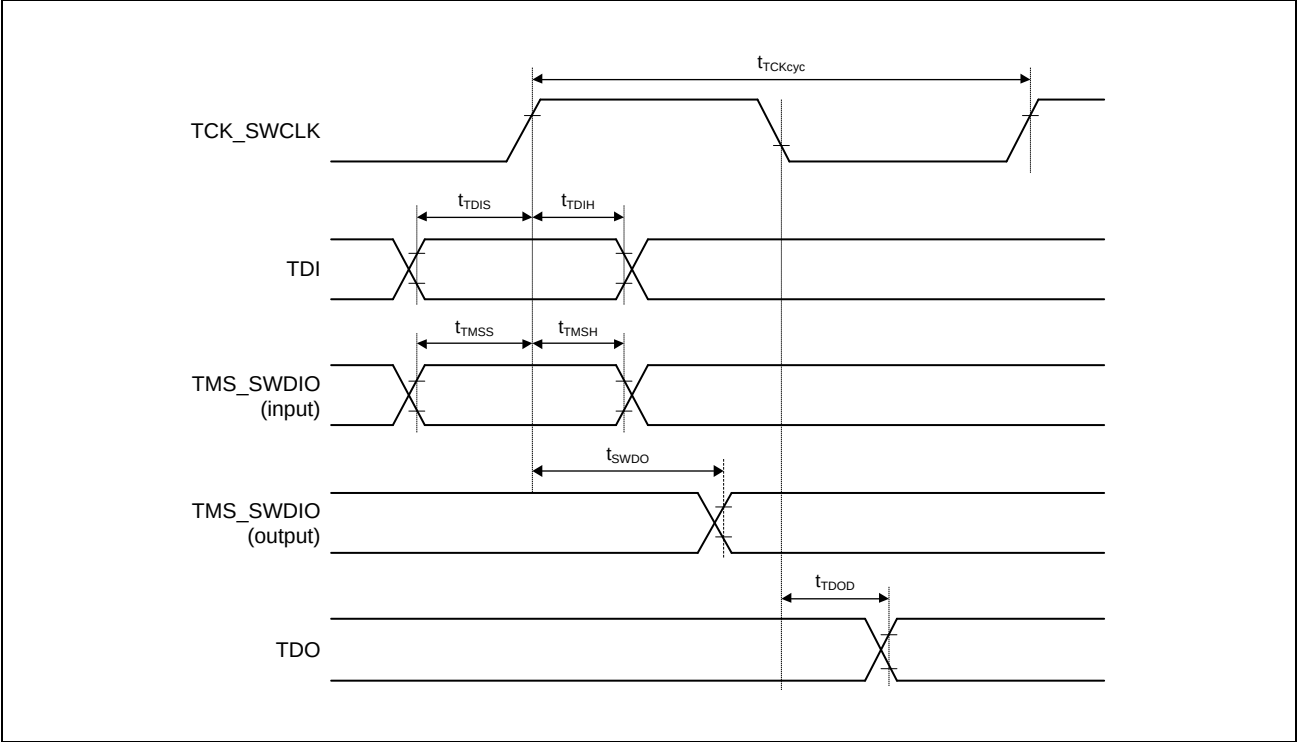


Figure 10.1-67 Data Transfer Timing

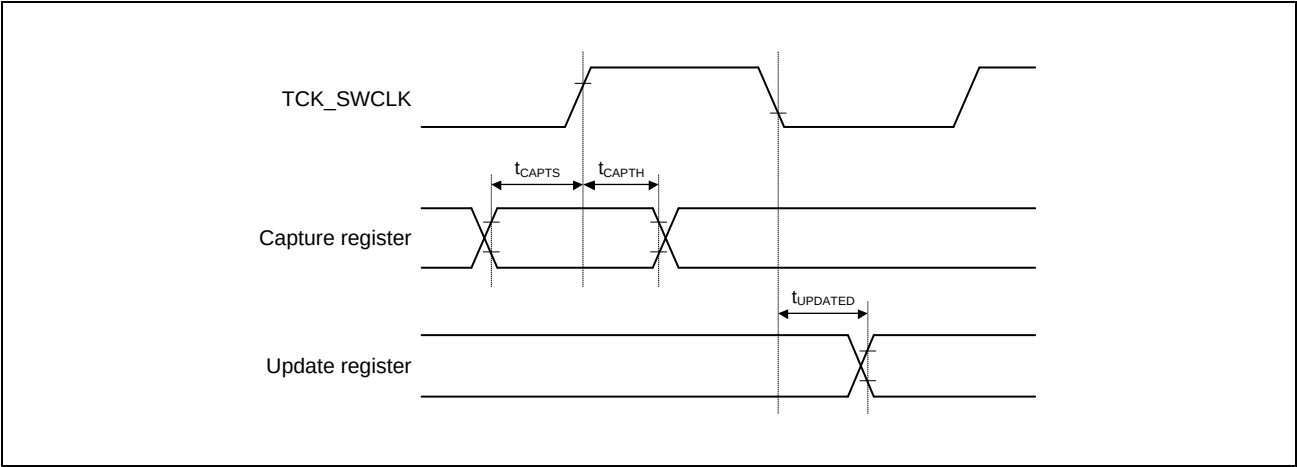


Figure 10.1-68 Boundary Scan Input/Output I/O Timing

10.1.6 Analog Characteristics

10.1.6.1 ADC Characteristics

Table 10.1-33 DC Characteristics

Item	Min.	Typ.	Max.	Unit
Resolution	—	12	—	Bit
Analog input capacitance	—	—	13	pF
Analog input range	0	—	ADCn_ADAVDD 1.8 ^{*2}	V
Conversion time ^{*1} Permissible signal source impedance Max. = 1.0 kΩ	0.4	—	4.0	μs
Offset error	0	—	100	LSB
Full-scale error	-100	—	0	LSB
Quantization error	—	±0.5	—	LSB
DNL differential non-linearity error	—	—	±3.0	LSB
INL integral nonlinearity error	—	—	±6.0	LSB

Note 1. The conversion time is the total of the sampling time and the comparison time.

Note 2. n = 0 to 2

Table 10.1-34 Recommended External Input Resistance

Item	Symbol	Min.	Typ.	Max.	Unit
External input resistance ^{*1} (ANIn00-ANIn07) ^{*2}	$R_{I_{ext}}$	—	—	1	kΩ

Note 1. Output resistance of signal generator + Series parasitic resistance between signal source and ADC input.

Note 2. n = 0 to 2

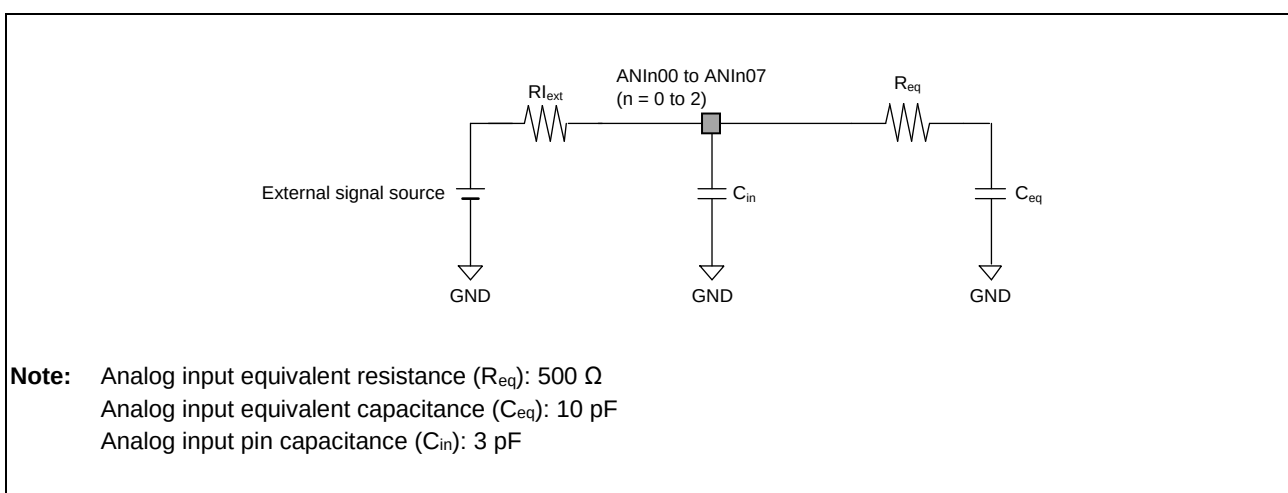


Figure 10.1-69 A/D Converter Equivalent Circuit and Peripheral Configuration Diagram

10.1.6.2 Temperature Sensor Characteristics

Table 10.1-35 Temperature Sensor Characteristics

Item	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Accuracy from -40°C to 125°C	Accm40_125	—	±3.0	±5.0	°C	—

10.1.7 Oscillation Circuits for Connecting Crystal Resonators (OSC)

This LSI chip includes two oscillation circuits (OSC) for connection to crystal resonators, specifically a 24-MHz crystal resonator for the system clock and a 32.768-kHz crystal resonator for the real-time clock. **Table 10.1-36** lists the pins for connecting the crystal resonators and the clock frequencies. **Figure 10.1-70** shows an example of the connections with crystal resonators.

Table 10.1-36 Pins for Connecting Crystal Resonators and Clock Frequency

External Pin Name	I/O	Clock Frequency
For the system clock		
QEXTAL	Input	24 MHz (frequency deviation: ±50 ppm)
QXTAL	Output	24 MHz
For the real-time clock		
RTXIN	Input	32.768 kHz (frequency deviation: ±50 ppm)
RTXOUT	Output	32.768 kHz
For the audio clocks		
AUDIO_EXTAL	Input	4 to 48 MHz
AUDIO_XTAL	Output	4 to 48 MHz

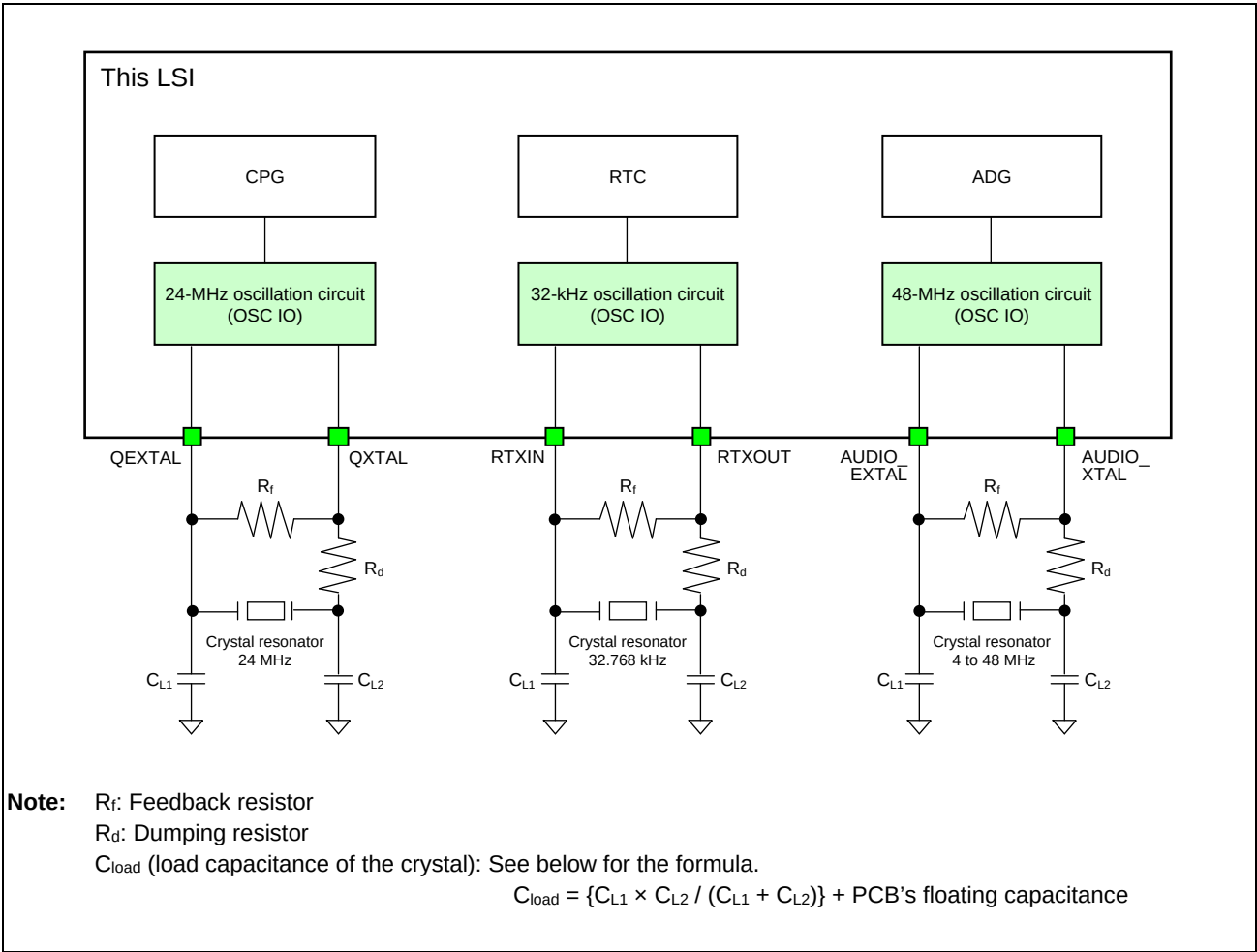


Figure 10.1-70 Example of Connections with Crystal Resonators

Place the crystal resonators and the capacitors C_{L1} and C_{L2} as close as possible to the pins to connect crystal resonators. To avoid interference and to ensure correct oscillation, the grounding points of the capacitors appended to the crystal resonators should be shared, and no wiring patterns should be placed near these components.

The characteristics of the crystal resonators are closely related to the design of the user board. Therefore, the user should sufficiently evaluate them with reference to the example of connection of crystal resonators in **Figure 10.1-70**.

The circuit rating of a crystal resonator depends on the crystal resonator and the stray capacitance of the mounting circuit. Therefore, contact the manufacturer of the crystal resonator before deciding upon the circuit rating. The user should thoroughly evaluate and then set the parameters (resistor and capacitor values).

Table 10.1-37 is a list of recommended values for the crystal resonators.

Table 10.1-37 Recommended Model Values for the Crystal Resonators

Clock Frequency	Model Values for the Crystal Resonators			
	Max. ESR* ¹	Max. C_L * ²	Max. C_0 * ³	Max. Drive Level
32.768 kHz	70 k Ω	12.5 pF	1.4 pF	1 μ W
24 MHz	60 Ω	12 pF	7 pF	100 μ W
48 MHz	50 Ω	10 pF	7 pF	100 μ W

Note 1. ESR means the equivalent series resistor of the crystal resonator.

Note 2. C_L is the load capacitance of the crystal resonator.

Note 3. C_0 is the parallel capacitance of the crystal resonator.

APPENDIX A PACKAGE DIMENSIONS

A.1 Package Dimensions

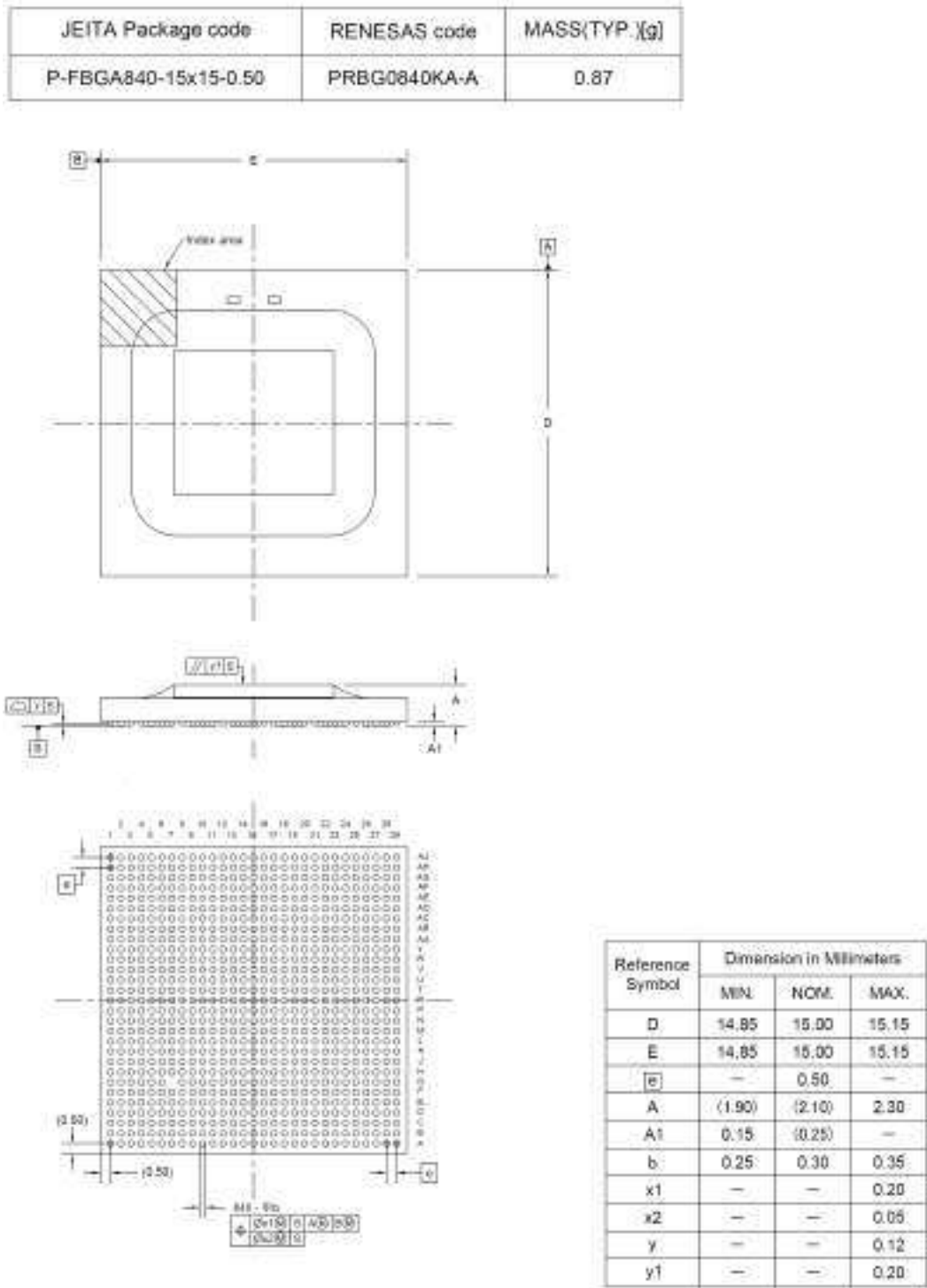


Figure A.1-1 Package Dimensions