FEUL63Q2500-01

ML63Q2500 Group
User's Manual

1st Edition
Issue Date: Jun. 20, 2025

Notice

Precaution on using ROHM Products

1) When using ROHM Products, refer to the latest product information and ensure that usage conditions (absolute maximum ratings[*1], recommended operating conditions, etc.) are within the ranges specified. ROHM disclaims any and all liability for any malfunctions, failure or accident arising out of or in connection with the use of ROHM Products outside of such usage conditions specified ranges, or without observing precautions. Even if it is used within such usage conditions specified ranges, semiconductors can break down and malfunction due to various factors. Therefore, in order to prevent personal injury, fire or the other damage from break down or malfunction of ROHM Products, please take safety at your own risk measures such as complying with the derating characteristics, implementing redundant and fire prevention designs, and utilizing backups and fail-safe procedures.
   *1: Absolute maximum ratings: a limit value that must not be exceeded even momentarily.

2) The Products specified in this document are not designed to be radiation tolerant.

3) Descriptions of circuits, software and other related information in this document are provided only to illustrate the standard operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. And the peripheral conditions must be taken into account when designing circuits for mass production. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, and other related information.

4) No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of ROHM or any third party with respect to ROHM Products or the information contained in this document (including but not limited to, the Product data, drawings, charts, programs, algorithms, and application examples, etc.). Therefore, ROHM shall have no responsibility whatsoever for any dispute, concerning such rights owned by third parties, arising out of the use of such technical information.

5) ROHM intends our Products to be used in a way indicated in this document. Please be sure to contact a ROHM sales office if you consider the use of our Products in different way from original use indicated in this document. For use of our Products in medical systems, please be sure to contact a ROHM representative and must obtain written agreement. Do not use our Products in applications which may directly cause injuries to human life, and which require extremely high reliability, such as aerospace equipment, nuclear power control systems, and submarine repeaters, etc. ROHM disclaims any and all liability for any losses and damages incurred by you or third parties arising by using the Product for purposes not intended by us without our prior written consent.

6) Please use the Products in accordance with any applicable environmental laws and regulations, such as the RoHS Directive. ROHM shall have no responsibility for any damages or losses resulting non-compliance with any applicable laws or regulations.

7) When providing our Products and technologies contained in this document to other countries, you must abide by the procedures and provisions stipulated in all applicable export laws and regulations, including without limitation the US Export Administration Regulations and the Foreign Exchange and Foreign Trade Act.

8) Please contact a ROHM sales office if you have any questions regarding the information contained in this document or ROHM's Products.

9) This document, in part or in whole, may not be reprinted or reproduced without prior consent of ROHM.

(Note) “ROHM” as used in this document means ROHM Co., Ltd.

Other Precaution

1) All information contained in this document is subject to change for the purpose of improvement, etc. without any prior notice. Before purchasing or using ROHM Products, please confirm the latest information with a ROHM sales office.

2) ROHM has used reasonable care to ensure the accuracy of the information contained in this document, however, ROHM shall have no responsibility for any damages, expenses or losses arising from inaccuracy or errors of such information.

TSZ72037・01・001

Notes for product usage

Notes on this page are applicable to the all LAPIS TECHNOLOGY microcontroller products.
For individual notes on each LAPIS TECHNOLOGY microcontroller product, refer to [Note] in the chapters of each user's manual.
The individual notes of each user's manual take priority over those contents in this page if they are different.

1. HANDLING OF UNUSED INPUT PINS
   Fix the unused input pins to the power pin or GND to prevent to cause the device performing wrong operation or increasing the current consumption due to noise, etc. If the handlings for the unused pins are described in the chapters, follow the instruction.

2. STATE AT POWER ON
   At the power on, the data in the internal registers and output of the ports are undefined until the power supply voltage reaches to the recommended operating condition and "L" level is input to the reset pin.
   On LAPIS TECHNOLOGY microcontroller products that have the power on reset function, the data in the internal registers and output of the ports are undefined until the power on reset is generated.
   Be careful to design the application system does not work incorrectly due to the undefined data of internal registers and output of the ports.

3. ACCESS TO UNUSED MEMORY
   If reading from unused address area or writing to unused address area of the memory, the operations are not guaranteed.

4. CHARACTERISTICS DIFFERENCE BETWEEN THE PRODUCT
   Electrical characteristics, noise tolerance, noise radiation amount, and the other characteristics are different from each microcontroller product.
   When replacing from other product to LAPIS TECHNOLOGY microcontroller products, please evaluate enough the apparatus/system which implemented LAPIS TECHNOLOGY microcontroller products.

5. USE ENVIRONMENT
   When using LAPIS TECHNOLOGY microcontroller products in a high humidity environment and an environment where dew condensation, take moisture-proof measures.

Preface

This manual describes the operation of the hardware of the 32-bit microcontroller.

Please ensure that you refer to the latest versions.

– Cortex®-M0+ Technical Reference Manual (DDI0484C)
– Cortex®-M0+ Generic User Guide (DUI0662B)

> The documents above are published by Arm Limited.
> Please ensure that you refer to the latest versions.

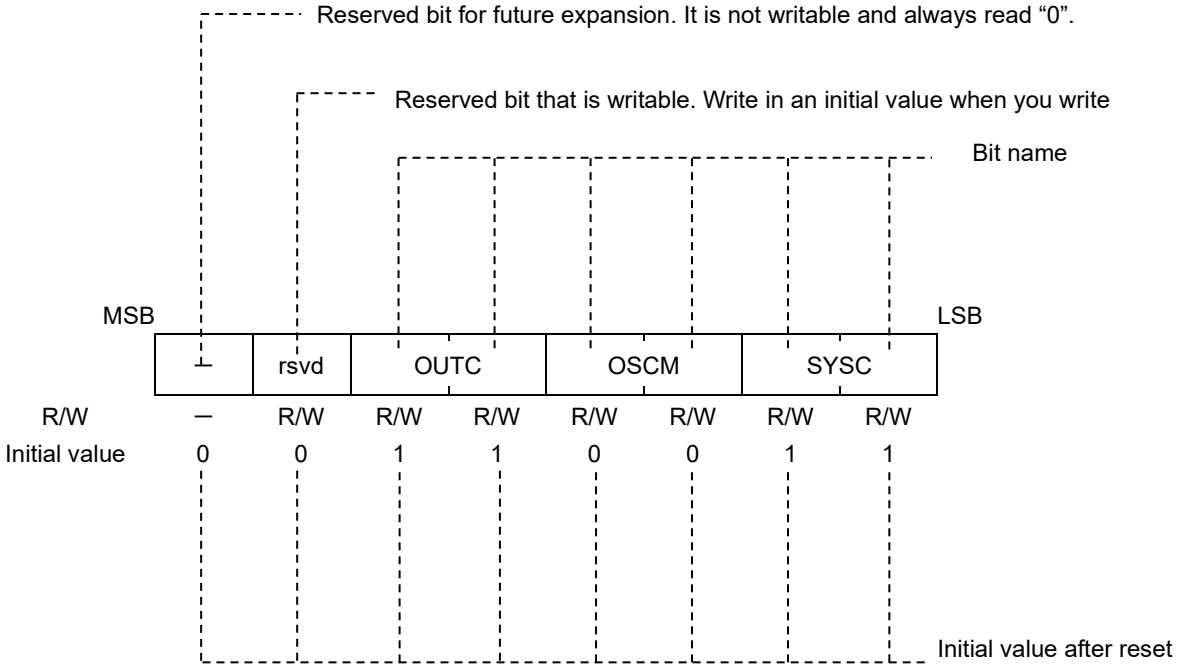
* This product uses Super Flash® technology licensed from Silicon Storage Technology, Inc. Super Flash® is a registered trademark of Silicon Storage Technology, Inc.
* Arm®,Cortex®,Thumb® are registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
* LAPIS TECHNOLOGY™, Solist-AI™ are trademark or registered trademark of ROHM Co., Ltd.

Notation

Classification	Notation	Description
● Numeric value	nnh, nnH, 0xnn	Indicates a hexadecimal number. n: Any value in the range of 0 to F
● Unit	word, W	32 bits
	half word, HW	16 bits
	byte, B	8 bits
	nibble, N	4 bits
	mega-, M	10^6
	kilo-, K	$2^{10} = 1024$
	kilo-, k	$10^3 = 1000$
	milli-, m	10^{-3}
	micro-, μ	10^{-6}
	nano, n	10^{-9}
	second, s	second

● Terminology
"H", "1": Indicates high voltage signal levels VIH and VOH as specified by the electrical characteristics.
"L", "0": Indicates low voltage signal levels VIL and VOL as specified by the electrical characteristics.
SFR: Indicates special function register.

● Register description
R/W: Indicates that Read/Write attribute. "R" indicates that data can be read and "W" indicates that data can be written. "R/W" indicates that data can be read or written.



Chapter 1

Overview

- **Please see "Notes for product usage" and "Notice" in this document on use with this product.**

1. Overview

This LSI is a high-performance low power 32-bit microcontroller. Equipped with a 32-bit CPU core Arm®Cortex®-M0+, it implements peripheral circuits, such as the Solist-AI™ accelerator and CAN controller.

- Applications
  Consumer and Industrial equipment (e.g., Household appliances, Housing equipment, Office equipment, Measurement instrumentation, etc)

  [Note]
  This product cannot be applicable for automotive use, automatic train control systems, and railway safety systems.
  Please contact ROHM sales office in advance if contemplating the integration of this product into applications that requires high reliability, such as transportation equipment for ships and railways, communication equipment for trunk lines, traffic signal equipment, power transmission systems, core systems for financial terminals and various safety control devices.

- Product list
  The ML63Q2500 group has products as show in the Table1-1 with multiple package and memory size combinations.

Table1-1 Product List

Program memory	Data memory (RAM)	Data Flash	48pin TQFP48 WQFN48*	64pin TQFP64 WQFN64*
256KByte	16KByte	8KByte	ML63Q2537	ML63Q2557
128KByte			ML63Q2534	ML63Q2554

* WQFN48 and WQFN64 package product is under development.

1.1 Features

- CPU
  - 32-bit RISC CPU (CPU name: Arm®Cortex®-M0+)
  - Arm®Thumb®/Thumb®-2 instruction supported
  - Serial Wire Debug Port
  - Minimum instruction execution time
    - 30.5 μs (@32.768 kHz system clock)
    - 20.83ns (@48 MHz system clock)
  - System timer (SysTick)
    24 bits × 1 channel, counting by System clock (SYSCLK) (Initial clock: LSCLK) External reference clock is not required

- Internal memory
  - Re-writing the program memory area by software
  - Background Operation (CPU can work while erasing and rewriting to the Data Flash memory area.)
  - Flash ROM
    - Program area: 128/256KB
    - Data area: 8KB
  - Data RAM
    - Work memory 16KB

- Solist-AI™ Acceralator (AxlCORE-ODL)
  - Not only prediction but also learning can be performed on the device. (No need for server/cloud/network connection)
    - Detect anomaly conditions by learning normal conditions for each individual.
  - Capable of high-speed execution of calculations used in AI processing.
    - Addition, subtraction and multiplication of scalars, vectors, and (non-square) matrices are possible.
    - Enables FFT calculation processing useful for vibration sensor data processing.
    - Calculations can be executed without CPU load.
  - Data format: bFloat16 (Built-in integer to bFloat16 conversion function.)
  - Low power consumption/low cost. (Compared to FPGA/GPU etc.), High-speed processing. (Compared to software processing.)
  - One-stop utility using model-based technology.
    Provides software library.
  - Built-in bFloat16-format uniform distribution pseudo-random number generator that can be used as a fixed table.
  - Application example.
    - Motor + acceleration sensor + AI: Early detection of bearing damage.
    - Motor + current sensor + AI: Detection of poor lubrication/contamination of foreign objects.
    - Thermography camera + AI: Accurate detection of abnormal heat generation.
    - FA sensor + AI: Early detection of random failures and abnormal conditions.

- CAN controller
  - 1 channel
  - Equipped with one channel of CAN_FD and 2.0B protocol compliant controller
  - Ability of real time communication control with up to 5Mbps (TBD)
  - ISO 11898-1:2015 compliant
  - SAE J1939 is supported
  - CAN FD up to 64 data bytes is supported
  - Up to 64 proprietaries receive buffers
  - Up to 32 proprietaries transmit buffers
  - Equipped with CAN error log function
  - Equipped with reception filter
  - Individual signals are transmitted when a high-priority message is received

- Interrupt controller (NVIC)
  - 1 non-maskable interrupt source and 31 maskable interrupt sources
  - Priority level (4-level) can be set for each interrupt

- DMA controller (DMAC)
  - 2 channels
  - Enable to allocate multiple DMA transfer request sources for each channel.
  - Channel priority: fixed mode/round robin mode
  - DMA transfer mode: cycle steal mode/burst mode
  - DMA request type: software requests/hardware requests
  - Maximum transfer count: 65,536
  - Data transfer size: 8 bits/16 bits/32 bits
  - Transfer request source: SSIOF, UARTF, I^2CF, SA-ADC

- Time base counter (TBC)
  - Low-speed time base counter × 1 channel with interrupt, × 1 channel for RTC.
  - The clock frequency adjustment in a range approx. -488ppm to +488ppm with 0.119ppm step.

- Real Time Clock (RTC)
  - 99 years calendar, alarm, adjustment of the clock

- 1 kHz Timer (TM1K)
  - 80Hz/60Hz/40Hz/20Hz/10Hz/1Hz interrupt function

- Timers (TMR)
  - 16-bit × 6 channels
  - 32-bit configuration available by using 2 channels
  - Selection of one-shot timer mode is possible
  - External clock can be selected as timer clock.

- Function Timers (FTM)
  - 16-bit × 2 channels
  - Equipped with the timer/capture/PWM functions using a 16-bit counter
  - An event trigger (external pin input interrupt or timer interrupt request) can control start/stop/clear of the timer (however, the minimum pulse width of pin input is timer clock 3φ)
  - 1 to 64 dividing of LSCLK/OSCLK/HSCLK/external input selectable as timer clock
  - Two types of PWM with the same period and different duties and complementary PWM with the dead time set can be output.

- Watchdog timer (WDT)
  - Non-maskable interrupt and reset
  - Free running
  - Overflow period: 4 types selectable (125ms, 500ms, 2s, and 8s when LSCLK = 32.768 kHz)

- Three-phase motor control PWM (NTMS)
  - 16-bit × 3 channels, 2 outputs each; total 6 outputs
  - Center-aligned waveform output with up-down counter
  - Generatable a trigger of the SA-ADC conversion starting
  - Emergency stop with analog-comparator or pin input

- Synchronous serial port (SSIOF)
  - 2 channels
  - 4-stage FIFO for each transmission and reception
  - Master/slave selectable
  - LSB first/MSB first selectable
  - Clock polarity and phase selectable
  - Supports slave-select signal

- UART (UARTF)
  - 4 channels
  - 4-stage FIFO for each transmission and reception
  - Full duplex buffer system
  - Communication speed: up to 115200bps.
  - Programmable interface (data length, parity, stop bits selectable)

- I^2C bus interface (I2CF)
  - 1 channel
  - 4-stage FIFO for each transmission and reception
  - Master/slave function
  - Fast mode (400 kHz), standard mode (100 kHz)

- General-purpose ports (GPIO)
  - Input/output port × up to 49 channels (including secondary or tertiary or quaternary).
  - Port interrupt: 8 sources selectable from all GPIOs

- Successive approximation type A/D converter (SA-ADC)
  - 2 units
  - Input × 12 channels
  - 12-bit resolution
  - Conversion time minimum 1μs/ channel @conversion clock is 24MHz
  - Simultaneous conversion of 2 inputs with 2 units is possible
  - Sampling time can be chosen
  - Consecutive scan conversion function for target input channels
  - Consecutive scan conversion with a specific interval time
  - One conversion result register for each channel
  - Upper /Lower limit is configurable for the conversion result, generates an interrupt
  - A/D converter self test function (full scale, zero scale, internal reference voltage)
  - Following triggers is available to start the A/D conversion
    16-bit timer interrupt requests, functional timer trigger and three-phase motor control PWM ADC conversion staring trigger.

- Analog comparator (CMP)
  - 3 units
  - Common-mode input voltage range: 0.1 to V_{DD}-1.2V
  - Propagation delay: Typ. 0.5μs

- Voltage Level Supervisor (VLS)
  - Monitoring V_{DD} level.
  - Threshold voltage: selectable from 10 level
  - Functional Voltage level detection reset or interrupt is generatable

- Reset
  - RESET_N pin reset
  - Power-on reset
  - Watchdog timer (WDT) overflow reset
  - Voltage Level Supervisor (VLS) reset
  - Crystal oscillation stop detection reset
  - SYSRESETREQ of Arm® Cortex®-M0+ (software reset)

- Clock
  - Low-speed clock:
    - Crystal oscillation (32.768 kHz)
    - Built-in RC oscillation (32.768kHz)
  - High-speed clock:
    - PLL (48MHz) generated from Low-speed clock
    - Crystal oscillation (40/20MHz) for system or CAN

- Power management
  - HALT mode: Instruction execution by CPU is suspended. All peripheral circuits can keep in operating states.
  - STOP mode: Stop of low-speed oscillation and high-speed oscillation (Operations of CPU and peripheral circuits are stopped.)
  - Clock gear: The frequency of high-speed system clock can be changed by software (1/1, 1/2, 1/4, 1/8, 1/16, 1/32 of the oscillation clock)
  - Block Control Function: Power down (reset registers and stop clock supply) the circuits of unused peripherals.

- Guaranteed operating range
  - Operating temperature: (ambient) -40°C to +105°C, (junction) -40°C to +115°C
  - Operating voltage: V_{DD}=2.3 to 5.5V

- Shipping package

Table1-2 Product name list

Package	Body size (including lead) [mm × mm]	Pin pitch [mm]	Packing form and Product name	
			Tray	Tape & Reel
48 pin plastic WQFN	7.0 × 7.0 (-)	0.5	ML63Q2534-NNNGDZW5AY ML63Q2537-NNNGDZW5AY	ML63Q2534-NNNGDZW5BY ML63Q2537-NNNGDZW5BY
48 pin plastic TQFP	7.0 × 7.0 (9.0 × 9.0)	0.5	ML63Q2534-NNNTBZWAY ML63Q2537-NNNTBZWAY	ML63Q2534-NNNTBZWBY ML63Q2537-NNNTBZWBY
64 pin plastic WQFN	9.0 × 9.0 (-)	0.5	ML63Q2554-NNNGDZW5AY ML63Q2557-NNNGDZW5AY	ML63Q2554-NNNGDZW5BY ML63Q2557-NNNGDZW5BY
64 pin plastic TQFP	10.0 × 10.0 (12.0 × 12.0)	0.5	ML63Q2554-NNNTBZWAY ML63Q2557-NNNTBZWAY	ML63Q2554-NNNTBZWBY ML63Q2557-NNNTBZWBY

1.1.1 How To Read The Part Number

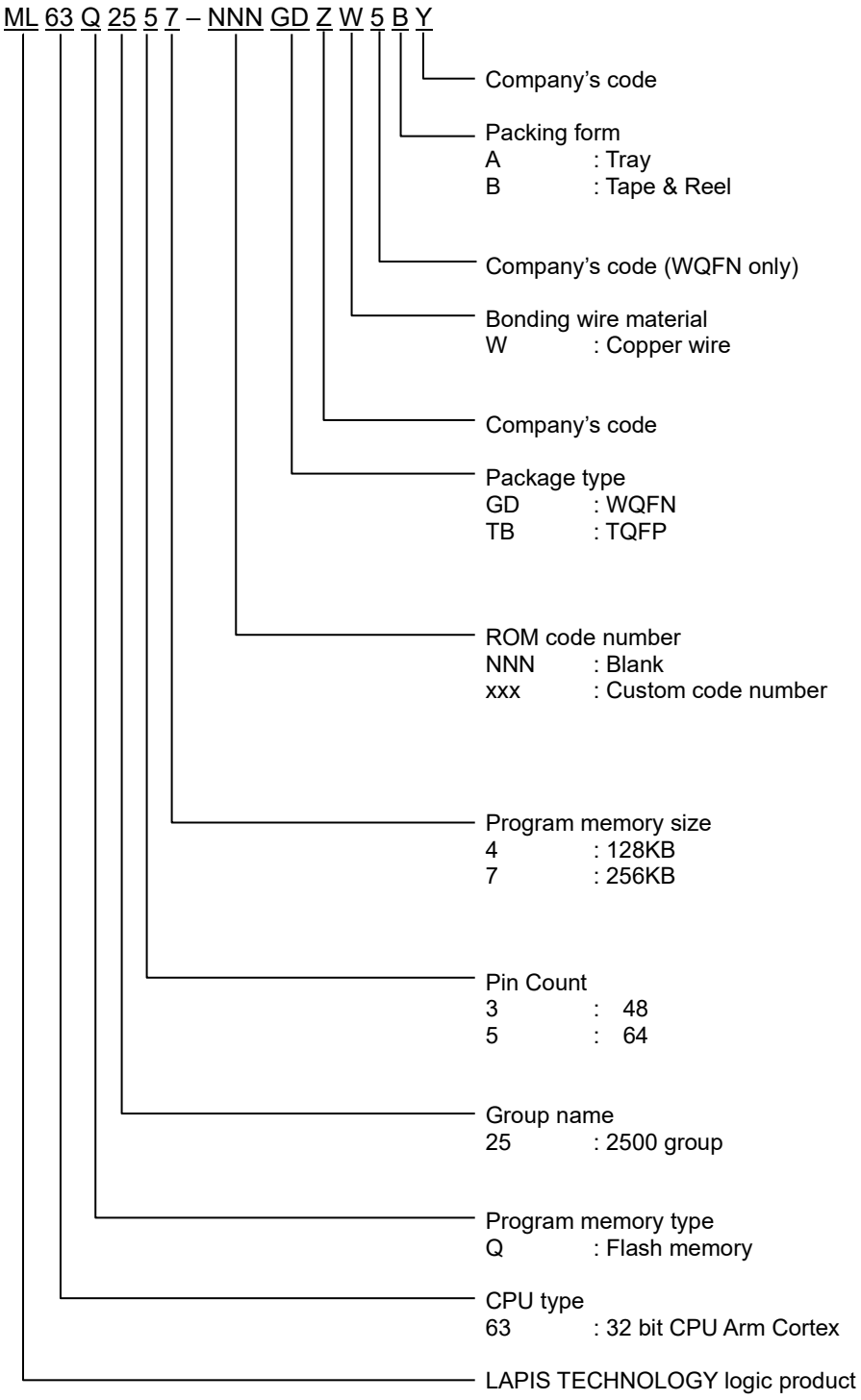


Figure1-1 Part Number

1.1.2 Function List

Table1-3 Product Specification

Category		ML63Q2534	ML63Q2537	ML63Q2554	ML63Q2557
CPU		Arm®Cortex®-M0+			
Memory	Program FLASH	128KB	256KB	128KB	256KB
	Data RAM	16KB			
	Data FLASH	8KB			
Pins	Total	48	48	64	64
	Power	4			
	Reset	1			
	GPIO	34		49	
interrupt	Non maskable	1			
	Maskable (External pin input)	31 (8)			
Timer	TMR	6 ch			
	FTM	2 ch			
	NTMS	1 ch			
	TBC	2 ch			
	RTC	1 ch			
	TM1K	1 ch			
	WDT	1 ch			
Communication	SSIOF	2 ch			
	UARTF	4 ch			
	I2CF	1 ch			
	CAN	1 ch			
Analog	VLS	1 unit			
	SA-ADC	2 unit 12 input			
	CMP	3 unit			
Other	DMAC	2 ch			
	Solist-AI™ Accelerator	1 ch			
Clock	Low speed	32.768kHz (built-in RC oscillation, crystal oscillation)			
	High speed	PLL (Up to 48MHz), crystal oscillation (20 or 40MHz)			
Reset	cause	Pin / POR / WDT / VLS / crystal oscillation stop detection / software			
Operating Condition	Temperature	Ta: -40°C to +105°C (Tj < +115°C)			
	Voltage	V_{DD}=2.3 to 5.5V			
Package		WQFN48, TQFP48		WQFN64, TQFP64	

1.2 Block Diagram

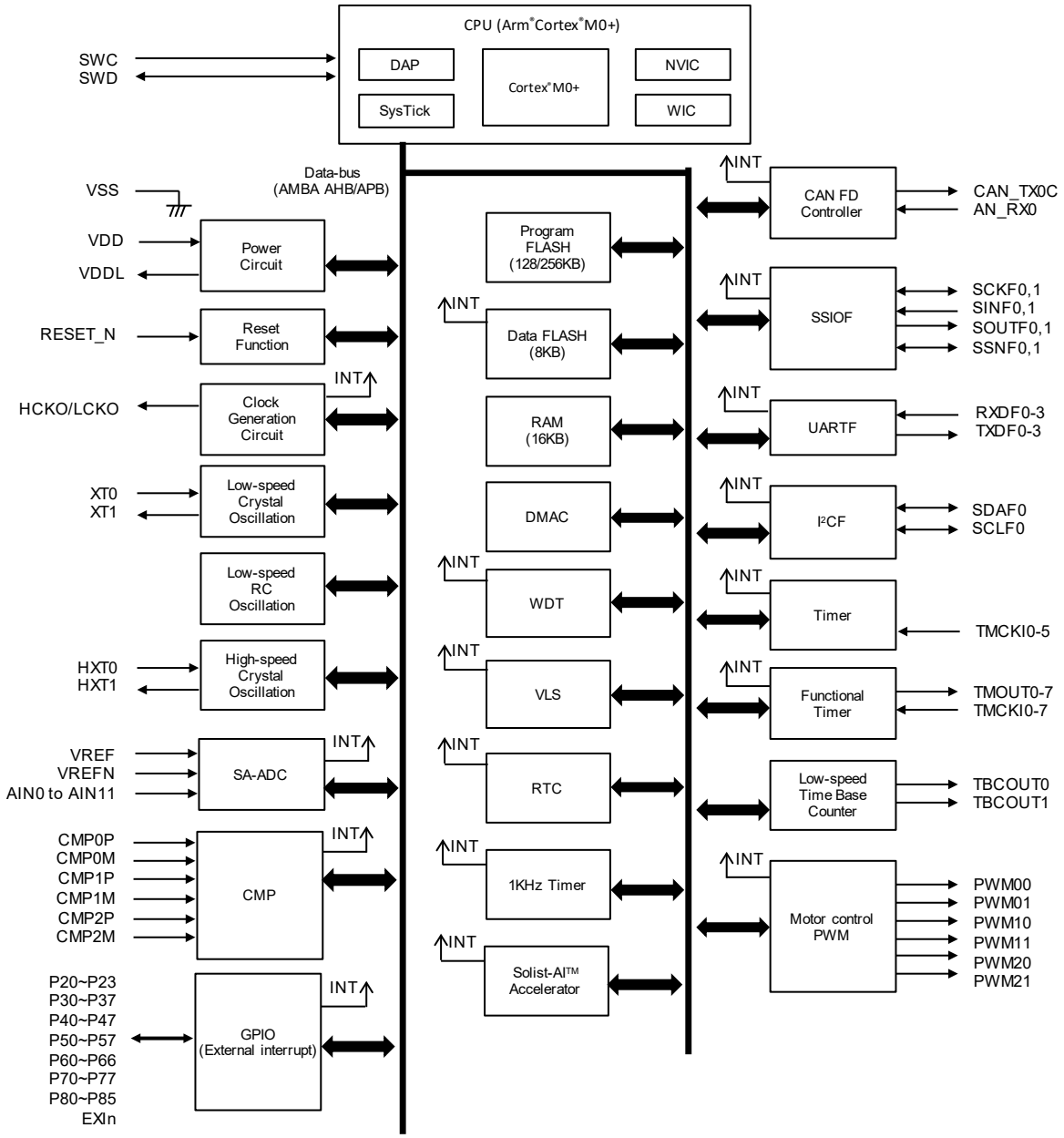


Figure1-2 BLOCK DIAGRAM

1.3 Pins

1.3.1 Pin Configuration

1.3.1.1 ML63Q2537/2534 48pin WQFN Package

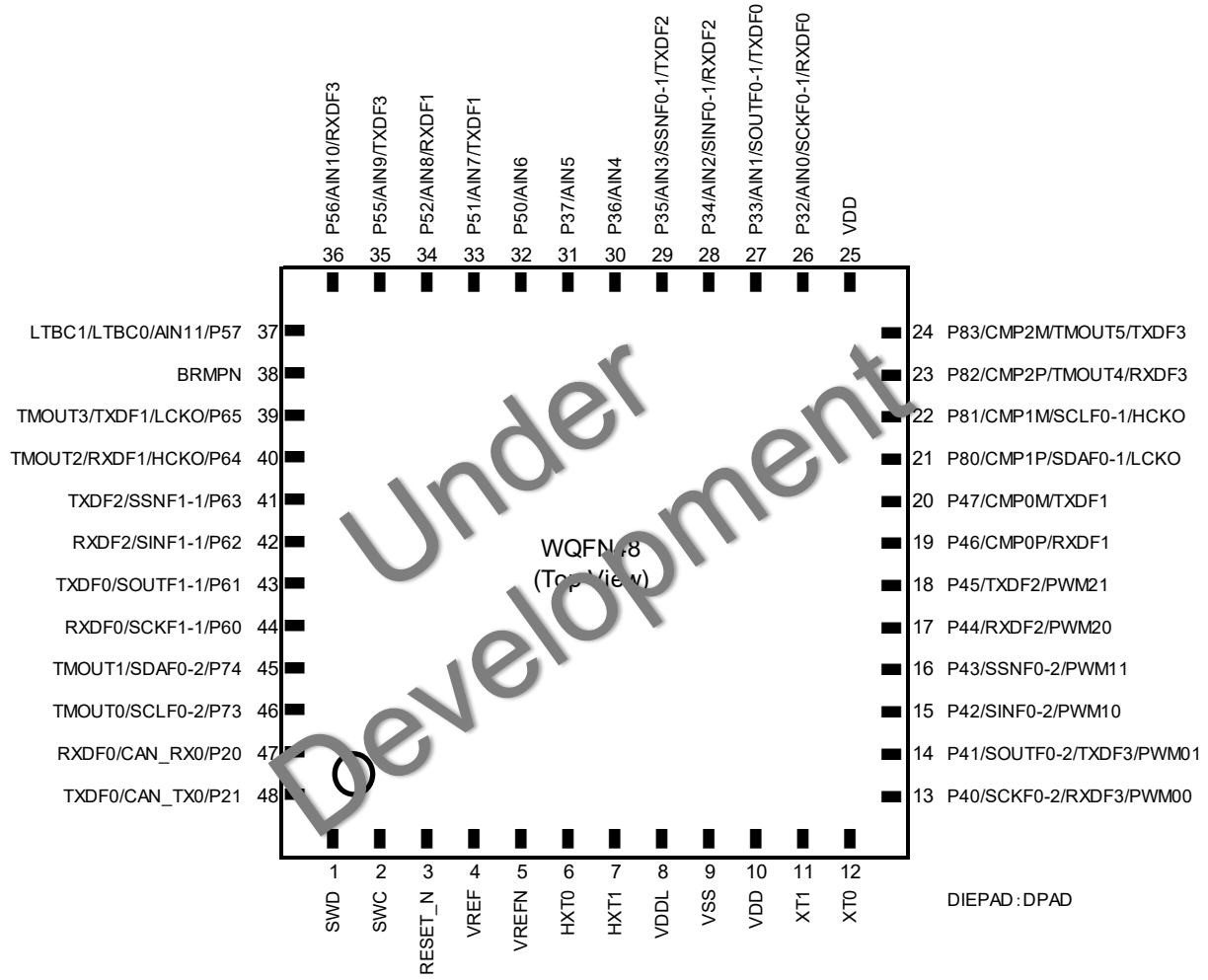


Figure1-3-1 48pin WQFN Package

1.3.1.2 ML63Q2537/2534 48pin TQFP Package

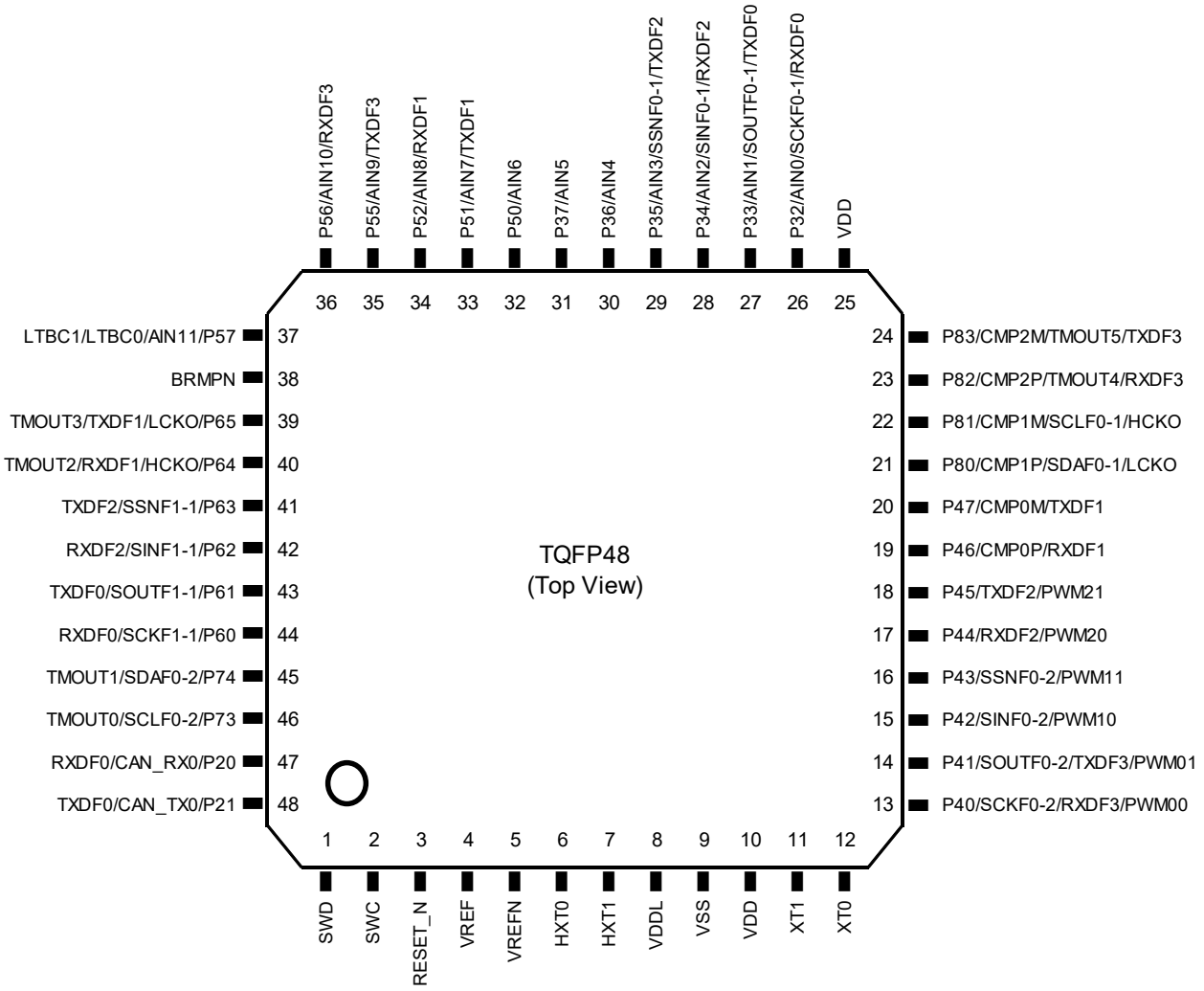


Figure1-3-2 48pin TQFP Package

1.3.1.3 ML63Q2557/2554 64pin WQFN Package

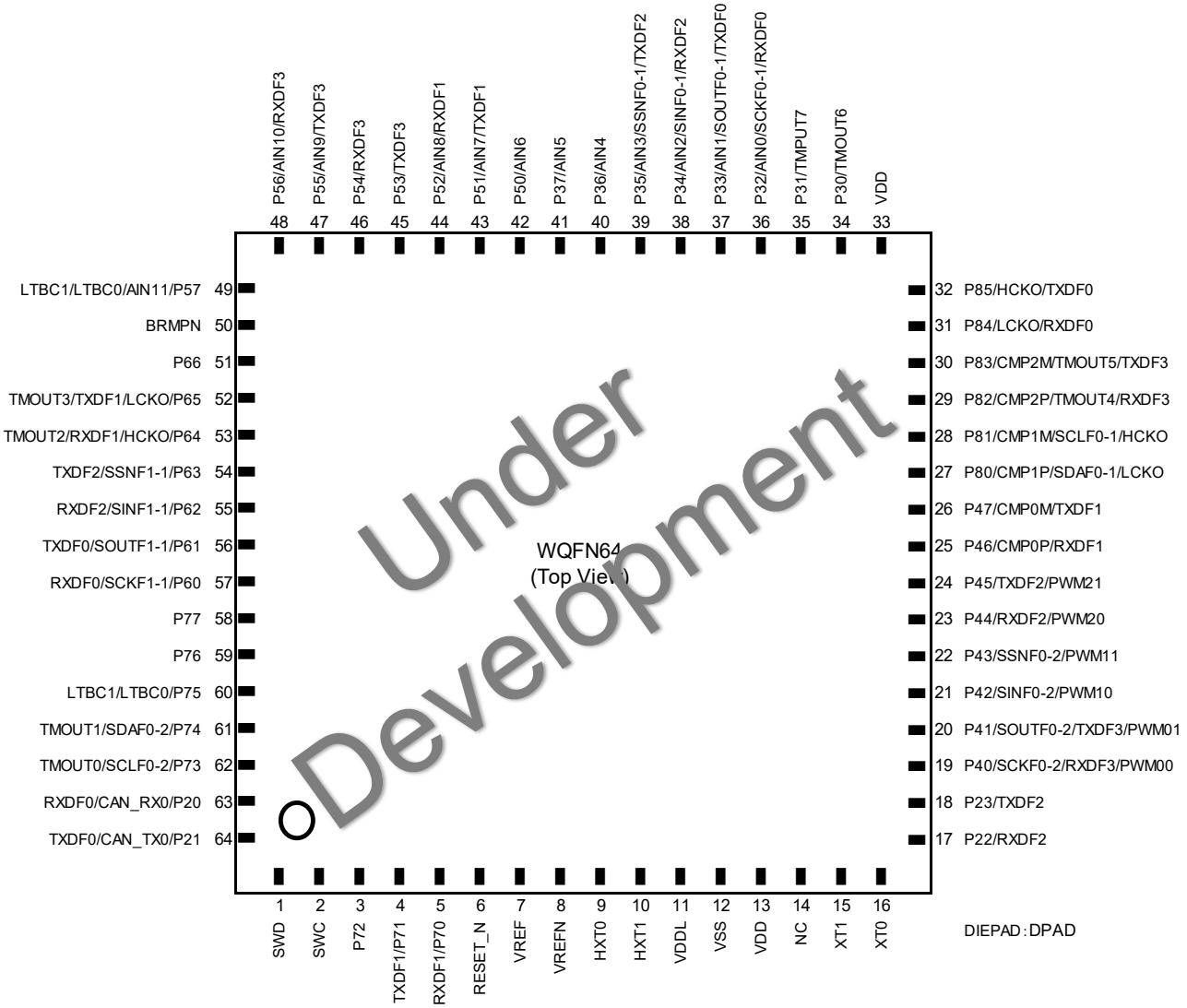


Figure1-3-3 64pin WQFN Package

1.3.1.4 ML63Q2557/2554 64pin TQFP Package

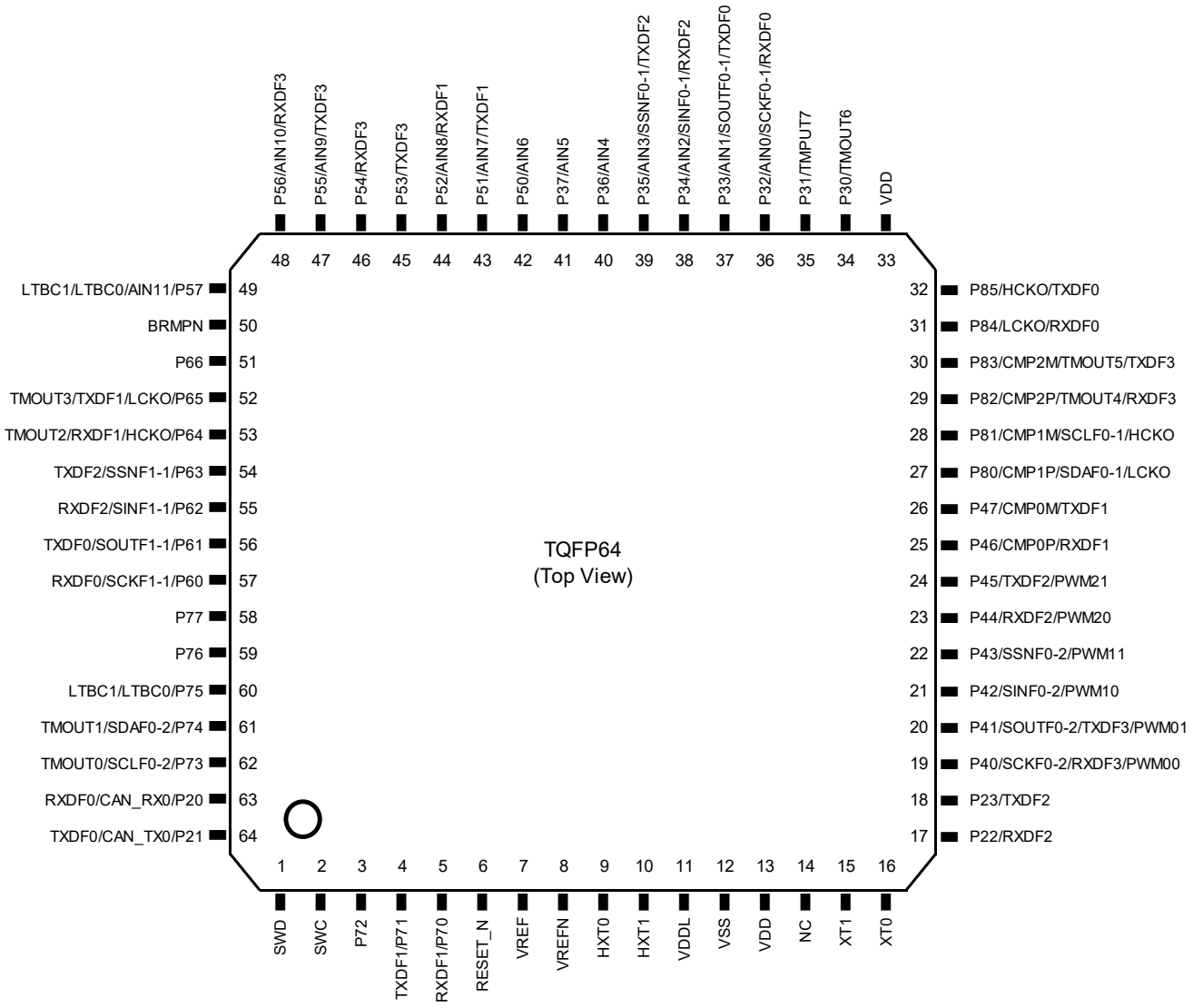


Figure1-3-4 64pin TQFP Package

1.3.2 List of Pins

Table1-4 List of Pins

PIN No.		LSI Pin name	Reset state	Primary Function		Secondary Function		Tertiary Function		Quaternary Function	
ML63Q253x	ML63Q255x			Functional pin name	I/O	Functional pin name	I/O	Functional pin name	I/O	Functional pin name	I/O
1	1	SWD	Pull-up input	SWD	I/O	–	–	–	–	–	–
2	2	SWC	Pull-up input	SWC	I	–	–	–	–	–	–
–	3	P72	Hi-Z output	P72	I/O	–	–	–	–	–	–
–	4	P71	Hi-Z output	P71	I/O	–	–	TXDF1	O	–	–
–	5	P70	Hi-Z output	P70	I/O	–	–	RXDF1	I	–	–
3	6	RESET_N	Hi-Z input	RESET_N	I	–	–	–	–	–	–
4	7	VREF	Hi-Z output	VREF	I/O	–	–	–	–	–	–
5	8	VREFN	Hi-Z output	VREFN	I/O	–	–	–	–	–	–
6	9	HXT0	Hi-Z output	HXT0	–	–	–	–	–	–	–
7	10	HXT1	Hi-Z output	HXT1	–	–	–	–	–	–	–
8	11	VDDL	–	VDDL	–	–	–	–	–	–	–
9	12	VSS	–	VSS	–	–	–	–	–	–	–
10	13	VDD	–	VDD	–	–	–	–	–	–	–
–	14	NC	–	NC	–	–	–	–	–	–	–
11	15	XT1	Hi-Z output	XT1	–	–	–	–	–	–	–
12	16	XT0	Hi-Z output	XT0	–	–	–	–	–	–	–
–	17	P22	Hi-Z output	P22	I/O	–	–	RXDF2	I	–	–
–	18	P23	Hi-Z output	P23	I/O	–	–	TXDF2	O	–	–
13	19	P40	Hi-Z output	P40	I/O	SCKF0-2	I/O	RXDF3	I	PWM00	O
14	20	P41	Hi-Z output	P41	I/O	SOUTF0-2	O	TXDF3	O	PWM01	O
15	21	P42	Hi-Z output	P42	I/O	SINF0-2	I	–	–	PWM10	O
16	22	P43	Hi-Z output	P43	I/O	SSNF0-2	I/O	–	–	PWM11	O
17	23	P44	Hi-Z output	P44	I/O	–	–	RXDF2	I	PWM20	O
18	24	P45	Hi-Z output	P45	I/O	–	–	TXDF2	O	PWM21	O
19	25	P46	Hi-Z output	P46/CMP0P	I/O	–	–	RXDF1	I	–	–
20	26	P47	Hi-Z output	P47/CMP0M	I/O	–	–	TXDF1	O	–	–
21	27	P80	Hi-Z output	P80/CMP1P	I/O	LCKO	O	SDAF0-1	I/O	–	–
22	28	P81	Hi-Z output	P81/CMP1M	I/O	HCKO	O	SCLF0-1	I/O	–	–
23	29	P82	Hi-Z output	P82/CMP2P	I/O	–	–	RXDF3	I	TMOUT4	O
24	30	P83	Hi-Z output	P83/CMP2M	I/O	–	–	TXDF3	O	TMOUT5	O
–	31	P84	Hi-Z output	P84	I/O	LCKO	O	RXDF0	I	–	–
–	32	P85	Hi-Z output	P85	I/O	HCKO	O	TXDF0	O	–	–
25	33	VDD	–	VDD	–	–	–	–	–	–	–
–	34	P30	Hi-Z output	P30	I/O	–	–	–	–	TMOUT6	O
–	35	P31	Hi-Z output	P31	I/O	–	–	–	–	TMOUT7	O
26	36	P32	Hi-Z output	P32/AIN0	I/O	SCKF0-1	I/O	RXDF0	I	–	–
27	37	P33	Hi-Z output	P33/AIN1	I/O	SOUTF0-1	O	TXDF0	O	–	–
28	38	P34	Hi-Z output	P34/AIN2	I/O	SINF0-1	I	RXDF2	I	–	–
29	39	P35	Hi-Z output	P35/AIN3	I/O	SSNF0-1	I/O	TXDF2	O	–	–
30	40	P36	Hi-Z output	P36/AIN4	I/O	–	–	–	–	–	–
31	41	P37	Hi-Z output	P37/AIN5	I/O	–	–	–	–	–	–
32	42	P50	Hi-Z output	P50/AIN6	I/O	–	–	–	–	–	–
33	43	P51	Hi-Z output	P51/AIN7	I/O	–	–	TXDF1	O	–	–
34	44	P52	Hi-Z output	P52/AIN8	I/O	–	–	RXDF1	I	–	–

PIN No.		LSI Pin name	Reset state	Primary Function		Secondary Function		Tertiary Function		Quaternary Function	
ML63Q253x	ML63Q255x			Functional pin name	I/O	Functional pin name	I/O	Functional pin name	I/O	Functional pin name	I/O
-	45	P53	Hi-Z output	P53	I/O	–	–	TXDF3	O	–	–
-	46	P54	Hi-Z output	P54	I/O	–	–	RXDF3	I	–	–
35	47	P55	Hi-Z output	P55/AIN9	I/O	–	–	TXDF3	O	–	–
36	48	P56	Hi-Z output	P56/AIN10	I/O	–	–	RXDF3	I	–	–
37	49	P57	Hi-Z output	P57/AIN11	I/O	LTBC0	O	LTBC1	O	–	–
38	50	BRMPN	Pull-up input	BRMPN	I	–	–	–	–	–	–
-	51	P66	Hi-Z output	P66	I/O	–	–	–	–	–	–
39	52	P65	Hi-Z output	P65	I/O	LCKO	O	TXDF1	O	TMOUT3	O
40	53	P64	Hi-Z output	P64	I/O	HCKO	O	RXDF1	I	TMOUT2	O
41	54	P63	Hi-Z output	P63	I/O	SSNF1-1	I/O	TXDF2	O	–	–
42	55	P62	Hi-Z output	P62	I/O	SINF1-1	I	RXDF2	I	–	–
43	56	P61	Hi-Z output	P61	I/O	SOUTF1-1	O	TXDF0	O	–	–
44	57	P60	Hi-Z output	P60	I/O	SCKF1-1	I/O	RXDF0	I	–	–
-	58	P77	Hi-Z output	P77	I/O	–	–	–	–	–	–
-	59	P76	Hi-Z output	P76	I/O	–	–	–	–	–	–
–	60	P75	Hi-Z output	P75	I/O	LTBC0	O	LTBC1	O	–	–
45	61	P74	Hi-Z output	P74	I/O	–	–	SDAF0-2	I/O	TMOUT1	O
46	62	P73	Hi-Z output	P73	I/O	–	–	SCLF0-2	I/O	TMOUT0	O
47	63	P20	Hi-Z output	P20	I/O	CAN_RX0	I	RXDF0	I	–	–
48	64	P21	Hi-Z output	P21	I/O	CAN_TX0	O	TXDF0	O	–	–
DIEPAD		DPAD	–	DPAD	–	–	–	–	–	–	–

Pxx can be used as GPIO, EXI and TMCKI.
The I^2C and SSIO interface use pins with a combination of the same suffix number after the hyphen.

1.3.3 Pin Description

The table below shows the pin descriptions for each function.
"I/O" Field in the below table define the pin type ("-" : power supply pin, "I" : Input pin, "O" : Out put pin, "I/O" bi-directional pin)

Table1-5 Pin Description

Function	Functional pin name	LSI pin name	I/O	Description
Power	—	VSS	—	Negative power supply pin (-) Define this terminal potential as V_{SS}
	—	VDD	—	Positive power supply pin (+). Connect a capacitor C_V (more than 1μF) between this pin and VSS. Define this terminal potential as $V_{DD.}$
	—	VDDL	—	Power supply for internal logic (internal regulator's output). Connect a capacitor C_L (1μF) between this pin and VSS.
System	SWC	SWC	I	Debugger clock input
	SWD	SWD	I/O	Debugger data input/output
	BRMPN	BRMPN	I	Remapping control input (for firmware update) Based on the BRMPN pin setting at the time of the reset release, Bank0 is remapped.
	RESET_N	RESET_N	I	Reset input. When this pin is set to a "L" level, system reset mode is set and the internal section is initialized. When this pin is set to a "H" level subsequently, program execution starts. A pull-up resistor is NOT internally connected.
General-purpose input/output port (GPIO)	P20 to P23	P20 to P23	I/O	General purpose input/output - High-impedance (initial value) - Input with Pull-up - Input without Pull-up - CMOS output - N channel (N-ch) open drain output
	P30 to P37	P30 to P37		
	P40 to P47	P40 to P47		
	P50 to P57	P50 to P57		
	P60 to P66	P60 to P66		
	P70 to P77	P70 to P77		
	P80 to P85	P80 to P85		
Clock input	XT0	XT0	I	32.768kHz crystal connection pin for low-speed clock. The oscillation unit is connected across XT0 and XT1. Capacitors C_{DL} and C_{GL} are connected across this pin and VSS as required. The XT1 is also used as an external clock input.
	XT1	XT1	I/O	
	HXT0	HXT0	I	20MHz or 40MHz crystal connection pin to use for CAN interface or high-speed clock. The oscillation unit is connected across HXT0 and HXT1. Capacitors C_{DL2} and C_{GL2} are connected across this pin and VSS as required. An external clock input is not supported.
	HXT1	HXT1	O	
Clock output	HCKO	P81 P64 P85	O	High-speed clock output
	LCKO	P80 P65 P84	O	Low-speed clock output
	LTBC0 /LTBC1	P57 P75	O	LTBC 1Hz/2Hz output
External interrupt	EXI0 to EXI7	P20 to P85	I	External maskable interrupt input. Select and assign from all GPIOs. There are also used for trigger of functional timers.
External timer clock	TMCKI0 toTMCKI7	P20 to P85	I	External timer clock input Select and assign from all GPIOs. There are used for clock of timers and functional timers.
Functional timer	TMOUT0	P73	O	Functional timer output
	TMPUT1	P74	O	
	TMOUT2	P64	O	
	TMOUT3	P65	O	
	TMOUT4	P82	O	
	TMOUT5	P83	O	
	TMOUT6	P30	O	
	TMOUT7	P31	O	

Function	Functional pin name	LSI pin name	I/O	Description
Three phase motor control PWM	PWM00	P40	O	PWM output
	PWM01	P41	O	
	PWM10	P42	O	
	PWM11	P43	O	
	PWM20	P44	O	
	PWM21	P45	O	
I^2C	SCLF0	P73 P81	I/O	I2CF0 clock input/output pin. This pin has an NMOS open drain output. When using this pin as a function of the I^2C, externally connect a pull-up resistor.
	SDAF0	P74 P80	I/O	I2CF0 data input/output pin. This pin has an NMOS open drain output. When using this pin as a function of the I^2C, externally connect a pull-up resistor.
CAN	CAN_RX0	P20	I	Serial data input of CAN bus 0
	CAN_TX0	P21	O	Serial data output pin of CAN bus 0
UART	RXDF0	P20 P84 P32 P60	I	UARTF0 reception data input
	TXDF0	P21 P85 P33 P61	O	UARTF0 transmission data output
	RXDF1	P70 P46 P52 P64	I	UARTF1 reception data input
	TXDF1	P71 P47 P51 P65	O	UARTF1 transmission data output
	RXDF2	P22 P34 P44 P62	I	UARTF2 reception data input
	TXDF2	P23 P35 P45 P63	O	UARTF2 transmission data output
	RXDF3	P40 P54 P56 P82	I	UARTF3 reception data input
	TXDF3	P41 P53 P55 P83	O	UARTF3 transmission data output
Synchronous serial port (SIOF)	SCKF0	P32 P40	I/O	SIOF0 clock input/output
	SOUTF0	P33 P41	O	SIOF0 data output
	SINF0	P34 P42	I	SIOF0 data input
	SSNF0	P35 P43	I/O	SIOF0 select input/output
	SCKF1	P60	I/O	SIOF1 clock input/output
	SOUTF1	P61	O	SIOF1 data output
	SINF1	P62	I	SIOF1 data input
	SSNF1	P63	I/O	SIOF1 select input/output
Successive approximation type A/D converter (SA-ADC)	VREF	VREF	I	Reference voltage positive input for SA-ADC
	VREFN	VREFN	I	Reference voltage negative input for SA-ADC
	AIN0	P32	I	SA-ADC analog input channel 0
	AIN1	P33	I	SA-ADC analog input channel 1
	AIN2	P34	I	SA-ADC analog input channel 2
	AIN3	P35	I	SA-ADC analog input channel 3
	AIN4	P36	I	SA-ADC analog input channel 4
	AIN5	P37	I	SA-ADC analog input channel 5
	AIN6	P50	I	SA-ADC analog input channel 6
	AIN7	P51	I	SA-ADC analog input channel 7
	AIN8	P52	I	SA-ADC analog input channel 8
	AIN9	P55	I	SA-ADC analog input channel 9
	AIN10	P56	I	SA-ADC analog input channel 10
	AIN11	P57	I	SA-ADC analog input channel 11
CMP	CMP0P	P46	I	Input (+) for analog comparator 0
	CMP0M	P47	I	Input (-) for analog comparator 0
	CMP1P	P80	I	Input (+) for analog comparator 1
	CMP1M	P81	I	Input (-) for analog comparator 1
	CMP2P	P82	I	Input (+) for analog comparator 2
	CMP2M	P83	I	Input (-) for analog comparator 2
Other	—	NC	—	Not connected anywhere. Do connect nothing.
	—	DPAD	—	This is pad on the bottom of WQFN package. Do connect nothing.

1.3.4 Termination of Unused Pins

Table1-6 Termination of unused pins

Pin	Pin termination
RESET_N	Connect to V_{DD}
BRMPN	Open
SWC	Connect a pull-up resistor.
SWD	Connect a pull-up resistor.
VREF	Connect to V_{DD}
VREFN	Connect to V_{SS}
HXT0, HXT1	Open
XT0, XT1	Open
P20 to P23 P30 to P37 P40 to P47 P50 to P57 P60 to P66 P70 to P77 P80 to P85	Open
DPAD	Open
NC	Open

[Note]

- **For unused input ports or unused input/output ports, if the corresponding pins are configured as high-impedance inputs and left open, the supply current may become excessively large. Therefore, terminate unused pins according to Table 1-6.**

Chapter 2

CPU

2. CPU

2.1 Overview

A RISC processor manufactured by Arm®.
It is a 32-bit processor for small size and low power consumption applications and has a 2-stage pipeline configuration. It implements the Arm®v6-M architecture, and operates with 16-bit Arm®Thumb® instructions and Arm®Thumb®-2 instructions.
For details, see "Cortex®-M0+ Technical Reference Manual".

2.1.1 Features

- Multiplier can process 32 bit × 32 bit in one cycle (holding the lower 32 bits of the operation result).
- Serial wire debug port
- Little-Endian
- Built-in debug component with four break points and two watch points
- Wakeup from any interrupt is possible.
- WFI (Wait for Interrupts) supported
- WFE (Wait for Events) supported
- SysTick supported
- MTB (Micro Trace Buffer) not supported

2.2 Description of Registers

2.2.1 List of Registers

Address	Name	Symbol	R/W	Size	Initial value
0xE000_E010	SysTick control and status register	SYST_CSR	R/W	32	0x0000_00004
0xE000_E014	SysTick reload value register	SYST_RVR	R/W	32	Unknown
0xE000_E018	SysTick current value register	SYST_CVR	R/W	32	Unknown
0xE000_E01C	SysTick calibration value register	SYST_CALIB	R	32	0xC000_0000
0xE000_ED00	CPUID register	CPUID	R	32	0x410C_C601
0xE000_ED04	Interrupt control and state register	ICSR	R/W	32	0x0000_0000
0xE000_ED0C	Application interrupt and reset control register	AIRCR	R/W	32	0xFA05_0000
0xE000_ED10	System control register	SCR	R/W	32	0x0000_0000
0xE000_ED14	Configuration and control register	CCR	R	32	0x0000_0208
0xE000_ED1C	System handler priority register 2	SHPR2	R/W	32	0x0000_0000
0xE000_ED20	System handler priority register 3	SHPR3	R/W	32	0x0000_0000

For details of the register, see "Cortex®-M0+ Technical Reference Manual".

2.3 AHB Bus Configuration

The internal bus is multi-layer bus architecture shown as Figure 2-1.
The priority of access is CAN, DMAC, and CPU, in that order.

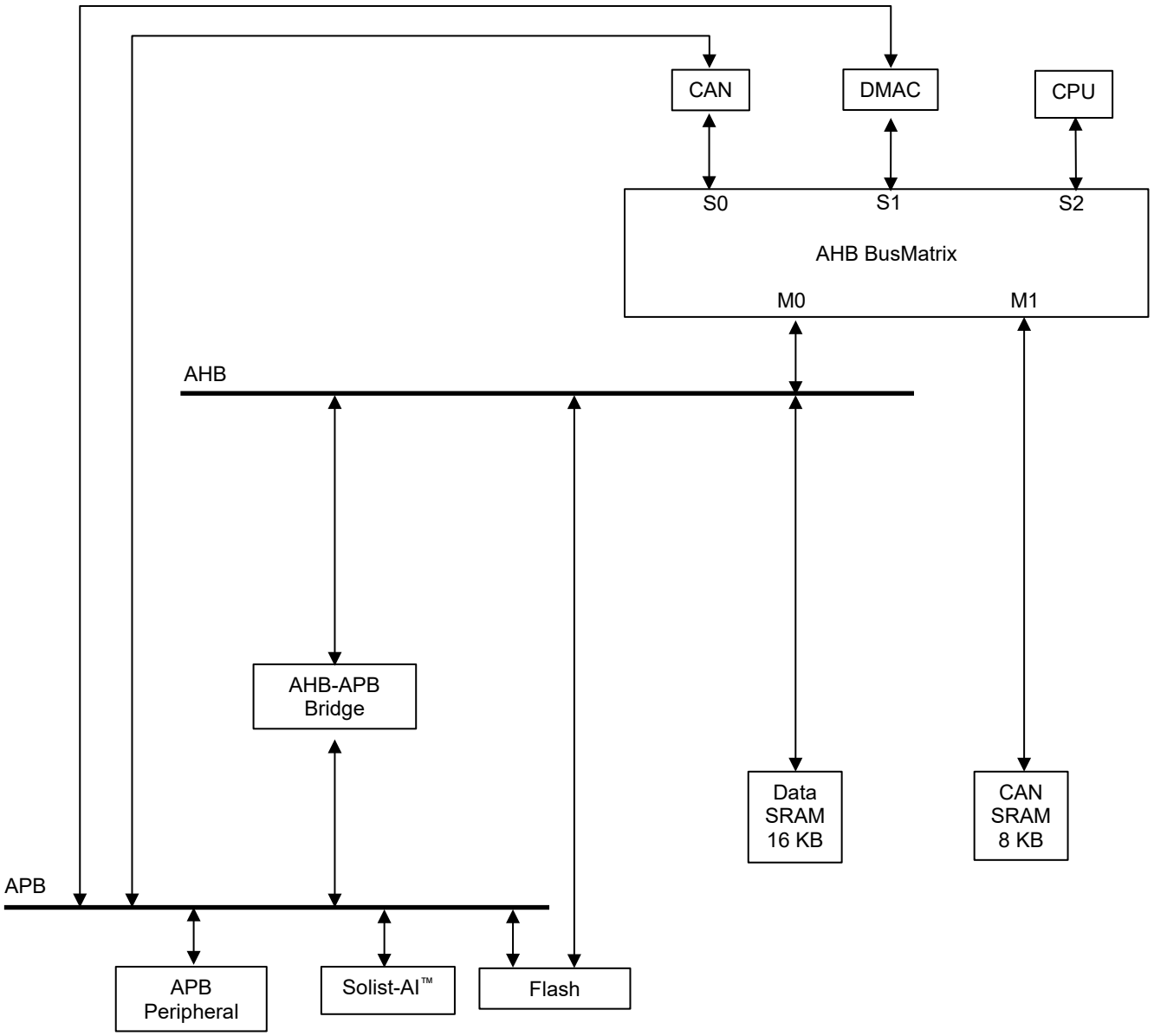


Figure 2-1 Configuration of internal bus

Chapter 3

Memory Space

3. Memory Space

3.1 Overview

In this LSI, various memories and registers are located in a 4 GB memory space, which is partitioned into 32 banks of 128 MB each.

3.2 Memory Map

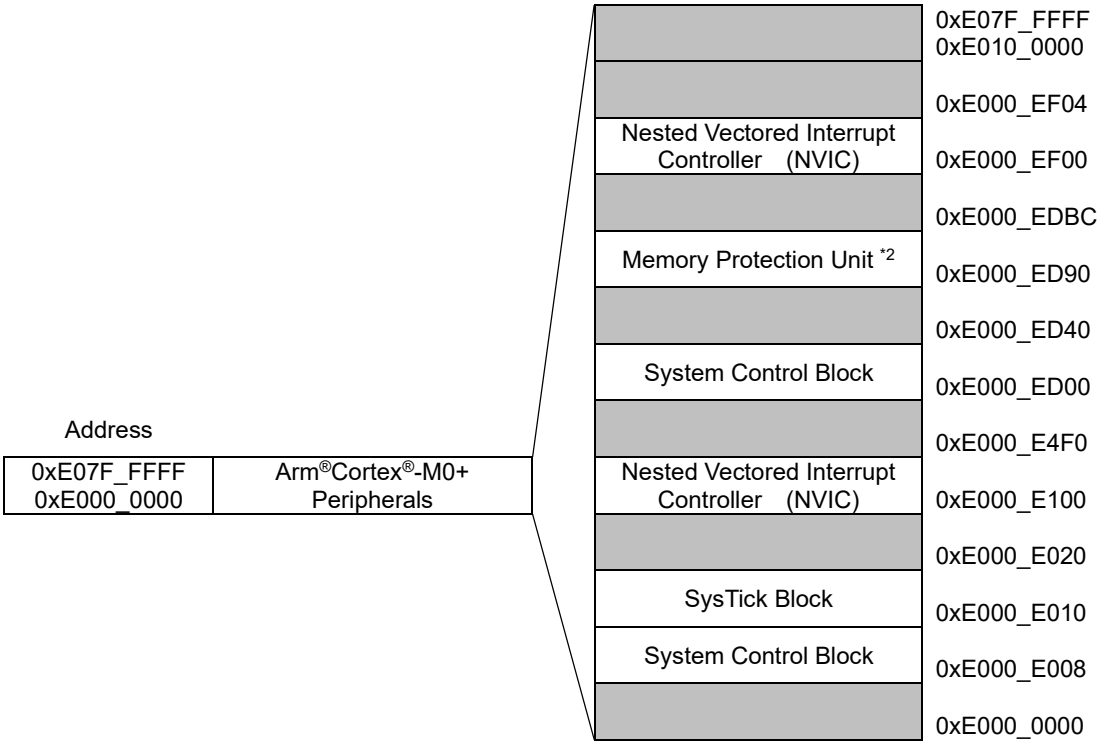
Figure 3-1 shows the memory map.

Bank	4GB	Address	Memory space	
31		0xF800_0000		
30		0xF001_0000		
		0xF000_0000	ROM Table	
29		0xE800_0000		
28		0xE080_0000		
		0xE000_0000	Arm®Cortex®-M0+ Peripherals	See Figure 3-2
27		0xD800_0000		
26		0xD000_0000		
25		0xC800_0000		
24	3GB	0xC000_0000		
23		0xB800_0000		
22		0xB000_0000		
21		0xA800_0000		
20		0xA000_0000		
19		0x9800_0000		
18		0x9000_0000		
17		0x8800_0000		
16	2GB	0x8000_0000		
15		0x7800_0000		
14		0x7000_0000		
13		0x6800_0000		
12		0x6000_0000		
11		0x5800_0000		
10		0x5000_0000		
9		0x4800_0000		
8	1GB	0x4000_0000	AHB/APB I/O	See Figure 3-3
7		0x3800_0000		
6		0x3000_0000		
5		0x2800_0000		
4		0x2000_4000		
		0x2000_0000	Work RAM (16KB)	
3		0x1800_2000		
		0x1800_0000	Data Flash (8KB)	
2		0x1004_0000		
		0x1002_0000	Flash memory program area (128KB) *2	
		0x1000_0000	Flash memory program area (128KB)	
1		0x0800_0000		
0	0GB	0x0000_0000	Remappable space	

: Reserved area *1

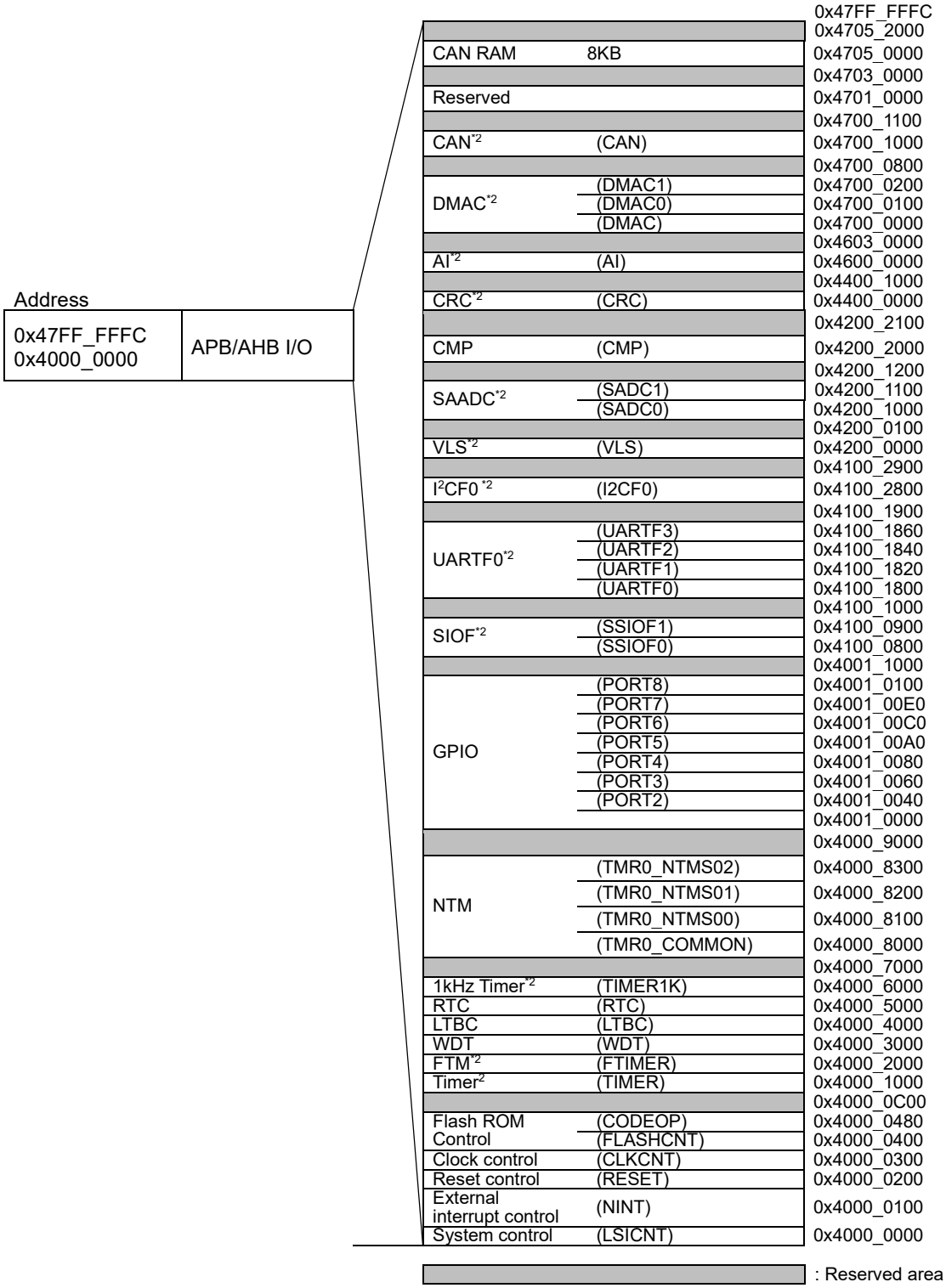
*1: Accessing any of reserved areas is prohibited. Proper operation cannot be guaranteed if accessed.
*2: ML63Q2537/ML63Q2557 only. It is reserved area in ML63Q2534/ML63Q2554.

Figure 3-1 Memory Map



*1: Accessing any of reserved areas is prohibited. Proper operation cannot be guaranteed if accessed.
*2: This LSI does not have the MPU.

Figure 3-2 Memory Map of the Arm®Cortex®-M0+ Peripherals area



*1 Accessing any of reserved areas is prohibited. Proper operation cannot be guaranteed if accessed.
*2 During individual block stop state, write data is ignored and read data is undefined. For details of individual block stop state, see Chapter 5.
( ) : The symbol of the base address is shown in parentheses.

Figure 3-3 Memory Map of the APB/AHB area

3.3 Internal Memory

3.3.1 Internal Flash Memory

Table 3-1 shows the address range of the Flash ROM of Bank2 and Bank3.
The program area contains the code option area. The code option area is 64 bytes of from address 0x1003_FFC0 to 0x1003_FFFF (ML63Q2537/ML63Q2557) or 0x1001_FFC0 to 0x1001_FFFF (ML63Q2534/ML63Q2554), and this area is not available for the program code area. For details of this area setting, see chapter 31 “Code Option”.

Table 3-1 Address Range of Bank2 and Bank3 Memory

Product	Program Area	Address range	Read Size	Programming Size
ML63Q2557 ML63Q2537	256KB	0x1000_0000 to 0x1003_FFFF *1	8/16/32 bit	32 bit
ML63Q2554 ML63Q2534	128KB	0x1000_0000 to 0x1001_FFFF *1	8/16/32 bit	32 bit

Product	Data Area	Address range	Read Size	Programming Size
ML63Q2557 ML63Q2537 ML63Q2554 ML63Q2534	8 KB	0x1800_0000 to 0x1800_1FFF *1	8/16 bit	8 bit

*1: Accessing an address out of this address range in the same bank is prohibited. Proper operation cannot be guaranteed if accessed.

3.3.2 Work RAM

Table 3-2 shows the address range of the work RAM of Bank4.

Table 3-2 Bank4 Address Range

Product	Work RAM	Address range	Access Size
ML63Q2557 ML63Q2537 ML63Q2554 ML63Q2534	16KB	0x2000_0000 to 0x2000_3FFF *1	8/16/32 bit

*1: Accessing an address out of this address range in the same bank is prohibited. Proper operation cannot be guaranteed if accessed.

3.4 Memory Controller Function

3.4.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0000	System control base address	LSICNT	-	-	-
0x10	Remapping control register	REMAPCON	R/W	32	0x0000_0000
0x14	Remapping base address register	REMAPBASE	R/W	32	(ML63Q2557, ML63Q2537) : 0x1003_E000 (ML63Q2554, ML63Q2534) : 0x1001_E000

3.4.2 Remapping Control Register (REMAPCON)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	REMAP_EN	REMAP			
R/W	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a special function register (SFR) used to set remapping.

Table 3-3 Remapping Address

REMAP_EN	REMAP	Description
0	xxxx	Based on the BRMPN pin that is set after the reset release, the following area is remapped to Bank0. H : The area of address 0x1000_0000 of Flash ROM is remapped to Bank0. L : (ML63Q2537/ML63Q2557) The area of 0x1003_E000 of Flash ROM is remapped to Bank0. (ML63Q2534/ML63Q2554) The area of 0x1001_E000 of Flash ROM is remapped to Bank0.
1	0000	The Program Flash ROM is remapped to Bank0.
1	xxx1	The work RAM is remapped to Bank0.
1	x100	The area starting at the address set by the REMAP_BASE register is remapped to Bank0.
1	Other	Setting prohibited

(x :Don't care)

[Note]

- **Operation cannot be guaranteed if remapping is performed when the remapping processing program (instruction to set the remapping control register) is placed in Bank0. Be sure to place the remapping processing program in a Bank other than Bank0 when performing the remapping. Remapping of Bank0 is performed as soon as this register is set.**
- **This register is not initialized by software reset when Remap register reset enable (REMAPRSTEN) is "0".**

3.4.3 Remapping Base Address Register (REMAPBASE)

Offset : 0x14
Initial value: 0x1003_E000 (ML63Q2557, ML63Q2537) , 0x1001_E000 (ML63Q2554, ML63Q2534)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	REMAP_BASE[29:16]													
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0/1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REMAP_BASE[15:12]				–	–	–	–	–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a special function register (SFR) used to set the base address of remapping. It is enabled only when REMAP_EN = "1" and REMAP = "x100" are set in the remapping control register.

Bit No	Bit name	Description
29 to 12	REMAP_BASE	The area starting at the address set by this register is assigned starting from address 0. The starting address must be an address in the memory space. Flash ROM : (ML63Q2557, ML63Q2537) : 0x1000_0000 to 0x1003_FFFF (ML63Q2554, ML63Q2534) : 0x1000_0000 to 0x1001_FFFF Work RAM : 0x2000_0000 to 0x2000_3FFF

[Note]

- **This register is not initialized by software reset when the remap register enable (REMAPRSTEN) is "0".**

3.4.4 Remapping Function

A Bank0 allocations is remapped by setting the BRMPN pin to L at booting. However, if the software reset is performed with REMAP_EN=1 in the REMAPCON register, the allocation by the REMAPCON register is maintained.
Remapping is performed as soon as the REMAP_EN of REMAPCON register is set to "1".
Table 3-4 shows the allocations of Bank0 during booting and remapping.
Operation cannot be guaranteed if remapping is performed when the remapping processing program (instruction to set the remapping control register) is placed in Bank0. Be sure to place the remapping processing program in a Bank other than Bank0 when performing the remapping.

Table 3-4 Bank0 Allocations during Booting and Remapping

Causes	BRMPN pin*	REMAP_ EN bit	REMAP bits *	Device	Beginning Address	Range of remapping	Remarks
Booting**	1	0	xxxx	Program Flash ROM	0x1000_0000	Bank2	-
	0				Address set in the REMAPBASE register	4KB	Wrapped around in valid area.
	x	1	xxxx	Not remapped	-	-	-
Software (REMAP_EN =1)	x	1	0000	Program Flash ROM	0x1000_0000	Bank2	-
			xxx1	Work RAM	0x2000_0000	Bank4	-
			x100	Work RAM / Program Flash ROM	Address set in the REMAPBASE register	4KB	Wrapped around in valid area.
			Other	Setting prohibited	-	-	-
-	x	0	xxxx	Not remapped	-	-	-

*: x = Don't care the data.
**: Booting with the REMAP_EN =1 is only when software reset. Because the REMAPCON register is initialized by system reset other than software reset .

3.5 Access Response for Memory Space

An access made to a bank that has been set as not allocated returns an error response (*). For specifications of the space in a bank exceeding the allocated memory size, see Section 3.3.
* Operation at error response : If an error response is returned for access from CPU, a hard fault exception is generated.

3.6 RAM Write Buffer

The work RAM has a build in one word write buffer. The writing data is stored into the write buffer at writing access to the RAM.
When the data is written to the same word address of RAM as the previous one, only the write buffer is updated.
By writing to a different word address in RAM, the data in the write buffer is written to the RAM physically.
The data in the write buffer that was last written to RAM is not preserved after the reset.

[Note]
- **When resetting by the software after writing to the RAM, write the data from the write buffer to the RAM physically by writing to an address where the word address of the RAM is different at the end of the writing. And also it is recommended to use the SRSCR register. See Chapter 4 "Reset function" for details on the SRSCR register.**

3.7 RAM Parity Error

At writing to RAM, one bit of parity is generated per byte of data and written with the data.
A flag is set in the MCUISTAT register and the MCU interrupt is notified, when a parity error is detected during RAM read if parity error interrupts are enabled in the system control function register (MCUINTEN).

[Note]
- **Since the initial value of RAM is undefined, a parity error may occur if it is read without writing it once.**

Chapter 4

Reset Function

4. Reset Function

4.1 Overview

This LSI has the 8 reset functions shown below. If any of these resets occur, this LSI enters system reset mode.

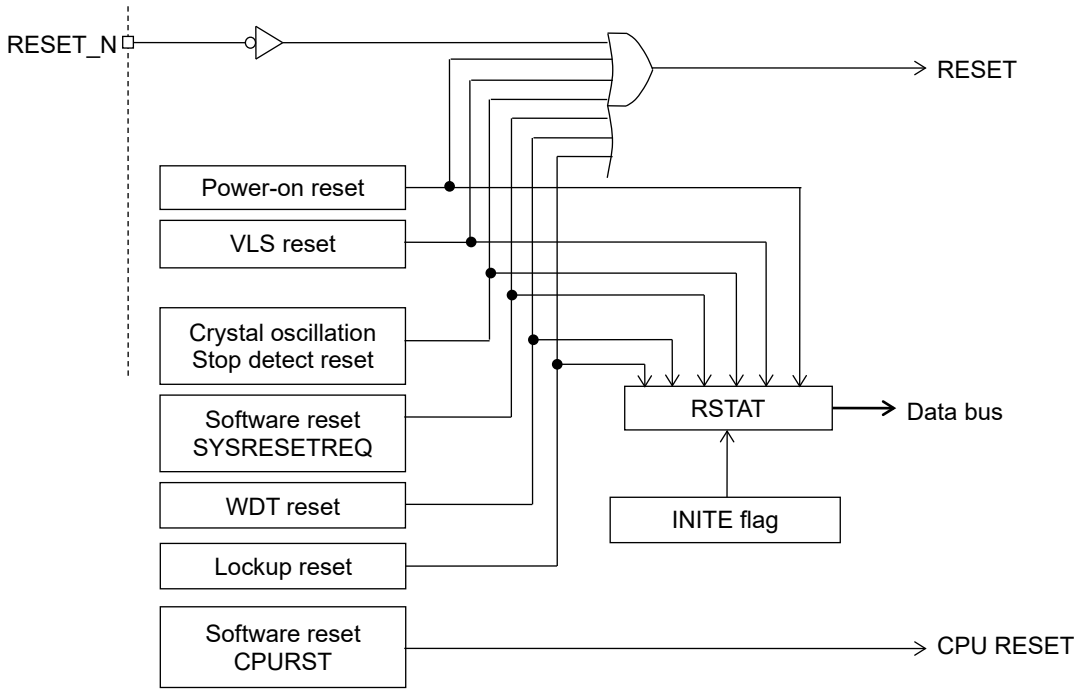
1）Reset by the RESET_N pin
2）Reset by power-on detection
3）Reset by the 2nd overflow of watchdog timer (WDT)
4）Reset by Voltage Level Supervisor (VLS)
5）Reset by the crystal oscillation stop detection
6）Reset by Lockup of processor
7）Software reset by the SYSRESETREQ bit
8）Software reset by the CPURST register

4.1.1 Features

- 125ms, 500ms, 2s, or 8s can be selected as the watchdog timer (WDT) overflow period.
- Built-in reset status register (RSTAT) indicating the reset generation causes
- It has the INITE flag function to detect abnormal start-up of the LSI.

4.1.2 Configuration

Figure 4-1 shows the configuration of the reset generation circuit.



RSTAT : Reset status register

Figure 4-1 Configuration of Reset Generation Circuit

4.1.1. List of Pins

Pin name	I/O	Description
RESET_N	I	Reset input pin

4.2 Description of Registers

4.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0200	Reset control base address	RESET	-	-	-
0x00	Reset status register	RSTAT	R/W	32	Undefined
0x04	LOCKUP reset setting register	LOCKUPEN	R/W	32	0x0000_0000
0x08	CPU reset register	CPURST	W	32	0x0000_0000
0x0C	Software reset scratch register	SRSCR	R/W	32	0x0000_0000
0x10	REMAP register reset enable	REMAPRSTEN	R/W	32	0x0000_0000

4.2.2 Reset Status Register (RSTAT)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	INITE	–	LOCKUP	–	VLSR	WDTR	XSTR	POR
R/W	–	–	–	–	–	–	–	–	R	–	R/W	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x

*) Initial value depends on the reset causes.

RSTAT is a special function register (SFR) that indicates the cause of a switch to the system reset mode.
At switch to the system reset mode, the bit for the reset cause is set to "1". RSTAT is not initialized during the system reset mode. The reset flags except INITE bit are cleared when this register is written any value.
When checking the reset cause using this function, clear each reset cause bit of RSTAT to "0" (not occurred) in advance.

Bit No.	Bit name	Description
7	INITE	This bit is a read-only bit to indicate that an abnormality occurred in starting LSI. If this bit is set to "1", restart the LSI by causing a reset to occur with either system reset except the CPURST. 0: LSI started-up normally 1: Abnormality occurred in start-up of LSI
5	LOCKUP	This bit is a flag that indicates that the processor fell in the lockup state. This bit is set to "1" when a lockup state occurs. This bit is set to "0" when the power is turned on. This bit is enabled only when the EN bit of the LOCKUPEN register is "1". 0: Lockup state not occurred 1: Lockup state occurred
3	VLSR	This bit is a flag that indicates that the voltage level supervisor reset (VLS reset) is occurred. This bit is set to “1” when the reset the voltage level detect circuit (VLS) generated. Also, the bit is undefined when the power is turned on. 0: VLS reset not occurred 1: VLS reset occurred
2	WDTR	This bit is a flag that indicates that the watchdog timer reset (WDT reset) is occurred. This bit is set to “1” when the reset by overflow of the watchdog timer is generated. Also, the bit is undefined when the power is turned on. 0: WDT reset not occurred 1: WDT reset occurred
1	XSTR	This bit is a flag that indicates the occurrence of crystal oscillation stop detect reset. When low-speed or high-speed crystal oscillation stops for the period specified by each crystal oscillation stop detection time or longer, this bit is set to “1”. The low-speed crystal oscillation stop reset is enabled by setting the LSTPSEL bit of FCON23 register to “00” and XSPEN bit of FCON01 register to “1”. The high-speed crystal oscillation stop reset is enabled by setting the HSTPSEL bit of FCON89 register to “00” and HXSPEN bit of FCON01 register to “1”. The FSTAT2 register indicates whether a low-speed oscillation or high-speed oscillation causes. See Chapter 6 for the FCON23/FCON89/FSTAT2 registers. Also, the bit is undefined when the power is turned on. 0: Crystal oscillation stop detect reset not occurred 1: Crystal oscillation stop detect reset occurred
0	POR	This bit is a flag that indicates that the power-on reset is generated. This bit is set to “1” when powered on. 0: Power-on reset not occurred 1: Power-on reset occurred

[Note]
- **No flag is provided that indicates the occurrence of reset by the RESET_N pin or software reset.**

4.2.3 LOCKUP Reset Setting Register (LOCKUPEN)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	EN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LOCKUPEN is a special function register (SFR) to enable/disable the LOCKUP reset.

Bit No.	Bit name	Description
0	EN	Sets whether to enable/disable the LOCKUP reset. This register is initialized by some system-reset included a LOCKUP reset. 0: Disabled (Initial value) 1: Enabled

4.2.4 CPU Reset Register (CPURST)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RST
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPURST is a special function register (SFR) to software reset.

Bit No.	Bit name	Description
0	RST	When set "1" to this bit, only the CPU is initialized and others are not initialized. See “4.3.2 Operation of System Reset Mode” for detail. This is write-only bit. When this bit is read, the read value is always "0". 0: writing is invalid. 1: The CPU is initialized.

4.2.5 Software Reset Scratch Register (SRSCR)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	d							
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRSCR is a special function register (SFR) that is not initialized when SYSRESETREQ or CPURST is occurred. This register is initialized by the other reset causes. This register can be used to identify software reset.

4.2.6 REMAP Register Reset Enable (REMAPRSTEN)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	EN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

REMAPRSTEN is a special function register (SFR) to control remap registers.

Bit No.	Bit name	Description
0	EN	When set this bit to "1", the REMAPCON and REMAPBASE register are initialized by the software reset (SYSRESETREQ or CPURST). 0: REMAPCON and REMAPBASE register are not initialized by the software reset (Initial value) 1: REMAPCON and REMAPBASE register are initialized by the software reset.

4.3 Description of Operation

4.3.1 Cause of Reset

In addition to a reset by RESET_N pin and software reset by the reset control register, this LSI enters the system reset mode according to the state in the LSI.

- **Reset by the RESET_N pin**

System reset occurs when “0” is input to RESET_N. The contents of reset status register (RSTAT) are not changed at system reset mode entered by the RESET_N pin.

- **Reset by power-on detection**

System reset occurs when LSI is powered on. POR bit of reset status register (RSTAT) becomes “1” at system reset mode entered by Power-on detection.

- **Reset by the overflow of watchdog timer (WDT)**

System reset occurs by the second WDT overflow. For the watchdog timer operation, see Chapter 14 "Watchdog Timer". WDTR bit of reset status register (RSTAT) is set to “1” at system reset occurred by WDT overflow.

- **Reset by Voltage Level Supervisor (VLS)**

System reset occurs when power supply voltage falls under the specified voltage level (VLS). The reset by VLS is disabled initially. To enable it, set the VLSSEL0 bit of the voltage level supervisor mode register (VLSMOD) to "1". For operations of the VLS function, see Chapter 30 "Voltage Level Supervisor". VLSR bit of the reset status register (RSTAT) is set to “1” at system reset occurred by VLS.

- **Reset by the crystal oscillation stop detection**

System reset occurs, when the low-speed crystal oscillation stops for the period specified by the low-speed oscillation stop detection time or longer, or when the high-speed crystal oscillation stops for the period specified by the high-speed oscillation stop detection time or longer. This function is disabled initially. At the time of the shift to a system reset mode by the low-speed or high-speed crystal oscillation stop detection, XSTR bit of the reset status register (RSTAT) is set to "1".
For the reset function by crystal oscillation stop detection, see Chapter 6 "Clock Generation Circuit".

- **Software reset by SYSRESETREQ**

System reset occurs by software.
The contents of reset status register (RSTAT) are not changed at system reset mode entered by SYSRESETREQ. For details, refer to "Cortex®-M0+ Devices Generic User Guide".

- **Reset by lockup of processor**

The reset occurs when the processor enters the lockup state. For details, refer to "Cortex®-M0+ Devices Generic User Guide"

- **Software reset by CPURST**

The reset occurs by software.
Only the CPU is initialized and others are not initialized.
The contents of reset status register (RSTAT) are not changed at system reset mode entered by CPURST.

4.3.2 Operation of System Reset Mode

System reset has the highest priority among all the processing and any other processing being executed up to then is cancelled.
In system reset mode, the following processing is performed.

1) The power circuit is initialized.
2) All the special function registers (SFRs) whose initial value is not undefined are initialized. However, SFRs of RTC are not initialized by except for the reset by the POR. See Appendix A "Registers" for the initial values of the SFRs.
3) CPU is initialized. All the registers in CPU are initialized.

[Note]

- **In system reset mode, the contents of data memory and those of any SFR whose initial value is undefined are not initialized and are undefined. Initialize them by software.**

The following table shows the causes of switch to the system reset mode and the initialization state in the LSI.

Table 4-1 Reset Range

(I :Initialized, N : Not initialized)

Causes / Module	RESET_N	POR	SYSRESETREQ	LOCKUP	CPURST	VLS	WDT	XTSTOP
SFR	I	I	I	I	N	I	I	I
SFR: SRSCR	I	I	N	I	N	I	I	I
SFR: REMAPCON, REMAPBASE	I	I	I *4	I	I *4	I	I	I
CPU	I	I	I	I	I	I	I	I
Data memory*3	N	N	N	N	N	N	N	N
RTC *1	N	I	N	N	N	N	N	N
Crystal oscillation circuit	I	I	I	I	N	I	I	I
VLS enable*2	I	I	I	I	N	N	I	I

*1: It is included SFRs in the RTC module and the SFRs related to RTC; LTBRR and LTBADJ. See to Chapter 9 and 12.
*2: See below for the initialization state at power-up and VLS reset assert.

Signal	At power up	VLS reset	Other than VLS reset
VLS enable (VLS0EN bit of the VLSCON register)	Disabled	Enabled	Disabled
VLS clock enable (CVLS bit of CLKCON register)	Disabled	Enabled	Disabled

The VLS0EN bit of the VLSCON register and the CVLS bit of the CLKCON register select enable/disable of the respective function. These are disabled when the power is turned on, but the state before the reset is retained only when a VLS reset occurs.
*3: The write buffer of RAM is cleared by system reset. See "3.6 RAM Write Buffer" for detail.
*4: It is not initialized when REMAPRSTEN register is 0.

Chapter5

System Control Function

5. System Control Function

5.1 Overview

This LSI has two power management modes listed below to save the current consumption. These power management modes are set by the Arm®Cortex®M0+ sleep mode and the standby control register (SBYCON).

1）HALT mode
2）STOP mode

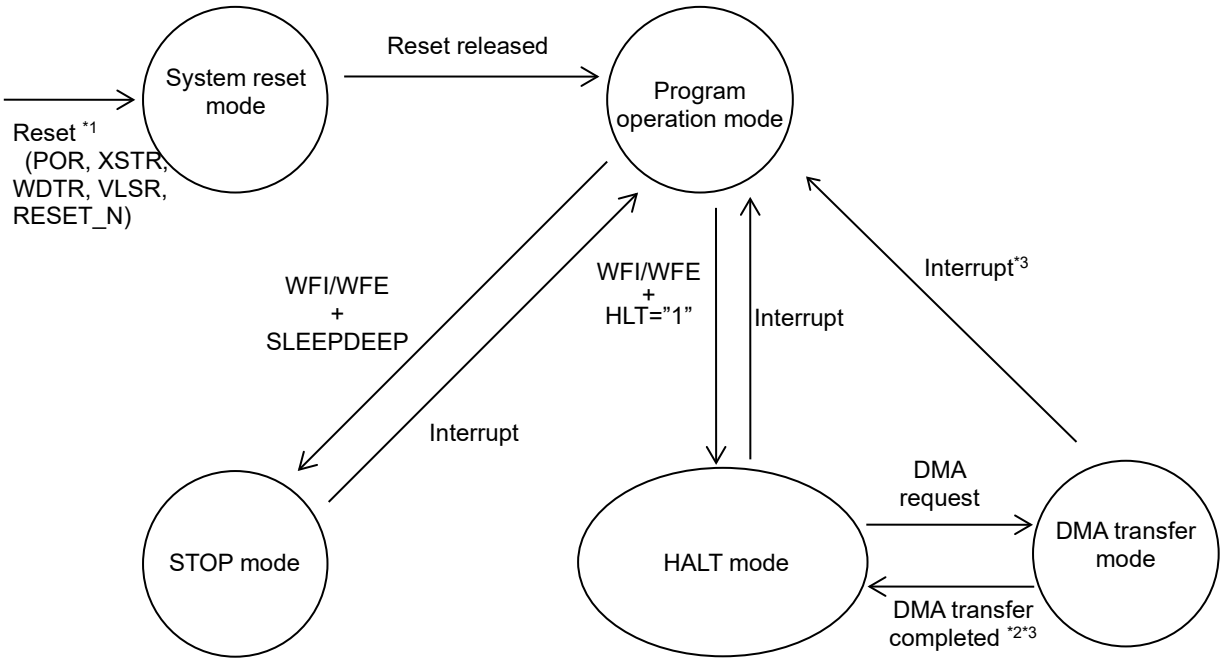
It also has a block control function, which power downs unused peripheral functions (reset registers and stop clock supplies) to further reduce the current consumption.
Also a regulator that generates internal voltage to reduce the power consumption inside the LSI is built-in.

5.1.1 Features

- HALT mode, where the CPU stops operating and only the peripheral circuit is operating
- STOP mode, which both low-speed oscillation and high-speed oscillation stop.
- Block control function, which power downs the circuits of unused function blocks (reset registers and stop clock supplies)

5.1.2 Configuration

Figure 5-1 shows an operating state transition diagram.



*1: At reset, the LSI switches to the system reset mode from any mode.
*2: After DMA transfer, the LSI switches to the HALT mode after waiting for 8 cycles of the system clock.
*3: When switching from the DMA transfer mode, the interrupt has a priority over the DMA transfer completion.

Figure 5-1 Operating State Transition Diagram

Figure5-2 shows the configuration of power source circuit.

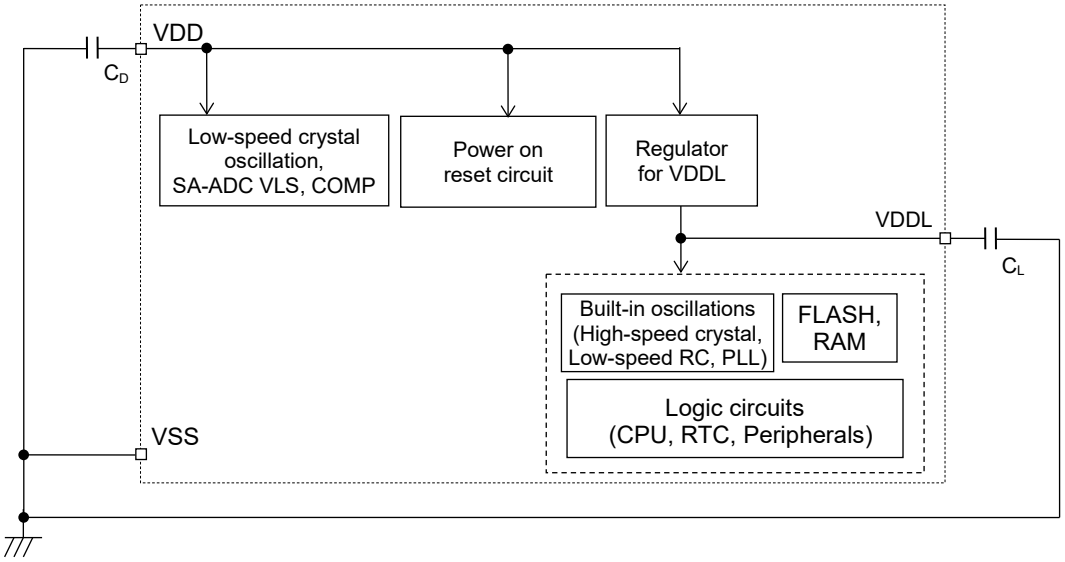


Figure5-2 Configuration of the power supply circuit

5.1.3 List of Pins

Connect VDDL pin and VDD pin to capacitors (C_L, C_D) between these pins and VSS pin to stabilize V_{DDL} and V_{DD} to improve the noise resistance.

Table 5-1 Pins

Pin name	I/O	Description
VDDL	—	Power supply for internal circuits (Generate internally)
VDD	—	Positive power supply
VSS	—	Negative power supply

[Note]

- **Place the capacitors (C_L, C_D) in the vicinity of LSI on the user board and wire them as short as possible avoiding via holes.**
- **The power supply for internal circuits (V_{DDL}) is unavailable to use for external devices.**

5.2 Description of Registers

5.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0000	System control base address	LSICNT	-	-	-
0x00	Revision register	IDR	R	32	0x0632_5XX0
0x04	DMA request select register	DREQSEL	R/W	32	0x0000_0000
0x08	BRMPN control register	BRMPNCON	R/W	32	0x0000_0001
0x0C	Unique ID register	UIDR	R	32	0xXXXX_XXXX
0x40	Flash Wait register	FLAWAIT	R/W	32	0x0000_0002
0x44	Standby control register	SBYCON	R/W	32	0x0000_0000
0x48	Reserved	-	-	-	-
0x4C	Clock control register	CLKCON	R/W	32	0x005C_0000
0x50	Reset control register	RSTCON	R/W	32	0xF4FF_F33F
0x54	Reserved	-	-	-	-
0x58	Reserved	-	-	-	-
0x5C	MCU status interrupt enable register	MCUINTEN	R/W	32	0x0000_0000
0x60	MCU status interrupt status register	MCUISTAT	R/W	32	0x0000_0000

5.2.2 Revision Register (IDR)

Offset : 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PID[27:12]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PID[11:0]												PRV			
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	1	0	1	x	x	x	1	0	1	x	x	0	0	0	0

IDR holds the 28-bit product ID and 4-bit revision.

Bit No	Bit name	Description
31 to 4	PID	This is indicated the product ID of this LSI. ML63Q2557 : 0x0632557 ML63Q2554 : 0x0632554 ML63Q2537 : 0x0632537 ML63Q2534 : 0x0632534
3 to 0	PRV	This is indicated the revision of this LSI.

5.2.3 DMA Request Select Register (DREQSEL)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	DREQ1SEL					–	–	–	DREQ0SEL				
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DREQSEL is a register used to select the external request of DMAC.

Bit No	Bit name	Description
12 to 8	DREQ1SEL	This is used to select the peripheral which uses channel 1 at transferring external requests of DMAC. One of peripherals shown in Table5-2 is selected by the DREQ1SEL setting.
4 to 0	DREQ0SEL	This is used to select the peripheral which uses channel 0 at transferring external requests of DMAC. One of peripherals shown in Table5-2 is selected by the DREQ0SEL setting.

Table5-2 List of Request

Setting value	Selected peripheral
00001	SSIOF0 (RX)
00010	SSIOF0 (TX)
00011	SSIOF1 (RX)
00100	SSIOF1 (TX)
00101	UARTF1 (RX)
00110	UARTF1 (TX)
00111	UARTF0 (RX)
01000	UARTF0 (TX)
01001	I2CF0 (RX)
01010	I2CF0 (TX)
01011	SA-ADC1
01100	SA-ADC0
01101	UARTF2 (RX)
01110	UARTF2 (TX)
01111	UARTF3 (RX)
10000	UARTF3 (TX)
Others	Not selected

5.2.4 BRMPN Control Register (BRMPNCON)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	PUEN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

BRMPNCON register is used to control the pull-up resistor of the BRMPN pin.

Bit No	Bit name	Description
0	PUEN	This bit is used to set the pull-up resistor of the BRMPN pin. 0: Disabled 1: Enabled (Initial value)

5.2.5 Unique ID Register (UIDR)

Offset : 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UID[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UID[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

UIDR holds the 32-bit unique ID per product.

5.2.6 Flash Wait Register (FLAWAIT)

Offset : 0x40

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	FLACC	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

FLAWAIT is a special function register (SFR) to set read cycle of Flash ROM when the system clock is selected as a high-speed clock.

Bit No	Bit name	Description
1 to 0	FLACC	This is used for setting read cycle of Flash ROM. This bit can be set only when the SYSCLK bit in FCON01 register is “0”. When the system clock is LSCLK, A read cycle of Flash ROM is fixed 1 cycle access. 00: 1 cycle access [SYSCLK frequency < 10MHz] 01: 2 cycles access [SYSCLK frequency < 30MHz] 10: 3 cycles access (Initial value) 11: Setting prohibited A description in [ ] indicate the setting conditions. The frequency of the system clock is adjusted by hardware to operate at a safe frequency. See the description of SYSC bit in section “6.2.2 Frequency Control Register 01 (FCON01)” for detail.

5.2.7 Standby Control Register (SBYCON)

Offset : 0x44

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	HLT
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SBYCON is a special function register (SFR) to control the operation mode of MCU.

Bit No	Bit name	Description
0	HLT	The HLT bit is used for setting the HALT mode. When the HLT bit is set to "1", and the WFI/WFE instruction is executed, the LSI switches to the HALT mode. 0: Disabled (Initial value) 1: Enabled

[Note]

- **It can switch to the STOP, HALT mode if a valid interrupt occurs. It returns from such a mode when another interrupt occurs which has a higher interrupt level than the valid interrupt.**
- **When this register is 0x0000_0000, and the CPU is operated the instruction of WFI/WFE, only the CPU is shifted sleep mode.**

5.2.8 Clock Control Register (CLKCON)

Offset : 0x4C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCRC	CAI	CDMAC	CCMP	CVLS	CRTC	–	–	CCAN	CUAF3	CI2CF0	CUAF2	CUAF1	CUAF0	CSIOF1	CSIOF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTM1K	CSAD1	CSAD	CNTMS	–	–	CFTM1	CFTM0	–	–	CTM5	CTM4	CTM3	CTM2	CTM1	CTM0
R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CLKCON is a special function register (SFR) used to control the clock of each peripherals.

[Description of each bits]
This is used to control supplying clocks to the corresponding peripheral circuit.
0: Stop clock to the corresponding peripheral circuit.
1: Supplied clock to the corresponding peripheral circuit.

Bit No	Bit name	Description (corresponding peripheral circuit)
31	CCRC	CRC
30	CAI	Solist-AI™ Accelerator
29	CDMAC	DMA controller
28	CCMP	Analog comparator
27	CVLS	Power supply voltage detection circuit (VLS) This bit is not initialized by VLS reset.
26	CRTC	Real time clock This bit is initialized by the power-on reset. It is not initialized by other reset causes.
25	-	Reserved bit
24	-	Reserved bit
23	CCAN	CAN FD interface
22	CUAF3	UART with FIFO 3
21	CI2CF0	I^2C bus interface with FIFO 0
20	CUAF2	UART with FIFO 2
19	CUAF1	UART with FIFO 1
18	CUAF0	UART with FIFO 0
17	CSIOF1	Synchronous serial port with FIFO 1
16	CSIOF0	Synchronous serial port with FIFO 0
15	CTM1K	1kHz timer
14	CSAD1	Successive approximation type (SA type) A/D converter 1
13	CSAD	Successive approximation type (SA type) A/D converter 0
12	CNTMS	Three-phase motor control PWM
11	-	Reserved bit
10	-	Reserved bit
9	CFTM1	Functional timer 1
8	CFTM0	Functional timer 0
7	-	Reserved bit
6	-	Reserved bit
5	CTM5	Timer 5 Timer 4 and Timer 5 are related. See Table 5-3.

Bit No	Bit name	Description (corresponding peripheral circuit)
4	CTM4	Timer 4 Timer 4 and Timer 5 are related. See Table 5-3.
3	CTM3	Timer 3 Timer 2 and Timer 3 are related. See Table 5-3.
2	CTM2	Timer 2 Timer 2 and Timer 3 are related. See Table 5-3.
1	CTM1	Timer 1 Timer 0 and Timer 1 are related. See Table 5-3.
0	CTM0	Timer 0 Timer 0 and Timer 1 are related. See Table 5-3.

As shown in Table 5-3, the timer is related to the configuration of two channels.

Table 5-3 Relation between setting and clock

CTMm	CTMn	Timer m			Timer n		
		SYSCLK	LSCLK	OSCLK	SYSCLK	LSCLK	OSCLK
0	0	Stop	Stop	Stop	Stop	Stop	Stop
0	1	Supply	Stop	Stop	Supply	Supply	Stop
1	0	Supply	Supply	Stop	Supply	Stop	Stop
1	1	Supply	Supply	Supply	Supply	Supply	Supply

Where is, “m and n” are combination of “5 and 4”, “3 and 2”, “1 and 0”.

[Note]

Setting any flag to "0" (disable operation) stops the clock supply of the applicable block. When the flag is set to "0", writing to the registers on the block is disabled.
To use the function of the block, follow the procedure below:
(1) Reset the applicable flag of the clock control register to "1" (clock supply).
(2) Reset the applicable flag of the reset control register to "0" (reset disabled).

To stop the function of the block, follow the procedure below:
(1) Set the applicable flag of the reset control register to "1" (reset enabled).
(2) Set the applicable flag of the clock control register to "0" (clock stop).

5.2.9 Reset Control Register (RSTCON)

Offset : 0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RCRC	RAI	RDMAC	RCMP	–	RRTC	–	–	RCAN	RUAF3	RI2CF0	RUAF2	RUAF1	RUAF0	RSIOF1	RSIOF0
R/W	R/W	R/W	R/W	R/W	–	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTM1K	RSAD1	RSAD	RNTMS	–	–	RFTM1	RFTM0	–	–	RTM5	RTM4	RTM3	RTM2	RTM1	RTM0
R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	1

RSTCON is a special function register (SFR) used to control the reset of each block.

[Description of each bits]
This is used to control reset of the corresponding peripheral circuit.
0: Disabled
1: Enabled (Initial value)

Bit No	Bit name	Description (corresponding peripheral circuit)
31	RCRC	CRC
30	RAI	Solist-AI™ Accelerator
29	RDMAC	DMA controller
28	RCMP	Analog comparator
27	-	Reserved bit
26	RRTC	Real time clock This bit is initialized by the power-on reset. It is not initialized by other reset causes.
25	-	Reserved bit
24	-	Reserved bit
23	RCAN	CAN FD interface
22	RUAF3	UART with FIFO 3
21	RI2CF0	I^2C bus interface with FIFO 0
20	RUAF2	UART with FIFO 2
19	RUAF1	UART with FIFO 1
18	RUAF0	UART with FIFO 0
17	RSIOF1	Synchronous serial port with FIFO 1
16	RSIOF0	Synchronous serial port with FIFO 0
15	RTM1K	1kHz timer
14	RSAD1	Successive approximation type (SA type) A/D converter 1
13	RSAD	Successive approximation type (SA type) A/D converter 0
12	RNTMS	Three-phase motor control PWM
11	-	Reserved bit
10	-	Reserved bit
9	RFTM1	Functional timer 1
8	RFTM0	Functional timer 0
7	-	Reserved bit
6	-	Reserved bit
5	RTM5	Timer 5
4	RTM4	Timer 4
3	RTM3	Timer 3

Bit No	Bit name	Description (corresponding peripheral circuit)
2	RTM2	Timer 2
1	RTM1	Timer 1
0	RTM0	Timer 0

[Note]

- **Setting any flag to "1"(disable operation) enables the reset of the applicable block. When the flag is set to "1", writing to the registers on the block is disabled.**
  To use the function of the block, follow the procedure below:
  (1) Reset the applicable flag of the clock control register to "1" (clock supply).
  (2) Reset the applicable flag of the reset control register to "0" (reset disabled).

  To stop the function of the block, follow the procedure below:
  (1) Set the applicable flag of the reset control register to "1" (reset enabled).
  (2) Set the applicable flag of the clock control register to "0" (clock stop).

5.2.10 Reserved

Offset : 0x54

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	rsvd							
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is Reserved. Don't write anything.

5.2.11 Reserved

Offset : 0x58

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	rsvd
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is Reserved. Don't write anything.

5.2.12 MCU Status Interrupt Enable Register (MCUINTEN)

Offset : 0x5C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	CAN_BUS_ERR_EN	–	CAN_PAR_EN	AI_PAR_EN	RAM_PAR_EN	–	–	–	FLCINT_EN
R/W	–	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCUINTEN is a SFR used to enable each request of MCU status interrupt.

[Description of each bits]
This is used to control request of the corresponding interrupt.
0: Disabled (Initial value)
1: Enabled

Bit No	Bit name	Description (Corresponding interrupt)
8	CAN_BUS_ERR_EN	An interrupt when bus error occurs by the CAN module accesses to outside of CAN RAM.
6	CAN_PAR_EN	A parity error interrupt for CAN RAM
5	AI_PAR_EN	A parity error interrupt for AI RAM
4	RAM_PAR_EN	A parity error interrupt for work RAM
0	FLCINT_EN	Data flash Erasing/Programming completion interrupt

[Note]

- **Initial values of the RAM area are undefined. A RAM parity error may occur if the parity error interrupt is enabled without RAM initialization. Initialize RAM area before RAM parity error interrupts are enabled by this register.**

5.2.13 MCU Status Interrupt Status Register (MCUISTAT)

Offset : 0x60

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	CAN_BUS_ERR	–	CAN_PAR	AI_PAR	RAM_PAR	–	–	–	FLCINT
R/W	–	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCUISTAT is a SFR used to control each request of MCU status interrupt.

[Description of each bits]
This bit indicates the request of the corresponding interrupt. Writing “1” clears this bit to “0”.
0: Not interrupt request (Initial value)
1: Occurred interrupt request

Bit No	Bit name	Description (Corresponding interrupt)
8	CAN_BUS_ERR	An interrupt when bus error occurs by the CAN module accesses to outside of CAN RAM. This interrupt is requested while this bit is “1”.
6	CAN_PAR	A parity error interrupt for CAN RAM It may be happened during the CPU, DMAC or CAN read to CAN RAM. This interrupt is requested while this bit is “1”.
5	AI_PAR	A parity error interrupt for AI RAM It may be happened during the Solist-AI module reads to AI RAM. This interrupt is requested while this bit is “1”.
4	RAM_PAR	A parity error interrupt for work RAM It may be happened during the CPU, DMAC read to work RAM or AI RAM. This interrupt is requested while this bit is “1”.
0	FLCINT	Data flash Erasing/Programming completion interrupt This interrupt request occurs only once.

5.3 Description of Operation

5.3.1 HALT Mode

During the HALT mode, the CPU interrupts execution of instructions and only the peripheral circuits are running.
When the HLT bit of the standby control register (SBYCON) is set to "1", and the WFI/WFE instruction is executed, the LSI switches to the HALT mode.
When a valid interrupt is occurred, the HALT mode is released, returning to the program run mode.
When a DMA request to DMAC is occurred, the HALT mode is released, switching to the DMA mode. The LSI switches to the HALT mode after DMA transfer is finished.
Figure5-2 shows the operation waveforms in the HALT mode.

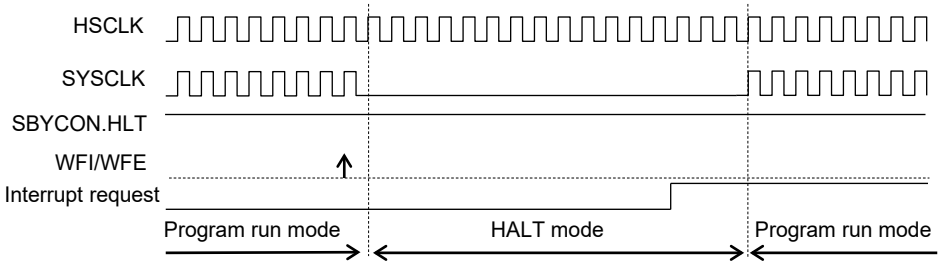


Figure5-2-1 Operation Waveforms in HALT Mode (FCON01.SYSCLK=1, FCON01.ENPLL=1)

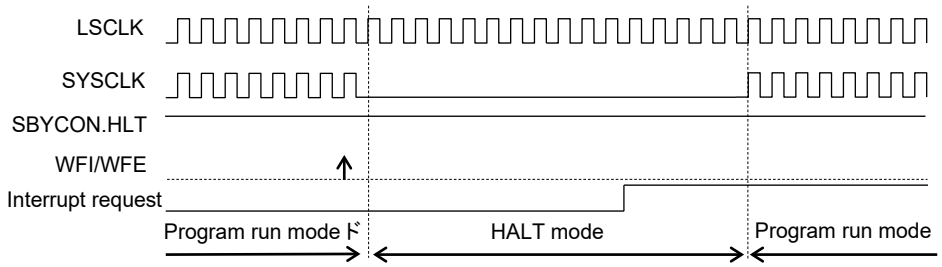


Figure5-2-2 Operation Waveforms in HALT Mode (FCON01.SYSCLK=0, FCON01.ENPLL=0)

5.3.2 STOP Mode

In the STOP mode, the low-speed oscillation and high-speed oscillation stop, and the CPU and peripheral circuits stop the operation.
When the SLEEPDEEP bit of the SCR register is set to "1", and the WFI/WFE instruction is executed, the LSI switches to the STOP mode. When a valid external pin interrupt request is generated, the STOP mode is released and the program run mode is returned after the wakeup time (T_{RTLS}).

[Note]

- **After the STOP mode is released, a valid interrupt processing is started.**

When the LSI switches to the STOP mode, the low-speed and high-speed oscillations stop. When a valid external pin interrupt request is generated in the STOP mode, it is released, restarting the low-speed oscillation. Even if the high-speed clock was oscillating before switching to the STOP mode, it does not start.
Regardless of the clock mode, the low-speed internal RC oscillation starts after an interrupt request is generated, and it starts supplying the low-speed internal RC clock to LSCLK.
Figure 5-3 shows the operation waveforms in the STOP mode when CPU operates with the low-speed clock.

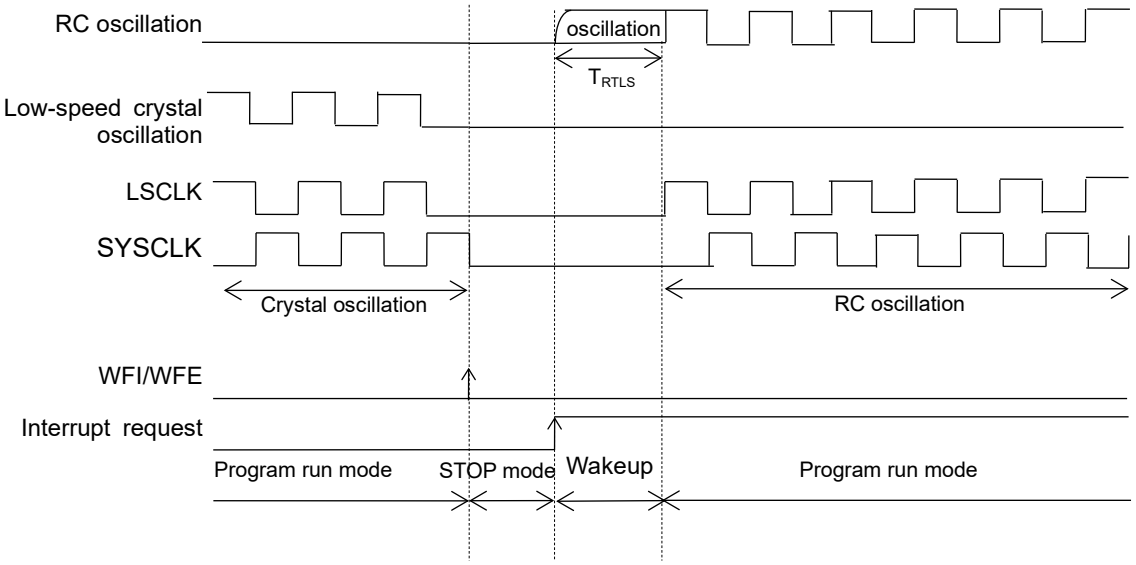


Figure 5-3 Operation Waveforms in STOP Mode When CPU Operates with Low-Speed Clock

When the LSI switches to the STOP mode with that the system clock is HSCLK, it returns to the program run mode with that the system clock is LSCLK. To use the system clock in the PLL oscillation mode after returning to the STOP mode, set ENPLL bit = "1" by software and wait for LPLL bit ="1", then switch to the mode from software.
Figure 5-4 shows the operation waveforms in the STOP mode with the PLL oscillation.

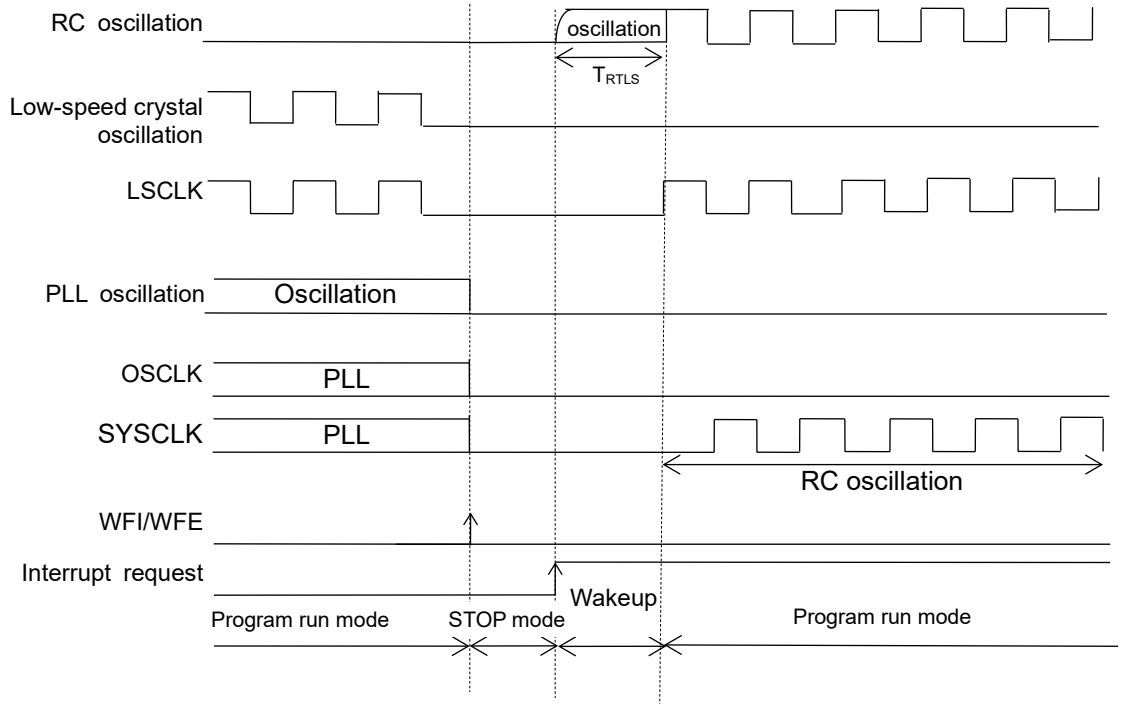


Figure 5-4 Operation Waveforms in STOP Mode With PLL Oscillation

5.3.3 Operation of Functions in STOP/HALT Mode

Table 5-4 shows the states of the functions in each of the STOP, HALT modes.

Table 5-4 State of Functions in STOP/HALT Modes
R : Operation A : Operable N : Not operable

Function	HALT	STOP
CPU	N	N
RAM	Retain	Retain
DMAC	A	N
LTBC	R	N
16-bit timer	A	N
Functional timer	A	N
RTC	A	N
1kHz timer	A	N
Watchdog timer	R	N
Three-phase motor control PWM	A *1	N
SSIO with FIFO	A *1	N
UART with FIFO	A *1	N
CAN	A *1	N
I^2C with FIFO	A *1	N
External interrupt	A	A
CRC	N	N
AI	A	N
SA-ADC	A	N
Analog comparator	A	A
Flash (BGO)	A *2	N
VLS	A	A

*1 : Operable only when the high-speed clock is active.
*2 : Operable only when the system clock is HSCLK.

5.3.4 Block Control Function

This LSI has a block control function, which stops clocks and completely turns operating circuits of unused peripherals off to make even more reducing current consumption.
For clock control register, the initial value of each flag is “0”, meaning the clock of each block is stopped. The initial value of the reset control register is “1”, meaning the reset of each block is enabled.

To use the function of the block, follow the procedure below:
(1) Set the applicable flag of the clock control register to “1” (clock supply).
(2) Set the applicable flag of the reset control register to “0” (reset disabled).

To stop the function of the block, follow the procedure below:
(1) Set the applicable flag of the reset control register to “1” (reset enabled).
(2) Set the applicable flag of the clock control register to “0” (clock stop).

5.3.5 Internal Power Supply Voltage (V_{DDL})

The built-in regulator generates approx. 1.6V as V_{DDL} to reduce power consumption.

Chapter 6

Clock Generation Circuit

6. Clock Generation Circuit

6.1 Overview

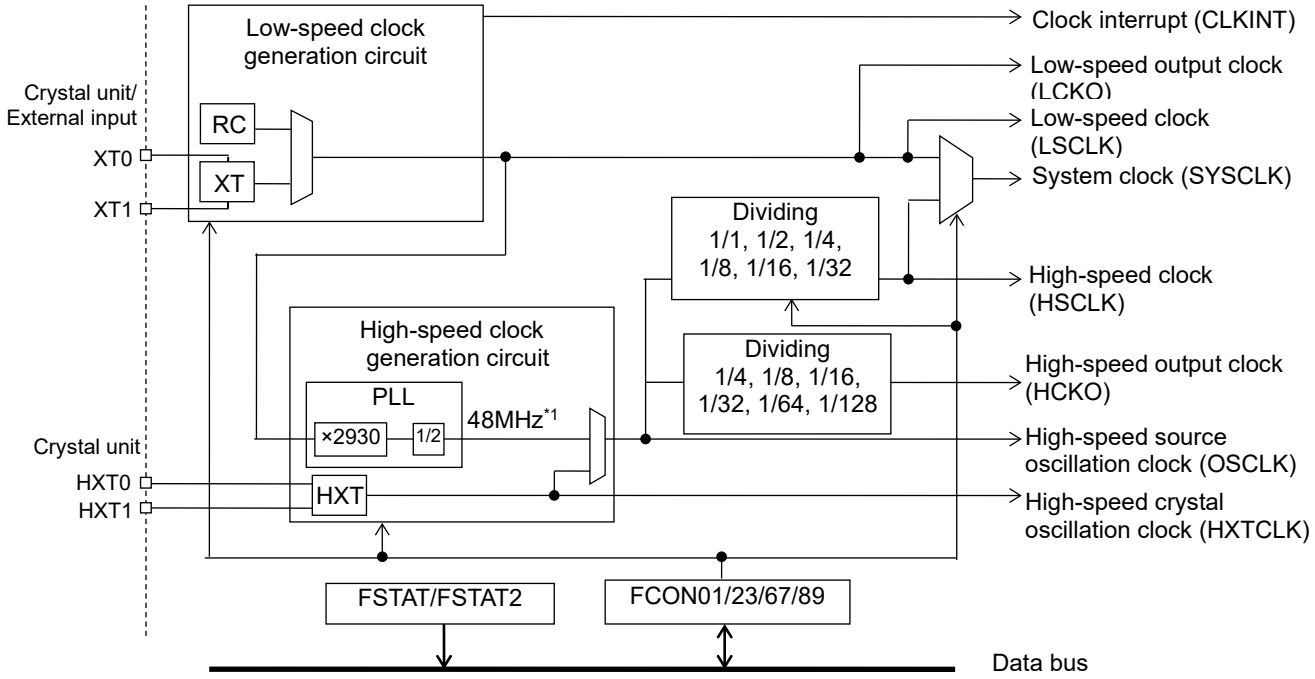
The clock generation circuit generates and provides the low-speed clock (LSCLK), the high-speed clock (HSCLK, OSCLK, HXTCLK), the system clock (SYSCLK), and the low-speed output clock (LCKO) and the high-speed output clock (HCKO). LSCLK, HSCLK, OSCLK and HXTCLK are base clocks for the peripheral circuits, SYSCLK is a basic operation clock of CPU, and LCKO and HCKO are clocks that are output from a port.
For clock output port, see Chapter 22 "Port".
For the STOP mode described in this chapter, see Chapter 5, "System Control Function".

6.1.1 Features

- Low-speed clock generation circuit :
  - Low-speed crystal oscillation mode (XT32K mode)
  - Low-speed clock input mode (EXT32K mode)
  - Built-in RC oscillation mode (RC32K mode)
  - Interrupt generation at switching low-speed clock mode
- High-speed clock generation circuit :
  - Built-in PLL oscillation mode (PLL mode)
  - High-speed crystal oscillation mode (HXT mode)

6.1.2 Configuration

Figure 6-1 shows a configuration of the clock generation circuit.



FCON01/23/67/89 : Frequency control register 01/23/67/89
FSTAT/FSTAT2 : Frequency status register / Frequency status register 2
*1 : The frequency of the 48MHz output is approximately 48.00512MHz at LSCLK = 32.768kHz.

Figure 6-1 Configuration of Clock Generation Circuit

[Note]
- **After power-on or system reset, the operation starts by the clock supplied from the built-in RC low-speed clock generation circuit. At initialization by software, set the FCON01 and FCON23 register to switch to the required clock.**

6.1.3 List of Pins

Pin name	I/O	Description
XT0	I	Pin for connecting a crystal for low-speed clock.
XT1	I/O	Pin for connecting a crystal for low-speed clock / external clock input
HXT0	I	Pin for connecting a crystal for high-speed clock.
HXT1	O	Pin for connecting a crystal for high-speed clock
LCKO	O	Low-speed clock output
HCKO	O	High-speed clock output

6.1.4 Clock Configuration Diagram

Figure 6-2 shows the clock system diagram.

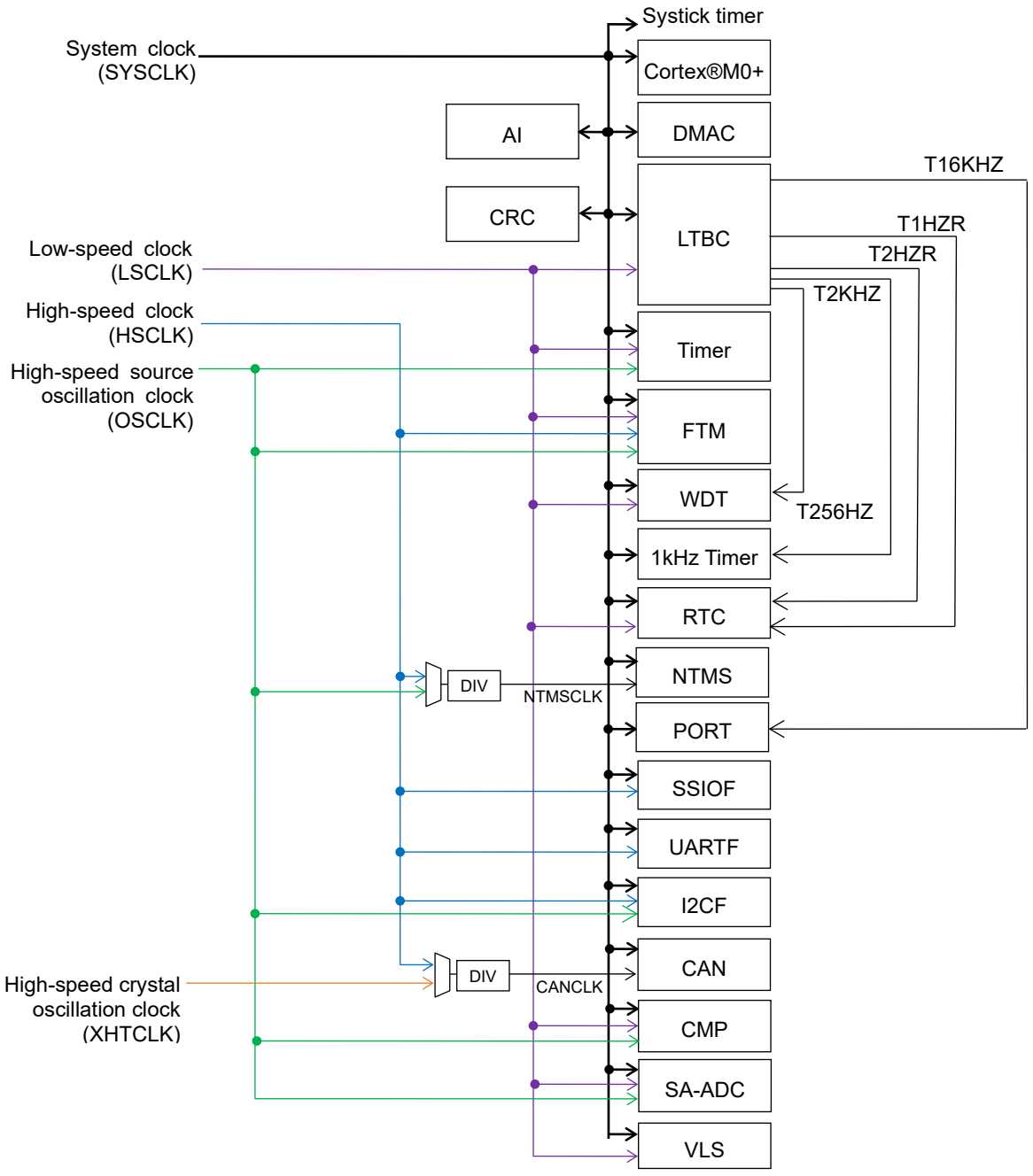


Figure 6-2 Clock System Diagram

6.2 Description of Registers

6.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0300	Clock control base address	CLKCNT	-	-	-
0x00	Frequency control register 01	FCON01	R/W	32	0x0000_0063
0x04	Frequency control register 23	FCON23	R/W	32	0x0000_0002
0x08	Frequency status register	FSTAT	R	32	0x0000_0104
0x14	Frequency control register 67	FCON67	R/W	32	0x0000_0000
0x18	High-speed clock wake-up time setting register	FHWUPT	R/W	32	0x0000_0000
0x1C	Reserved	-	-	-	-
0x20	Frequency control register 89	FCON89	R/W	32	0x0000_0200
0x24	Frequency status register 2	FSTAT2	R/W	32	0x0000_0000
0x50	CANCLK mode register	SYSCANMOD	R/W	32	0x0001_0002
0x54	NTMSCLK mode register	SYSNTMSMOD	R/W	32	0x0001_0002

6.2.2 Frequency Control Register 01 (FCON01)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LPLL	–	HXSPEN	XSPEN	–	ENPLL	ENOSC	SYSCLK	OUTC			OSCM		SYSC		
R/W	R	–	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

FCON01 is a special function register (SFR) used to control the high-speed clock generation circuit and to select system clock.

Bit No	Bit name	Description
15	LPLL	LPLL is used to check the lock state of the PLL oscillator circuit. 0: Non-locked state (Initial value) 1: Locked state
13	HXSPEN	HXSPEN is used to enable a function of the high-speed crystal oscillation stop detection. This function performs controls that is set in FCON89 when the high-speed crystal oscillation stop is detected. 0: Disabled (Initial value) 1: Enabled
12	XSPEN	XSPEN is used to enable a function of the low-speed crystal oscillation stop detection. This function performs controls that is set in FCON23 when the low-speed crystal oscillation stop is detected. 0: Disabled (Initial value) 1: Enabled
10	ENPLL	ENPLL is used to enable the PLL oscillation circuit. This bit can be set only when OSCM[0] = 0. In addition, A writing to this bit is unavailable when a timeout flag waiting for lock (PTO bit of the FSTAT2 register) is “1”. 0: Disabled (Initial value) 1: Enabled
9	ENOSC	ENOSC is used to enable the high-speed crystal oscillation circuit. This bit should be enabled after setting the high-speed crystal oscillation circuit with FCON89. In addition, A writing to this bit is unavailable when a timeout flag waiting for oscillation stability or a flag of oscillation stop detection (HTO or HSTP of the FSTAT2 register) is “1”. 0: Disabled (Initial value) 1: Enabled
8	SYSCLK	SYSCLK is used to select the system clock. The system clock can be selected from low-speed clock (LSCLK) or the HSCLK selected by SYSC. When the oscillation of high-speed clock stops (that is, ENPLL is changed to “0” when OSCM=“00” or ENOSC is changed to “0” when OSCM=“01”), this bit is automatically cleared to “0” and the low-speed clock (LSCLK) is selected for system clock. 0: LSCLK (Initial value) 1: HSCLK
7 to 5	OUTC	OUTC is used to select a division ratio of the frequency of the high-speed output clock (HCKO). This clock frequency is limited 12MHz or lower. 000: Setting prohibited (OSCLK) 001: OSCLK / 2 010: OSCLK / 4 011: OSCLK / 8 (Initial value) 100: OSCLK / 16 101: OSCLK / 32 110: OSCLK / 64 111: OSCLK / 128

Bit No	Bit name	Description
4 to 3	OSCM	OSCM is used to select high-speed clock. This bit can be set only when ENOSC and ENPLL and SYSCLK bits is all 0. 00: PLL oscillation (Initial value) 01: High-speed crystal oscillation 10: Setting prohibited (PLL oscillation) 11: Setting prohibited (High-speed crystal oscillation)
1 to 0	SYSC	SYSC is used to select a division ratio of the frequency of the high-speed clock (HSCLK) for the system and peripheral circuits. Sets the divider value for OSCLK. As shown to Table 6-1, this division ratio is automatically adjusted by the setting of OSCLK frequency and flash wait-cycle, for operation at safe frequencies. 000: OSCLK 001: OSCLK / 2 010: OSCLK / 4 011: OSCLK / 8 (Initial value) 100: OSCLK / 16 101: OSCLK / 32 110: Setting prohibited (OSCLK / 64) 111: Setting prohibited (OSCLK / 128)

Table 6-1 Adjustment of System clock (: Adjusted value)

Setting			Adjusted division ratio and frequency [Hz] (at OSCLK=48MHz)					
OSCM	HOSCF	SYSC	FLACC=2		FLACC=1 (< 30MHz)		FLACC=0 (< 10MHz)	
0	-	0	0	48M	1	24M	3	6M
		1	1	24M	1	24M	3	6M
		2	2	12M	2	12M	3	6M
		3	3	6M	3	6M	3	6M
		4	4	3M	4	3M	4	3M
		5	5	1.5M	5	1.5M	5	1.5M
1	0	0	0	40M	1	20M	2	10M
		1	1	20M	1	20M	2	10M
		2	2	10M	2	10M	2	10M
		3	3	5M	3	5M	3	5M
		4	4	2.5M	4	2.5M	4	2.5M
		5	5	1.25M	5	1.25M	5	1.25M
	1	0	0	20M	0	20M	1	10M
		1	1	10M	1	10M	1	10M
		2	2	5M	2	5M	2	5M
		3	3	2.5M	3	2.5M	3	2.5M
		4	4	1.25M	4	1.25M	4	1.25M
		5	5	0.63M	5	0.63M	5	0.63M

OSCM: OSCM of this register
HOSCF: HOSCF of the FCON89 register
FLACC: FLACC of FLAWAIT register. (See “5.2.6. Flash Wait Register (FLAWAIT)”)

[Note]

- **When the mode to stop the high-speed clock such as STOP mode is set, ENOSC/ENPLL are automatically cleared to "0". After resuming from the mode in which the high-speed clock is stopped, the system clock is the low-speed clock (SYSCLK=0). Set these bits to “1” if the high-speed clock will be used again.**

6.2.3 Frequency Control Register 23(FCON23)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	LSTPSEL		–	–	–	–	–	–	–	LFLTSEL	–	–	XTM	
R/W	–	–	R/W	R/W	–	–	–	–	–	–	–	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

FCON23 is a special function register (SFR) used to select the clock for the low-speed clock generation circuit.

Bit No	Bit name	Description
13 to 12	LSTPSEL	LSTPSEL is used to select an operation when the low-speed crystal oscillation stop is detected. This bit can be set only in the low-speed built-in RC oscillation mode. 00: Reset (Initial value) 01: Reset 10: Flag only 11: Interrupt
4	LFLTSEL	LFLTSEL is used to enable a noise filter for the low-speed crystal oscillation. This bit can be set only in the low-speed built-in RC oscillation mode. 0: Disabled (Initial value) 1: Enabled
1 to 0	XTM	XTM is used to select the low-speed clock mode. This is can be set only when both ENPLL and SYSCLK is “0”. XTM is initialized by entry into the STOP mode. That is, the built-in low-speed RC oscillation mode is selected at that time. 00: Setting prohibited (A setting is invalid, so previous value is kept.) 01: Low-speed crystal oscillation (XT32K) mode or Low-speed external input (EXT32K) mode. The choice between XT32K and EXT32K can be made with LOSCMD in the FCON67 register. 10: Low-speed built-in RC oscillation (RC32K) mode (Initial value) 11: Setting prohibited (A setting is invalid, so previous value is kept.)

6.2.4 Frequency Status Register (FSTAT)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	L0XT	L0RC	–	–	–	–	–	LOSCS	–	–
R/W	–	–	–	–	–	–	R	R	–	–	–	–	–	R	–	–
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

FSTAT is a special function register (SFR) used to show the clock generation circuit state.

Bit No	Bit name	Description
9	L0XT	It indicates that LSCLK is an XT32K or EXT32K clock when L0XT is “1”.
8	L0RC	It indicates that LSCLK is RC32K clock when L0RX is “1”.
2	LOSCS	LOSCS indicates a completion to wait for stability time in the low-speed crystal oscillation circuit. When the low-speed oscillation mode switching is completed after changing XTM of FCON23 register, LOSCS will change from 1 to 0 and LSCLK will switch to clock of the crystal oscillation circuit. LOSCS is always “1” when the setting XTM of FCON23 register is RC32K mode. In addition, this bit become to “1” by entry to the STOP mode. In the case that an interrupt or a flag is selected as the low-speed crystal oscillation stop detection operation, LOSCS changes to “1” by the stop detection, then it will change to “0”, when count of stability wait is completed again. At this time, there is no interrupt waiting for oscillation stability. Also, the LSCLK continues to supply during LOSCS =1. 0: Completed to wait for stability time 1: Selected RC32K mode (Initial value) In XT32K/EXT32K mode, stopped the oscillation or state that stabilization time is counting.

6.2.5 Frequency Control Register 67 (FCON67)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	LOSCMD	–	LMOD		
R/W	–	–	–	–	–	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FCON67 is a special function register (SFR) used to set for the low-speed crystal oscillation circuit.
This register can be written only when XTM[0] of FCON23 register is “0”.

Bit No	Bit name	Description
4	LOSCMD	LOSCMD is used to choose the low-speed crystal oscillation or external input. 0: Low-speed oscillation (XT32K) mode (Initial value) 1: Low-speed external clock input (EXT32K) mode
2 to 0	LMOD	LMOD is used to choose the mode in low-speed crystal oscillation mode. This setting is invalid in the external clock input mode. 000: Standard (STD) mode (Initial value) 001: Low power consumption (LP) mode 010: Tough mode 011: Setting prohibited 1xx: Setting prohibited

6.2.6 High-speed Clock Wake-up Time Setting Register (FHWUPT)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	FHDT			–	FHUT			–	–	FPDT		–	–	–	FPUT
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W	–	–	R/W	R/W	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FHWUPT is a special function register used to control start-up/wake-up for the high-speed clock generation circuit. This register can be written only when both of ENOSC and ENPLL is “0”.

Bit No	Bit name	Description
14 to 12	FHDT	FHDT is used to select the number of counts for the high-speed crystal oscillation supplied timeout. This timeout is counted by the low-speed built-in RC oscillation. This counting starts after the high-speed crystal oscillation circuit is enabled, then HTO becomes to “1” when the high-speed crystal oscillation has not been supplied before count reaches to FHDT setting value. At this time, ENOSC is cleared, and SYSCLK is cleared too if the high-speed crystal oscillation clock is selected as system clock. 000: No timeout (Initial value) 001: 128 counts (Approx. 3.9ms) 010: 192 counts (Approx. 5.9ms) 011: 256 counts (Approx. 7.8ms) 100: 384 counts (Approx. 11.7ms) 101: 512 counts (Approx. 15ms) 110: 768 counts (Approx. 23.4ms) 111: 1024 counts (Approx. 31.2ms)
10 to 8	FHUT	FHUT is used to set the number of counts for the high-speed crystal oscillation clock supplied wait. This is counted by the high-speed crystal oscillation. This counting starts approx. 30µs after the high-speed crystal oscillation circuit is enabled, then the clock is supplied when the count reaches to FHUT setting value. 000: 16 counts (Initial value) 001: 256 counts 010: 1024 counts 011: 2048 counts 100: 4096 counts 101: 8192 counts 110: 16384 counts 111: 32768 counts
5 to 4	FPDT	FPDT is used to select the number of counts for the PLL oscillation supplied timeout. This timeout is counted by the low-speed built-in RC oscillation. This counting starts after the PLL oscillation is enabled, then PTO becomes to “1” when the PLL oscillation has not been supplied before count reaches to FPDT setting value. At this time, ENPLL is cleared if FPUT is “0”, and SYSCLK is cleared too if the PLL oscillation is selected as system clock. 00: No timeout (Initial value) 01: 256 counts (Approx. 7.8ms) 10: 384 counts (Approx. 11.7ms) 11: 512 counts (Approx. 15ms)
0	FPUT	FPUT is used to be configured the timing of starting to supply the PLL oscillation clock. 0: Waits the PLL is locked (Initial value) 1: No waits the PLL is locked

[Note]

- **FPUT=1, set SYSC to something other than 0x0 before setting ENOSC to 1. If SYSC=0x0, operation is not guaranteed.**

6.2.7 Frequency Control Register 89(FCON89)

Offset : 0x20

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	HSTPSEL		–	HSTPT			–	–	HFLTSEL		–	–	HOSCF	
R/W	–	–	R/W	R/W	–	R/W	R/W	R/W	–	–	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

FCON89 is a special function register used to set for the high-speed clock generation circuit.
This register can be written only when both of ENOSC and ENPLL is “0”.

Bit No	Bit name	Description
23 to 16	rsvd	These are reserved. Do not change from initial value.
13 to 12	HSTPSEL	HSTPSEL is used to select an operation when the high-speed crystal oscillation stop is detected. 00: Reset (Initial value) 01: Reset 10: Flag only 11: Interrupt. If the system clock is HXTCLK, ENOSC is cleared to “0” by the stop detection, then the system clock is changed to LSCLK.
10 to 8	HSTPT	HSTPT is used to select the number of counts for the high-speed crystal oscillation stop detection. This detect time is counted by the low-speed built-in RC oscillation. This counting starts after the analog circuit detects stop, then stop detection is notified when count reach to HSTPT setting value. 000: Setting prohibited (0 count) 001: 1 count 010: 2 counts (Initial value) 011: 4 counts 100: 8 counts 101: 16 counts 110: 32 counts 111: Setting prohibited (32 counts)
5 to 4	HFLTSEL	HFLTSEL is used to enable a noise filter for the high-speed crystal oscillation. x0: Disabled (Initial value) 01: Enabled filter for 20MHz oscillation 11: Enabled filter for 40MHz oscillation
1 to 0	HOSCF	HOSCF is used to select frequency of the high-speed crystal oscillation unit connected. 00: 40MHz (Initial value) 01: 20MHz 10: Setting prohibited 11: Setting prohibited

6.2.8 Frequency Status Register 2 (FSTAT2)

Offset : 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	H0PLL	H0XT	–	–	–	–	–	LSTP	LSTAB	HSTP	HTO	–	–	–	PTO
R/W	–	R	R	–	–	–	–	–	R/W	R/W	R/W	R/W	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSTAT2 is a special function register (SFR) used to show state and interrupt request of the clock generation circuit.

Bit No	Bit name	Description
14	H0PLL	H0PLL indicates that the PLL oscillation is supplied. This bit is cleared when ENPLL changes “0”.
13	H0XT	H0XT indicates that the high-speed crystal oscillation stability wait is completed. This bit is cleared when ENOSC changes “0”.
7	LSTP	LSTP indicates that the low-speed crystal oscillation stop has been detected. This bit is cleared by writing “1” to this bit. Writing “0” is invalid. In addition, this bit is not cleared by system reset other than the POR or RESET_N pin reset. 0: No detected, or cleared state. 1: Has detect. The interrupt is requested if LSTPSEL=3.
6	LSTAB	LSTAB indicates an interrupt request that the low-speed crystal oscillation stability wait is completed. This bit is cleared by writing “1” to this bit. Writing “0” is invalid. 0: No request 1: Has request
5	HSTP	HSTP indicates that the high-speed crystal oscillation stop has been detected. This bit is cleared by writing “1” to this bit. Writing “0” is invalid. In addition, this bit is not cleared by system reset other than the POR or RESET_N pin reset. 0: No detected, or cleared state. 1: Has detect. The interrupt is requested if HSTPSEL=3.
4	HTO	HTO indicates that timeout for the high-speed crystal oscillation stability wait is occurred. This bit is cleared by writing “1” to this bit. Writing “0” is invalid. 0: No occurred, or cleared state. 1: Has occurred
0	PTO	PTO indicates that timeout for the PLL oscillation supplied wait is occurred. This bit is cleared by writing “1” to this bit. Writing “0” is invalid. 0: No occurred, or cleared state. 1: Has occurred

6.2.9 CANCLK Mode Register (SYSCANMOD)

Offset : 0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	DIV		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	HXTSEL	–	HSSEL	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R/W	–	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SYSCANMOD is a special function register used to set for CANCLK.

Bit No	Bit name	Description
18 to 16	DIV	DIV is used to select dividing ratio for CANCLK. 000: Setting prohibited (No dividing) 100: Divided by 8 001: No dividing (Initial value) 101: Divided by 16 010: Divided by 2 110: Divided by 32 011: Divided by 4 111: Divided by 32
3, 1	HXTSEL HSSEL	HXTSEL and HSSEL are used to select clock source for CANCLK. Only one of the bits can be set to “1”. The writing other than the following combinations are disabled. HXTSEL=0, HSSEL=1 : HSCLK HXTSEL=1, HSSEL=0 : HXTCLK

6.2.10 NTMSCLK Mode Register (SYSNTMSMOD)

Offset : 0x54

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	DIV		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	OSSEL	HSSEL	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SYSCANMOD is a special function register used to set for NTMSCLK.

Bit No	Bit name	Description
18 to 16	DIV	DIV has no function. The NTMSCLK is not divided.
3, 1	OSSEL HSSEL	OSSEL and HSSEL are used to select clock source for CANCLK. Only one of the bits can be set to “1”. The writing other than the following combinations are disabled. OSSEL=0, HSSEL=1 : HSCLK OSSEL=1, HSSEL=0 : OSCLK

6.3 Description of Operation

6.3.1 Low-speed Clock

6.3.1.1 Low-speed Built-in RC Oscillation Mode

Figure 6-3 shows the low-speed clock generation circuit configuration in the low-speed built-in RC oscillation (RC32K) mode.
When a system reset is released, the low-speed built-in RC oscillation starts up. After the CPU operation start-up start-up time (t_{CPUI}), the LSCLK is output, and the CPU runs program. When the STOP mode is released, the low-speed clock is enabled and the LSCLK is supplied after the wake-up time (T_{RTLS}: 18 counts).

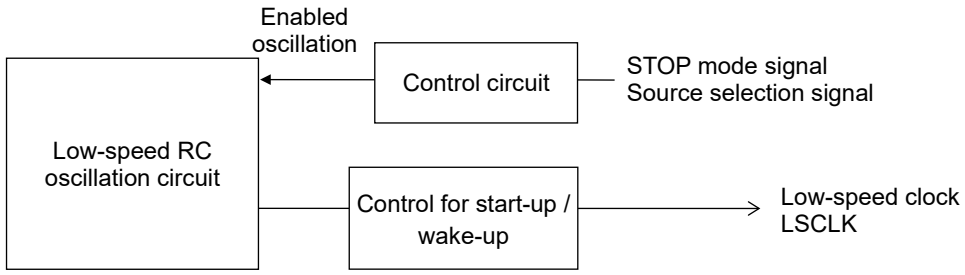


Figure 6-3 Circuit Configuration in the Low-speed Built-in RC Oscillation Mode

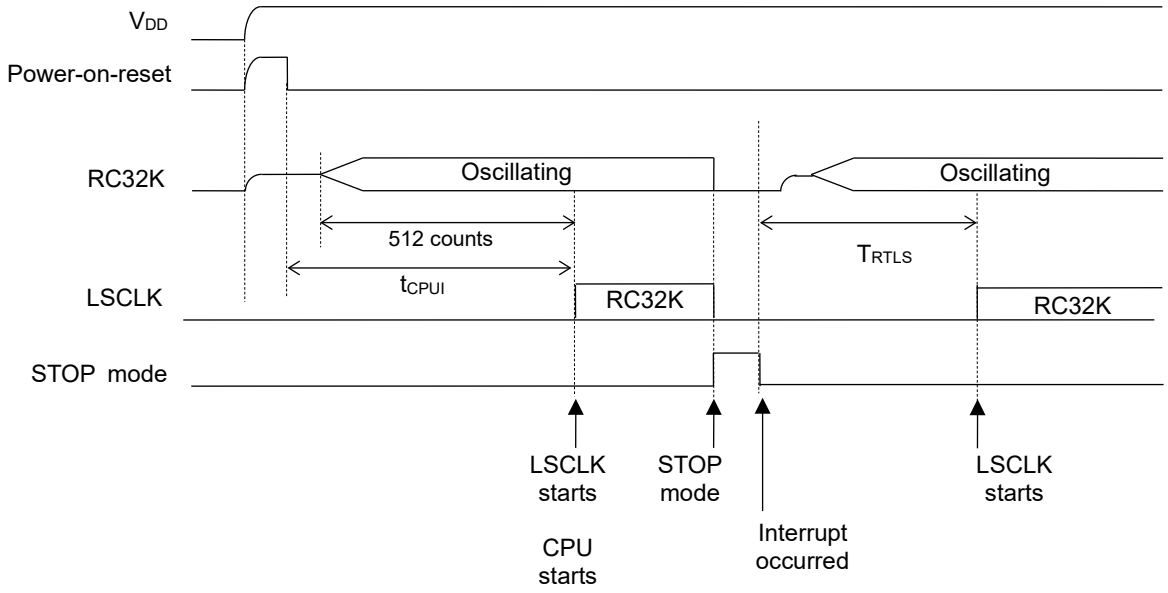


Figure 6-4 Low-speed Clock Operation Waveforms at Start of Low-speed RC Oscillation Circuit and in STOP Mode

6.3.1.2 Low-speed Crystal Oscillation Mode

Figure 6-5 shows the low-speed clock generation circuit configuration in the crystal oscillation (XT32K) mode. The low-speed clock generation circuit is provided with an external 32.768kHz crystal. To match the oscillation frequency by using a trimmer capacitor, connect external capacitors (C_{GL} and C_{DL}) as required.

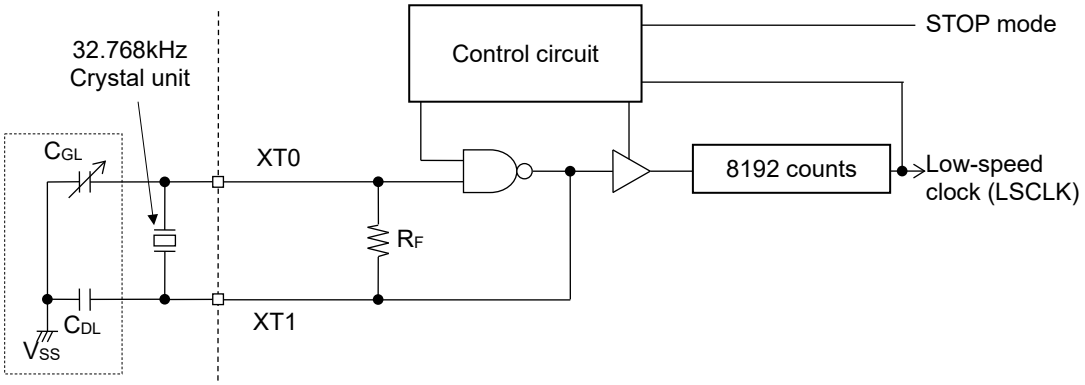


Figure 6-5 Circuit Configuration of the Crystal Oscillation Mode

[Note]

- **Carefully design a board so that the crystal oscillator does not stop.**
  - **Place the crystal oscillator near LSI as much as possible, and do not place a signal and power supply wiring that it becomes a noise source near the crystal oscillator and the wiring.**
  - **The impedance between XT1 and XT0 might decrease by moisture uptake of circuit board in high moisture environment and condensation on the circuit board, and then the oscillation trouble may occur. Please make moisture measures such as coating the circuit board when used in such environments. The oscillation stop might be caused due to condensation.**

  Refer to the application note; "Precautions for MCU board design" for details.

For the low-speed crystal oscillation, the oscillation start/stop can be controlled by the frequency control register 23 (FCON23).
If it sets the XTM of FCON23 to "01", crystal oscillation circuit starts oscillation. After waiting for the low-speed crystal oscillation start-up time (T_{XTL}) and the low-speed crystal oscillation stability time (8192 counts), the low-speed clock (LSCLK) switches from the build-in RC oscillation clock to the low-speed crystal oscillation clock. At this time, the clock interrupt (CLKINT) is generated. See Chapter 5 “System Control Function” for the operation in each stand-by mode.
When the software enters STOP mode, XTM is initialized After the STOP mode is released, LSCLK is the built-in RC oscillation. A setting of LSCLK is necessary again if the low-speed crystal oscillation is used as LSCLK.

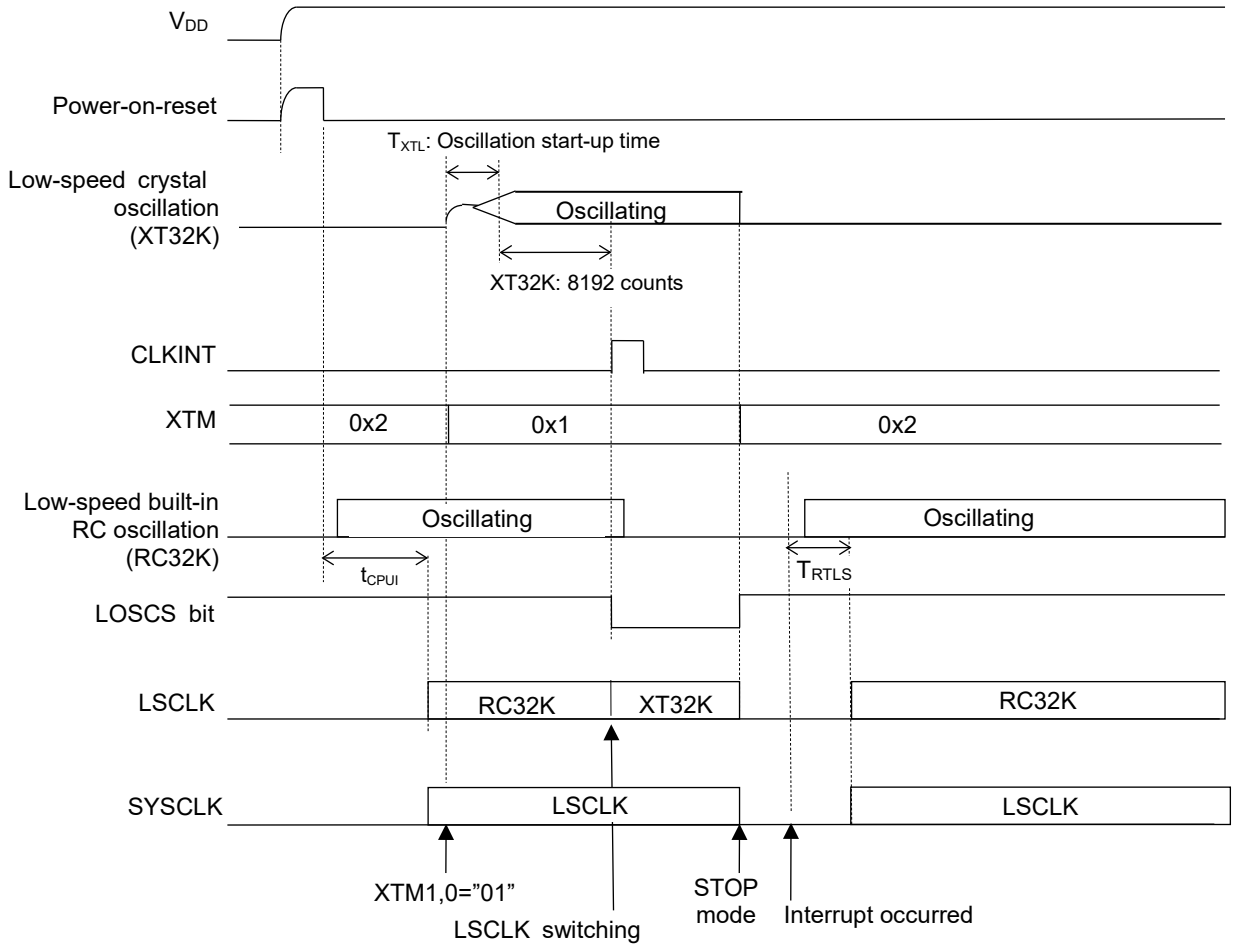


Figure 6-6 Low-Speed Clock Generation Circuit Operation (Crystal Oscillation Mode)

6.3.1.3 Low-speed External Clock Input Mode

Figure 6-7 shows the low-speed clock generation circuit configuration in the external clock input (EXT32K) mode. In this mode, input 32.768kHz clock from XT1 pin. The operation is basically as same as the XT32K mode.

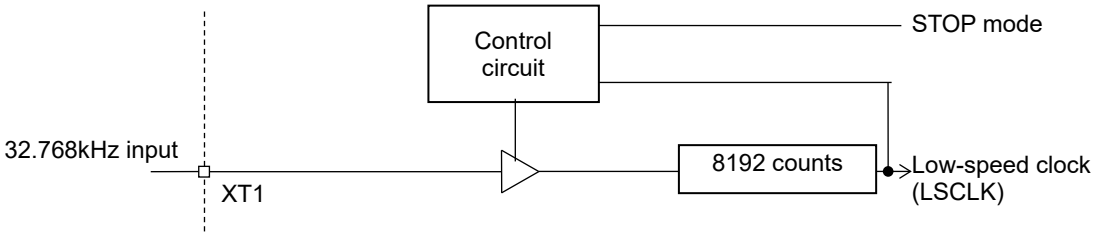


Figure 6-7 Circuit Configuration in the External Clock Input Mode

6.3.1.4 Low-speed Clock Switching

Figure 6-8 shows a flow of checking and setting of the Low-Speed Crystal Oscillation.
As for "Setting for XT32k/EXT32k with FCON01/23/67",

(1) Set FCON67 register; choosing either XT32K or EXT32K and mode in the XT32K.
(2) Set FCON01 register; setting the function of the low-speed crystal oscillation stop detection with XSPEN bit.
(3) Set FCON23 register; choosing noise filter on/off and enabling the oscillation.

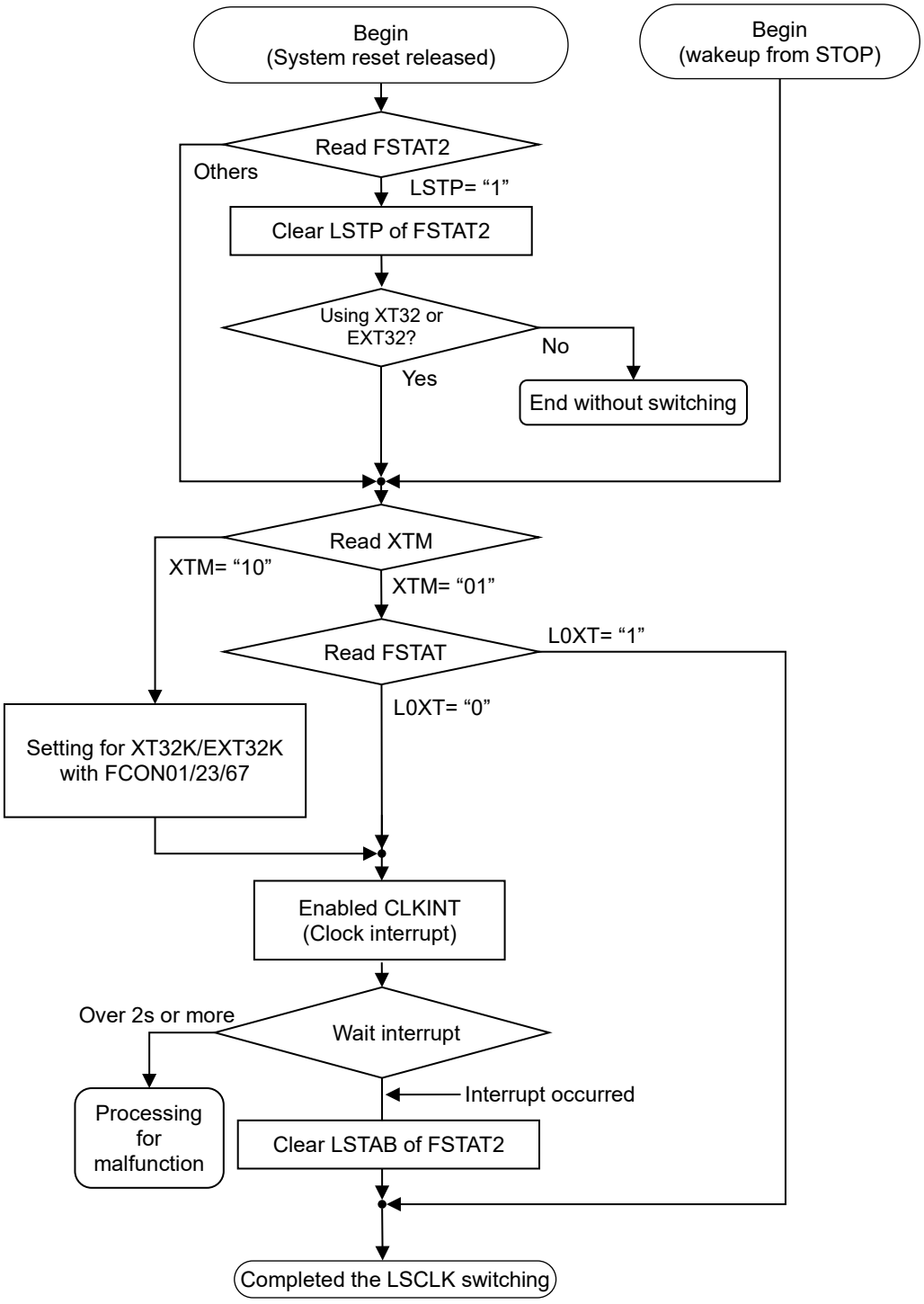


Figure 6-8 Low-speed Clock Switching Flowchart

6.3.2 High-speed Clock

The high-speed clock generation circuit has the following mode that is selected with OSCM of FCON01 register.

- PLL oscillation mode (PLL mode)
- High-speed crystal oscillation mode (HXT mode)

The high-speed clock (HSCLK) and the high-speed output clock (HCKO) are clocks that divide the high-speed source oscillation clock (OSCLK) by setting SYSC and OUTC in the FCON01 register, respectively.
The HCKO can be output from a general-purpose port. See "List of Pins" in Table 1-3 for details on port assignment.
Figure 6-9 shows circuit configuration in the high-speed clock generation circuit.

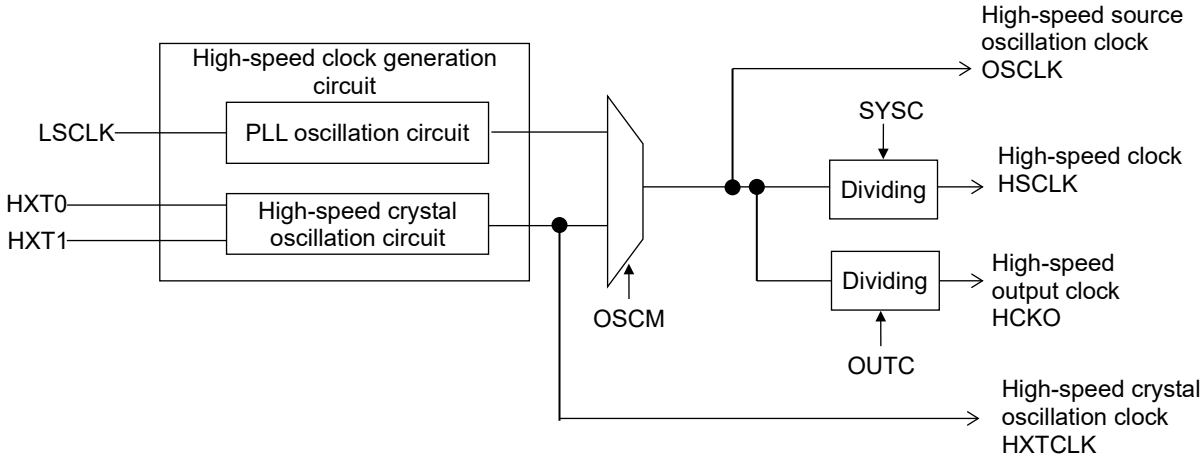


Figure 6-9 Circuit Configuration in the High-speed Clock Generation Circuit

6.3.2.1 Built-in PLL Oscillation Mode

Figure 6-10 shows the high-speed clock generation circuit configuration in the built-in PLL oscillation mode.
The PLL outputs LSCLK multiplied by 2930 and then divided by 2 as OSCLK.
Therefore, the frequency of OSCLK is approx. 48.00512 MHz at LSCLK = 32.768 kHz.

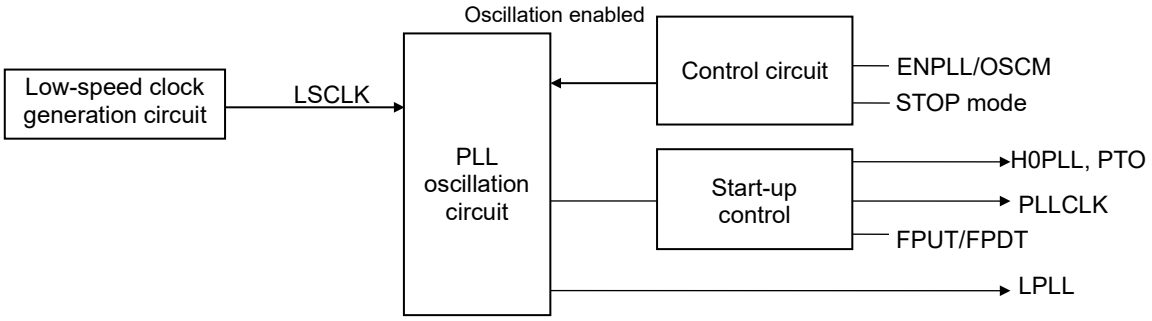


Figure 6-10 Circuit Configuration in the Built-In PLL Oscillation Mode

The PLL oscillation is supplied as OSCLK when the ENPLL is set to "1" and the conditions set in the FHWUPT register are satisfied. At this time, H0PLL of FSTAT2 register becomes "1" and is held until ENPLL becomes "0".
If the oscillation stability wait is not completed within the set time due to LSCLK being unstable, PTO of FSTAT2 register will be set to "1". An interrupt does not occur when the clock is supplied, or when the timeout is detected.
When the software enters STOP mode, ENPLL and H0PLL are cleared and the high-speed clock is stopped.
If the PLL is used as a high-speed clock after the STOP is released, it is necessary to set the ENPLL to "1" again.

Figure 6-11 shows the operation waveforms of the high-speed clock generation circuit in the built-in PLL oscillation mode.

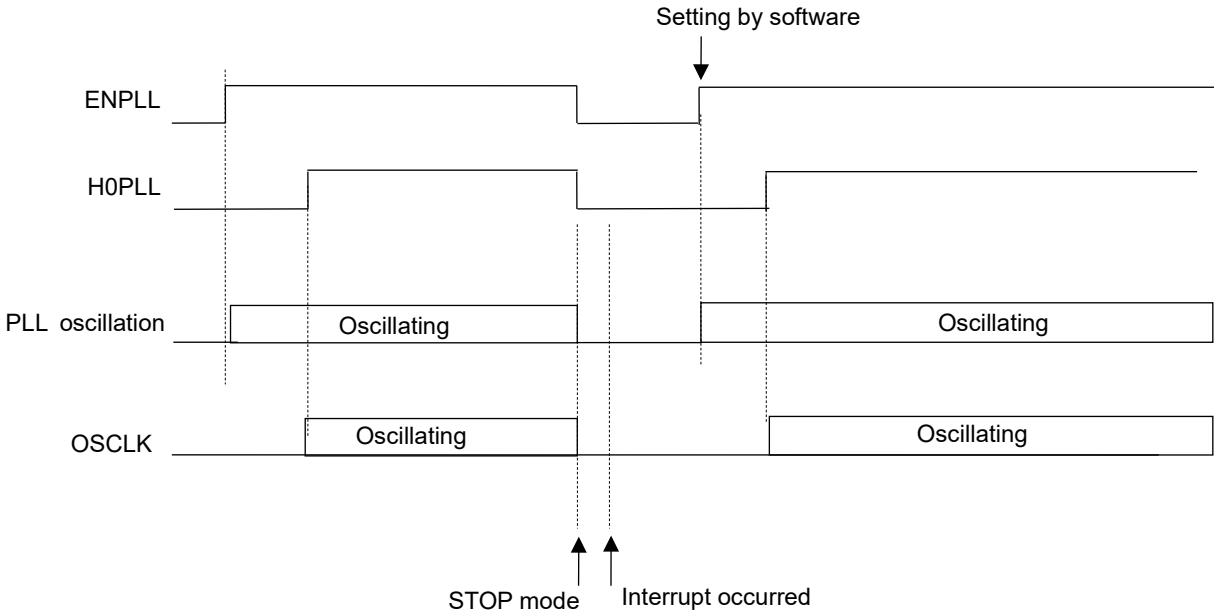


Figure 6-11 Operation Waveforms of PLL circuit

Figure 6-12 shows a flow of setting of the Built-In PLL.

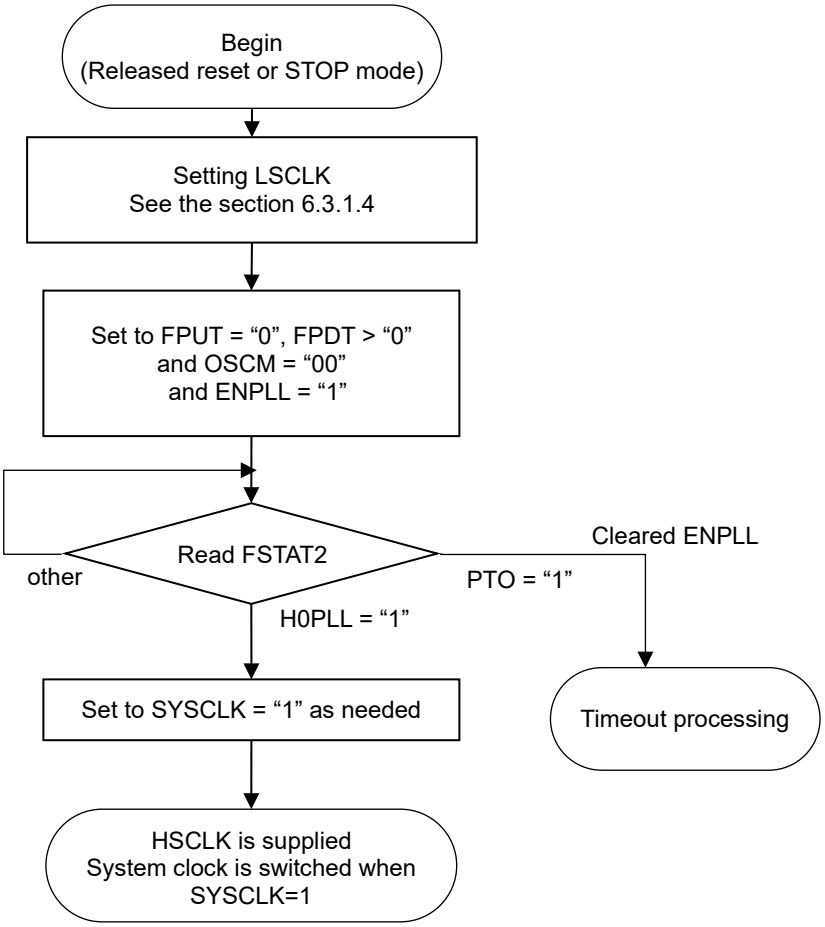


Figure 6-12-1 Setting flow of Built-in PLL Oscillation (Waiting for lock)

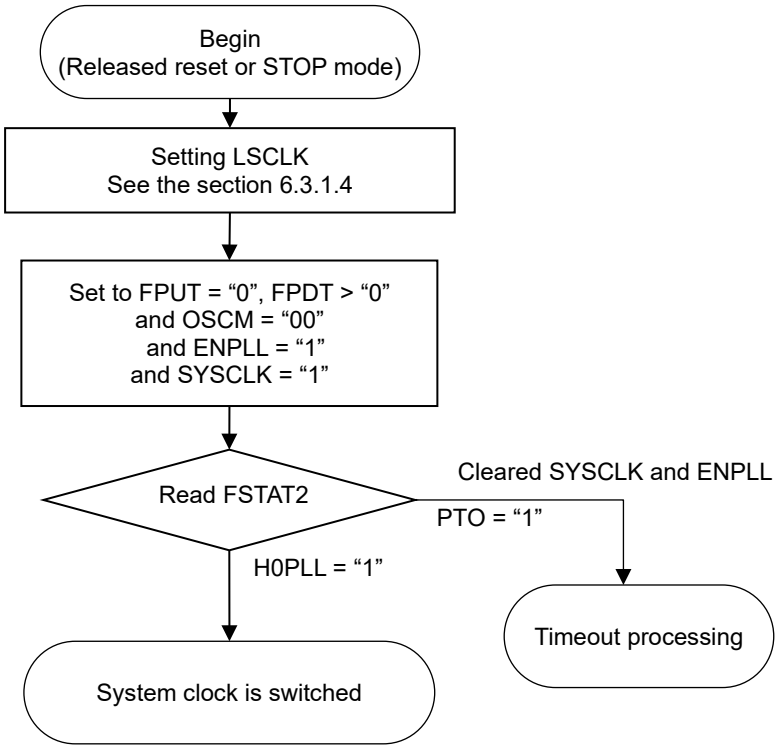


Figure 6-12-2 Setting flow of Built-in PLL Oscillation (Waiting for system clock supply after PLL locked)

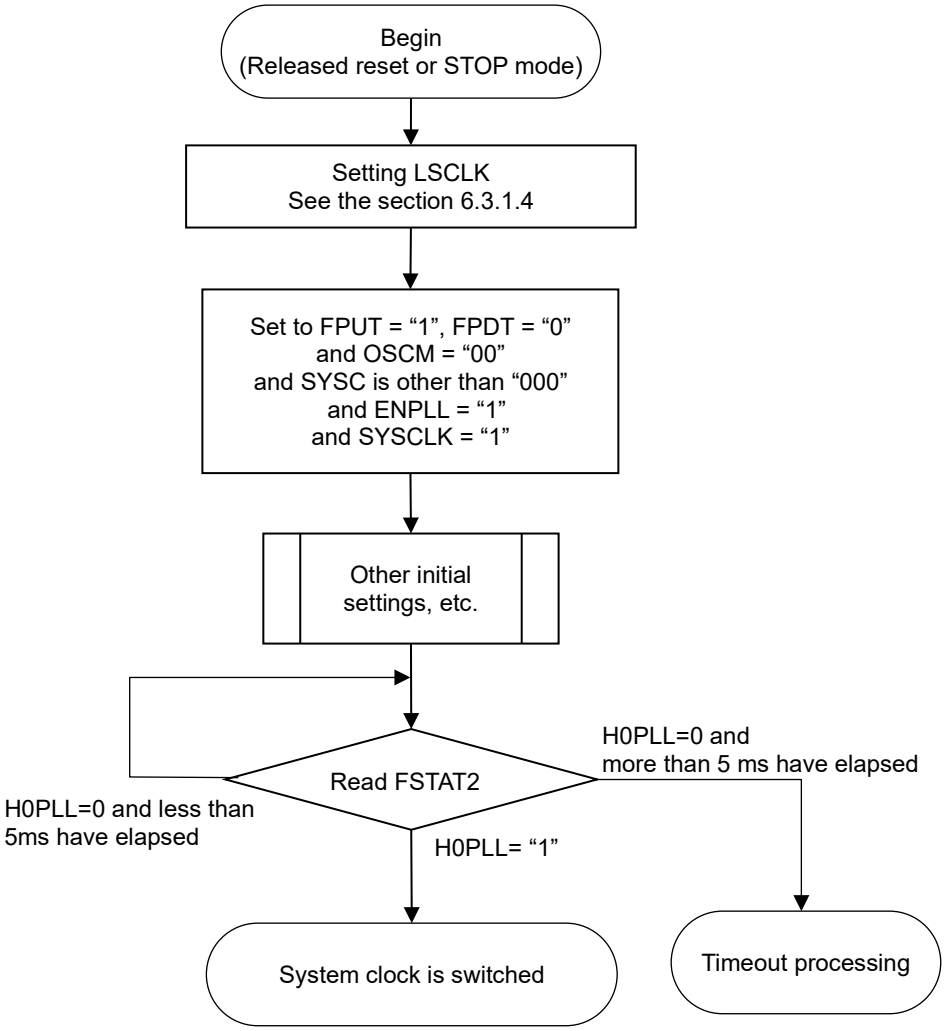


Figure 6-12-3 Setting flow of Built-in PLL Oscillation (Use the system clock without waiting for a lock)

6.3.2.2 High-speed Crystal Oscillation Mode

Figure 6-13 shows the high-speed crystal oscillation circuit configuration.
When the high-speed crystal oscillation mode is selected, the high-speed crystal oscillation can be connected to HXT0 and HXT1 pins.

When ENOSC of FCON01 register is set to "1" with OSCM = "01", the high-speed clock generation circuit starts to count by the high-speed crystal oscillation. After this count reaches setting value in the FHWUPT register, HXTCLK and HSCLK is output. When with OSCM = "00", only HXTCLK is output. Then the high-speed crystal oscillation stop detection circuit is enabled if HXSPEN of FCON01 register is "1". See "6.3.4 Oscillation Stop Detection" for this function.
When the software enters STOP mode, ENOSC and H0XT are cleared and the high-speed clock is stopped.
If the high-speed crystal oscillation is used as a high-speed clock after the STOP is released, it is necessary to set the ENOSC to "1" again.

Figure 6-14 shows the operation waveforms of the high-speed clock generation circuit in the high-speed crystal oscillation mode.

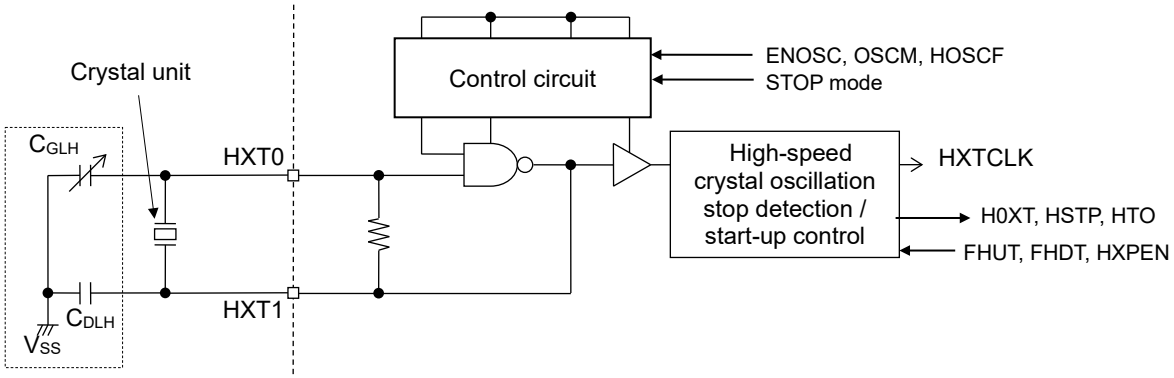


Figure 6-13 High-speed Crystal Oscillation Circuit Configuration

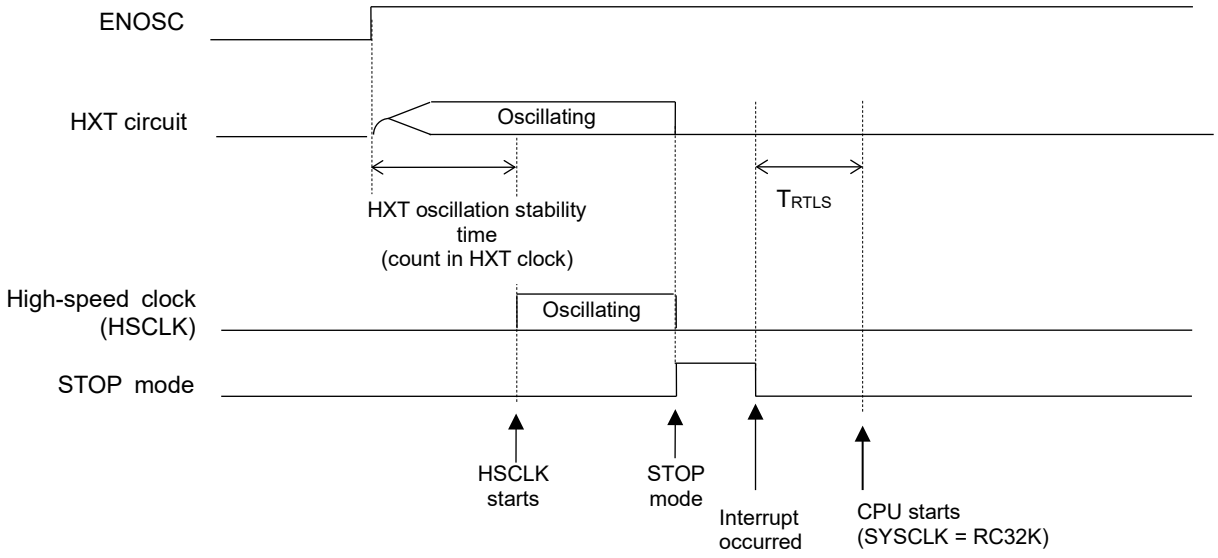


Figure 6-14 Operation Waveforms of high-speed crystal oscillation circuit.

Figure 6-15 shows a flow of setting of the high-speed crystal oscillation.

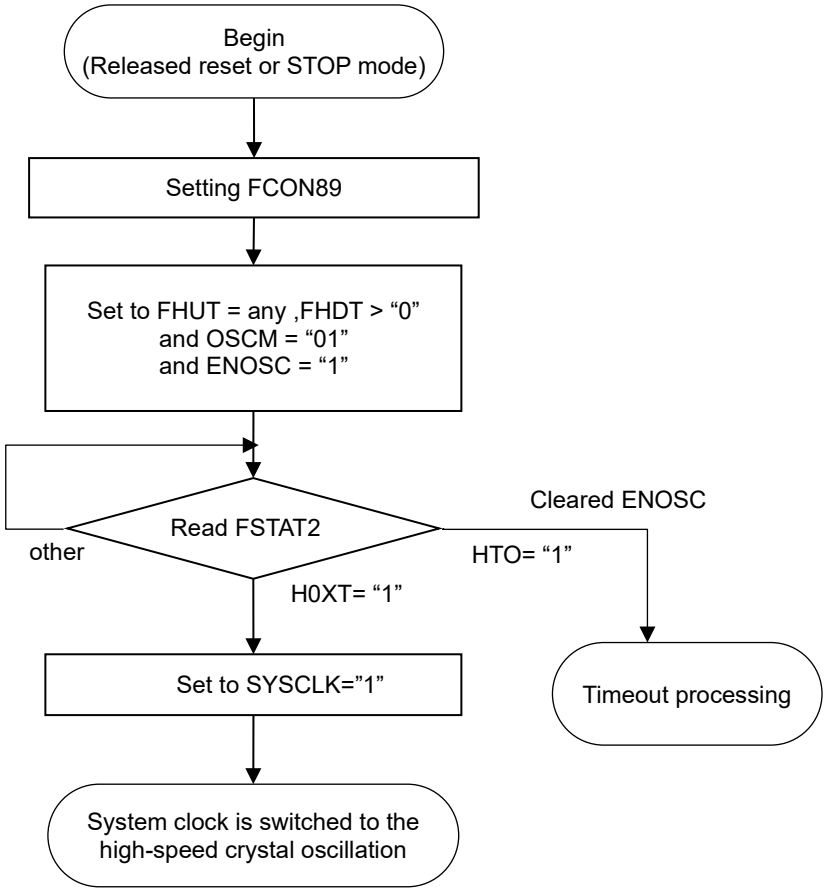


Figure 6-15-1 Setting flow of the High-speed Crystal Oscillation (Waiting for stabilization)

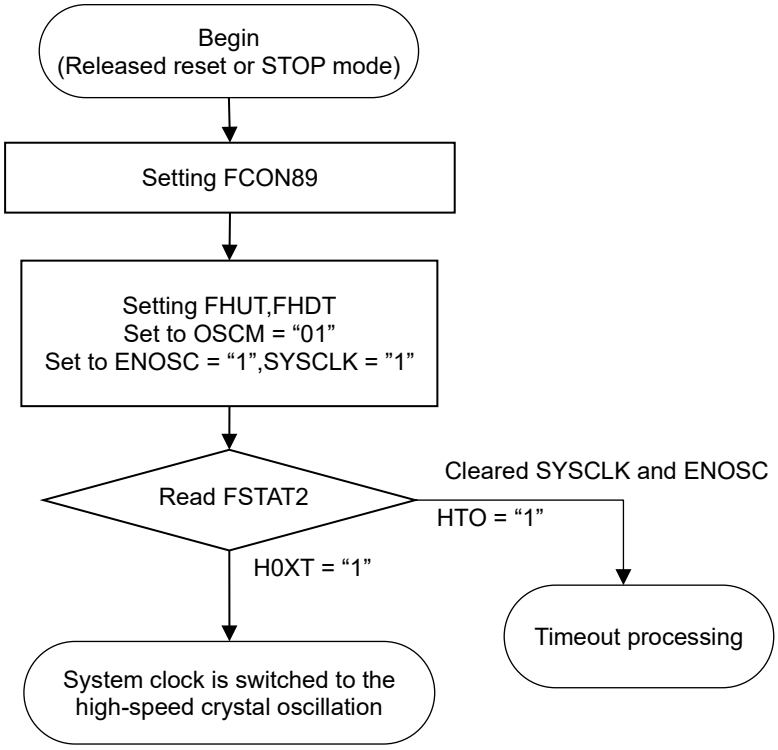


Figure 6-15-2 Setting flow of the High-speed Crystal Oscillation (Waiting for system clock supply after HXT stabilization))

6.3.3 Switching of System Clock

The system clock can be switched between high-speed clock (HSCLK) and low-speed clock (LSCLK) by using the FCON01 register.
Figure 6-16 shows the flow chart of the system clock switching processing (HSCLK to LSCLK).
See section “6.3.2 High-speed Clock” for the system clock switching processing (LSCLK to HSCLK).

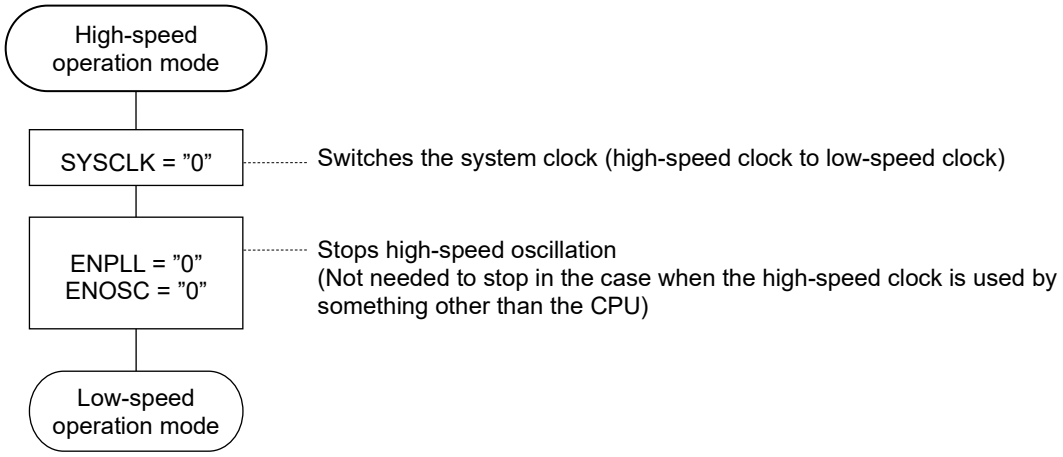


Figure 6-16 Flowchart of System Clock Switching Processing (HSCLK to LSCLK)

6.3.4 Oscillation Stop Detection

When the LSCLK is stopped for approx. 2ms while LSCLK is operated with the low-speed crystal oscillation / external clock, LSTP of FSTAT2 register is set to “1” and a reset or interrupt is occurred. If an interrupt request is selected, interrupt is requested while LSTP is “1”. Write “1” to LSTP to clear the interrupt request.

When the HXTCLK is stopped for time set with HSTPT after the high-speed crystal oscillation is supplied, HSTP of FSTAT2 register is set to “1” and a reset or interrupt is occurred. If an interrupt request is selected, interrupt is requested while HSTP is “1”. Write “1” to HSTP to clear the interrupt request. Also, ENOSC is cleared if stop detection function other than reset is selected, and SYSCLK is cleared and system clock is changed LSCLK if system clock is HXTCLK. ENOSC can not be “1” when HSTP is “1”.

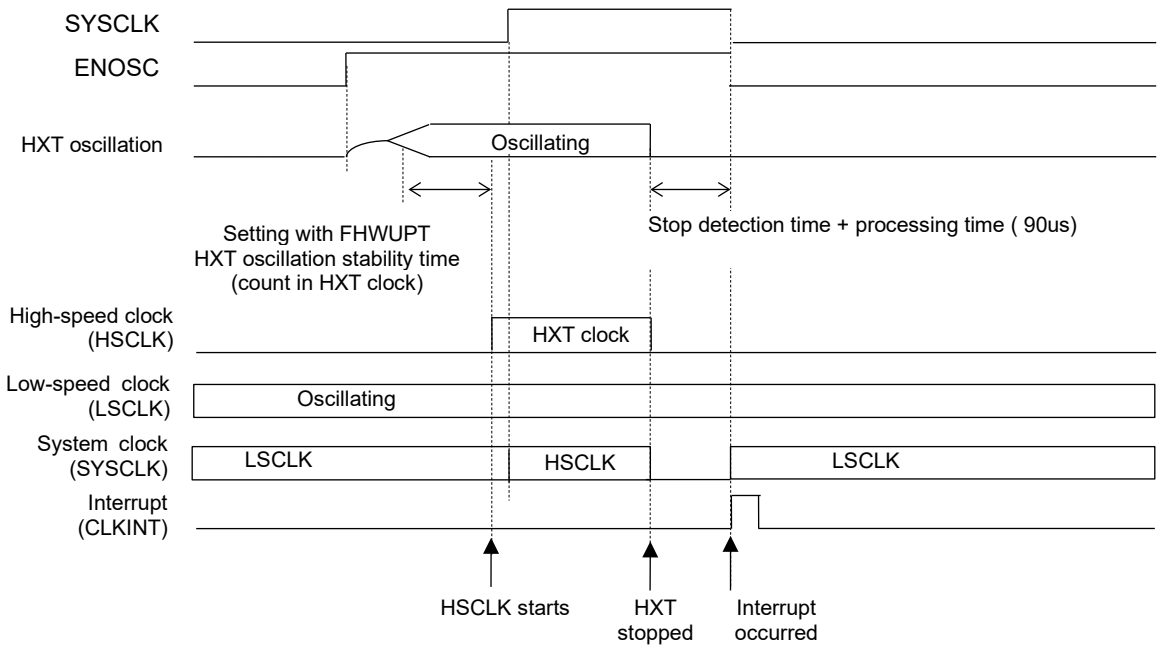


Figure 6-17 Oscillation Stop Detection Operation Waveforms of the HXT circuit.

6.3.5 Timeout

When the PLL is enabled with that FPUT of FHWUPT register is "1" and FPDT is something other than 0, then PTO of FSTAT2 register is set to "1" and ENPLL is cleared to "0" if the PLL locked wait is not completed within time set with FPDT. In addition, SYSCLK is cleared to "0" too if the system clock is PLL oscillation. ENPLL can not be "1" when PTO is "1".
The reasons why PLL does not lock may be that the underlying low-speed clock may not be stable. The reasons of the low-speed clock instability may be that there is a connection with the low-speed crystal resonator, or characteristic abnormality of the crystal resonator, or the power supply fluctuation, etc.

When the high-speed crystal oscillation is enabled with that FHDT of FHWUPT register is something other than 0, then HTO of FSTAT2 register is set to "1" and ENOSC is cleared to "0" if the oscillation stability wait is not completed within time set with FHDT. In addition, SYSCLK is cleared to "0" too if the system clock is the high-speed crystal oscillation. ENOSC can not be "1" when HTO is "1".
The reasons why the stability wait is not completed may be that there is a connection with the high-speed crystal resonator, or characteristic abnormality of the crystal resonator, etc.

If a timeout occurs, the system should process for malfunction appropriately.

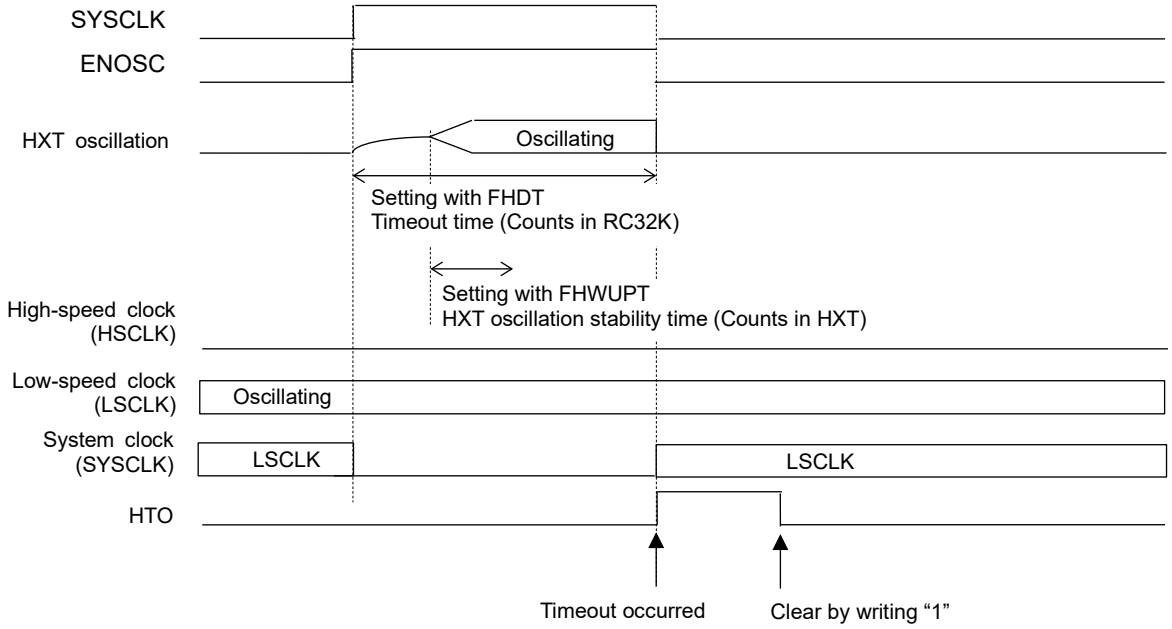


Figure 6-18 Waveforms of Timeout Operation in the HXT circuit

6.3.6 Interrupt

The clock interrupt is occurred at following conditions.

- When the low-speed clock stability wait is completed in the XT32K/EXT32K mode
- When the low-speed crystal oscillation stop is detected
- When the high-speed crystal oscillation stop is detected

The interrupt does not occur when the high-speed clock stability wait is completed.
While any source bit in the FSTAT2 register is "1", the interrupt is requested. This interrupt request is kept until all source bits is cleared.

Chapter 7

Interrupt

7. Interrupt

7.1 Overview

This LSI has 31 interrupt sources (External interrupts: 1 source, Internal interrupts: 30 sources) and a software interrupt (SVC).
For details of each interrupt, see the following chapters:

"Chapter 5 System Control Function "
"Chapter 6 Clock Generation Circuit"
"Chapter 8 DMAC "
"Chapter 9 Time Base Counter "
"Chapter 10 Timers"
"Chapter 11 Functional Timer (FTM)"
"Chapter 12 Real Time Clock (RTC)"
"Chapter 13 1kHz Timer "
"Chapter 14 Watchdog Timer"
"Chapter 15 Three-Phase Motor Control PWM (NTMS) "
"Chapter 16 Synchronous Serial Port with FIFO (SSIOF)"
"Chapter 18 UART with FIFO (UARTF)"
"Chapter 19 CAN FD Controller"
"Chapter 20 I^2C Bus Interface with FIFO (I2CF)"
"Chapter 22 General Purpose Port (GPIO)"
"Chapter 25 Solist-AI Accelerator"
"Chapter 26 Successive Approximation Type A/D Converter (SA-ADC)"
"Chapter 28 Analog Comparator "
"Chapter 29 Flash Memory"
"Chapter 30 Voltage Level Supervisor (VLS)"

7.1.1 Features

- Non-maskable interrupt source: 1 (WDT)
- Maskable interrupt sources: 31 (Internal sources: 30, External sources: 1)
- Software interrupt (SVC)

7.2 Description of Registers

7.2.1 List of Registers

Address	Name	Symbol	R/W	Size	Initial value
0xE000_E100	Interrupt set-enable register	NVIC_ISER	R/W	32	0x0000_0000
0xE000_E180	Interrupt clear-enable register	NVIC_ICER	R/W	32	0x0000_0000
0xE000_E200	Interrupt set-pending register	NVIC_ISPR	R/W	32	0x0000_0000
0xE000_E280	Interrupt clear-pending register	NVIC_ICPR	R/W	32	0x0000_0000
0xE000_E400	Interrupt priority register 0	NVIC_IPR0	R/W	32	0x0000_0000
0xE000_E404	Interrupt priority register 1	NVIC_IPR1	R/W	32	0x0000_0000
0xE000_E408	Interrupt priority register 2	NVIC_IPR2	R/W	32	0x0000_0000
0xE000_E40C	Interrupt priority register 3	NVIC_IPR3	R/W	32	0x0000_0000
0xE000_E410	Interrupt priority register 4	NVIC_IPR4	R/W	32	0x0000_0000
0xE000_E414	Interrupt priority register 5	NVIC_IPR5	R/W	32	0x0000_0000
0xE000_E418	Interrupt priority register 6	NVIC_IPR6	R/W	32	0x0000_0000
0xE000_E41C	Interrupt priority register 7	NVIC_IPR7	R/W	32	0x0000_0000

For details of the interrupt registers of NVIC (0xE000_E100 to 0xE000_E41C), see the section about NVIC of "Cortex®-M0+ Devices Generic User Guide".

7.3 Description of Operation

7.3.1 Interrupt Source

For 32 sources which do not include the watchdog timer interrupt (WDTINT), interrupt enable/disable is controlled by NVIC_ISER and NVIC_ICER. WDTINT is a non-maskable interrupt.
When the interrupt conditions are satisfied, the CPU reads the exception handler start address from the vector table address determined for each interrupt source and starts executing the exception handler.
Table7-1 lists the interrupt sources.

Table7-1 Interrupt Sources

Interrupt No	Interrupt source	Symbol	Vector table address
NMI	Watchdog timer interrupt	WDTINT	0x0000_0008
IRQ[0]	EXI interrupt	EXIINT	0x0000_0040
IRQ[1]	Timer 0 interrupt	TM0INT	0x0000_0044
IRQ[2]	Timer 1 interrupt	TM1INT	0x0000_0048
IRQ[3]	Functional timer 0 interrupt	FTM0INT	0x0000_004C
IRQ[4]	DMAC ch.0 interrupt	DMAC0INT	0x0000_0050
IRQ[5]	Three-phase motor PWM interrupt	NTMSINT	0x0000_0054
IRQ[6]	Synchronous serial port with FIFO 1 interrupt	SIOF1INT	0x0000_0058
IRQ[7]	MCU Status Interrupt (Flash erasing/programming completion, RAM error, etc.)	MCUINT	0x0000_005C
IRQ[8]	UART with FIFO 1 interrupt	UAF1INT	0x0000_0060
IRQ[9]	Timer 2 interrupt	TM2INT	0x0000_0064
IRQ[10]	Timer 3 interrupt	TM3INT	0x0000_0068
IRQ[11]	Functional timer 1 interrupt	FTM1INT	0x0000_006C
IRQ[12]	DMAC ch.1 interrupt	DMAC1INT	0x0000_0070
IRQ[13]	UART with FIFO 3 interrupt	UAF3INT	0x0000_0074
IRQ[14]	Synchronous serial port with FIFO 0 interrupt	SIOF0INT	0x0000_0078
IRQ[15]	I2CF0 interrupt	I2CF0INT	0x0000_007C
IRQ[16]	UART with FIFO 0 interrupt	UAF0INT	0x0000_0080
IRQ[17]	Timer 4 interrupt	TM4INT	0x0000_0084
IRQ[18]	Timer 5 interrupt	TM5INT	0x0000_0088
IRQ[19]	Analog comparator interrupt	CMPINT	0x0000_008C
IRQ[20]	Clock interrupt	CLKINT	0x0000_0090
IRQ[21]	VLS interrupt	VLSINT	0x0000_0094
IRQ[22]	Successive approximation type A/D converter 0 interrupt	SADINT	0x0000_0098
IRQ[23]	UART with FIFO 2 interrupt	UAF2INT	0x0000_009C
IRQ[24]	Reserved	-	0x0000_00A0
IRQ[25]	CAN controller interrupt 0	CAN0INT0	0x0000_00A4
IRQ[26]	CAN controller interrupt 1	CAN0INT1	0x0000_00A8
IRQ[27]	Successive approximation type A/D converter 1 interrupt	SAD1INT	0x0000_00AC
IRQ[28]	RTC interrupt	RTCINT	0x0000_00B0
IRQ[29]	Low-speed time base counter interrupt	LTBCINT	0x0000_00B4
IRQ[30]	1kHz timer interrupt	TM1KINT	0x0000_00B8
IRQ[31]	AI Accelerator interrupt	AIINT	0x0000_00BC

[Note]
- **When multiple interrupts are generated concurrently, they are processed starting from the highest priority level, and the lower-priority interrupts are pending. If they have the same priority level, the interrupt with a smaller interrupt number has higher priority.**
- **Please define vector tables for all unused interrupts for fail safe.**

Chapter 8

DMAC

8. DMAC

8.1 Overview

The DMAC (DMA controller) is a 2-channel direct memory access controller.
By incorporating the DMAC, data transfer between a memory unit and another memory unit, between I/O and memory unit, and between I/O and another I/O can be performed at high speed instead of using the CPU, reducing the burden on CPU operation as well as increasing the efficiency of LSI operation.
DMA transfer is supported for the various peripherals.

8.1.1 Features

- Channel count: 2 channels
- Channel priority
  - Fixed mode : Channel 0 always has a higher priority than channel 1.
  - Round robbing mode : The cannel that has accepted a transfer request has the lowest priority.
- Maximum transfer count : 65536 (64K) times
- Data transfer size: Byte (8 bit) / Half-word (16 bit) / Word (32 bit)
- Dual address access
  Data read from the transfer source and data write to the transfer destination are performed separately.
- Bus ownership request system
  - Cycle steal mode
    Bus ownership request signals are asserted for each DMA transfer.
  - Burst mode
    Bus ownership request signals are asserted until termination of the number of transfers specified.
    (Read to Write for a transfer unit is continuously performed up to the transfer count specified.)
- DMA transfer request system
  - Automatic request
    Transfer requests are automatically generated in the DMAC.
  - External request
    Transfer requests are accepted by the request from the peripheral circuit connected to the channel.
- Interrupt requests
  An interrupt request is generated to the CPU after termination of the number of DMA transfers specified.
  - Interrupt request signals are output to each channel individually.
  - Interrupt request signals can be masked for each channel.

[Note]
- **For selection of a peripheral circuit that issues external requests, see Chapter 5, “System Control Function”.**

8.1.2 Configuration

Figure 8-1 shows the configuration of the DMAC.

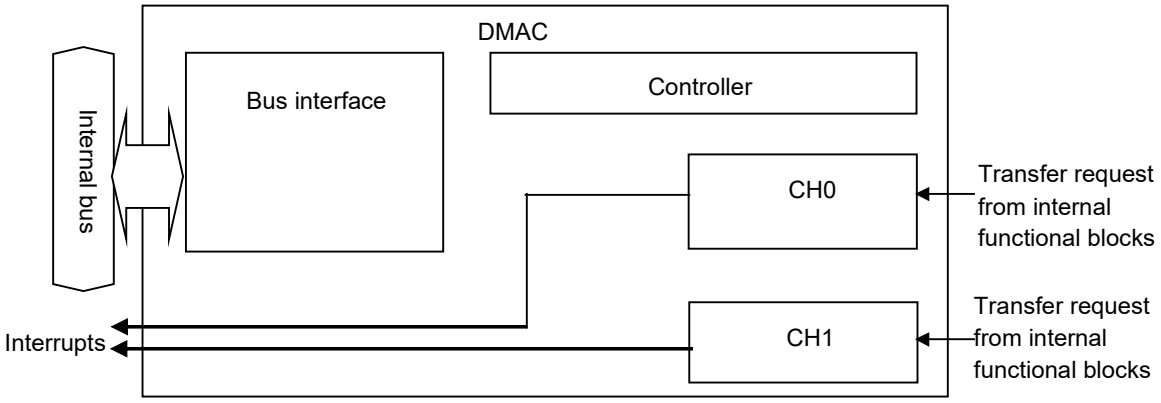


Figure 8-1 Configuration of DMAC

8.2 Description of Registers

8.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4700_0000	DMA common base address	DMAC	-	-	-
0x00	DMA mode register	DMAMOD	R/W	32	0x0000_0000
0x04	DMA status register	DMASTA	R	32	0x0000_0000
0x08	DMA end status register	DMAINT	R	32	0x0000_0000
0x4700_0100	DMA channel 0 base address	DMAC0	-	-	-
0x00	DMA channel mask register	DMACMSK0	R/W	32	0x0000_0001
0x04	DMA transfer mode register	DMACTMOD0	R/W	32	0x0000_0040
0x08	DMA transfer source address register	DMACSAD0	R/W	32	0x0000_0000
0x0C	DMA transfer destination address register	DMACDAD0	R/W	32	0x0000_0000
0x10	DMA transfer count register	DMACSIZ0	R/W	32	0x0000_0000
0x14	DMA end status clear register	DMACCINT0	W	32	0x0000_0000
0x4700_0200	DMA channel 1 base address	DMAC1	-	-	-
0x00	DMA channel mask register	DMACMSK1	R/W	32	0x0000_0001
0x04	DMA transfer mode register	DMACTMOD1	R/W	32	0x0000_0040
0x08	DMA transfer source address register	DMACSAD1	R/W	32	0x0000_0000
0x0C	DMA transfer destination address register	DMACDAD1	R/W	32	0x0000_0000
0x10	DMA transfer count register	DMACSIZ1	R/W	32	0x0000_0000
0x14	DMA end status clear register	DMACCINT1	W	32	0x0000_0000

8.2.2 DMA Mode Register (DMAMOD)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	PRI
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAMOD is a common setting register for DMAC.

Bit No	Bit name	Description
0	PRI	PRI is used to set channel priority. 0: Fixed (Initial value) Channel 0 has the highest channel priority and channel 1 has the lowest. 1: Round robin The channel that was used last among valid channels has the lowest channel priority.

8.2.3 DMA Status Register (DMASTA)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	STA	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMASTA is a register that indicates the status of DMA transfer.

[Description of each bits]
It indicates the DMA transfer status of corresponding channel.
If a value other than “0” is set in the transfer count registers (DMACSIZ0, 1), “1” is set in the applicable channel bit. If a DMA transfer has been completed normally for the specified DMA transfer count or if it has been ended as an error during a DMA transfer, the applicable channel bit is cleared.

0: No un-transferred data.
1: Un-transferred data remaining.

Bit No	Bit name	Description
1	STA[1]	DMA channel 1
0	STA[0]	DMA channel 0

8.2.4 DMA Completion Status Register (DMAINT)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ISTP	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	ISTA		–	–	–	–	–	–	IREQ	
R/W	–	–	–	–	–	–	R	R	–	–	–	–	–	–	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAINT is a register that indicates the DMA completion status.

[Description of each bits]
When arbitrary data is written into the DMA completion status clear registers (DMACCINT0 or DMACCINT1) provided in each channel, the IREQ, ISTA and ISTP bits of the corresponding channel are cleared.

Bit No	Bit name	Description
17	ISTP[1]	This indicates an abnormal end cycle of channel 1. 0: read cycle 1: write cycle
16	ISTP[0]	This indicates an abnormal end cycle of channel 0. 0: read cycle 1: write cycle
9	ISTA[1]	This indicates an end status of channel 1. 0: Normal end 1: Abnormal end
8	ISTA[0]	This indicates an end status of channel 0. 0: Normal end 1: Abnormal end
1	IREQ[1]	This indicates an interrupt request status of channel 1. 0: No request 1: Requested
0	IREQ[0]	This indicates an interrupt request status of channel 1. 0: No request 1: Requested

[Note]

- **To reference this register during interrupt processing, reference the ISTP bit when an abnormal end occurs. At the time of normal end or forced abort, the ISTP bits retain the initial value.**

8.2.5 DMA Channel Mask Register (DMACMSK0 to 1)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	MSK
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DMACMSK0-1 are registers that set the masks of channels. The DMAC does not perform transfer for masked channels.
Applicable channels are masked after a reset.

Bit No	Bit name	Description
0	MSK	This bit is used to set mask. 0: released mask (Initial value) 1: masked

[Note]

- **In processing to program or erase of the flash memory, access to flash memory by DMAC is prohibited.**

8.2.6 DMA Transfer Mode Register (DMACTMOD0 to 1)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	IMK	BRQ	DDP	SDP	TSIZ		ARQ
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DMACTMOD0-1 registers are registers that set various DMA transfer modes.

Bit No	Bit name	Description
6	IMK	This bit is used to set interrupt mask. 0: Released mask (Initial value) An interrupt can be generated when a DMA transfer ends (normal end/abnormal end). 1: masked
5	BRQ	This bit is used to set the bus ownership request mode. 0: Burst mode (Initial value) 1: Cycle steal mode
4	DDP	This bit is used to set the device type of the transfer destination. 0: Fixed address device (Initial value) 1: Increment address device
3	SDP	This bit is used to set the device type of the transfer source. 0: Fixed address device (Initial value) 1: Increment address device
2 to 1	TSIZ	These bits used to set the transfer size. 00: Byte (Initial value) 01: Half-word 10: Word 11: Setting prohibited
0	ARQ	This bit is used to set the DMA transfer request. 0: External transfer request (Initial value) 1: Automatic transfer request

[Note]

- **Auto request is enabled (external request is disabled) when an auto request is specified for a transfer request with the ARQ bit and an external request signal is generated.**
- **The transfer size is restricted according to the device type and the bus width. See Section "8.4.1 Notes at Transfer Size Setting".**

8.2.7 DMA Transfer Source Address Register (DMACSAD0 to 1)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSAD[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSAD[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMACSAD0-1 are registers that set the address of the DMA transfer source.
A DMA transfer (read) is started based on the values set in these registers.

Also, if the transfer source is an increment address device, the address is incremented starting at the address set in this register according to the transfer size (byte/half-word/word), and if it is a fixed address device, fixed addresses are used as the DMAC accesses. Therefore note the following:

<In the Case of an Increment Address Device>
For word transfer : Set the lower two bits to “00”.
For half-word transfer : Set the lower one bit to “0”.
The address is updated when a read from the transfer source is completed normally.

[Note]

- **This DMAC increments addresses according to the settings of the device type and transfer size. At that time, the lower bits of an address are ignored internally according to the settings; however, upon reading these registers, the lower bits output the values written.**

8.2.8 DMA Transfer Destination Address Register (DMACDAD0 to 1)

Offset : 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDAD[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDAD[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMACDAD0-1 are registers that set the address of the DMA transfer destination.
A DMA transfer (write) is started based on the values set in these registers.
Also, if the transfer destination is an increment address device, the address is incremented starting at the address set in this register according to the transfer size (byte/half-word/word), and if it is a fixed address device, fixed addresses are used as the DMAC accesses. Therefore note the following:

<In the Case of an Increment Address Device>
For word transfer : Set the lower two bits to “00”.
For half-word transfer : Set the lower one bit to “0”.
The address is updated when a write to the transfer destination is completed normally.

[Note]

- **This DMAC increments addresses according to the settings of the device type and transfer size. At that time, the lower bits of an address are ignored internally according to the settings; however, upon reading these registers, the lower bits output the values written.**

8.2.9 DMA Transfer Count Register (DMACSIZ0 to 1)

Offset : 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	CSIZ[16]
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIZ[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DMACSIZ0-1 registers are 32-bit readable/writable registers that set the DMA transfer count. When the transfer source (read) terminates properly, the values are decremented.
To perform DMA transfer n times, set this register to “n”.
Since the maximum setting value of these registers is “0001_0000H” (65536), the operation is not guaranteed if a value larger than the maximum setting value is specified.

8.2.10 DMA Completion Status Clear Register (DMACCINT0 to 1)

Offset : 0x14

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCINT[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCINT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMACCINT0-1 are registers that are used to clear the interrupt information of the applicable channel of the DMA completion status register (DMAINT).
When arbitrary data is written into these registers, the interrupt request bit (IREQ), interrupt source bit (ISTA) and abnormal end cycle bit (ISTP) of the corresponding channel of the DMAINT are cleared.

8.3 Description of Operation

8.3.1 Channel Priority

If a transfer request is issued to multiple channels at the same time, this DMAC performs transfers according to the predetermined priority order. The channel with the highest priority at that time can perform a DMA transfer in units of one transfer (byte, half-word, word, or one burst transfer).
The following methods are available to determine the priority order of channels, and either one can be selected by register setting.

8.3.1.1 Fixed Mode

In the fixed mode, the channel priority is channel 0 and channel 1 in this order.

8.3.1.2 Round Robin Mode

In the round robin mode, when a transfer in units of one transfer (byte, half-word, word, or one burst transfer) is completed in one channel, the priority order of that channel becomes the lowest, and the priority order is changed so that the channel with the second highest priority order becomes the highest. The channel priority after reset is channel 0 and channel 1 in this order.

1. Initial state — Channel 0 > Channel 1
2. When channel 0 is selected — Channel 1 > Channel 0
3. When channel 1 is selected — Channel 0 > Channel 1

8.3.2 Dual Address Access

In dual access, the DMAC accesses both the transfer source and the destination source via addresses, and performs DMA transfer using two bus cycles of a read cycle and a write cycle. The DMAC accesses the transfer source using a read cycle, and accesses the transfer destination using a write cycle, and then transfer data is temporarily stored in the register inside the DMAC.

8.3.3 Bus Ownership Request Mode

If peripheral circuits are selected by DMAC Control Register (DMARQCNT), use not Burst mode but Cycle steal mode. For the DMAC control register, see Chapter 5, “System Control Functions”.

8.3.3.1 Cycle Steal Mode

The DMAC de-asserts the bus ownership request signal to the system bus for making accesses to the inside of the LSI every time 1 unit of transfer (1 byte, half-word or 1 word) is executed. Next, if there are any DMA transfers left that need to be executed, the DMAC asserts the bus ownership request signal, and executes 1 unit of transfer again after acquiring the bus ownership, and then de-asserts the bus ownership request signal. This operation is repeated until the transfer end condition is satisfied.
If a transfer request is issued from a channel with a higher priority during a cycle steal transfer, the transfer of that channel is executed after one DMA transfer is finished.

8.3.3.2 Burst Mode

Once the DMAC obtains the bus ownership, it continuously asserts a bus ownership request signal until the specified number of transfers is complete. Therefore, in burst transfer, the bus ownership may not be passed to the CPU during the period of burst transfer; secure system performance by software processing.
If a transfer request is issued from a channel with a higher priority during a burst mode transfer, the transfer of that channel is not performed until the current burst transfer (for the specified number of transfers) is finished.

8.3.4 Completion of DMA Transfer

The following describes the DMA transfer completion and suspension conditions:

(a) Normal end
If transfers are executed as many times as the value specified in the DMA transfer count register (DMACSIZ), the DMAC terminates the DMA transfer of the corresponding channel and can generate an interrupt.

(b) Abnormal end
If the DMAC accesses a reserved area and receives an error response from the system bus, the DMAC immediately terminates DMA transfer and generates an interrupt.
Error information contains information about the channel in which the error occurred and about whether the error occurred in read cycle (occurred at the transfer source) or write cycle (occurred at the transfer destination). Error information can be acquired by referencing the values of IREQ, ISTA and ISTP of the DMA completion status register (DMAINT).
For the address areas causing error responses, see Chapter 3, “Memory Space”.

(c) Forced suspension
When “1” is written (masked) in the mask channel bit of the DMA channel mask register (DMACMSK), the DMAC suspends transfer after the current transfer is terminated (a write to the transfer destination). In this case, an interrupt is not generated. Moreover, transfer can be resumed by canceling the mask.

Table 8-1 Register Values after DMA Transfer Completion

Register		(a) After normal end	(b) After abnormal end	(c) After forced suspension
DMA status register (DMASTA)		“0” (No un-transferred data)	“0” (No un-transferred data)	“1” (Un-transferred data present)
DMA completion status register (DMAINT)	IREQ	“1” (*1)	“1” (*1)	“0” (No change)
	ISTA	“0” (Normal end)	“1” (Abnormal end)	“0” (No change)
	ISTP	“0” (Initial status)	“0” (At transfer source error) “1” (At transfer destination error)	“0” (No change)
DMA transfer source/destination address register (DMACSAD, DMACDAD)		Retains the address following the final transfer address	Retains the error address	Retains the address following the suspended transfer address
DMA transfer count register (DMACSIZ)		“0”	*2	Retains the number of remaining transfers

*1 : The IREQ bit is set to “1” regardless of the IMK (interrupt mask) bit setting of the DMA transfer mode register.
*2 : Note that the value varies depending on whether an error occurs at the transfer source or the transfer destination. The DMAC decrements the DMA transfer count register (DMACSIZ) when access to the transfer source is properly terminated.

8.3.5 DMA Transfer Request

There are two request modes for DMA transfer requests: the external request mode that uses the DMA transfer request signal from internal peripheral circuits, and the auto request mode that does not use such a signal. Either mode can be selected by register setting.

8.3.5.1 Auto Request Mode

The auto request mode is used to automatically generate transfer requests inside the DMAC. The CPU achieves this by setting the auto request bit to “1” inside the DMAC in the case of a transfer between memory and memory, or between a memory and a peripheral circuit that cannot generate transfer requests.
In this case, once the auto request bit is set to “1”, a DMA transfer is automatically performed every time the DMAC is started.

8.3.5.2 External Request Mode

The external request mode is used to generate transfer requests by inputting a DMA transfer request signal from one of the peripheral function blocks. For the peripheral circuit which is requested the external request, see Chapter 5, “System Control Function”.

8.3.6 Procedure for setting DMA transfer mode

(1) Procedure for setting DMA transfer mode

1. Setting the DMA channel mask register (DMACMSK)
   - Channel mask
   (It is not necessary to set channel mask because channels are masked in the case of a transfer immediately after a reset.)
2. Setting the DMA mode register (DMAMOD)
   - Set in the state that all channels is masked
   - Set the priority (fixed/round robin).
3. Setting the DMA channel transfer mode register (DMACTMOD) of the corresponding channel
   - Set in the state that the corresponding channel is masked
   - Set the transfer request (auto request/external request).
   - Set the transfer size (byte/half-word/word).
   - Set the device type of the transfer source and destination (fixed address device/increment address device).
   - Set the bus ownership request method (set the cycle steal mode/burst mode).
   - Set whether to output or not to output an interrupt request.
4. Setting the DMAC Request Selection Register (DREQSEL)
   (In case of setting an external request in the DMACTMOD register)
   - Set the request signal.

The following figure shows an example of setting:

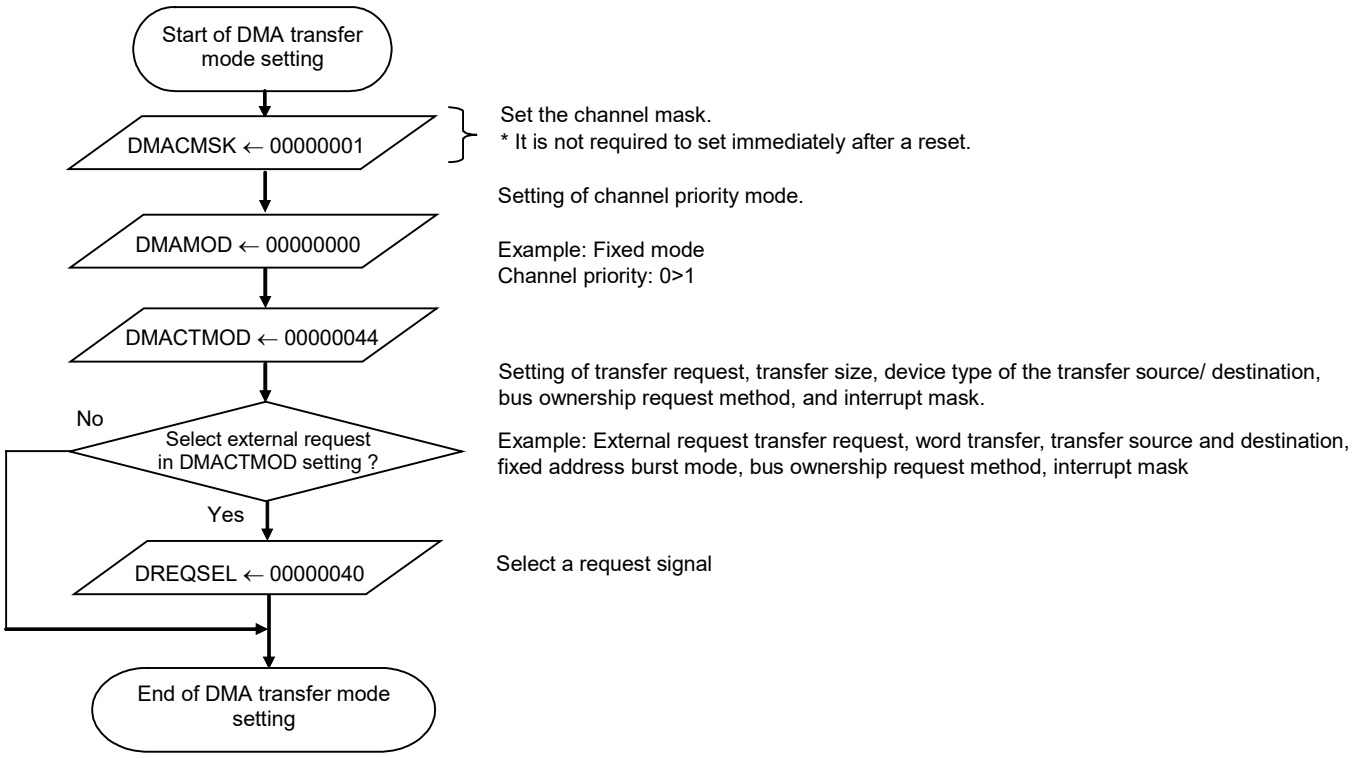


Figure 8-2 Example of setting

(2) Procedure for starting up DMAC

1. Setting the DMA channel mask register (DMACMSK) of the corresponding channel
   - Channel mask
   (It is not necessary to set this register in the case of a transfer immediately after a reset. For other cases, be sure to mask channels.)
2. Clearing an interrupt
   - Write arbitrary data into the DMA completion status clear register (DMACCINT).
   (It is not necessary to set this register in the case of a transfer immediately after a reset. For other cases, be sure to clear the interrupt.)
3. Setting the DMA channel start address registers (DMACSAD, DMACDAD) of the corresponding channel
   - Set the transfer source address register.
   - Set the transfer destination address register.
4. Setting the DMA channel transfer count register (DMACSIZ) of the corresponding channel
   - Set the transfer count.
5. Setting the DMA channel mask register (DMACMSK) of the corresponding channel
   - Start the DMAC by canceling the channel mask.

The following figure shows an example of setting:

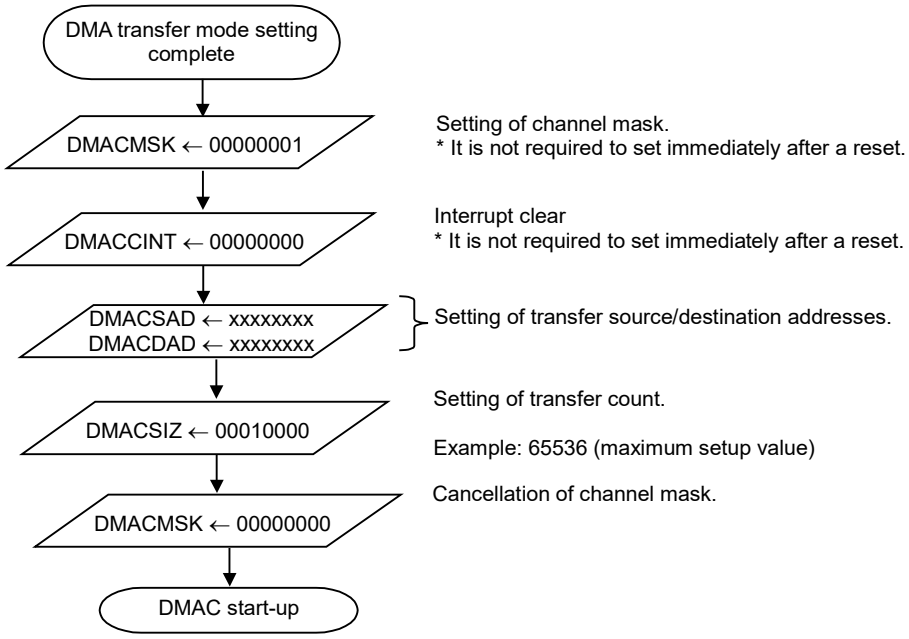


Figure 8-3 Example of setting

[Note]

- **If continuous DMA transfers are performed for the increment address device, the DMAC holds the next address in their respective registers (DMACSAD, DMACDAD) after the final transfer is terminated. Therefore, to transfer into successive address areas, the DMAC can be started without the setting of the transfer source and destination addresses (step 3 above). In this case, be sure to perform interrupt clear processing (step 2 above). The DMAC does not perform transfer unless interrupt clear processing is performed.**

8.4 Notes

8.4.1 Notes at Transfer Size Setting

The DMAC cannot set the bus width of the transfer source and destination individually. Therefore, the data transfer size that can be used for DMA transfer may be limited according to the connecting device (transfer source/transfer destination). Determine the transfer size by considering the following items.

- Data bus width of transfer source and transfer destination devices
  8 bits, 16 bits or 32 bits
- Connection bus of transfer source and transfer destination devices
  Peripheral circuit, or internal memory
- Device type of transfer source and transfer destination
  Increment address device or fixed address device

8.4.2 Notes at Accessing

It is prohibited to set the address of the register in the DMAC to the source address and the destination address. If such an access is attempted, the operation cannot be guaranteed.

Chapter 9

Time Base Counter

9. Time Base Counter

9.1 Overview

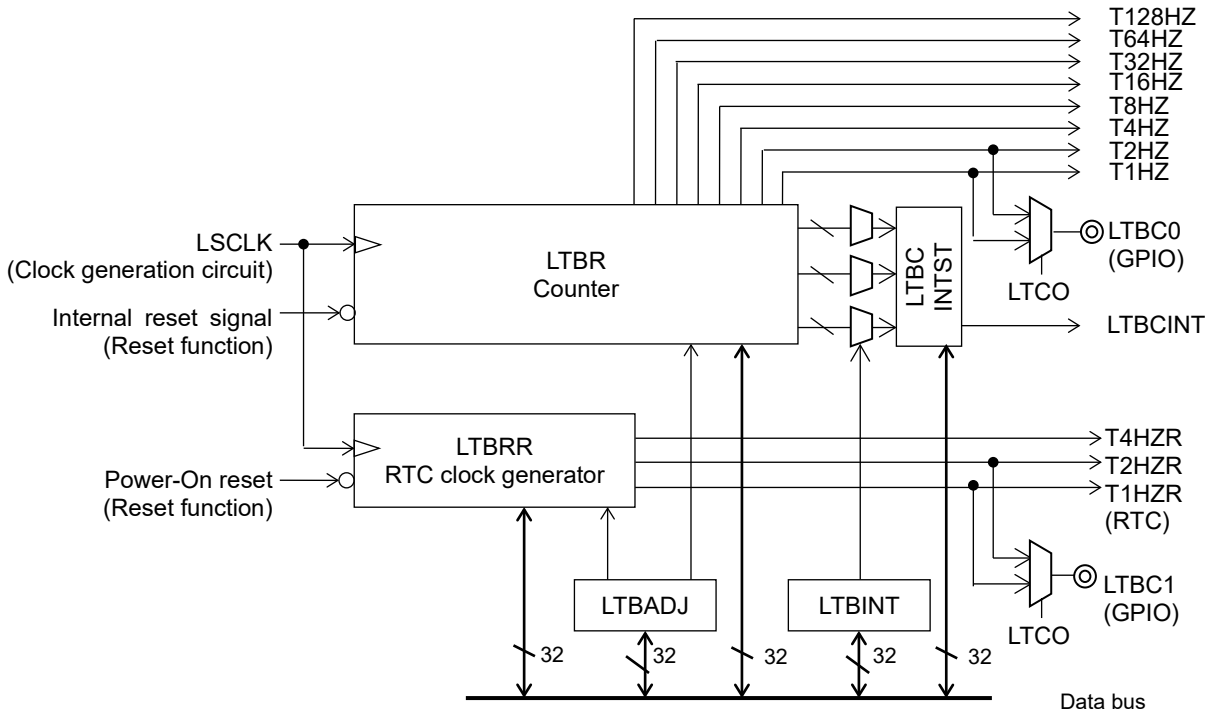
The time base counter generates base clocks for peripheral circuits, and generates interrupt periodically.

9.1.1 Features

- Generate eight frequency (128Hz, 64Hz, 32Hz, 16Hz, 8Hz, 4Hz, 2Hz and 1Hz) of pulse signals by dividing the low-speed clock (LSCLK).
- Three interrupt requests can be chosen among eight periodical interrupt requests.
- The 1Hz or 2Hz signal can be output from GPIO.
- The clock frequency adjust function
  - Allows to adjust in a range approx.-488ppm to +488ppm with the resolution approx.0.119ppm.
- The 1Hz or 2Hz signal is used for the RTC clock

9.1.2 Configuration

Figure 9-1 show the configuration of a low-speed time base counter (LTBC) respectively.



LTBR	: Low-speed time base counter register
LTBADJ	: Low-speed time base counter frequency adjust register
LTBINT	: Low-speed time base counter interrupt select register
LTBCINTST	: Low-speed time base counter interrupt status register
LTBRR	: Low-speed time base counter for RTC register

Figure 9-1 Configuration of Low-Speed Time Base Counter (LTBC)

9.1.3 List of Pins

Table 9-1 List of Pins

Pin name	I/O	Description
LTBC0	O	T1HZ/T2HZ output
LTBC1	O	T1HZR/T2HZR output

9.2 Description of Registers

9.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_4000	LTBC base address	LTBC	-	-	-
0x00	Low-speed time base counter register	LTBR	R/W	32	0x0000_0000
0x04	Low-speed time base counter frequency adjustment register	LTBADJ	R/W	32	0x0000_0000
0x08	Low-speed time base counter interrupt select resister	LTBINT	R/W	32	0x0000_0888
0x0C	Low-speed time base counter interrupt status register	LTBCINTST	R/W	32	0x0000_0000
0x10	Low-speed time base counter for RTC register	LTBRR	R	32	0x0000_0000

9.2.2 Low-Speed Time Base Counter Register (LTBR)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	T1HZ	T2HZ	T4HZ	T8HZ	T16HZ	T32HZ	T64HZ	T128HZ
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LTBR is a special function register (SFR) to read the T128HZ-T1HZ outputs of the low-speed time base counter. When write to LTBR, the content of LTBR becomes “0” regardless of the write data.

[Note]

- **LTBC interrupts may occur depending on the LTBR write timing. See “9.3.1 Low-Speed Time Base Counter”.**

9.2.3 Low-Speed Time Base Counter Frequency Adjustment Register (LTBADJ)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	LADJS	LADJ											
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LTBADJ is a special function register (SFR) to set the frequency adjustment values of the low-speed time base clock. See “9.3.2 Time Base Counter Frequency Adjustment Function” for the relation of the setting data and the adjustable ppm.

Bit No	Bit name	Description
12	LADJS	LADJS is used to specify sign of the frequency adjustment ratio. 0: Positive (Initial value) 1: Negative
11 ~ 0	LADJ	LADJ is used to specify the frequency adjustment ratio.

9.2.4 Low-Speed Time Base Counter Interrupt Select Register (LTBINT)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	LTCO
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	LTI2S				LTI1S				LTI0S			
R/W	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

LTBINT is a special function register (SFR) which specify low speed time base clock which is used as an interrupt signal.

Bit No	Bit name	Description
16	LTCO	LTCO is used to select clock that is output from LTBC0/LTBC1. 0: T1HZ/T1HZR (Initial value) 1: T2HZ/T2HZR
11 ~ 8	LTI2S	LTI2S is used to select clock that is assigned to LTBINT2. See a description of LTI0S bit about setting value.
7 ~ 4	LTI1S	LTI1S is used to select clock that is assigned to LTBINT1. See a description of LTI0S bit about setting value.
3 ~ 0	LTI0S	LTI0S is used to select clock that is assigned to LTBINT0. 0000: T128HZ 0001: T64HZ 0010: T32HZ 0011: T16HZ 0100: T8HZ 0101: T4HZ 0110: T2HZ 0111: T1HZ 1000: Disabled interrupt (Initial value) Others : Setting prohibited

9.2.5 Low-speed Time Base Counter Interrupt Status Register (LTBCINTST)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	LTBINT2	LTBINT1	LTBINT0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LTBCINTST is a special function register (SFR) which indicates the cause of the interrupt.

Bit No	Bit name	Description
2	LTBINT2	This bit is set to "1" when LTBINT2 interrupt request is generated. The LTBINT2 interrupt can be clear by writing "1" to this bit.
1	LTBINT1	This bit is set to "1" when LTBINT1 interrupt request is generated. The LTBINT1 interrupt can be clear by writing "1" to this bit.
0	LTBINT0	This bit is set to "1" when LTBINT0 interrupt request is generated. The LTBINT0 interrupt can be clear by writing "1" to this bit.

9.2.6 Low-Speed Time Base Counter for RTC Register (LTBRR)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	T1HZR	T2HZR	T4HZR	T8HZR	T16HZR	T32HZR	T64HZR	T128HZR
R/W	–	–	–	–	–	–	–	–	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LTBRR is a special function register (SFR) to read the T128HZR-T1HZR outputs of the low-speed time base counter for RTC.
When writing to one of the registers of RTCSEC, RTCMIN, RTCHOUR, RTCWEEK, RTCDAY, RTCMON and RTCYEAR, RTC time base counter register (LTBRR) is cleared.

9.3 Description of Operation

9.3.1 Low-Speed Time Base Counter

The low-speed time base counter (LTBC) starts counting from 0x0000 on the LSCLK falling edge after system reset. Three of LTBC interrupt are generated by falling edge of clock output which was assigned by the low-speed time base counter interrupt select register (LTBINT).
The T128HZ to T1HZ output of LTBC can be read from the low-speed time base counter register (LTBR). When reading the data, read LTBR twice and check that the two values coincide to prevent reading of undefined data during counting. Figure 9-2 shows an example of program to read LTBR.

```
unsigned int ltbr_t;
do {
      ltbr_t = LTBR;              ; First read
}while(ltbr_t != LTBR);           ; Second read and compare
```

Figure 9-2 Programming Example for Reading LTBR

LTBR is reset when write operation is performed and the T128HZ to T1HZ outputs are set to “0”. At this time, interrupt occurs when clock is assigned to LTBC interrupt changing from “1” to “0”. Therefore, when LTBR is reset, after prohibits LTBC interrupt of interrupt controller, LTBR is reset and the processing which clears LTBR interrupt request which occurred by reset is needed. Figure 9-3 shows the sequence to clear the LTBC interrupt request.
The system clock needs to be the low-speed clock, when a writing the LTBR register.

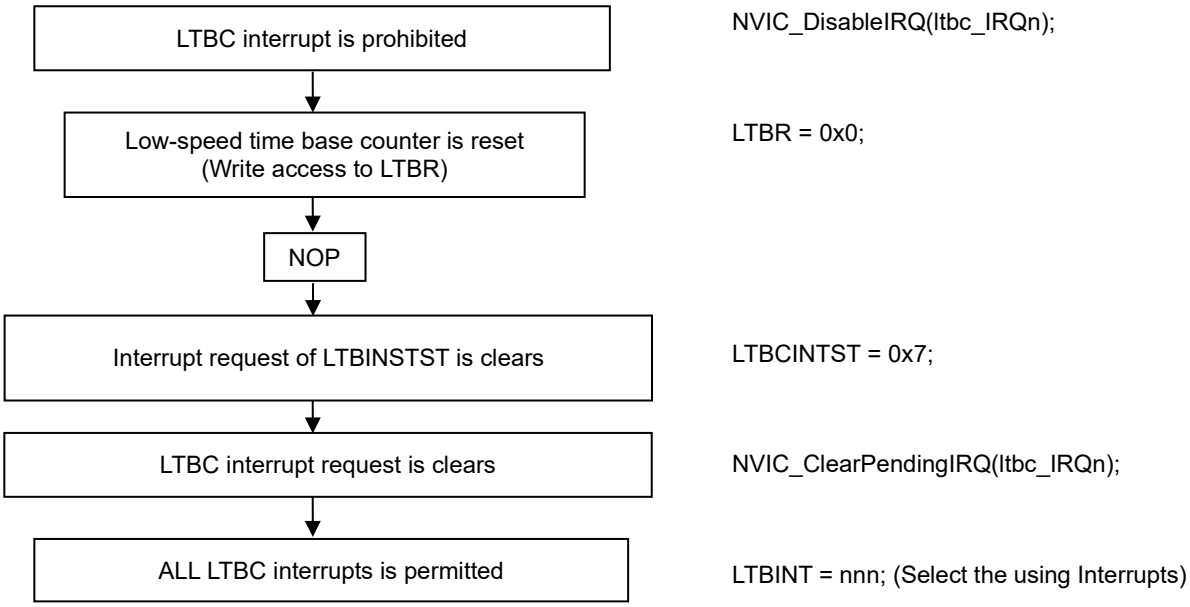


Figure 9-3 Sequence to Clear the LTBC Interrupt Request which Occurred by LTBR Reset

1 CPU cycle is needed after LTBR interrupt occurs until LTBC interrupt request flag of interrupt controller is set. When LTBC interrupt request is cleared after the writing LTBR, please do not put the order to clear request flag just after an order to write in LTBR at. Please clear request flag after placing NOP and putting time.

Figure 9-4 shows interrupt generation timing of the time base counter output by writing to LTBR.

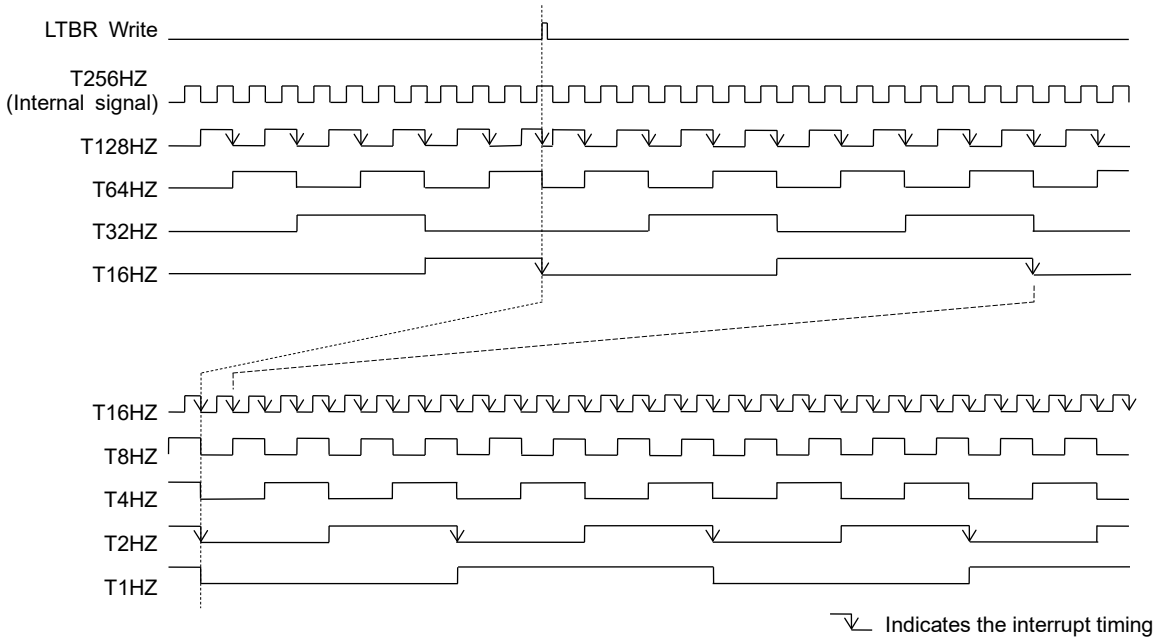


Figure 9-4 Interrupt Timing by Writing to LTBR

9.3.2 Low-speed Time Base Counter Frequency Adjustment Function

For T128HZ to T1HZ, T128HZR and T1HZR of the low-speed time base counter, the frequency can be adjusted using the low-speed time base counter frequency adjustment register (LTBADJ).
Measure the signal output from the LTBC0 or LTBC1 pin, then adjust the frequency using the LTBADJ register.
The adjustment range and resolution are as follows:

- Adjustment range : Approx. -488 ppm to +488 ppm
- Adjustment resolution : Approx. 0.119 ppm

The following mode is available to confirm the adjusted frequency:

Frequency adjustment mode	Description
Normal frequency adjustment mode	This mode is used to confirm that 256 seconds includes exactly 256 cycles/512 cycles of T1HZR/T2HZR, which is output form pin as LTBC1 under operating with actual adjusted low-speed clock.

Table 9-2 shows adjustment ratio corresponding to setting value set in the LTBADJ.

Table 9-2 Frequency adjustment value set in the LTBADJ and Adjustment ratio

LADJS	LADJ[11: 0]												Hexadecimal	Frequency adjustment ratio (ppm)
0	1	1	1	1	1	1	1	1	1	1	1	1	0x0FFF	+488.162
0	1	1	1	1	1	1	1	1	1	1	1	0	0x0FFE	+488.043
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
0	0	0	0	0	0	0	0	0	0	0	1	1	0x0003	+0.358
0	0	0	0	0	0	0	0	0	0	0	1	0	0x0002	+0.238
0	0	0	0	0	0	0	0	0	0	0	0	1	0x0001	+0.119
0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0x1000	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0x1001	–0.119
1	0	0	0	0	0	0	0	0	0	0	1	0	0x1002	–0.238
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	1	1	1	1	0	0x1FFE	–488.043
1	1	1	1	1	1	1	1	1	1	1	1	1	0x1FFF	–488.162

The correction values (LADJS, LADJ[11:0]) set in the LTBADJ register can be calculated using the following formula.

Correction value = Frequency adjustment ratio x 8388608 (decimal)
= Frequency adjustment ratio x 800000H (hexadecimal)

Example 1:	Example 2:
When adjusting +15.0 ppm (when the clock loses)	When adjusting -25.5 ppm (when the clock gains)
Correction value = +15.0 ppm x 8388608 (decimal) = +125.82912 (decimal) ≃ 0x7E (hexadecimal)	Correction value = 25.5 ppm x 8388608 (decimal) = 213.909504 (decimal) ≃ 0xD6 (hexadecimal) When setting LTBADJ, add a sign bit.
LTBADJ = 0x7E (hexadecimal)	LTBADJ = 0x10D6 (hexadecimal)

[Note]

- **The frequency adjustment accuracy does not guarantee the accuracy including the frequency variation of the low-speed oscillation (32.768 kHz) due to temperature variations.**

Chapter10

Timers

10. Timers

10.1 Overview

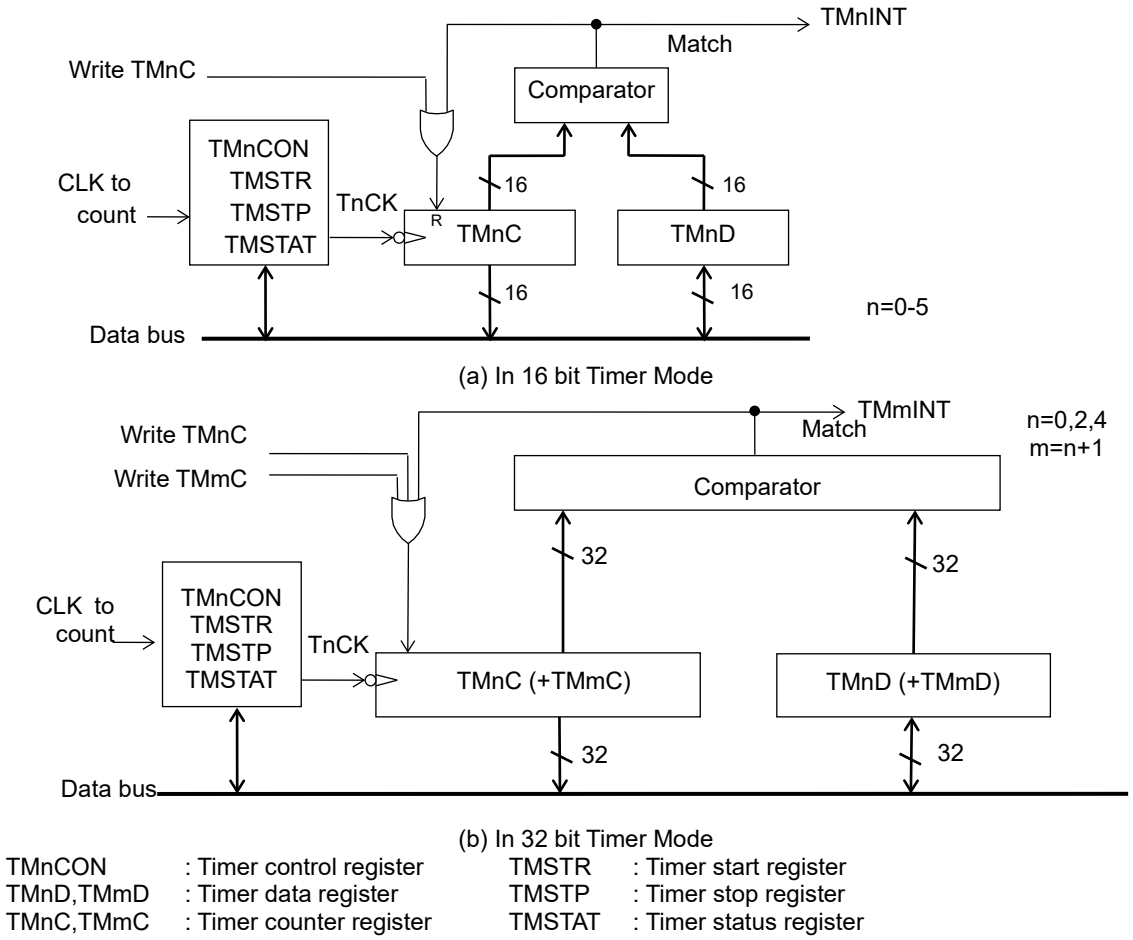
This LSI includes 6 channels of 16 bit timers. A pair of 2 timers functions as 32 bit timer.

10.1.1 Features

- The timer interrupt (TMnINT, n=0 to 5) is generated when the values of timer counter register (TMnC, n=0 to 5) and timer data register (TMnD, n=0 to 5) coincide.
- A timer configured by combining timer 0 and timer 1 / timer 2 and timer 3 / timer 4 and timer 5 can be used as a 32 bit timer.
- Low-speed clock (LSCLK), high-speed clock (OSCLK), and external input (TMCKI 0 to 5) are selectable as timer clock (selectable clock is different every channel).
- Timer clock can be divided by 1, 2, 4, 8, 16, 32, and 64 by divider function.

10.1.2 Configuration

Figure 10-1 shows the configuration of the timers.



TMnCON	: Timer control register	TMSTR	: Timer start register
TMnD,TMmD	: Timer data register	TMSTP	: Timer stop register
TMnC,TMmC	: Timer counter register	TMSTAT	: Timer status register

Figure 10-1 Configuration of Timers

10.2 Description of Registers

10.2.1 List of Registers

Ch	Address/Offset	Name	Symbol	R/W	Size	Initial value
-	0x4000_1000	Timer base address	TIMER	-	-	-
0	0x00	Timer 0 data register	TM0D	R/W	32	0x0000_FFFF
	0x04	Timer 0 counter register	TM0C	R/W	32	0x0000_0000
	0x08	Timer 0 control register	TM0CON	R/W	32	0x0000_0000
1	0x10	Timer 1 data register	TM1D	R/W	32	0x0000_FFFF
	0x14	Timer 1 counter register	TM1C	R/W	32	0x0000_0000
	0x18	Timer 1 control register	TM1CON	R/W	32	0x0000_0000
2	0x20	Timer 2 data register	TM2D	R/W	32	0x0000_FFFF
	0x24	Timer 2 counter register	TM2C	R/W	32	0x0000_0000
	0x28	Timer 2 control register	TM2CON	R/W	32	0x0000_0000
3	0x30	Timer 3 data register	TM3D	R/W	32	0x0000_FFFF
	0x34	Timer 3 counter register	TM3C	R/W	32	0x0000_0000
	0x38	Timer 3 control register	TM3CON	R/W	32	0x0000_0000
4	0x40	Timer 4 data register	TM2D	R/W	32	0x0000_FFFF
	0x44	Timer 4 counter register	TM2C	R/W	32	0x0000_0000
	0x48	Timer 4 control register	TM2CON	R/W	32	0x0000_0000
5	0x50	Timer 5 data register	TM3D	R/W	32	0x0000_FFFF
	0x54	Timer 5 counter register	TM3C	R/W	32	0x0000_0000
	0x58	Timer 5 control register	TM3CON	R/W	32	0x0000_0000
Common	0xF0	Timer start register	TMSTR	W	32	0x0000_0000
	0xF4	Timer stop register	TMSTP	W	32	0x0000_0000
	0xF8	Timer status register	TMSTAT	R	32	0x0000_0000

10.2.2 Timer n Data Register (TMnD : n=0, 2, 4)

Offset : 0x00 (TM0D), 0x20 (TM2D), 0x40 (TM4D)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TnD[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TnD[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TMnD (n=0, 2, 4) is a special function register (SFR) to set the value to be compared with the timer n counter register (TMnC) value.
The TMnD[15:0] is compared with the TMnC[15:0] in 16bit timer mode, and the TMnD[31:0] is compared with the TMnC[31:0] in 32bit timer mode.
In 32bit timer mode, TnD[31:16] is able to be accessed and the initial value is 0xFFFF.

[Note]
- **Set TMnD when the timer stops.**
- **In 16bit timer mode, TMnD is set to 0x0001 when 0x0000 is written to TMnD.**
- **In 32bit timer mode, TMnD is set to 0x0000_0001 when 0x0000_0000 is written to TMnD.**

10.2.3 Timer m Data Register (TMmD : m=1, 3, 5)

Offset : 0x10 (TM1D), 0x30 (TM3D), 0x50 (TM5D)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TmD															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TMmD (m=1, 3, 5) is a special function register (SFR) to set the value to be compared with the timer m counter register (TMmC) value.
The TmD[15:0] is compared with the TmC[15:0] in 16bit timer mode. The value of TMmD[15:0] is displayed TnD[31:16] (n=0, 2, 4).

[Note]
- **Set TMmD when the timer stops.**
- **In 16bit timer mode, TMmD is set to 0x0001 when 0x0000 is written to TMmD.**

10.2.4 Timer n Counter Register (TMnC : n=0, 2, 4)

Offset : 0x04(TM0C), 0x24(TM2C), 0x44(TM4C)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TnC[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TnC[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMnC (n=0, 2, 4) is a special function register (SFR) that functions as a 16/32 bit binary counter.
When some data is written to TMnC, TMnC is cleared to “0x0000_0000”.
In 32 bit timer mode, even if write to either TMnC timer counter or TM(n+1)C timer counter, both timer counters are cleared to 0x0000_0000.
In the case of the selection of timer clock and system clock shown in Table 10-1, TnC can be read even during timer operation.

10.2.5 Timer m Counter Register (TMmC : m=1, 3, 5)

Offset : 0x14(TM1C), 0x4000_1034(TM3C), 0x4000_1054(TM5C)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TmC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMmC (m=1, 3, 5) is a special function register (SFR) that functions as a 16 bit binary counter.
When some data is written to TmC is cleared to “0x0000”.
In 32 bit timer mode, even if write to either TMnC timer counter or TM(n+1)C timer counter, both timer counters are cleared to 0x0000_0000.
In the 32 bit timer mode, the counter value can be read by TnC[31:0] in the timer n counter register (N = 0, 2, 4).
In the case of the selection of timer clock and system clock shown in Table 10-1, TmC can be read even during timer operation.

Table 10-1 TnC/TmC Read Enable condition during Timer Operation

System clock	Timer clock
LSCLK	LSCLK and divided LSCLK
HSCLK	OSCLK and divided OSCLK However, when frequency of SYSCLK is more than timer clock.

[Note]

- **When using the 32bit timer mode, and also if you restart the timer after the timer is stopped by the software or automatically stopped in one shot timer mode, always reset the timer counter register (TMmC, TMnC) to 0x0000_0000 by making a write even if these registers value is 0x0000.**

10.2.6 Timer n Control Register (TMnCON : n=0, 2, 4)

Offset : 0x08(TM0CON), 0x28(TM2CON) , 0x48(TM4CON)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TnmM32	-	-	-	TnOST	-	TnDIV			-	-	TnCS	
R/W	–	–	–	R/W	–	–	–	R/W	–	R/W	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMnCON is a special function register(SFR) to set timer modes.

Bit No	Bit name	Description
12	TnmM32	TnmM32 is used for selecting a 32 bit timer mode. 0: 16 bit timer mode (Initial value) 1: 32 bit timer mode. Two timers are connected and function as a 32 bit timer. The TnmM32 of each timer control register, a pair of timers to be concatenated, and the interrupts used are as follows.
8	TnOST	TnOST is used for selecting timer mode. 0: Normal timer mode (Initial value) 1: One-shot timer mode In 32bit timer mode, Timer function mode is selected by TnOST.
6 to 4	TnDIV	TnDIV is used for selecting dividing rate of operation clock. 000: No divided (Initial value) 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Setting prohibited The TnDIV setting is used in both of 16 bit timer mode and 32bit timer mode.
1 to 0	TnCS	TnCS is used for selecting the operation clock of timer n. 00: LSCLK (Initial value) 01: OSCLK 10: LSCLK 11: TMCKI0 (Timer 0), TMCKI2 (Timer 2), TMCKI4 (Timer 4) See Chapter 22, “GPIO” for TMCKI. The TnCS setting is used in both of 16 bit timer mode and 32bit timer mode.

Register name	Combination channels	Interrupt
TM0CON	Timer1 and Timer0	Timer1
TM2CON	Timer3 and Timer2	Timer3
TM4CON	Timer5 and Timer4	Timer5

[Note]

- **Timer control register needs to be set while target timer is stop(TMSTAT register TnSTAT state is “0”)**
- **To supply OSCLK, both CTMn and CTMm bits in the CLKCON register need to be 1. See the Chapter 5 for the CLKCON register.**

10.2.7 Timer m Control Register (TMmCON : m=1, 3, 5)

Offset : 0x18(TM1CON), 0x38(TM3CON), 0x58(TM5CON)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TmOST	-	TmDIV			-	-	TmCS	
R/W	–	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMmCON is a special function register (SFR) to set timer modes.
This register is not used in 32-bit timer mode.

Bit No	Bit name	Description
8	TmOST	TmOST is used for selecting timer mode. 0: Normal timer mode (Initial value) 1: One-shot timer mode
6 to 4	TmDIV	TmDIV is used for selecting dividing rate of operation clock. 000: No divided (Initial value) 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Setting prohibited
1 to 0	TmCS	TmCS is used for selecting the operation clock of timer m. 00: LSCLK (Initial value) 01: OSCLK 10: LSCLK 11: TMCKI1 (Timer 1), TMCKI3 (Timer 3), TMCKI5 (Timer 5) See Chapter 22, “GPIO” for TMCKI.

[Note]
- **Timer control register needs to be set while target timer is stop (TMSTAT register TmSTAT state is “0”)**
- **To supply OSCLK, both CTMn and CTMm bits in the CLKCON register need to be 1. See the Chapter 5 for the CLKCON register.**

10.2.8 Timer Start Register (TMSTR)

Offset : 0xF0

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
R/W	–	–	–	–	–	–	–	–	–	–	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMSTR is a special function (SFR) to start the counter of timer 0 to timer 5.

[Description of each bit]
This is bit used to control counter of the corresponding timer. In initial state, the counter is stopped.
0: Writing is invalid
1: Start count

Bit No	Bit name	Description
5	T5RUN	Timer 5 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
4	T4RUN	Timer 4
3	T3RUN	Timer 3 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
2	T2RUN	Timer 2
1	T1RUN	Timer 1 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
0	T0RUN	Timer 0

10.2.9 Timer Stop Register (TMSTP)

Offset : 0xF4

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	T5STP	T4STP	T3STP	T2STP	T1STP	T0STP
R/W	–	–	–	–	–	–	–	–	–	–	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMSTP is a special function (SFR) to stop the counter of timer 0 to timer 5.

[Description of each bit]
This is bit used to control counter of the corresponding timer. In initial state, the counter is stopped.
0: Writing is invalid
1: Stop count

Bit No	Bit name	Description
5	T5STP	Timer 5 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
4	T4STP	Timer 4
3	T3STP	Timer 3 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
2	T2STP	Timer 2
1	T1STP	Timer 1 It is used only in 16 bit timer mode and setting prohibited in 32 bit timer mode.
0	T0STP	Timer 0

10.2.10 Timer Status Register (TMSTAT)

Offset : 0xF8

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	T5STAT	T4STAT	T3STAT	T2STAT	T1STAT	T0STAT
R/W	–	–	–	–	–	–	–	–	–	–	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMSTAT is a special function (SFR) to indicate the status of timer 0 to timer 5.

[Description of each bit]
This bit indicates a counting status of the corresponding timer.
0: Stopped (Initial value)
1: Counting

Bit No	Bit name	Description
5	T5STAT	Timer 5 It is used only in 16 bit timer mode. It always indicates “0” in 32 bit timer mode.
4	T4STAT	Timer 4
3	T3STAT	Timer 3 It is used only in 16 bit timer mode. It always indicates “0” in 32 bit timer mode.
2	T2STAT	Timer 2
1	T1STAT	Timer 1 It is used only in 16 bit timer mode. It always indicates “0” in 32 bit timer mode.
0	T0STAT	Timer 0

10.3 Description of Operation

10.3.1 Normal timer mode operation

When the TnRUN bit of timer n register (TMSTR) are set to “1”, the timer counter (TMnC) is in operation state (TnSTAT = “1”) by the first falling edge of the timer clock (TnCK) that are selected by the timer control register (TMnCON), and start to count up by the second falling edge.
When the count value of TMnC coincide with the timer data register (TMnD), timer interrupt (TMnINT) occurs on the next falling edge of timer clock, at same time TMnC are reset to 0x0000 and continues incremental count.

When the TnSTP bit are set to “1”, TMnC stop a count after one fall count of the timer clock (TnCK) , and TnSTAT bit of timer status register (TMSTAT) becomes “0”.

When the TnRUN bit are set to “1” again, TMnC restart an incremental count from the previous values. To initialize TMnC to “0x0000”, perform write operation in TMnC.

The timer interrupt period (T_{TMI}) is expressed by the following equation.

$$T_{TMI} = \frac{TMnD + 1}{TnCK\ (Hz)}$$

TMnD: Timer n data register (TMnD) setting value (0x0001 to 0xFFFF)
TnCK: Clock frequency selected by the Timer n control register (TMnCON)

After TnRUN bit are set to "1", timer counter is synchronized by the timer clock and counting starts so that an error of a maximum of 1 clock period occurs until the first timer interrupt. The timer interrupt periods from the second time are constant.
Figure 10-2 shows the normal timer mode operation timing diagram of Timer n.

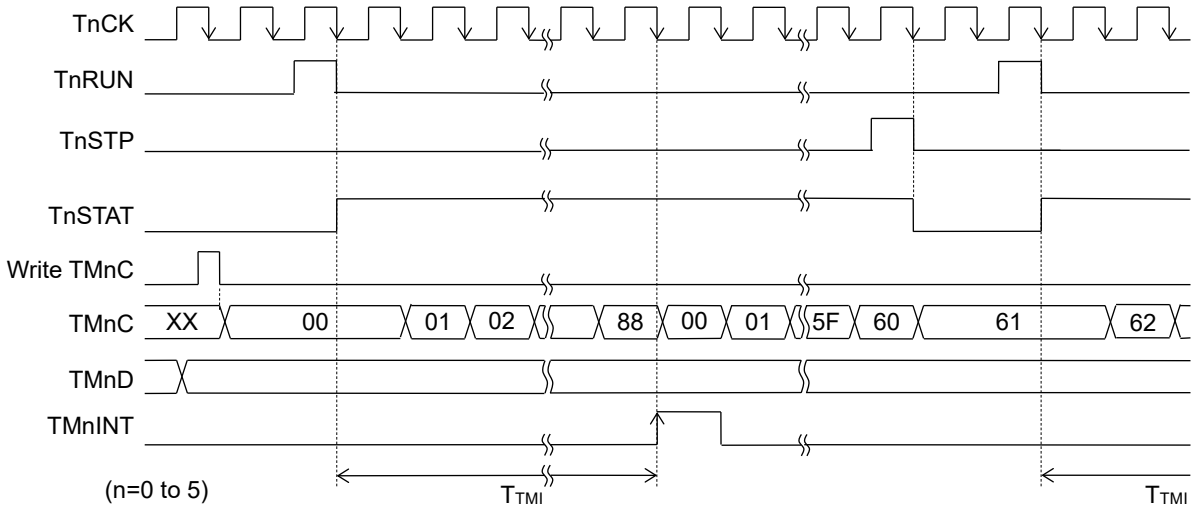


Figure 10-2 Normal Timer Mode Operation Timing Diagram of Timer n

[Note]

- **Count stop and timer interrupt may occur at same time because counter stop operation is performed synchronizing with count operation.**

10.3.2 One shot timer mode operation

When TMnCON register TnOST bit set to "1", the timer operates one-shot timer mode.
In one-shot timer mode, when the count value (TMnC) and the timer n data register (TMnD) coincide, TnRUN bit is cleared automatically.

Figure 10-3 shows the one-shot timer mode operation timing diagram

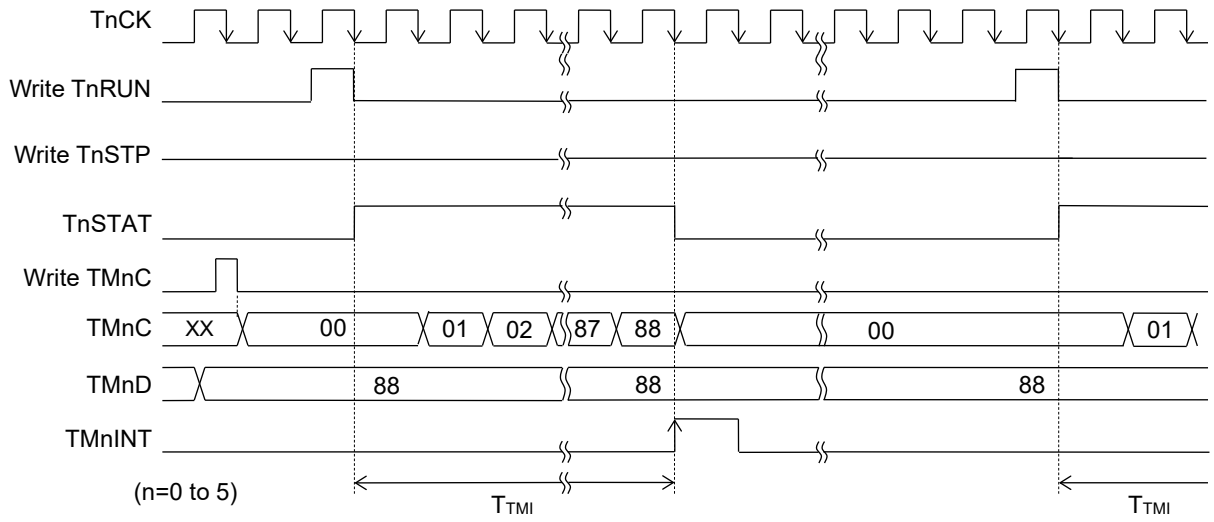


Figure 10-3 One-Shot Timer Mode Operation Timing Diagram

10.3.3 32 bit timer mode

Two of 16bit timer can be used as 32bit timer by TnmM32 bit of TMnCON (n = 0, 2, 4) register.
The following shows a corresponding list of timer-channels and related registers.

Table 10-2 Used registers in 32 bit timer mode

Control \ Channel	0,1	2,3	4,5
Data register	TM0D	TM2D	TM4D
Counter register	TM0C	TM2C	TM4C
Control register	TM0CON	TM2CON	TM4CON
RUN bit	T0RUN	T2RUN	T4RUN
STOP bit	T0STP	T2STP	T4STP
STAT bit	T0STAT	T2STAT	T4STAT
Interrupt	TM1INT	TM3INT	TM5INT

Chapter 11

Functional Timer (FTM)

11. Functional Timer (FTM)

11.1 Overview

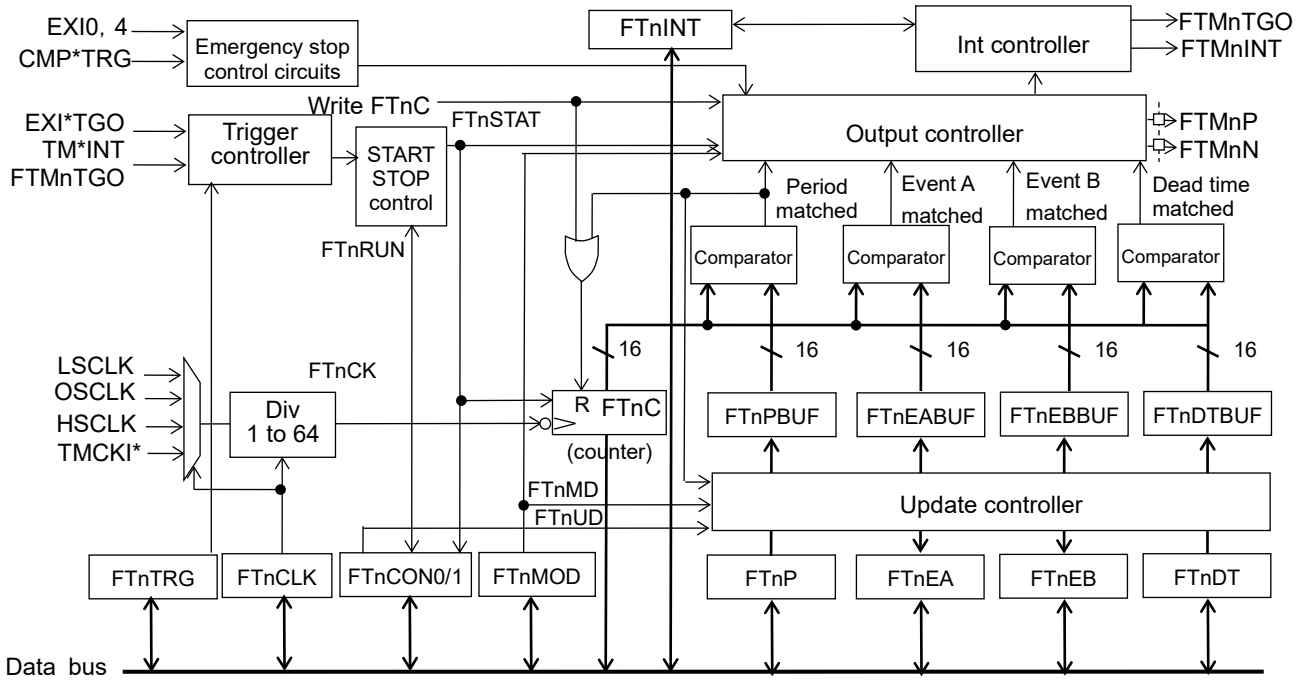
FTM is a 16 bit functional timer with the capture and PWM functions in addition to the timer function. It can be started/stopped using an external input signal and a signal from another timer as a trigger.
The LSI includes two channels of the functional timer.

11.1.1 Features

- Equipped with the timer/capture/PWM functions using a 16 bit counter
- 1 to 64 dividing of LSCLK/OSCLK/HSCLK/external input clock selectable as timer clock
- The timer output signal can be switched between the positive and negative logics
- Duty interrupt and coincident interrupt with the setting value as well as the cyclic interrupt generated
- Equipped with one-shot mode
- An event trigger (external pin input interrupt or timer interrupt request) can control start/stop/clear of the timer (however, the minimum pulse width of pin input is timer clock 3ϕ)
- An external input and comparator output can generate an emergency stop and emergency stop interrupt.
- Two types of PWM with the same period and different duties and complementary PWM with the dead time set can be output
- The capture function can measure the duty/cycle of the input signal
- Interrupt source to be notified can be set

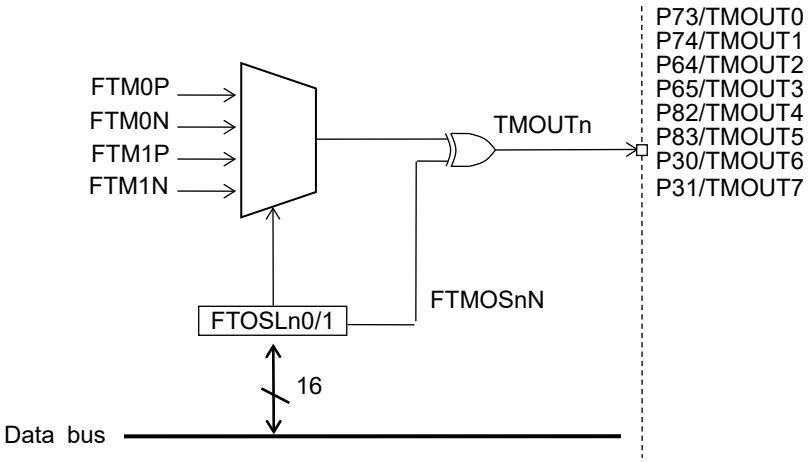
11.1.2 Configuration

Figure 11-1 shows the configuration of the FTM circuit.



FTnP : FTMn period register
FTnEA/B : FTMn event A/B register
FTnDT : FTMn dead time register
FTnC : FTMn counter register
FTnCON0/1 : FTMn control register 0,1
FTnTRG : FTMn trigger register
FTnINTE/S/C : FTMn interrupt enable/status/clear register
FTnPBUF : FTMn period buffer
FTnEA/BBUF : FTMn event A/B buffer
FTnDTBUF : FTMn dead time buffer
FTnMOD : FTMn mode register
FTnCLK : FTMn clock register

(a) Configuration of Circuit



FTOSLn : FTM output selection register n (n = 0,4)

(b) Output Selection Part

Figure 11-1 Configuration of FTM

11.1.3 List of Pins

Table 11-1 Pins

Pin name	I/O	Description
TMOUT0-7	O	Timer output (Selectable from FTM0-1)

11.2 Description of Registers

11.2.1 List of Registers

Ch	Address/Offset	Name	Symbol	R/W	Size	Initial value
-	0x4000_2000	FTM base address	FTIMER	-	-	-
0	0x000	FTM0 period register	FT0P	R/W	32	0x0000_FFFF
	0x004	FTM0 event register A	FT0EA	R/W	32	0x0000_0000
	0x008	FTM0 event register B	FT0EB	R/W	32	0x0000_0000
	0x00C	FTM0 dead time register	FT0DT	R/W	32	0x0000_0000
	0x010	FTM0 counter register	FT0C	R/W	32	0x0000_0000
	0x014	FTM0 control register 0	FT0CON0	R/W	32	0x0000_0000
	0x018	FTM0 control register 1	FT0CON1	R/W	32	0x0000_0000
	0x01C	FTM0 mode register	FT0MOD	R/W	32	0x0000_0000
	0x020	FTM0 clock register	FT0CLK	R/W	32	0x0000_0000
	0x024	FTM0 trigger register	FT0TRG	R/W	32	0x0000_0000
	0x030	FTM0 interrupt enable register	FT0INTE	R/W	32	0x0000_0000
	0x034	FTM0 interrupt status register	FT0INTS	R	32	0x0000_0000
	0x038	FTM0 interrupt clear register	FT0INTC	W	32	0x0000_0000
1	0x100	FTM1 period register	FT1P	R/W	32	0x0000_FFFF
	0x104	FTM1 event register A	FT1EA	R/W	32	0x0000_0000
	0x108	FTM1 event register B	FT1EB	R/W	32	0x0000_0000
	0x10C	FTM1 dead time register	FT1DT	R/W	32	0x0000_0000
	0x110	FTM1 counter register	FT1C	R/W	32	0x0000_0000
	0x114	FTM1 control register 0	FT1CON0	R/W	32	0x0000_0000
	0x118	FTM1 control register 1	FT1CON1	R/W	32	0x0000_0000
	0x11C	FTM1 mode register	FT1MOD	R/W	32	0x0000_0000
	0x120	FTM1 clock register	FT1CLK	R/W	32	0x0000_0000
	0x124	FTM1 trigger register	FT1TRG	R/W	32	0x0000_0000
	0x130	FTM1 interrupt enable register	FT1INTE	R/W	32	0x0000_0000
	0x134	FTM1 interrupt status register	FT1INTS	R	32	0x0000_0000
	0x138	FTM1 interrupt clear register	FT1INTC	W	32	0x0000_0000
Port	0xF00	FTM output select register 0	FTOSL0	R/W	32	0x0000_0000
	0xF04	FTM output select register 4	FTOSL4	R/W	32	0x0000_0000

11.2.2 FTMn Period Register (FTnP : n=0,1)

Offset : 0x000(FT0P), 0x100(FT1P)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnP															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FTnP is a special function register (SFR) used to set the cycle (clock count of one cycle) of FTMn.
Set this register after setting the operation mode using FTnMD.

Bit No	Bit name	Description
15 to 0	FTnP	This is used to set the one period of counter. 0x0001 to 0xFFFF : setting value + 1 clock When 0x0000 is written to FTnP, one period is set to 2 clocks.

11.2.3 FTMn Event Register A (FTnEA : n=0,1)

Offset : 0x004(FT0EA), 0x104(FT1EA)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnEA															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnEA is a special function register (SFR) used to set the event timing of FTMn or indicate the captured data.
Set this register after setting the operation mode using FTnMD.

Bit No	Bit name	Description	
15 to 0	FTnEA	TIMER mode 0x0000-0xFFFF:	Set the count value to generate an interrupt. (Interrupt timing is FTnEA setting value + 1) This value must be less than the period register FTnP.
		CAPTURE mode 0x0000-0xFFFF:	The captured count value is stored. When it is read, FTnFLGA/FTnISA is cleared. However, FTnFLGA is cleared only when FTnTGEN = “1”. See section 11.3.6.2 as for clearing FTnFLGA. In the CAPTURE mode, writing to FTnEA is disabled.
		PWM1 mode 0x0000-0xFFFF:	Set the duty of PWM output FTMnP of FTMn.
		PWM2 mode 0x0000-0xFFFF:	Set the duty of PWM output FTMnP and FTMnN of FTMn.

11.2.4 FTMn Event Register B (FTnEB : n=0,1)

Offset : 0x008(FT0EB), 0x108(FT1EB)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnEB															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnEB is a special function register (SFR) used to set the event timing of FTMn or indicate the captured data.
Set this register after setting the operation mode using FTnMD.

Bit No	Bit name	Description	
15 to 0	FTnEB	TIMER mode 0x0000-0xFFFF:	Set the count value to generate an interrupt. (Interrupt timing is FTnEB setting value + 1) This value must be less than the period register FTnP.
		CAPTURE mode 0x0000-0xFFFF:	The captured count value is stored. When it is read, FTnFLGB/FTnISB is cleared. However, FTnFLGB is cleared only when FTnTGEN = “1”. See section 11.3.6.2 as for clearing FTnFLGB. In the CAPTURE mode, writing to FTnEB is disabled.
		PWM1 mode 0x0000-0xFFFF:	Set the duty of PWM output FTMnN of FTMn.
		PWM2 mode	Set FTnIEB/FTnIOB to 0 in this mode.

11.2.5 FTMn Dead Time Register (FTnDT : n=0,1)

Offset : 0x00C(FT0DT), 0x10C(FT1DT)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnDT															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnDT is a special function register (SFR) used to set the dead time of the timer output.
Set this register after setting the operation mode using FTnMD.

Bit No	Bit name	Description
15 to 0	FTnDT	This is used to set dead time of timer output The dead-time width is FTnDT setting value + 1. This is enabled when FTnDTEN bit of FTnMOD register is set to "1". This is always disabled in the CAPTURE mode.

11.2.6 FTMn Counter Register (FTnC : n=0,1)

Offset : 0x010(FT0C), 0x110(FT1C)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnC is a special function register (SFR) used to indicate the counter value of FTMn.
When writing to this register, the counter is cleared to 0x0000.
This register should be accessed while the counter is stopped.

11.2.7 FTMn Control Register 0 (FTnCON0 : n=0,1)

Offset : 0x014(FT0CON0), 0x114(FT1CON0)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	FTnSDN	-	-	-	FTnEMGEN	-	-	-	FTnTGEN	-	-	-	FTnRUN
R/W	–	–	–	R/W	–	–	–	R/W	–	–	–	R/W	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnCON0 is a special function register (SFR) used to set the function of FTMn.

Bit No	Bit name	Description
12	FTnSDN	This is used to mask the PWM output to “L”. 0: Release mask (Initial value) 1: masked This is disabled in the CAPTURE mode.
8	FTnEMGEN	This is used to enable emergency stop function. 0: Disabled (Initial value) 1: Enabled This is disabled in the CAPTURE mode.
4	FTnTGEN	This is used to enable a function to start/stop by event trigger. 0: Disabled (Initial value) 1: Enabled
0	FTnRUN	This is used to start/stop counter by software. 0: Stop counting (Initial value) 1: Start counting or during counting

11.2.8 FTMn Control Register 1 (FTnCON1 : n=0,1)

Offset : 0x018(FT0CON1), 0x118(FT1CON1)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	FTnSTAT	FTnFLGC	FTnFLGB	FTnFLGA	-	-	-	FTnUD
R/W	-	-	-	-	-	-	-	-	R	R	R	R	-	-	-	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnCON1 is a special function register (SFR) used to set the function of FTMn.

Bit No	Bit name	Description
7	FTnSTAT	This indicates operation status of the counter. 0: Counter stopped (Initial value) 1: Counter running
6	FTnFLGC	This indicates the control state by the CST bit of FTnTRG register. This is cleared by reading the FTnC register. 0: State enabled to start by event trigger. (Initial value) 1: State unabled to start by event trigger.
5	FTnFLGB	This indicates a state of event timing B. [TIMER, PWM1, PWM2 mode] 0: Counter value < value of event register B (Initial value) 1: Counter value ≥ value of event register B [CAPTURE mode] 0: Capture data not available (Initial value) 1: Capture data available It is cleared when FTnEB is read if FTnTGEN = “1”. See section 11.3.6.2 as for clearing FTnFLGB.
4	FTnFLGA	This indicates a state of event timing A. [TIMER, PWM1, PWM2 mode] 0: Counter value < value of event register A (Initial value) 1: Counter value ≥ value of event register A [CAPTURE mode] 0: Capture data not available (Initial value) 1: Capture data available It is cleared when FTnEA is read if FTnTGEN = “1”. See section 11.3.6.2 as for clearing FTnFLGA.
0	FTnUD	This bit is used to update FTnP, FTnEA, FTnEB and FTnDT of FTMn during operation. To update FTnP, FTnEA, FTnEB and FTnDT, write "1" to this bit after setting these registers. Writing "1" transfers the setting values to the internal buffer of FTnP, FTnEA, FTnEB and FTnDT at the same time. When the transfer completes, this bit is cleared automatically. 0: Update completed (Initial value) 1: Requesting update

11.2.9 FTMn Mode Register (FTnMOD : n=0,1)

Offset : 0x01C(FT0MOD), 0x11C(FT1MOD)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	FTnSTPO	FTnOST	FTnDTEN	-	-	-	-	FTnMD	
R/W	-	-	-	-	-	-	-	R/W	R/W	R/W	-	-	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnMOD is a special function register (SFR) used to set the function of FTMn.

Bit No	Bit name	Description
8	FTnSTPO	This is used to set the output state when the counter stops. 0: Output to "L" at stop. (Initial value) If restarted without clearing the counter, it is "L" until the next period. 1: The current output state is kept after stop. When restarted without clearing the counter, the output depends on the counter value This bit is disabled in the CAPTURE mode.
7	FTnOST	This is used to select auto-reload / one-shot mode. [TIMER, PWM1, PWM2 mode] 0: Auto-reload mode (Initial value) 1: One-shot mode [CAPTURE mode] 0: Auto mode Even if the capture is performed once, the data of EA and EB is overwritten (updated) when the next capture is performed. When the counter goes around, it restarts from 0. 1: Single mode Once captured into EA or EB, the next capture is not performed before read. When the counter goes around, it stops.
6	FTnDTEN	This is used to enable dead time. 0: Disabled (Initial value) 1: Enabled This bit is disabled in the CAPTURE mode.
1 to 0	FTnMD	This is used to set function mode. 00: TIMER mode (Initial value) 01: CAPTURE mode 10: PWM1 mode 11: PWM2 mode

[Note]

- **When using the One-shot mode / Single mode, set to “1” FTnIEP of FTnINTE register and confirm that FTnISP of FTnINTS register is “0”.**
- **Initialize this peripheral with block reset before changing to another mode, if it is in the operation state once.**

11.2.10 FTMn Clock Register (FTnCLK : n=0,1)

Offset : 0x020(FT0CLK), 0x120(FT1CLK)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	FTnXCK			-	FTnCKD			-	-	FTnCK	
R/W	-	-	-	-	-	R/W	R/W	R/W	-	R/W	R/W	R/W	-	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnCLK is a special function register (SFR) used to set the function of FTMn.

Bit No	Bit name	Description
10 to 8	FTnXCK	This is used to select the source when selecting EXTCLK as a timer clock source. See Chapter22 “GPIO” for TMCKI. 000: TMCKI0 (Initial value) 001: TMCKI1 010: TMCKI2 011: TMCKI3 100: TMCKI4 101: TMCKI5 110: TMCKI6 111: TMCKI7
6 to 4	FTnCKD	This is used to select the dividing ratio of the timer clock source. 000: No dividing (Initial value) 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Setting prohibited
1 to 0	FTnCK	This is used to select timer clock source. 00: LSCLK (Initial value) 01: OSCLK 10: HSCLK 11: EXTCLK (selected by FTnXCK)

11.2.11 FTMn Trigger Register (FTnTRG : n=0,1)

Offset : 0x024(FT0TRG), 0x124(FT1TRG)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	FTnEST		-	-	FTnTRM	
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnSTSS	-	-	-	FTnSTS				-	-	-	-	FTnCST	FTnEXCL	FTnST1	FTnST0
R/W	R/W	–	–	–	R/W	R/W	R/W	R/W	–	–	–	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnTRG is a special function register (SFR) used to set the function of FTMn.

Bit No	Bit name	Description
21 to 20	FTnEST	This is used to select the emergency stop trigger source. This bit is effective only when FTnEMGEN is “1”. 00: Rising edge of EXI0TGO (Initial value) 01: Rising edge of EXI4TGO 10: Rising edge of CMP0 11: Rising edge of CMP1
17 to 16	FTnTRM	This is used to select the edge of the event trigger. It is enabled only when EXI0-7TGO is selected as the event trigger source. Otherwise, it is fixed to the rising edge. Counter start / Counter stop/clear 00: Rising edge / Rising edge (Initial value) 01: Falling edge / Rising edge 10: Rising edge / Falling edge 11: Falling edge / Falling edge
15, 11 to 8	FTnSTSS FTnSTS	This is used to select the source of the event trigger. Do not select itself, for example, FTM0 for the FTM0 setting. FTnSTS FTnSTS 0 0000: EXI0TGO (Initial value) 0 0001: EXI1TGO 0 0010: EXI2TGO 0 0011: EXI3TGO 0 0100: EXI4TGO 0 0101: EXI5TGO 0 0110: EXI6TGO 0 0111: EXI7TGO 1 0000: TM0INT 1 0001: TM1INT 1 0010: TM2INT 1 0011: TM3INT 1 0100: TM4INT 1 0101: TM5INT 1 1000: FTM0TGO 1 1001: FTM1TGO Others: Setting prohibited EXInTGO is the trigger signal from external input. The timer interrupt request (TMnINT) is an interrupt request signal independent of the interrupt enabled/disabled setting of the interrupt enable register. FTM trigger output (FTMnTGO) is used only for event trigger.
3	FTnCST	This is used to select the operation mode of starting the counter by event trigger. 0: An event trigger always starts the counter when it is stopped (except for emergency stop) (Initial value) 1: An event trigger does not start the counter before FTnC is read when it is stopped except for emergency stop
2	FTnEXCL	This is used to select whether the counter is cleared when an event trigger stops it. It is not cleared at emergency stop regardless of this bit setting. 0: Disabled clearing the counter (Initial value) 1: Enabled clearing the counter

Bit No	Bit name	Description
1	FTnST1	This is used to select whether an event trigger stops the counter. 0: Disabled stopping counter (Initial value) 1: Enabled stopping counter
0	FTnST0	This is used to selects whether an event trigger starts the counter. 0: Disabled starting counter (Initial value) 1: Enabled starting counter

11.2.12 FTMn Interrupt Enable Register (FTnINTE : n=0,1)

Offset : 0x030(FT0INTE), 0x130(FT1INTE)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	FTnIOB	FTnIOA	FTnIOP	-	-	-	FTnIETR	FTnIETS	FTnIEB	FTnIEA	FTnIEP
R/W	-	-	-	-	-	R/W	R/W	R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnINTE is a special function register (SFR) used to control the interrupt of FTMn.

[Description of each bit]
This is used to enable the corresponding interrupt / function.
0: Disabled (Initial value)
1: Enabled

Bit No	Bit name	Description (corresponding function)
10	FTnIOB	Outputs an event timing B interrupt request as a trigger (FTMnTGO) for other peripherals. Set FTnIOB to “0” in PWM2 mode.
9	FTnIOA	Outputs an event timing A interrupt request as a trigger (FTMnTGO) for other peripherals.
8	FTnIOP	Outputs a periodic interrupt request as a trigger (FTMnTGO) for other peripherals.
4	FTnIETR	Trigger counter start interrupt
3	FTnIETS	Trigger counter stop interrupt
2	FTnIEB	Event timing B interrupt / Capture B interrupt Set FTnIEB to “0” in PWM2 mode.
1	FTnIEA	Event timing A interrupt / Capture A interrupt
0	FTnIEP	Periodic interrupt

11.2.13 FTMn Interrupt Status Register (FTnINTS : n=0,1)

Offset : 0x034(FT0INTS), 0x134(FT1INTS)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	FTnISES	FTnISTR	FTnISTS	FTnISB	FTnISA	FTnISP
R/W	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnINTS is a special function register (SFR) used to indicate the interrupt status of FTMn.

[Description of each bit]
This is used to indicate the corresponding interrupt request.
0: It has not occurred (Initial value)
1: It has occurred

Bit No	Bit name	Description (corresponding function)
5	FTnISES	Emergency stop interrupt This bit is cleared when writing “1” to FTnICES.
4	FTnISTR	Trigger counter start interrupt This bit is cleared when writing “1” to FTnICTR.
3	FTnISTS	Trigger counter stop interrupt This bit is cleared when writing “1” to FTnICTS.
2	FTnISB	[TIMER/PWM1 mode] Event timing B interrupt This bit is cleared when writing “1” to FTnICB. [CAPTURE mode] Capture B interrupt It indicate that a captured data is stored to FTnEB. This bit is cleared when writing “1” to FTnICB or reading FTnEB
1	FTnISA	[TIMER/PWM1/PWM2 mode] Event timing A interrupt This bit is cleared when writing “1” to FTnICA. [CAPTURE mode] Capture A interrupt It indicates that a captured data is stored to FTnEA. This bit is cleared when writing “1” to FTnICA or reading FTnEA
0	FTnISP	Periodic interrupt This bit is cleared when writing “1” to FTnICP.

[Note]

- **New interrupts are not notified, if exiting the interrupt handler with some interrupt status remaining. Make sure FTnINTS is “0” before exiting the interrupt handler of FTMn. Or request interrupt of FTMn again with FTnIR bit in the FTnINTC register.**

11.2.14 FTMn Interrupt Clear Register (FTnINTC : n=0,1)

Offset : 0x038(FT0INTC), 0x138(FT1INTC)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTnIR	-	-	-	-	-	-	-	-	-	FTnISES	FTnISTR	FTnISTS	FTnISB	FTnISA	FTnISP
R/W	W	-	-	-	-	-	-	-	-	-	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTnINTC is a special function register (SFR) used to clear the interrupt status of FTMn.
When reading it, 0x0000_0000 is always read.

[Description of each bit 5 to 0]
It is used to clear corresponding interrupt request.
Writing “0”: Invalid
Writing “1”: Clear the corresponding interrupt request

Bit No	Bit name	Description (corresponding function)
15	FTnIR	This is used to request remaining interrupts. Write "1" to this bit at the end of an interrupt routine. Writing "0": Invalid Writing "1": If there is any unhandled interrupt source, the interrupt request is generated again.
5	FTnISES	Emergency stop interrupt
4	FTnISTR	Trigger counter start interrupt
3	FTnISTS	Trigger counter stop interrupt
2	FTnISTB	Event timing B interrupt / Capture B interrupt
1	FTnISTA	Event timing A interrupt / Capture A interrupt
0	FTnISTP	Periodic interrupt

11.2.15 FTM Output Select Register n (FTOSLn : n = 0, 4)

Offset : 0xF00(FT0INTC), 0xF04(FT1INTC)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	FTOSN(n+3)	–	FTOS (n+3)			–	–	–	FTOSN(n+2)	–	FTOS (n+2)		
R/W	–	–	–	R/W	–	R/W	R/W	R/W	–	–	–	R/W	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	-	FTOSN(n+1)	-	FTOS (n+1)			-	-	-	FTOSNn	–	FTOSn		
R/W	–	–	-	R/W	-	R/W	R/W	R/W	-	-	–	R/W	-	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTOSLn is a special function register (SFR) used to control assignment and output polarity of output FTMnP/FTMnN of FTM to TMOUT0 - 7.
FTOSLn (n=0, 4) corresponds to TMOUT0 - 7 as follows:

Table 11-2 TMOUT setting

Pin name	Functional pin name	Registers	FTOSNn	FTOSn
P73	TMOUT0	FTOSL0	FTOSN0 (bit 4)	FTOS0 (bit 2 to 0)
P74	TMOUT1		FTOSN1 (bit 12)	FTOS1 (bit 10 to 8)
P64	TMOUT2		FTOSN2 (bit 20)	FTOS2 (bit 18 to 16)
P65	TMOUT3		FTOSN3 (bit 28)	FTOS3 (bit 26 to 24)
P82	TMOUT4	FTOSL4	FTOSN4 (bit 4)	FTOS4 (bit 2 to 0)
P83	TMOUT5		FTOSN5 (bit 12)	FTOS5 (bit 10 to 8)
P30	TMOUT6		FTOSN6 (bit 20)	FTOS6 (bit 18 to 16)
P31	TMOUT7		FTOSN7 (bit 28)	FTOS7 (bit 26 to 24)

Bit No	Bit name	Description
28 20 12 4	FTOSN(n+3) FTOSN(n+2) FTOSN(n+1) FTOSNn	This is used to invert the FTM output. 0: Not invert (Initial value) 1: Invert
26 to 24 18 to 16 10 to 8 2 to 0	FTOS(n+3) FTOS(n+2) FTOS(n+1) FTOSn	This is used to select the FTM output to be assigned to TMOUTx (x = 0 to 7). 000: FTM0P (Initial value) 001: FTM0N 010: FTM1P 011: FTM1N 1xx: Setting prohibited

11.3 Description of Operation

This operates as timer, capture, or PWM according to the mode set in FTnMD.
This section describes start/stop by software/event trigger, emergency stop, interrupt processing, and output control for each mode.
FTMn has four types of function mode: TIMER, CAPTURE, PWM1, and PWM2.

Table 11-3 Function mode

Mode	Description
TIMER mode	It controls the interrupt generation and output signal using the counter overflow.
CAPTURE mode	It stores the count value when the selected event trigger is generated to the FTMn event register A (FTnEA) and FTMn event register B (FTnEB).
PWM1 mode	It can generate two types of PWM waveform with the same period and aligned start edges, using the FTMn event register A (FTnEA) as the DUTY value of the output signal FTMnP and the FTMn event register B (FTnEB) as the DUTY value of the output signal FTMnN.
PWM2 mode	It can generate a complementary PWM waveform where the output signal FTMnN operates exclusively, using the FTMn event register A (FTnEA) as the DUTY value of the output signal FTMnP. Also, the dead time can be set using the FTMnDeadTimer register (FTMnDT).

11.3.1 Common Sequence

The FTM starts the operation by FTnCON0 after setting 1-6 described below as needed.
Then it processes interrupts and updates cycle/event settings and so on.

1: Mode setting (FTnMOD)
Select the mode using the mode register (FTnMOD). Also, set the dead time to output waveform, and so on.

2: Clock setting (FTnCLK)
Select the counter clock. This sets the source clock and the dividing ratio.

3: Trigger setting (FTnTRG)
Use this setting when starting/stopping the counter by event trigger. Select the event trigger source and action, and the edge of the event trigger/emergency stop/capture for FTnTRG.

4: Interrupt setting (FTnINTE)
Set the interrupt source. Select from period/event (counter coincide, duty, capture) and trigger start/stop interrupt.
When using the One-shot mode / Single mode, set to “1” FTnIEP of FTnINTE register and confirm that FTnISP of FTnINTS register is “0”.

5: Period/event setting (FTnP,FTnEA,FTnEB,FTnDT)
Set the period, data for counter coincide, duty and dead time.

Table 11-4 Period/event setting

	TIMER	CAPTURE	PWM1	PWM2
FTnP	Auto-reload period or timeout of one-shot			
FTnEA	Coincident interrupt setting value	(Capture data)	FTMnP duty	Duty
FTnEB	Coincident interrupt setting value		FTMnN duty	(Unused)
FTnDT	Dead time for output	(Unused)	Dead time for output	Dead time for output

The period is calculated as follows:

$$T_{priod} = \frac{FTnP + 1}{FTnCK\ [Hz]} \quad (FTnP : 0x0001 \text{ to } 0xFFFF)$$

6: Output setting (FTOSL*, Each Port Setting)
Set which output to which port, and reverse.

7: Control start/stop (FTnCON0)
Allow the software start or event trigger reception. Also, set the emergency stop enable.
The counter operates at a falling edge of FTnCK. The software start/stop are synchronized by FTnCK. FTnSTAT is set to “H” after one cycle in FTnCK at start, and the counter starts operating after two cycles. At stop, the counter is stopped in one cycle in FTnCK, and FTnSTAT is set to “L”. The counter value is kept at this time. If started again, it restarts after one cycle. To clear the counter, use write access to FTnC.

8: Processing during operation (FTnCON0/1, FTnINTS/C)
The state during operation can be seen in FTnCON1 or FTnINTS. To change the waveform of PWM, etc., set the period/event and set FTnUD of FTnCON1. Then, it is updated in the next period. Also, setting FTnSDN of FTnCON0 forces the output to be masked to “L”.

11.3.2 Counter Operation

The internal counter of FTM operates in the same way in all the modes.
It counts up until the setting value of the FTMn period register (FTnP).
At overflow in auto-reload mode (FTnOST bit of the FTMn mode register (FTnMOD) is “0”), the counter is cleared and continues counting again. At overflow in one-shot mode (FTnOST bit of FTnMOD is “1” and FTnIPE bit of FTnINTE is “1”), the counter is cleared and stops counting.
The software or event trigger can start/stop counting.

11.3.2.1 Starting/Stopping Counter by Software

When FTnRUN bit of the FTMn control register 0 (FTnCON0) is set to “1”, the counter starts.
In one-shot mode (FTnOST bit of the FTMn mode register (FTnMOD) is “1”), FTnRUN bit is automatically set to "0" when the counter stops due to overflow.

If the counter is operating (FTnSTAT bit of the FTMn control register 1 (FTnCON1) is “1”), the counter stops when FTnRUN is set to “0”. At this time, the counter keeps the value when it stops. When FTnRUN bit is set to “1” again, the counter continues from the stopped value.
To clear the counter, write to the FTMn counter register (FTnC) when it is not operating. (This written data is meaningless.)

11.3.2.2 Starting/Stopping Counting by Event Trigger

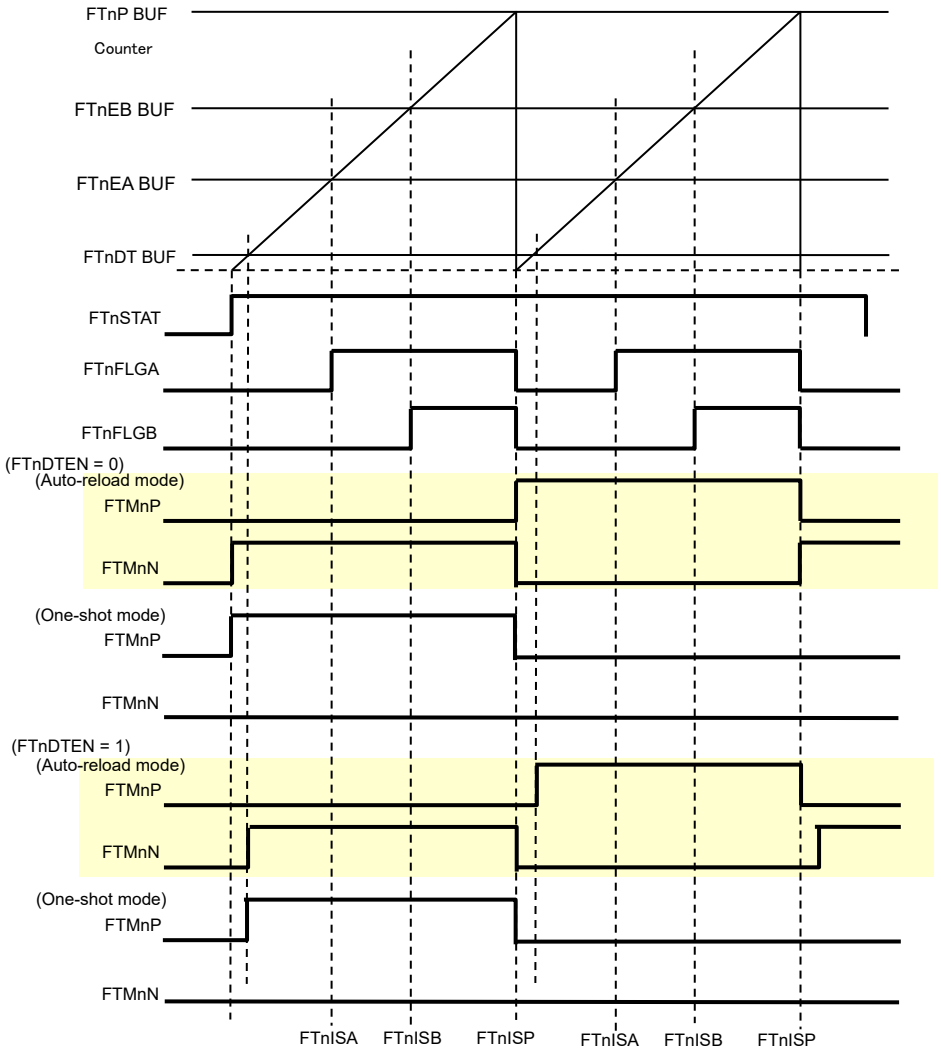
When FTnTGEN bit of the FTMn control register 0 (FTnCON0) is set to “1”, the counter is made controllable by triggers.
Set the FTMn trigger setting register 0 and 1 (FTnTRG) to select a trigger and so on.
The event trigger source can be selected from the external interrupts, the timer interrupts, and another FTM triggers.
The counter start, counter stop, or counter start/ stop can be selected by selecting an event trigger.

11.3.3 TIMER Mode Operation

The TIMER mode controls the interrupt generation and output signal using the counter overflow.

11.3.3.1 Output Waveform in TIMER Mode

In the timer output auto-reload mode, the output is toggled for each period.
If the counter value is 0x0000, FTMnP starts with L and FTMnN starts with “H” when FTnRUN bit of the FTMn control register 0 (FTnCON0) is set to "1".
In the one-shot mode, it stops after outputting “H” pulse of one period from FTMnP. FTMnN is fixed to “L”.
When the dead time is set using the FTMn dead time register (FTnDT), the output is L after starting the counter before passing the count set in FTnDT.
When the FTnDTEN of the FTMn mode register (FTnMOD) is set to “1”, the output is “L” from the start of the count until the counter reaches the count values set in the FTMn dead time register (FTnDT).



(It is case of auto-reload mode except output waveform)

$$T_{priod} = \frac{FTnP + 1}{FTnCK\ [Hz]} \quad (FTnP : 0x0001 \text{ to } 0xFFFF)$$

$$T_{eventA/B} = \frac{FTnEA(or\ FTnEB) + 1}{FTnCK\ [Hz]} \quad (FTnA/FTnB : 0x0000 \text{ to } 0xFFFE)$$

$$T_{deadtime} = \frac{FTnDT + 1}{FTnCK\ [Hz]} \quad (FTnDT : 0x0000 \text{ to } 0xFFFE)$$

Figure 11-2 TIMER mode

11.3.4 PWM1 Mode Operation

The PWM1 mode generates synchronization output pulses with the period set in FTnP. The duties of the output FTMnP and FTMnN are set in FTnEA and FTnEB respectively.

11.3.4.1 Output Waveform in PWM1 Mode

In the Auto-reload mode, the initial values of FTMnP and FTMnN are “L”, and they change to “H” at start. Each of them changes to “L” at the duty value. It changes to “H” in the next period. This is repeated until they stop. In the one-shot mode, they automatically stop and change to “L” after one period.
If the dead time is enabled, the FTMnP and FTMnN are “L” output from start of the counter until the dead time is passed.

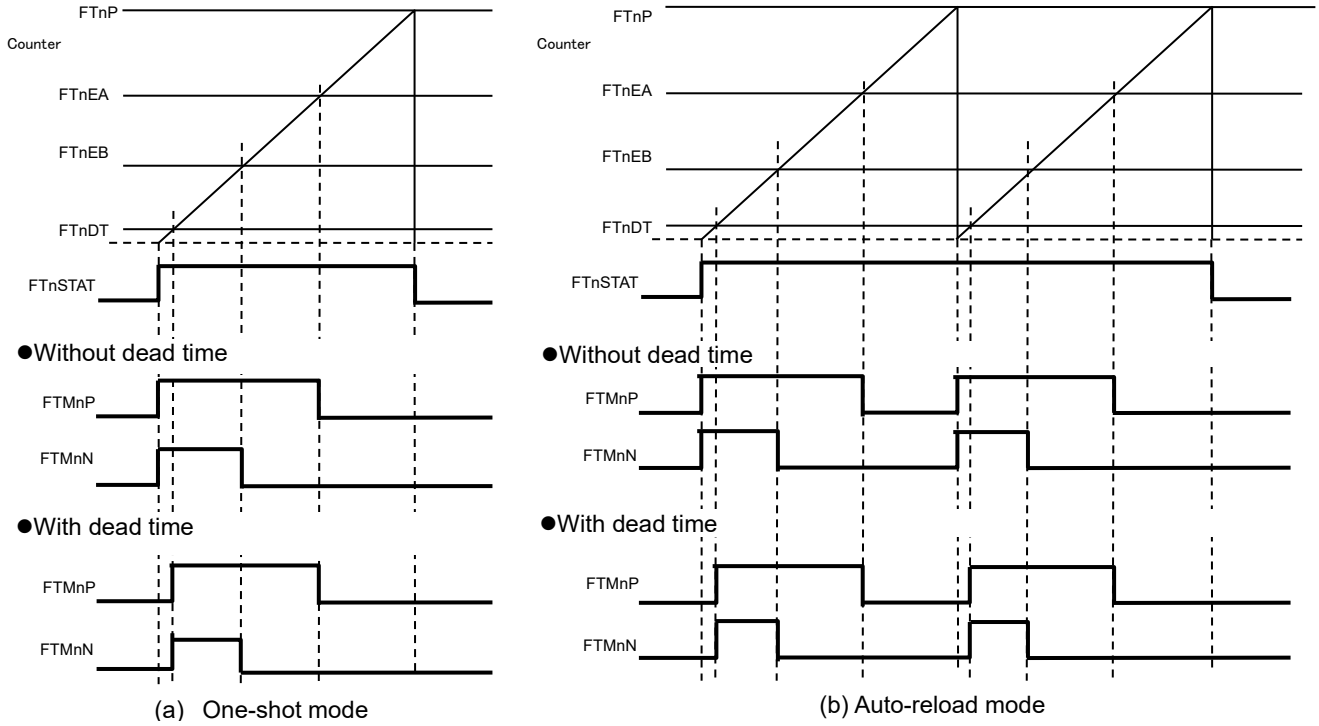


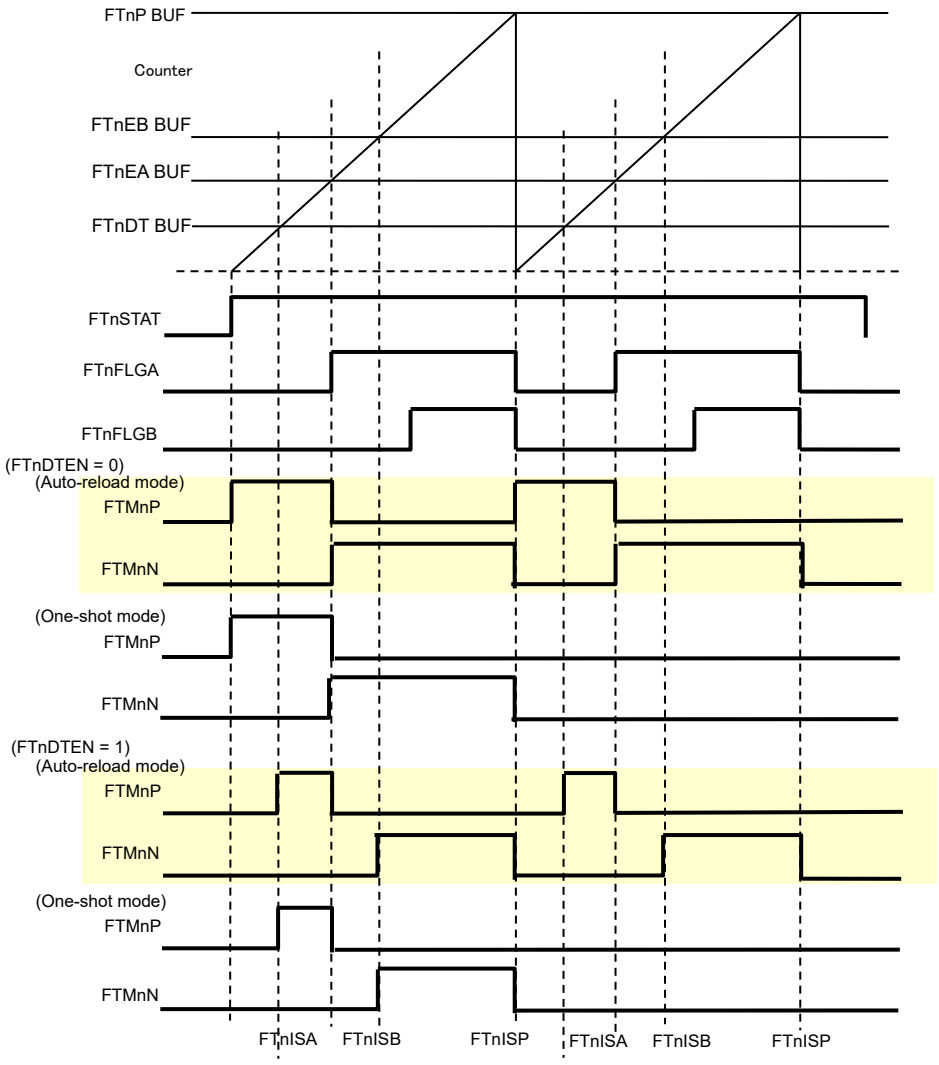
Figure 11-3 PWM1 mode

11.3.5 PWM2 Mode Operation

The PWM2 mode generates a complementary output pulse with the cycle set in FTnP. Set the duties of the output FTMnP/FTMnN in FTnEA. FTnEB is not used.

11.3.5.1 Output Waveform in PWM2 Mode

In the Auto-reload mode, the initial values of FTMnP and FTMnN are “L”, and FTMnP changes to “H” at start. FTMnP changes to L and FTMnN changes to “H” at the duty value. FTMnP changes to “H” and FTMnN changes to “L” in the next period. This is repeated until they stop. In the one-shot mode, they automatically stop and change to “L” after one period.
If the dead time is enabled, the FTMnP output is “L” from start of the counter until the dead time is passed, and the FTMnN output is “L” from coincidence of duty until the dead time is passed.



(It is case of auto-reload mode except output waveform)

$$T_{priod} = \frac{FTnP + 1}{FTnCK\ [Hz]}$$ (FTnP : 0x0001 to 0xFFFF)

$$T_{duty} = \frac{FTnEA + 1}{FTnCK\ [Hz]}$$ (FTnA/FTnB : 0x0000 to 0xFFFE)

$$T_{deadtime} = \frac{FTnDT + 1}{FTnCK\ [Hz]}$$ (FTnDT : 0x0000 to 0xFFFE)

Figure 11-4 PWM2 mode

11.3.6 CAPTURE Mode Operation

The CAPTURE mode stores the count value at the time when an event trigger source is generated, to the FTnEA/FTnEB register. The event trigger source to be captured is common to that used at counter start/stop.

Stored data in FTnEA	Counter value at the time when an event trigger rising edge is generated
Stored data in FTnEB	Counter value at the time when an event trigger falling edge is generated

11.3.6.1 Measurement Example in the CAPTURE Mode

The following example shows the measurement of the period and duty of PWM input from EXI0 using CAPTURE mode and counter start/stop by trigger events.

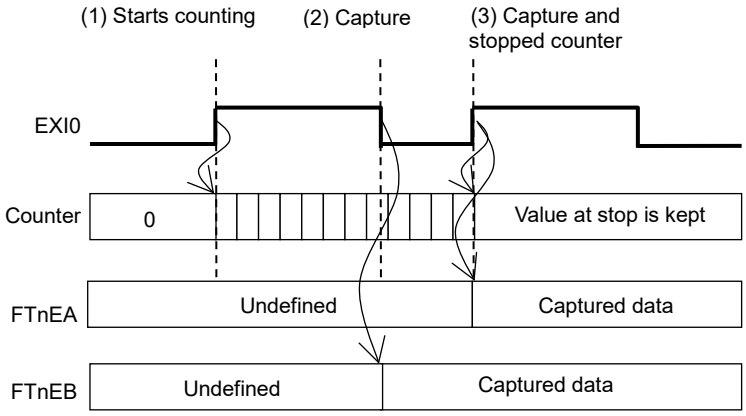


Figure 11-5 Measurement example in the CAPTURE mode

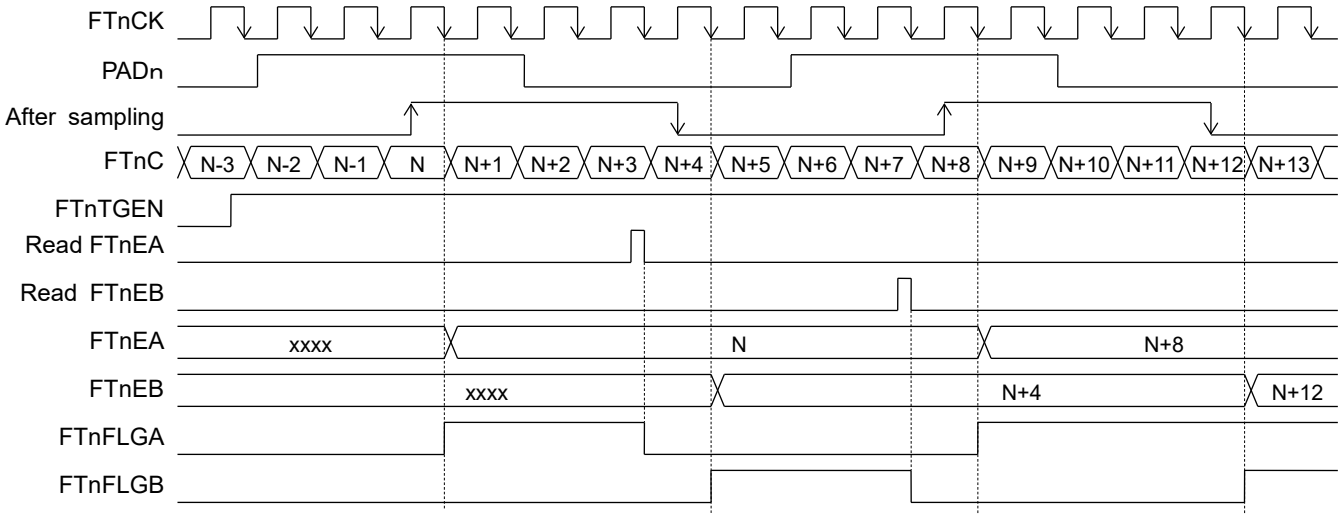
Set the FTnMOD register to the capture mode (FTnMD = “01”).
Use the FTnINTE register (FTnIETS = “1”) to enable the trigger counter stop interrupt.
Use the FTnTRG register to set the trigger event source to EXI0TGO (FTnSTSS = “0”, FTnSTS = “0000”), enable counter start (FTnST0 = “1”), and enable counter stop (FTnST1=1).
Use the FTnTRG register to set counter start and stop to rising edge (FTnTRM = “00”).
Use the FTnCON0 register to enable the trigger operation (FTnTGEN = “1”).

The counter starts at rising of EXI0. (1)
Then the counter value is stored to the FTnEB register at falling of EXI0. (2)
When the rising of EXI0 is detected again, the counter stops, and an interrupt occurs. (3)
And the counter value is stored to the FTnEA register at rising of EXI0.

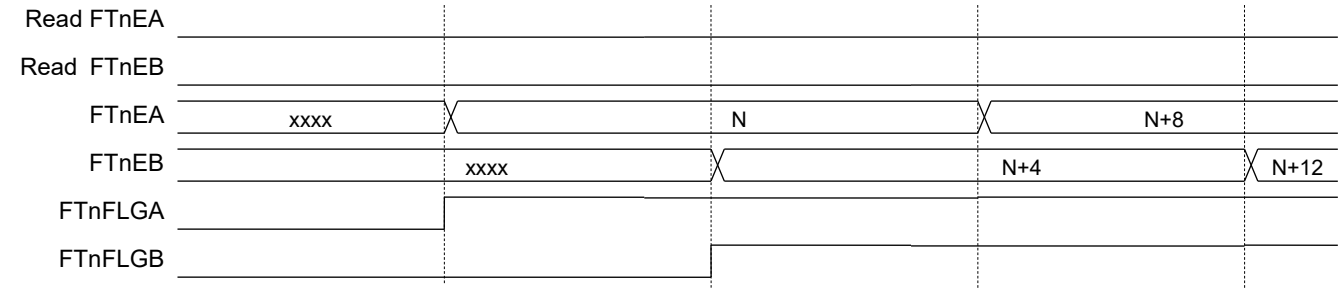
At this time, the values of FTnEA and FTnEB correspond to the period and the duty of EXI0 respectively.

The operation after capture depends on the FTnOST bit of the FTnMOD register.

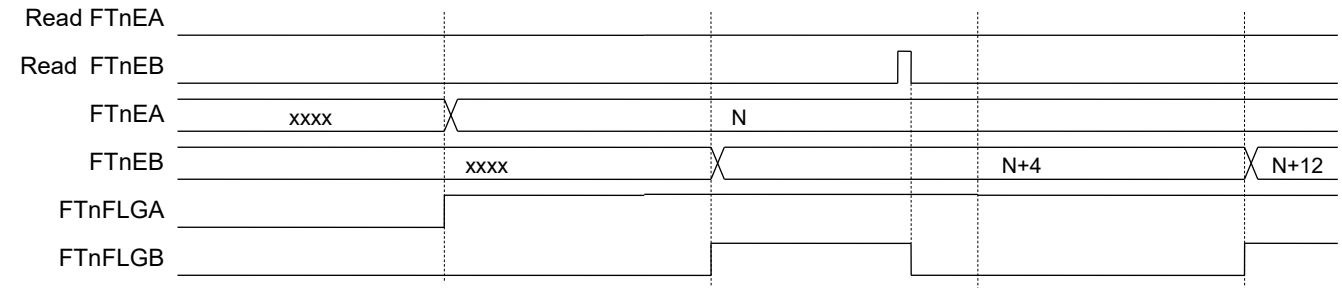
- When FTnOST = “0” (auto mode)
  After the counter restarts at the next rising of EXI0, the value of FTnEA is updated at falling of EXI0.
- When FTnOST = “1” (single mode)
  After the counter restarts at the next rising of EXI0, the value of FTnEA is not updated at falling of EXI0.



(a) FTnEA, FTnEB register read before next trigger (FTnOST=0,1)



(b) Next trigger occurred without register read (FTnOST=0)



(c) Next trigger occurred without register read (FTnOST=1)
Figure 11-6 Operation Timing in the CAPTURE mode

11.3.6.2 Clearing FTnFLGA/FTnFLGB bit

In one-shot mode, FTnEA/FTnEB data is not updated if FTnFLGA/FTnFLGB is “1” each.
The FTnFLGA/FTnFLGB is cleared by reading FTnEA/FTnEB, respectively. However, it is invalid when FTnTGEN = “0”.
Even if FTnFLGA/FTnFLGB is cleared and FTnTGEN is set to “0”, a trigger may be entered during control and FTnFLGA/FTnFLGB may be set to “1”.

To avoid this, set FTnTGEN to “0” after making sure that FTnFLGA/FTnFLGB is “0” with no trigger input. For example, set FTnST to 0 and stop the trigger start. Or initialize this peripheral circuit by block reset after setting FTnTGEN to 0.

11.3.7 Event/Emergency Stop Trigger Control

11.3.7.1 Trigger Signal

FTMn can receive two types of trigger signal: event trigger and emergency stop trigger.
The event trigger is used as counter start/stop or trigger of capture. EXI0-7TGO (external interrupts), TIMER0-5 interrupts, or FTM0-1 triggers can be selected as the trigger source.
The emergency stop trigger is used to stop the timer operation. It stops the counter and sets output FTMnP/FTMnN to "L". EXI0TGO, EXI4TGO, CMP0TRG or CMP1TRG can be selected as the trigger source.
The external interrupt controller can apply a filter function to the EXI0-7TGO.
The timer interrupt source and the FTM trigger source are set using the register of each timer.

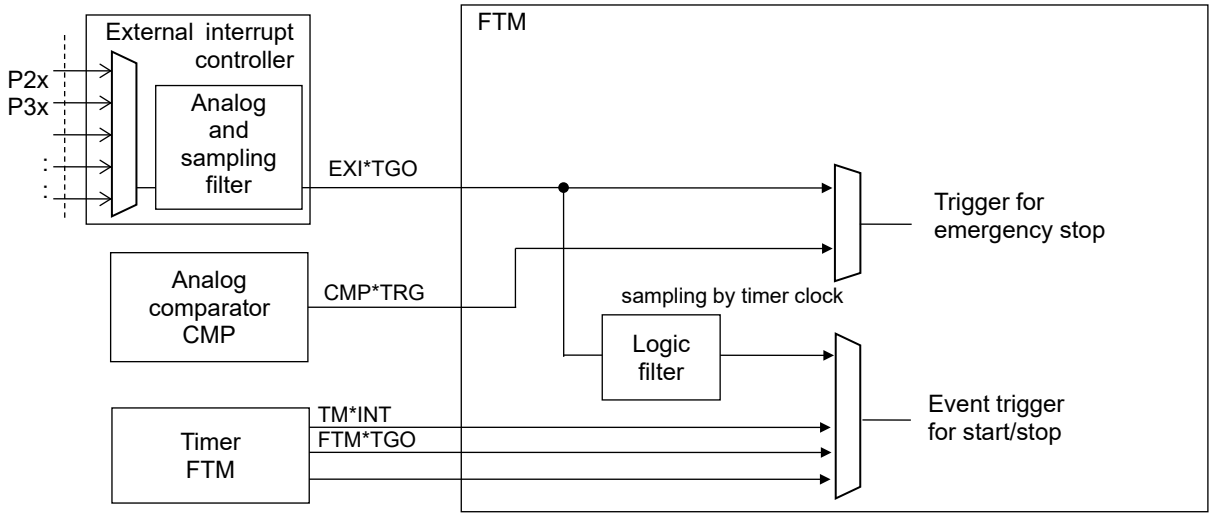


Figure 11-7 Input Path of Trigger Signal

11.3.7.2 Start/Stop Operating by Event Trigger

Here is the setting used to control the counter by event triggers.
1) FTnTRG setting
Enable/Disable counter start/stop by event triggers
Set whether or not to clear the counter at stop by an event trigger
Set whether or not to accept the next counter start after stop by an event trigger
Set the event trigger source (EXI0-7TGO, TIMER0-5INT, FTM0-1TGO)
Set the edge of the event trigger which generates counter start
Set the edge of the event trigger which generates counter stop
2) Controlling FTnCON0
Set FTnTGEN to "1" to enter the waiting state for event triggers.
Then, set FTnRUN to "1" to start the counter by the software.
Set FTnRUN to "0" during the counter operation to stop the counter by the software.

Because the trigger signal is sampled at FTnCK when the external input trigger (EXInTGO) is selected as counter control by event triggers, the input pulse width should be set to Analog filter 200ns and three or more sampling clocks. Pulses shorter than three sampling clocks may be or may not be removed. Note that the sampling is not performed when the timer interrupt is selected as the event trigger.
Figure 11-8 shows the sampling timing of the external input.

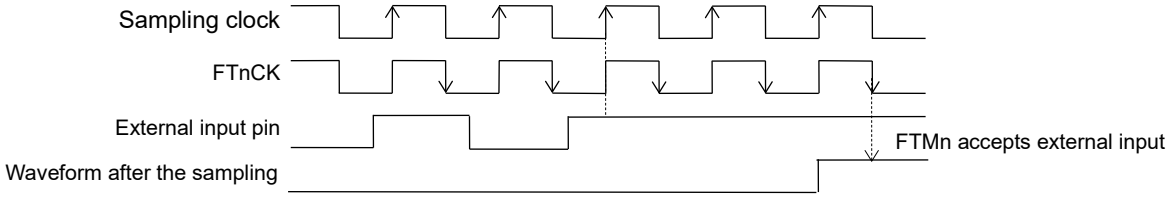
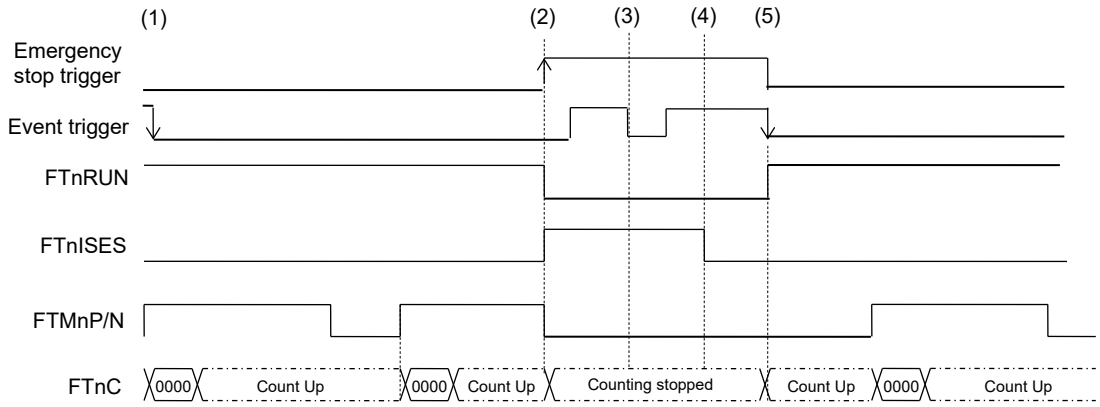


Figure 11-8 Sampling Timing of External Input

11.3.7.3 Emergency Stop Operation

When FTnEMGEN is set to "1", the emergency stop function is enabled. Set this bit after the trigger source is selected in FTnEST.

If an emergency stops trigger input (rising edge) is detected, the counter stops, the output is set to “L”, and an emergency stop interrupt occurs.
To restart the counter, clear the emergency stop interrupt status (write “1” to FTnICES) and “1” is set to run bit.
Figure 11-9 shows the operation timing at emergency stop.
After the emergency stop, the RUN bit is cleared to “0”, the counter stops after one timer clock, and the STAT bit is cleared to 0. When the STAT bit is “1”, setting the RUN to “1” is not accepted. Confirm that the STAT has changed to “0” after clearing the interrupt status before running the next RUN.



(1) The counter operation starts at an event trigger (falling edge).
(2) The counter stops at an emergency stop trigger (rising edge). An emergency stop interrupt occurs.
(3) The event trigger is disabled due to the emergency stop in progress.
(4) Clear the emergency stop interrupt to enable the operation.
(5) The counter operation restarts at an event trigger (falling edge).
(In this example, the pulse output restarts after one cycle because the counter is not cleared)

Figure 11-9 Operation Timing Diagram at Emergency Stop

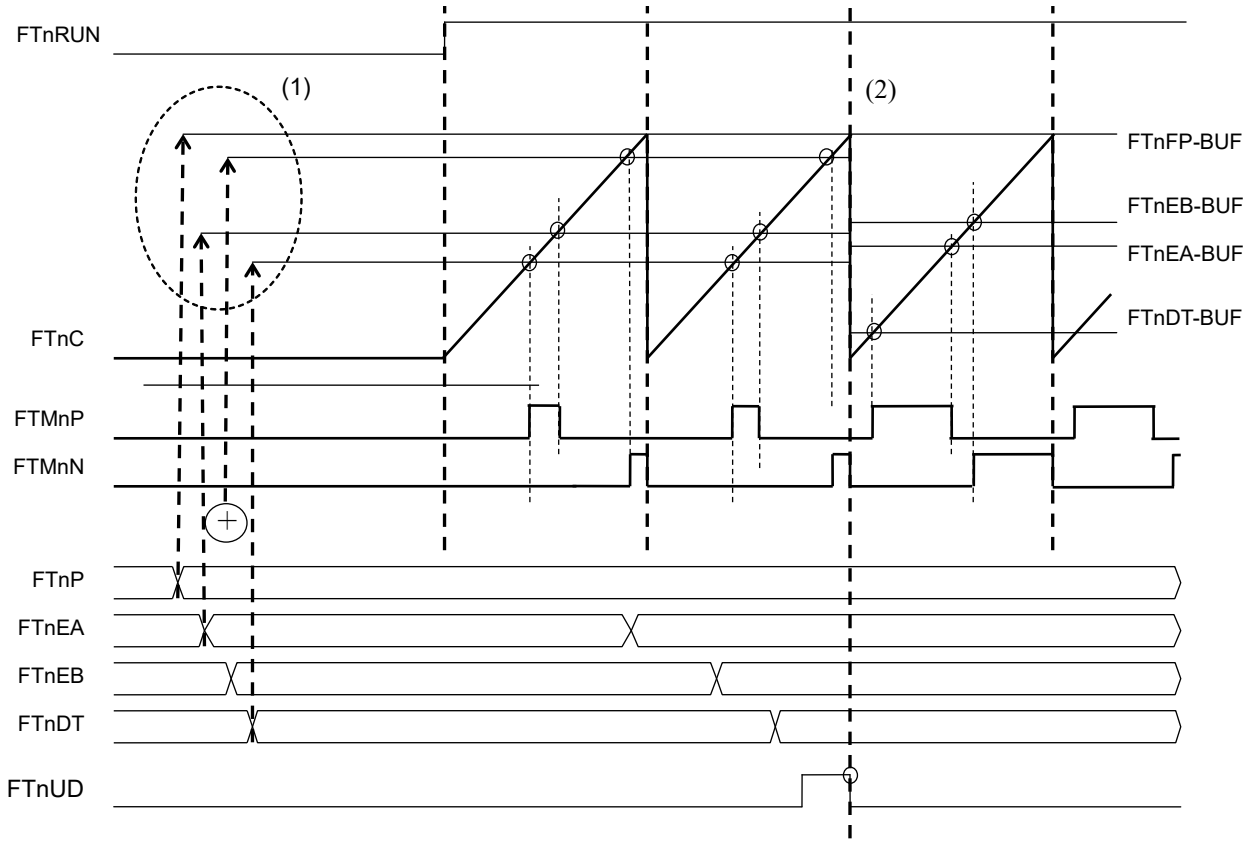
11.3.8 Output at Counter Stop

The FTMnP and FTMnN states depend on the setting of FTnSTPO when the counter stops by a software/Event trigger. If FTnSTPO is “0”, FTMnP/FTMnN is set to “L” at the same time as stop. If the counter is restarted in this state, the FTMnP/FTMnN outputs keep “L” during that period, and they change according to the counter value from the next period.
When FTnSTPO is “1”, FTMnP/FTMnN keep the state at stop. When the counter is restarted, their states change according to the counter value.
When the FTnEXCL bit of the FTnTRG register is set to “1” or the software clears the counter after counter stop, the counter value is counted up from 0x0000, and the output depends on the counter value.

11.3.9 Changing Period, Event A/B, and Dead Time during Operation

The period, event A/B, and dead time can be changed in the next cycle when the timer is counting.
To do so, set desired registers (FTnP, FTnEA, FTnEB, FTnDT, etc.), and then write "1" to the FTnUD bit of the FTnCON1 register to request the update. The values in the buffers for the period, event A/B, and dead time are updated at the beginning of the next period, and the FTnUD bit is set to "0".

Here is an example in the PWM2 mode (DTEN = "1").



(1) Each buffer is updated at set timing during operation stopped.
(2) Each buffer is updated at the beginning of the next period where FTnUD is set to "1" during operation.

Figure 11-10 Update Timing during Operation

11.3.10 Interrupt Source

This section describes the interrupt source and how to clear it.
When a target interrupt enables (FTnIE*) is set to "1", the interrupt status is enabled, and the interrupt controller is notified of the source.
Note that the emergency stop interrupt enable does not exist. When the emergency stop is enabled, its interrupt is also enabled.
If the interrupt status is set to "1" for a source, clear it by an appropriate processing.
When the interrupt handler is used, write "1" to FTnIR bit at the end of the interrupt processing (when exiting the interrupt handler).

Table 11-5 interrupt Source

Name	Mode	Status	How to clear
Periodic interrupt	ALL	FTnISP	Write "1" to FTnICP
Event timing A interrupt	TIMER/PWM1/PWM2	FTnISA	Write "1" to FTnICA
Capture A interrupt	CAPTURE	FTnISA	Write "1" to FTnICA or read FTnEA
Event timing B interrupt	TIMER/PWM1	FTnISB	Write "1" to FTnICB
Capture B interrupt	CAPTURE	FTnISB	Write "1" to FTnICB or read FTnEB
Trigger stop interrupt	ALL	FTnISTS	Write "1" to FTnICTS
Trigger start interrupt	ALL	FTnISTR	Write "1" to FTnICTR
Emergency stop interrupt	ALL	FTnISES	Write "1" to FTnICES

The periodic interrupt/event timing A interrupt/event timing B interrupt can be selected as the interrupt trigger output.

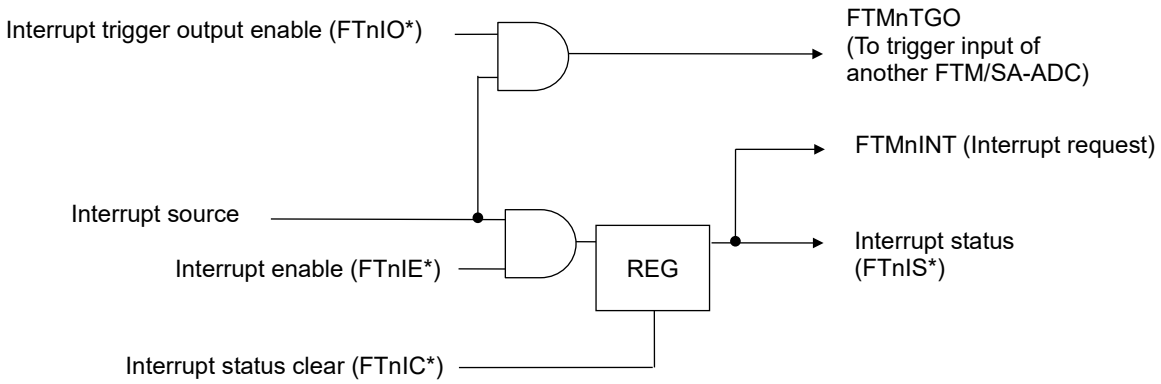


Figure 11-11 Interrupt Control Signal

Chapter 12

Real Time Clock (RTC)

12. Real Time Clock (RTC)

12.1 Overview

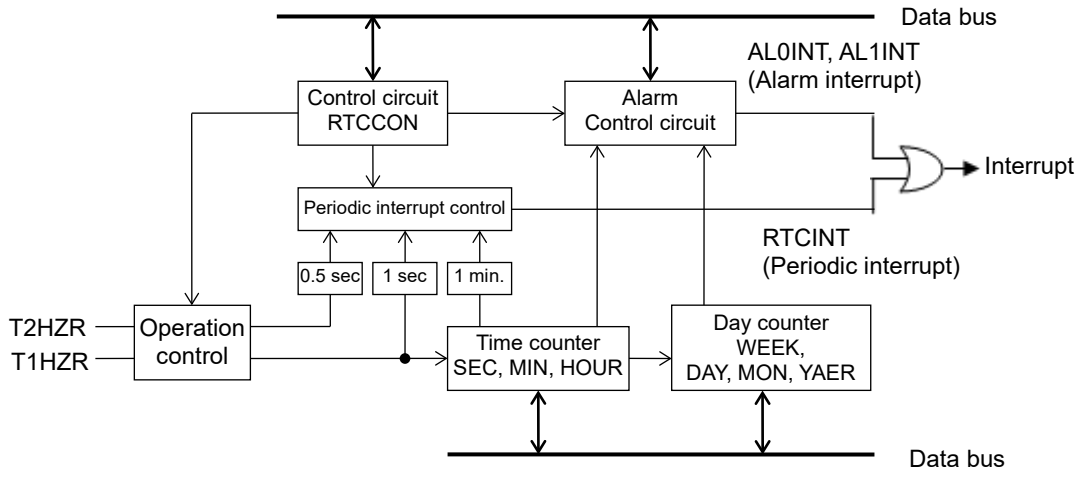
This LSI includes real time clock (RTC).
For input clocks, see Chapter 9, “Time Base Counter”. For interrupt permission and interrupt request flags described in this chapter, see Chapter 7, “Interrupts”.

12.1.1 Features

- Date counting function including year, month, day, and day of the week, and clock counting function including hour, minute, and second
- Auto calendar function with leap year checking function
- Periodic interrupt function with selection range of 0.5 seconds, 1 second, and 1 minute
- Alarm interrupt function according to coincidence of day of the week, hour, and minute
- Alarm interrupt function according to coincidence of month, day, hour, and minute
- The RTC counter continues counting operation even when a reset (other than power-on reset) is generated.

12.1.2 Configuration

Figure 12-1 shows the configuration of the real time clock.



RTCCON : Real time clock control register

Figure 12-1 Configuration of Real Time Clock

12.2 Description of Registers

12.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_5000	Real time clock base register	RTC	-	-	-
0x00	Real time clock second register	RTCSEC	R/W	32	0x0000_0000
0x04	Real time clock minute register	RTCMIN	R/W	32	0x0000_0000
0x08	Real time clock hour register	RTCHOUR	R/W	32	0x0000_0000
0x0C	Real time clock week register	RTCWEEK	R/W	32	0x0000_0007
0x10	Real time clock day register	RTCDAY	R/W	32	0x0000_0001
0x14	Real time clock month register	RTCMON	R/W	32	0x0000_0001
0x18	Real time clock year register	RTCYEAR	R/W	32	0x0000_0000
0x1C	Real time clock control register	RTCCON	R/W	32	0x0000_0000
0x20	Real time clock alarm 0 minute register	AL0MIN	R/W	32	0x0000_0000
0x24	Real time clock alarm 0 hour register	AL0HOUR	R/W	32	0x0000_0000
0x28	Real time clock alarm 0 week register	AL0WEEK	R/W	32	0x0000_0000
0x2C	Real time clock alarm 1 minute register	AL1MIN	R/W	32	0x0000_0000
0x30	Real time clock alarm 1 hour register	AL1HOUR	R/W	32	0x0000_0000
0x34	Real time clock alarm 1 day register	AL1DAY	R/W	32	0x0000_0000
0x38	Real time clock alarm 1 month register	AL1MON	R/W	32	0x0000_0000
0x50	Real time clock hour/minute/second register	RTCHMS	R	32	0x0000_0000
0x54	Real time clock year/month/day/week register	RTCYMDW	R	32	0x0001_0107
0x60	Real time clock interrupt status register	RTCINTST	R/W	32	0x0000_0000

12.2.2 Real Time Clock Second Register (RTCSEC)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	RS						
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCSEC is a special function register (SFR) to store decimal second data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
6 to 0	RS	RS is used to store binary-coded decimal second data (0x00 to 0x59). During RTC operation, RTCSEC increments on the rising edge of the T1Hz signal of the low-speed time base counter (LTBC).

[Note]
- **When setting second data in RTCSEC, stop RTC (RTCEN=0).**
- **Do not write unrealistic second data to RTCSEC.**
- **When reading RTCSEC, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.3 Real Time Clock Minute Register (RTCMIN)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	RM						
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCMIN is a special function register (SFR) to store decimal minute data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
6 to 0	RM	RM is used to store binary-coded decimal minute data (0x00 to 0x59). RTCMIN increments at occurrence of carry from RTCSEC (Second: 0x59 to 0x00).

[Note]
- **When setting minute data in RTCMIN, stop RTC (RTCEN=0).**
- **Do not write unrealistic minute data to RTCMIN.**
- **When reading RTCMIN, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.4 Real Time Clock Hour Register (RTCHOUR)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	RH					
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCHOUR is a special function register (SFR) to store decimal hour data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
5 to 0	RH	RH is used to store binary-coded decimal hour data (0x00 to 0x23). RTCHOUR increments at occurrence of carry from RTCMIN (Minute: 0x59 to 0x00).

[Note]
- **When setting hour data in RTCHOUR, stop RTC (RTCEN=0).**
- **Do not write unrealistic hour data to RTCHOUR.**
- **When reading RTCHOUR, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.5 Real Time Clock Week Register (RTCWEEK)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	RW		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

RTCWEEK is a special function register (SFR) to store decimal day-of-the-week data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
2 to 0	RW	RW is used to store day-of-the-week data (0x01 to 0x07). RTCWEEK is incremented at occurrence of carry from RTCHOUR (Hour: 0x23 to 0x00). Day-of-the-week data can be associated with any actual day of the week since their relation is not fixed.

[Note]
- **When setting day-of-the-week data in RTCWEEK, stop RTC (RTCEN=0).**
- **Do not write unrealistic day-of-the-week data to RTCWEEK.**
- **When reading RTCWEEK, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.6 Real Time Clock Day Register (RTCDAY)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	RD					
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RTCDAY is a special function register (SFR) to store decimal day data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
5 to 0	RD	RD is used to store binary-coded decimal day data (0x01 to 0x31). RTCDAY increments at occurrence of carry from RTCHOUR (Hour: 0x23 to 0x00).

[Note]
- **When setting day data in RTCDAY, stop RTC (RTCEN=0).**
- **Do not write unrealistic day data to RTCDAY.**
- **When reading RTCDAY, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.7 Real Time Clock Month Register (RTCMON)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	RMO				
R/W	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RTCMON is a special function register (SFR) to store decimal month data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
4 to 0	RMO	RMO is used to store binary-coded decimal month data (0x01 to 0x12). RTCMON increments at occurrence of carry from RTCDAY (Day: 0x28/0x29/0x30/0x31 to 0x01).

[Note]
- **When setting month data in RTCMON, stop RTC (RTCEN=0).**
- **Do not write unrealistic month data to RTCMON.**
- **When reading RTCMON, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.8 Real Time Clock Year Register (RTCYEAR)

Offset : 0x18

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	RY							
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCYEAR is a special function register (SFR) to store decimal year data.
When this register is written, the time base counter register for RTC (LTBRR) is cleared.

Bit No	Bit name	Description
7 to 0	RY	RY is used to store binary-coded decimal year data (0x00 to 0x99). RTCYEAR increments at occurrence of carry from RTCMON (Month: 0x12 to 0x01).

[Note]

- **When setting month data in RTCYEAR, stop RTC (RTCEN=0).**
- **Do not write unrealistic year data to RTCYEAR.**
- **When reading RTCYEAR, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.9 Real Time Clock Control Register (RTCCON)

Offset : 0x1C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	RIN		RTCEN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCCON is a special function register (SFR) to control the real time clock.

Bit No	Bit name	Description
2 to 1	RIN	RIN is used to enable a periodic interrupt. 00: Disabled (Initial value) 01: Enabled 0.5-second interrupt 10: Enabled 1-second interrupt 11: Enabled 1-minute interrupt
0	RTCEN	RTCEN is a bit to control the operation start/stop of the RTC. 0: Stops RTC (Initial value) 1: Starts RTC

[Note]

- **Before Starting RTC operation, setting each RTC registers and clearing the interrupts.**
- **A 0.5-second interrupt and a 1-second interrupt request a RTC periodic interrupt (RTCINT) even if RTC is being stopped (RTCEN is "0"). A 1-minute interrupt does not request a RTC periodic interrupt while RTC is stopped.**
- **Even if the T1HZ bit of LTBR register of time base counter is "1", RTCSEC is not +1 when RTCEN bit of the RTCCON register is "1" to "0".**
- **Set "0" to RTCEN bit and the confirm that reading value of RTCEN bit is "0", before re-configuration of the RTC. It takes up to one cycle at the low-speed clock for the RTCEN bit to return "0".**

12.2.10 Real Time Clock Alarm 0 Minute Register (AL0MIN)

Offset : 0x20

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	AL0M						
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL0MIN is a special function register (SFR) to set an alarm 0 interrupt time.

Bit No	Bit name	Description
6 to 0	AL0M	AL0M is used to store binary-coded decimal minute data (0x00 to 0x59) for generation of an alarm.

[Note]

- **Setting an unrealistic time in this register does not generate an alarm 0 interrupt (AL0INT).**

12.2.11 Real Time Clock Alarm 0 Hour Register (AL0HOUR)

Offset : 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	AL0H					
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL0HOUR is a special function register (SFR) to set an alarm 0 interrupt time.

Bit No	Bit name	Description
5 to 0	AL0H	AL0H is used to store binary-coded decimal hour data (0x00 to 0x23) for generation of an alarm.

[Note]

- **Setting an unrealistic time in this register does not generate an alarm 0 interrupt (AL0INT).**

12.2.12 Real Time Clock Alarm 0 Week Register (AL0WEEK)

Offset : 0x28

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	AL0W		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL0WEEK is a special function register (SFR) to set an alarm 0 interrupt time.

Bit No	Bit name	Description
2 to 0	AL0W	AL0W is used to store day-of-the-week data (0x00 to 0x07) for generation of an alarm. When AL0WEEK is not used as comparison data for generating an alarm, set AL0WEEK to “0x00”. That setting means every day.

12.2.13 Real Time Clock Alarm 1 Minute Register (AL1MIN)

Offset : 0x2C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	AL1M						
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL1MIN is a special function register (SFR) to set an alarm 1 interrupt time.

Bit No	Bit name	Description
6 to 0	AL1M	AL1M is used to store binary-coded decimal minute data (0x00 to 0x59) for generation of an alarm.

[Note]

- **Setting an unrealistic time in this register does not generate an alarm 1 interrupt (AL1INT).**

12.2.14 Real Time Clock Alarm 1 Hour Register (AL1HOUR)

Offset : 0x30

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	AL1H					
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL1HOUR is a special function register (SFR) to set an alarm 1 interrupt time.

Bit No	Bit name	Description
5 to 0	AL1H	AL1H is used to store binary-coded decimal hour data (0x00 to 0x23) for generation of an alarm.

[Note]

- **Setting an unrealistic time in this register does not generate an alarm 1 interrupt (AL1INT).**

12.2.15 Real Time Clock Alarm 1 Day Register (AL1DAY)

Offset : 0x34

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	AL1D					
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL1DAY is a special function register (SFR) to set an alarm 1 interrupt time.

Bit No	Bit name	Description
5 to 0	AL1D	AL1D is used to store binary-coded decimal day data (0x01 to 0x31) for generation of an alarm. When AL1DAY is not used as comparison data for generating an alarm, set AL1nDAY to 0x00. That setting means every day in the month selected by AL1MON.

[Note]

- **Setting an unrealistic time except 0x00 in this register does not generate an alarm 1 interrupt (AL1INT).**

12.2.16 Real Time Clock Alarm 1 Month Register (AL1MON)

Offset : 0x38

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	AL1MO				
R/W	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AL1MON is a special function register (SFR) to set an alarm 1 interrupt time.

Bit No	Bit name	Description
4 to 0	AL1MO	AL1MO is used to store decimal month data (0x01 to 0x12) for generation of an alarm. When AL1MON is not used as comparison data for generating an alarm, set AL1MON to 0x00. That setting means every month.

[Note]

- **Setting an unrealistic time except 0x00 in this register does not generate an alarm 1 interrupt (AL1INT).**

12.2.17 Real Time Clock Hour/Minute/Second Register (RTCHMS)

Offset : 0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	RHR					
R/W	–	–	–	–	–	–	–	–	–	–	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	RMR							–	RSR						
R/W	–	R	R	R	R	R	R	R	–	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCHMS is a read-only special function register (SFR) to indicate binary-coded decimal values of second data, minute data, and hour data.

Bit No	Bit name	Description
21 to 16	RHR	RHR is same the value as RH of RTCHOUR.
14 to 8	RMR	RMR is same the value as RM of RTCMIN.
6 to 0	RSR	RSR is same the value as RS of RTCSEC.

[Note]

- **When reading RTCHMS, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.18 Real Time Clock Year/Month/Day/Week Register (RTCYMDW)

Offset : 0x54

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RYR								–	–	–	RMOR				
R/W	R	R	R	R	R	R	R	R	–	–	–	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	RDR						–	–	–	–	–	RWR		
R/W	–	–	R	R	R	R	R	R	–	–	–	–	–	R	R	R
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1

RTCYMDW is a read-only special function register (SFR) to indicate binary-coded decimal values of day-of-the-week data, day, month, and year.

Bit No	Bit name	Description
31 to 24	RYR	RYR is same the value as RY of RTCYEAR.
20 to 16	RMOR	RMOR is same the value as RMO of RTCMON.
13 to 8	RDR	RDR is same the value as RD of RTCDAY.
2 to 0	RWR	RWR is same the value as RW of RTCWEEK.

[Note]

- **When reading RTCYMDW, read consecutively it twice until the last data coincides the previous data to prevent reading of uncertain data while counting up the clock.**

12.2.19 Real Time Clock Interrupt Status Register (RTCINTST)

Offset : 0x60

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	AL1INT	AL0INT	RTCINT
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTCINTST register is used to display the source of interrupt.

Bit No	Bit name	Description
2	AL1INT	This bit is set to "1" when AL1INT interrupt request is generated. The AL1INT interrupt can be clear by writing "1" to this bit.
1	AL0INT	This bit is set to "1" when AL0INT interrupt request is generated. The AL0INT interrupt can be clear by writing "1" to this bit.
0	RTCINT	This bit is set to "1" when RTCINT interrupt request is generated. The RTCINT interrupt can be clear by writing "1" to this bit.

12.3 Description of Operation

RTC stops operation after reset release. As each bit of the date and time registers (RTCSEC, RTCMIN, RTCHOUR, RTCWEEK, RTCDAY, RTCMON, and RTCYEAR) is the initial value, set a date and a time through the software. Set a date and a time after stopping RTC operation by setting the RTCEN bit of RTCCON to ”0”.
By setting the RTCEN bit to "1" immediately after clearing the low-speed time base counter for RTC by writing to the date/time register, it will count up the second register (RTCSEC) approximately 1 second after the start of RTC operation.

Table 12-1 lists the count values of each register.

Table 12-1 Count Values of Each Register

Register name		Count value	Remarks
Second register	RTCSEC	0x00 to 0x59	-
Minute register	RTCMIN	0x00 to 0x59	-
Hour register	RTCHOUR	0x00 to 0x23	24-hour system
Week register	RTCWEEK	0x01 to 0x07	-
Day register	RTCDAY	0x01 to 0x31	Jan., Mar., May, Jul., Aug., Oct., Dec.
		0x01 to 0x30	Apr., Jun., Sep., Nov.
		0x01 to 0x28	Feb. without leap day
		0x01 to 0x29	Feb. with leap day
Month register	RTCMON	0x01 to 0x12	-
Year register	RTCYEAR	0x00 to 0x99	-

When reading data of each register of RTC, read data from all the registers twice and check that the values coincide to prevent the reading of undefined data during counting.

RTC can generate a periodic interrupt (RTCINT) that occurs at a fixed cycle, an alarm 0 interrupt (AL0INT) that occurs as a result of the coincidence of week, hour, or minute data and an alarm 1 interrupt (AL1INT) that occurs as a result of the coincidence of month, day, hour, or minute data.

Options available for periodic interrupts by using the RINT1 and RINT0 bit of RTCCON include periodic interrupt prohibition, 0.5-second interrupt, 1-second interrupt, and 1-minute interrupt.

An alarm 0 interrupt (AL0INT) occurs when carry (RTCSEC: 59 seconds to 00 seconds) from the second register (RTCSEC) occurs and the value of the real time clock alarm 0 register (AL0WEEK, AL0HOUR or AL0MIN) and the value of the week, hour, or minute register (RTCWEEK, RTCHOUR, or RTCMIN) coincide.

An alarm 1 interrupt (AL1INT) occurs when carry (RTCSEC: 59 seconds to 00 seconds) from the second register (RTCSEC) occurs and the value of the real time clock alarm 1 register (AL1MON, AL1DAY, AL1HOUR, or AL1MIN) and the value of the month, day, hour, or minute register (RTCMON, RTCDAY, RTCHOUR, or RTCMIN) coincide.

When AL0WEEK, AL1MON, or AL1DAY is not used as comparison data of the alarm, set 0x00. For instance, to set the alarm to 8:30 a.m. of each day, set 0x00 in AL0WEEK and set data of hour 8 and minute 30 in AL0HOUR and AL0MIN respectively.

Chapter 13

1kHz Timer (1kHzTM)

13. 1kHz Timer (1kHzTM)

13.1 Overview

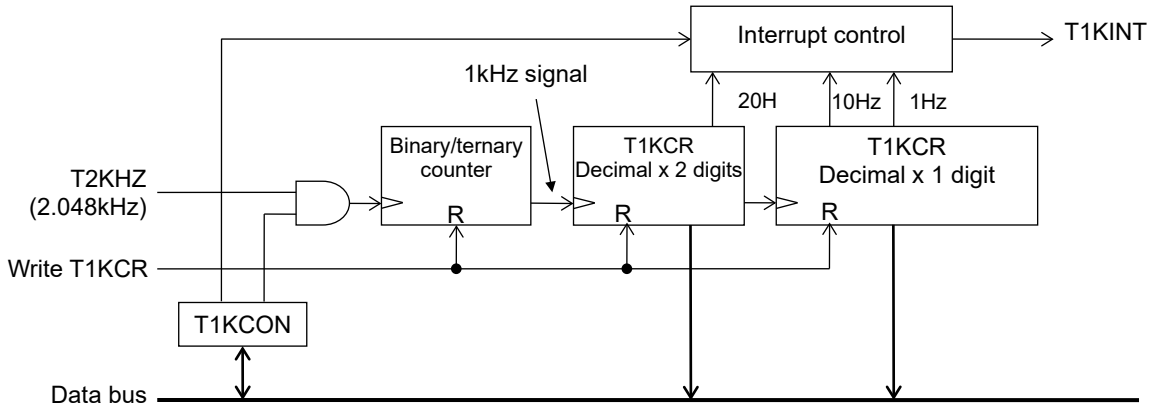
This LSI includes a 1 kHz timer to measure 1/1000 seconds.
The 1 kHz timer counts the 1 kHz signal created by dividing the T2KHZ output frequency (2.048 kHz) of the low-speed time base counter (LTBC) and generates 80Hz/60Hz/40Hz/20Hz/10Hz/1Hz interrupt as 1 kHz timer interrupt.
With the 1 kHz timer, 1/1000 second, which is difficult to generate on a time-base-counter basis, represented by a decimal number can be obtained easily. The timer can be applied to period measurement for stopwatches.
For the timer base counter, see Chapter 9, “Time Base Counter”.

13.1.1 Features

- 80Hz/60Hz/40Hz/20Hz/10Hz/1Hz interrupt select function

13.1.2 Configuration

Figure 13-1 shows the configuration of the 1 kHz timer.



T1KCON : 1 kHz timer control register
T1KCR : 1 kHz timer count register

Figure 13-1 Configuration of 1 kHz Timer

13.2 Description of Registers

13.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_6000	1 kHz timer base address	TIMER1K	-	-	-
0x00	1 kHz timer count register	T1KCR	R/W	32	0x0000_0000
0x04	1 kHz timer control register	T1KCON	R/W	32	0x0000_0000

13.2.2 1kHz Timer Count Register (T1KCR)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	T1KC												
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

T1KCR is special function registers (SFRs) to read the decimal count values of the 1 kHz timer.
When the write operation to T1KCR, the valid bit of T1KCR is "0".

Bit No	Bit name	Description
12 to 0	T1KC	T1KC indicates the count values of the 1 kHz timer with binary-coded decimal (0 to 1999) T1KC[12] : a figure for 1 second. T1KC[11:8] : a figure for 1/10 second. T1KC[7:4] : a figure for 1/100 second. T1KC[3:0] : a figure for 1/1000 second.

13.2.3 1kHz Timer Control Register (T1KCON)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	T1KSEL			T1KRUN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

T1KCON is a special function register (SFR) to control the 1 kHz timer.

Bit No	Bit name	Description
3 to 1	T1KSEL	T1KSEL is used to select the interrupt period of the 1 kHz timer. 000: 10Hz (Initial value) 001: 20Hz 010: 1Hz 011: 1Hz 100: 40Hz 101: 60Hz 110: 80Hz 111: 1Hz
0	T1KRUN	T1KRUN bit is used to control start/stop of the count operation of the 1 kHz timer counter. 0: Stops timer operation (Initial value) 1: Starts timer operation

13.3 Description of Operation

By setting the T1KRUN bit of the 1kHz timer control register (T1KCON) to “1”, the 1kHz timer starts counting of the 1kHz timer counter registers (T1KCR).
By dividing the T2KHz signal (2.048kHz) of the low-speed timer base counter (LTBC) by the binary/ternary counter, the timer generates a 1kHz signal. Based on the 1kHz signal, a 1kHz timer interrupt request signal (T1KINT) is generated by the decimal counters of T1KCR. The period of the 1kHz timer interrupt can be selected from 80Hz/60Hz/40Hz/20Hz/10 Hz/1 Hz using the T1KSEL bit of T1KCON.
When write operation is performed for T1KCR, the value of the binary/ternary counter and the value of T1KCR is cleared to “0”.
Data can be read from T1KCR. When reading data from T1KCR in the 1kHz timer operation start state, read T1KCR twice and check that the values match to prevent the reading of undefined data during counting.

13.3.1 Start Timing

The 1kHz timer starts operating at first falling edge T2KHZ after T1KRUN rises, then it counts up at second falling edge.

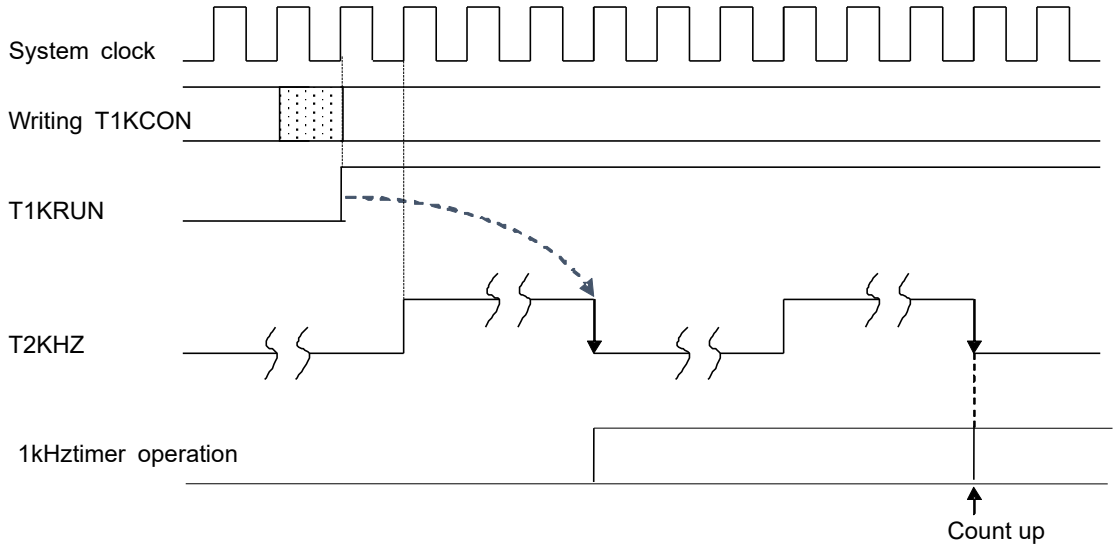


Figure13-2 Start Timing of 1kHz Timer

Chapter 14

Watchdog Timer (WDT)

14. Watchdog Timer (WDT)

14.1 Overview

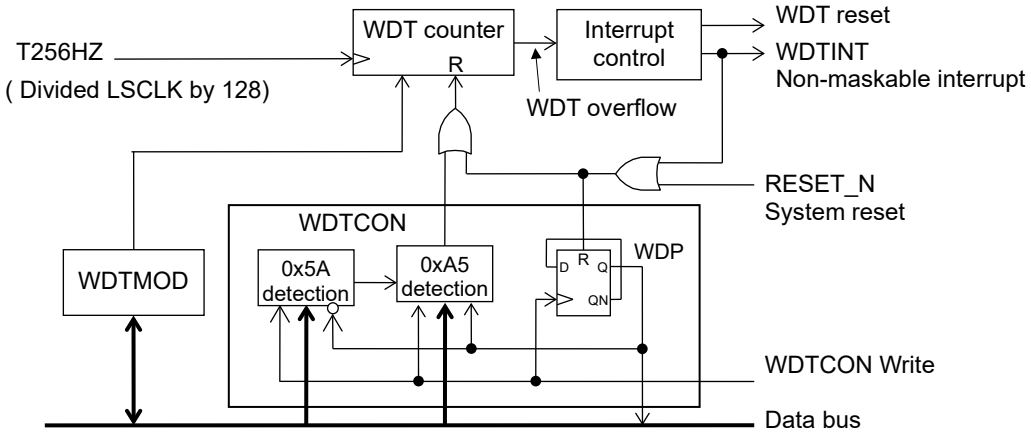
Watchdog timer is free run counter that is used for detection of program abnormal behavior.
The watchdog timer starts counting automatically after system reset release and requests WDT interrupt when the first overflow occurs.
When the second overflow occurs, the watchdog timer generates a WDT reset signal and shifts the mode to a system reset mode.
For WDT interrupt see Chapter 7, “Interrupts,” and for WDT reset see Chapter 4, “Reset Function”.

14.1.1 Features

- Free running (stop setting at any time or in the HALT-mode is available by code-option)
- Count low-speed clock 128 period.
- One of four types of overflow periods (125ms, 500ms, 2s, and 8s @LSCLK=32.768kHz) selectable by software
- Requests a WDT interrupt (non-maskable interrupt) by the first overflow
- Reset generated by the second overflow

14.1.2 Configuration

Figure 14-1 shows the configuration of the watchdog timer.



WDTCON : Watchdog timer control register
WDTMOD : Watchdog timer mode register

Figure 14-1 Configuration of Watchdog Timer

14.2 Description of Registers

14.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_3000	Watchdog timer base address	WDT	-	-	-
0x00	Watchdog timer control register	WDTCON	R/W	32	0x0000_0000
0x04	Watchdog timer mode register	WDTMOD	R/W	32	0x0000_0002

14.2.2 Watchdog Timer Control Register (WDTCON)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	d[7:1]							WDP/ d[0]
R/W	–	–	–	–	–	–	–	–	W	W	W	W	W	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDTCON is a special function register (SFR) to control the Watchdog Timer.
When some data is written to WDTCON, the value of the internal pointer (WDP) is reversed.
WDT counter is cleared when 0x5A and 0xA5 is written to WDTCON in succession.
The value in WDP is read from bit0 when WDTCON is read. In this time "0" is read from bit 7 to bit 1.
WDP is set to "0" at system reset and at WDT counter overflow.

To clear the WDT counter, write "0x5A" when WDP state is "0", and then write "0xA5" when WDP state is "1".
The WDT counter will not be cleared if each clear data is written with a wrong WDP.

[Note]

- **When the WDT interrupt (WDTINT) is occurred by the WDT counter first overflow, the WDT counter and the internal pointer (WDP) are initialized for 1/2 clock period of low-speed clock (approximately 15.26us@32.768kHz). Therefore, the writing to the WDTCON becomes invalid during the period and WDP is not reversed. In processing clear WDT when WDT interrupt occur and system clock is in high-speed clock state, confirm that WDP is inverted by writing to WDTCON and confirm the writing to WDTCON is done normally.**

14.2.3 Watchdog Timer Mode Register (WDTMOD)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	WDTP	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

WDTMOD is a special function register to set the overflow period of the WDT counter.

Bit No	Bit name	Description
1 to 0	WDTP	WDTP is used to select an overflow period of the watchdog timer (T_{WOV}). 00: 4096 counts at LSCLK (125ms*) 01: 16384 counts at LSCLK (500ms*) 10: 65536 counts at LSCLK (2s*) (Initial value) 11: 262144 counts at LSCLK (8s*) * : where calculated at LSCLK = 32.768kHz

14.3 Description of Operation

The WDT counter starts counting after the system reset has been released and the low-speed clock (LSCLK) oscillation starts.
The WDT counter can be cleared by writing 0x5A with the internal pointer (WDP) "0", then writing 0xA5 with the internal pointer (WDP) "1".
The WDP is set to “0” at the time of system reset or the WDT counter overflow. And it is inverted whenever data is written to WDTCON.
A watchdog timer interrupt (WDTINT) occurs when the WDT counter is not cleared within the WDT counter overflow period (T_{WOV}). And then WDT reset occurs and the mode shifts to a system reset mode, when the watchdog timer interrupt and overflow occur again before the WDT counter is cleared by the software processing performed.
For the overflow period (T_{WOV}) of the WDT counter, one of 125ms, 500ms, 2s, and 8s can be selected by the watchdog mode register (WDTMOD).
Clear the WDT counter within the clear period of the WDT counter shown in Table14-1.

Table14-1 Clear Period of WDT Counter

WDTP	T_{WOV}	T_{WCL}
00	4096 counts at LSCLK (125ms*)	3968 counts at LSCLK (Approx.121ms*)
01	16384 counts at LSCLK (500ms*)	16256 counts at LSCLK (Approx.496ms*)
10	65536 counts at LSCLK (2s*)	65408 counts at LSCLK (Approx.1996ms*)
11	262144 counts at LSCLK (8s*)	262016 counts at LSCLK (Approx.7996ms*)

* : where calculated at LSCLK = 32.768kHz

[Note]

- **WDT counter clock is T256HZ that is divided by 128 of LSCLK. Therefore, keep on supply the LSCLK during not reset and STOP-mode conditions.**
- **The time of Table14-1 is changed with the frequency of the LSCLK to be used. It is calculable by the frequency of low speed crystal oscillator to be used as follows.**
  Example) In case of WDT=00 and LSCLK=32.768kHz,
  Twov = 1 / ((32.768[kHz]) / 128[dividing]) × 32 [clock] = 125 [msec]

Figure 14-2 shows an example of watchdog timer operation.

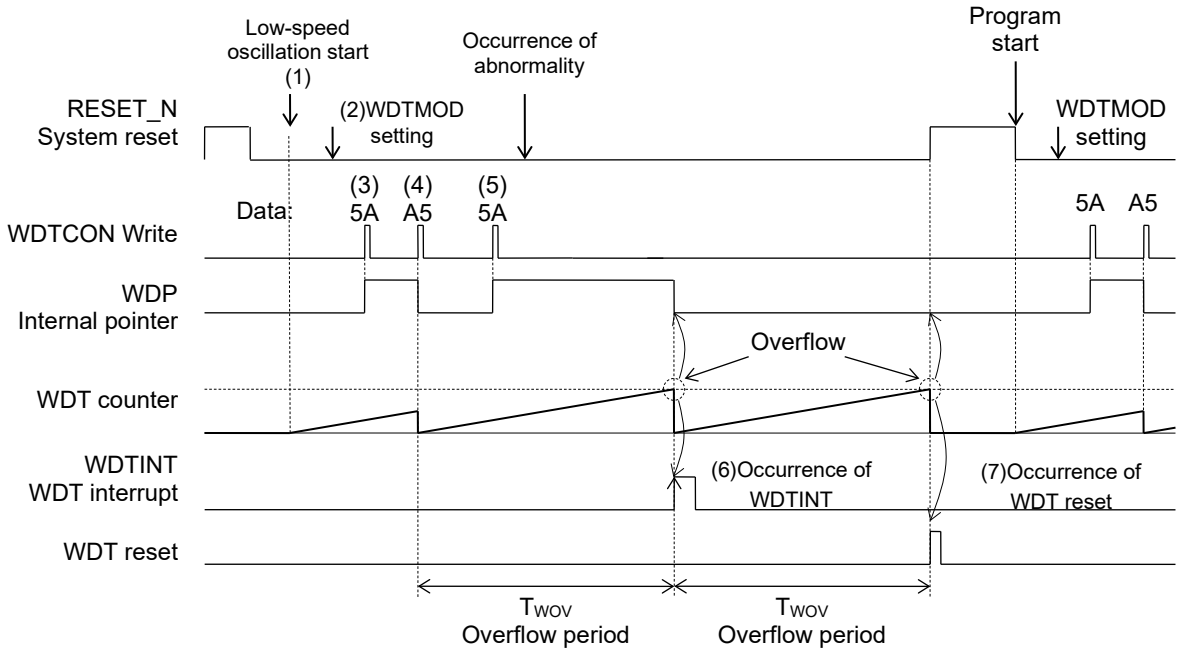


Figure 14-2 Example of Watchdog Timer Operation

1) The WDT counter starts counting after the system reset has been released and the low-speed clock oscillation start.
2) The overflow period of the WDT counter (T_{WOV}) is set to WDTMOD.
3) Write 0x5A to WDTCON. (Internal pointer changes from "0" to "1")
4) Write 0xA5 to WDTCON and clear the WDT counter. (Internal pointer changes from "1" to "0")
5) Write 0x5A to WDTCON. (Internal pointer changes from "0" to "1")
6) If abnormalities occur and the writing of 0xA5 is not performed, WDT counter overflows. Watchdog timer interrupt occurs because the overflow is the first overflow after reset of WDT counter. In addition, during the period of the half clock of LSCLK, WDT counter and internal pointer are initialized. While it is initialized, the writing to WDTCON becomes invalid, and internal pointer doesn't invert.
7) If the WDT counter is not cleared even by the software processing performed following a watchdog timer interrupt and the WDT counter overflows again, WDT reset occurs and the mode is shifted to a system reset mode.

[Note]

- **In STOP mode, the watchdog timer operation also stops. The HALT mode is released when the WDT interrupt occurs in HALT mode.**
- **The watchdog timer cannot detect all the abnormal operations. Even if the CPU loses control, the watchdog timer cannot detect the abnormality in the operation state in which the WDT counter is cleared.**

14.3.1 The process example when not using Watchdog Timer

Watchdog timer is a free-run counter, so it cannot be stopped. Even when the watchdog timer function is not used as fail-safe function, it is necessary to clear the WDT counter.
The example programming code is shown, in order to clear the WDT counter when WDT interrupt occurs and control the system reset by WDT.

The example of programming code:

```
do
 {
    WDT->WDTCON = 0x5a;
 } while((WDT->WDTCON) & 0x1 != 0x1)
    WDT->WDTCON = 0xa5;
```

14.3.2 Code option

Normally, the watchdog timer operates when not in the STOP mode. The watchdog timer function stops if setting the WDTMD bit in the code option to “0”. The watchdog timer function stops in the HALT mode if setting the WDTPWMD0 bit in the code option to “0” and WDTPMD bit to “1”. When entering HALT mode, the counter is cleared and counting starts when HALT mode is released (including transition to DMA transfer mode).

Chapter 15

Three-Phase Motor Control PWM (NTMS)

15. Three-Phase Motor Control PWM (NTMS)

15.1 Overview

This block is equipped with 3 channels of 16-bit timers with two PWM outputs as three-phase motor control PWM outputs.

15.1.1 Features

- Counting type
  Up-count (sawtooth wave mode) or Up/Down-count (triangle wave mode)
- PWM output waveform
  Sawtooth wave mode that is edge-aligned output
  Triangle wave mode that is center-aligned output

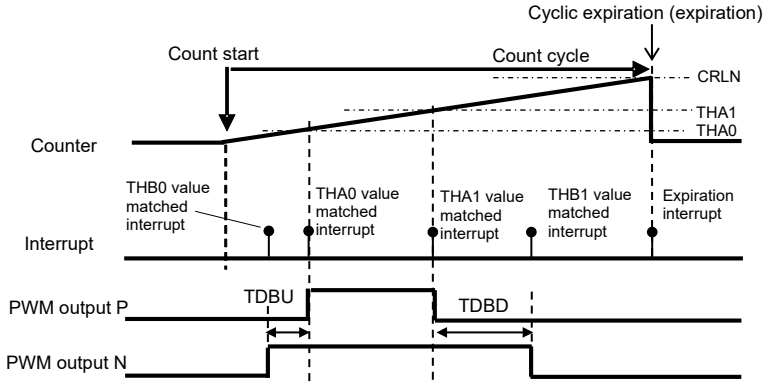


Figure 15-1 PWM Waveform Generation (Sawtooth Wave Mode)

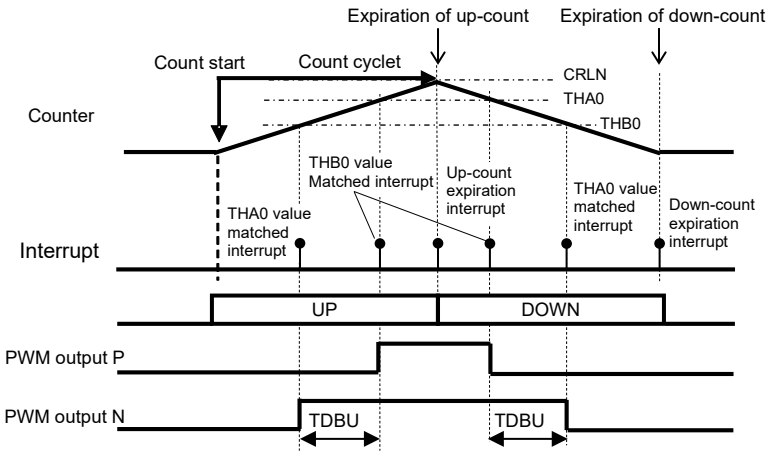


Figure 15-2 PWM Generation (Triangle Wave Mode)

Triangle

- PWM output control
  Inverted output
  Emergency stop control by analog-comparator and external pin input
  Emergency stop control by software
  Dynamic parameter update during operation

- Request function
  Compare match (cycle, threshold value): an interrupt is occurred at match the setting value
  AD conversion interlocking function : starting trigger to SA-ADC is occurred at match the setting value

15.1.2 Configuration

Figure 15-3 shows the configuration of the PWM (NTMS).

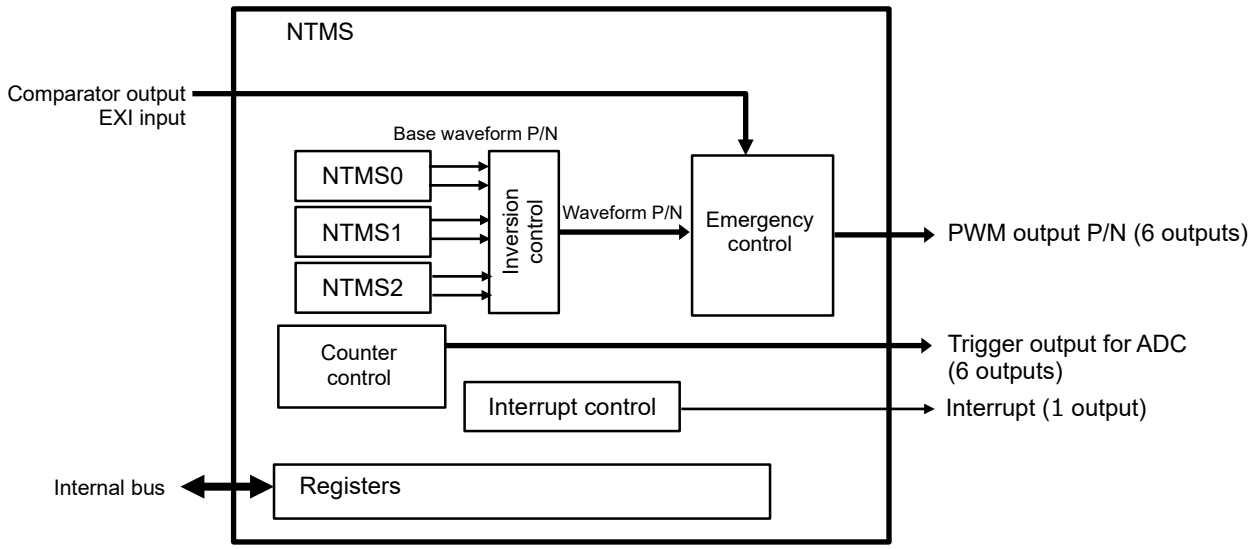


Figure 15-3 Configuration of the PWM (NTMS)

15.1.3 Pins

Table 15-1 PWM Output

NTMS internal name	LSI functional pin name
PWM output P	PWM00, PWM10, PWM20
PWM output N	PWM01, PWM11, PWM21

15.2 Description of Registers

15.2.1 List of Registers

Table15-2 List of Common Registers

Address /Offset	Name	Symbol	R/W	Writing on run	Initial value
0x40008000	NTMS common base address	TMR0_COMMON	-	-	-
0x00	Timer start register	TMON	R/W	OK	0x00000000
0x04	Timer stop register	TMOFF	W	OK	0x00000000
0x40	Reserved	-	-	-	-
0x44	Timer break setting register	TMBRK	R/W	OK	0x00000000
0x48	Timer buffer control register	TMBE	R/W	OK	0x00000000
0x50	Timer emergency processing start register	TMES	R/W	OK	0x00000000
0x54	Timer emergency processing stop register	TMEE	W	OK	0x00000000
0x58	Reserved	-	-	-	-
0xC0	Timer emergency stop trigger select register	TMGMSEL	R/W	NG	0x00000000
0xC4	Timer emergency stop trigger control register	TMGMC	R/W	OK	0x00000000
0xC8	Reserved	-	-	-	-

Table15-3 List of Registers by channel

Address /Offset	Name	Symbol	R/W	Writing on run	Initial value
0x40008100	NTMS0 base address	TMR0_NTMS00	-	-	-
0x40008200	NTMS1 base address	TMR0_NTMS01	-	-	-
0x40008300	NTMS2 base address	TMR0_NTMS02	-	-	-
0x00	NTMSn timer mode select register	TMD	R/W	NG	0x00000000
0x04	NTMSn timer control register	TMCTL	R/W	NG	0x00000000
0x08	NTMSn timer cycle setting register	TMLEN	R/W	OK	0x00000000
0x10	NTMSn timer input trigger setting register	TMTG	R/W	NG	0x00777777
0x14	NTMSn timer cycle setting buffer register	TMLENB	R/W	OK	0x00000000
0x20	NTMSn timer waveform threshold setting register A0	TMTHA0	R/W	OK	0x00000000
0x24	NTMSn timer waveform threshold setting register A1	TMTHA1	R/W	OK	0x00000000
0x28	Reserved	-	-	-	-
0x2C	Reserved	-	-	-	-
0x30	NTMSn timer waveform threshold setting buffer register A0	TMTHA0B	R/W	OK	0x00000000
0x34	NTMSn timer waveform threshold setting buffer register A1	TMTHA1B	R/W	OK	0x00000000
0x38	NTMSn timer emergency stop setting register	TMGM	R/W	NG	0x00000000
0x40	NTMSn timer interrupt mask setting register	TMIMSK	R/W	OK	0x00000000
0x44	NTMSn timer interrupt source storing register	TMISTS	R/W	OK	0x00000000
0x48	NTMSn timer RAW interrupt source storing register	TMISTR	R/W	OK	0x00000000
0x50	NTMSn timer count monitor register	TMMON	R/W	NG	0x00000000
0x5C	NTMSn timer status register	TMEST	R	-	0x00000000
0x60	Reserved	-	-	-	-
0x64	Reserved	-	-	-	-
0x68	NTMSn timer waveform dead-time setting register U	TMDBU	R/W	OK	0x00000000
0x6C	NTMSn timer waveform dead-time setting register D	TMDBD	R/W	OK	0x00000000
0x70	NTMSn timer waveform dead-time setting buffer register U	TMDBUB	R/W	OK	0x00000000
0x74	NTMSn timer waveform dead-time setting buffer register D	TMDBDB	R/W	OK	0x00000000
0x78	NTMSn timer AD conversion control register	TMADCT	R/W	NG	0x00000000

Address /Offset	Name	Symbol	R/W	Writing on run	Initial value
0x7C	NTMSn timer AD conversion start request timing register A	TMADA	R/W	OK	0x00000000
0x80	NTMSn timer AD conversion start request timing buffer register A	TMADAB	R/W	OK	0x00000000
0x84	NTMSn timer AD conversion start request timing register B	TMADB	R/W	OK	0x00000000
0x88	NTMSn timer AD conversion start request timing buffer register B	TMADBB	R/W	OK	0x00000000

15.2.2 NTMSn Timer Mode Select Register (TMD)

Offset :0x0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	rsvd		rsvd		BADB		BADA		BLEN		BTDE	rsvd	BTHA
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	rsvd	–	–	–	PDTE	TMMD			–
R/W	–	–	–	–	–	–	–	R/W	–	–	–	R/W	R/W	R/W	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMD is SFR to set mode.

Bit No	Bit name	Description
31 – 29	-	Reserved bits
28 – 25	rsvd	Reserved bits. Do not change from Initial value.
24 – 23	BADB	TMADB buffer operation setting (See the description of “BADA” for detail.) Update of TMADB is controlled by BADB and TMBE.N*_ADC.
22-21	BADA	TMADA buffer operation setting Sets the timing at which the value of the AD conversion start request timing buffer register is transferred to the AD conversion start request timing register during timer operation. 0: No transferred (Sawtooth wave mode) / No transferred (Triangle wave mode 3/4) 1: Transferred at expiration / Transferred at expiration of up-count 2: Setting prohibited / Transferred at expiration of down-count 3: Setting prohibited / Transferred at both expiration of up/down-count Update of TMADA is controlled by BADA and TMBE.N*_ADC.
20 – 19	BLEN	TMLEN buffer operation setting 0 : No buffer 1 : Single buffer 2 : Setting prohibited 3 : Setting prohibited The timing at which the value of the TMLENB register is transferred to the TMLEN register during timer operation: Sawtooth wave mode : Transferred at expiration Triangle wave mode : Transferred at expiration of down-count Update of TMLEN and TMLENB is controlled by this bit and TMBE.N*_LEN.
18	BTDE	TMDBU and TMDBD buffer operation setting 0 : No buffer 1 : Single buffer Update of TMDBU and TMDBD is controlled by this bit and TMBE.N*_TDE, TMBE.N*TAB.
17	rsvd	Reserved bit. Do not change from Initial value.
16	BTHA	TMTHA0 and TMTHA1 buffer operation setting 0 : No buffer 1 : Single buffer Update of TMTHA0 and TMTHA1 is controlled by this bit and TMBE.N*TAB.
15 – 9	-	Reserved bits
8	rsvd	Reserved bits. Do not change from Initial value.
7 – 5	-	Reserved bits
4	PDTE	Set this bit to “1” before operating.

Bit No	Bit name	Description
3 – 1	TMMD	Timer mode select 0 : Setting prohibited 1 : Sawtooth wave mode 2 : Setting prohibited 3 : Setting prohibited 4 : Setting prohibited 5 : Setting prohibited 6 : Triangle wave mode 3; Same threshold for rising and falling, Transferred at expiration of down-count 7 : Triangle wave mode 4; Same threshold for rising and falling, Transferred at both expiration of up/down-count
0	-	Reserved bit

15.2.3 NTMSn Timer Mode Control Register (TMCTL)

Offset :0x4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	UPNS	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rsvd					–	rsvd	–	–	–	–	CSCS	rsvd		MPRS	rsvd
R/W	R/W	R/W	R/W	R/W	R/W	–	R/W	–	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMCTL is SFR to control mode.

Bit No	Bit name	Description
31 – 18	-	Reserved bits
17 – 16	UPNS	Waveform inversion control 0 : Non inverted output 1 : Inverted output Bit [16] controls 'base waveform P' to generate 'Waveform P'. Bit [17] controls 'base waveform N' to generate 'Waveform N'.
15 – 11	rsvd	Reserved bits. Do not change from the initial value.
10	-	Reserved bit
9	rsvd	Reserved bit. Do not change from the initial value.
8– 5	-	Reserved bits
4	CSCS	Count start initialization enable 0 : Counter is not initialized at counter stop. The counter can be resumed to continue after counter stop. However counter is cleared in the following case. When counter value at start of the count is larger than or equal to CRLN of the TMLEN. 1 : Counter is initialized at counter stop. The counter is always started from the initial state.
3– 2	rsvd	Reserved bits. Do not change from Initial value.
1	MPRS	Selection for condition for ending count when timer enable is OFF 0 : Stop immediately 1 : Stop at cyiclic expiration in the sawtooth wave mode. Stop at expiration of down-count in the triangle wave mode.
0	rsvd	Reserved bit. Do not change from Initial value.

15.2.4 NTMSn Timer Cycle Setting Register (TMLEN)

Offset :0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRLN															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMLEN is SFR to set a cycle.

Bit No	Bit name	Description
31 – 16	-	Reserved bits
15 – 0	CRLN	Timer count cycle setting Sets timer cycle. Actual timer cycle is 'setting value + 1'. Setting range : 0x0001 - 0xFFFF (Cycle :2 - 65536)

15.2.5 NTMSn Timer Cycle Setting Buffer Register (TMLENB)

Offset :0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRLNB															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMLENB is SFR to operate as buffer for TMLEN.

Bit No	Bit name	Description
31 – 16	-	Reserved bits
15 – 0	CRLNB	Buffer for TMLEN This is used only when TMBE.N*_LEN = "0" and TMD.BLEN = "1".

15.2.6 NTMSn Timer Waveform Threshold Setting Register A0/A1 (TMTHA0/ TMTHA1)

Offset :0x20 (TMTHA0), 0x24 (TMTHA1)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THA0/THA1															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMTHA0/TMTHA1 is SFR to set timing for waveform generation.
The waveform is generated by calcurating settings of this register and TMDBU/TMDBD registers.

Table15-4 Description of Each Paramter

Parameter	Sawtooth wave mode (See Figure 15-4)	Triangle wave mode 3/4 (See Figure 15-5)
THA0	Rising of the base waveform P	Rising/Falling of the base waveform P
THA1	Falling of the base waveform P	-
THB0	Rising of the base waveform N = THA0 – TDBU	Rising/Falling of the base waveform N
THB1	Falling of the base waveform N = THA1 + TDBD	-

Table15-5 Restrictions

Parameter	Sawtooth wave mode	Triangle wave mode 3/4
THA0	0 ≤ THA0 < THA1	0 < THA0 < CRLN
THA1	THA0 < THA1 ≤ CRLN	-

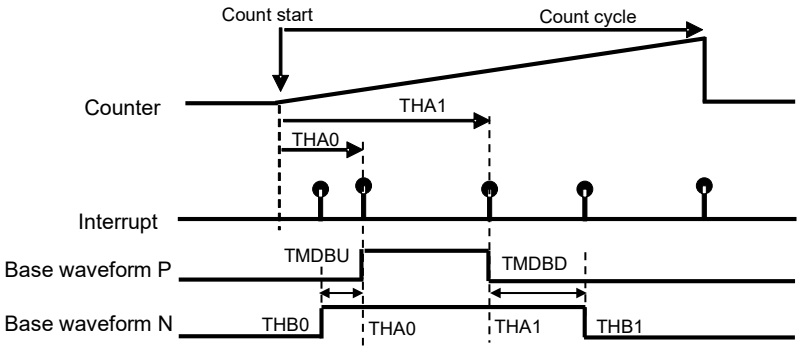


Figure 15-4 Waveforms and Threshold in the Sawtooth Wave Mode

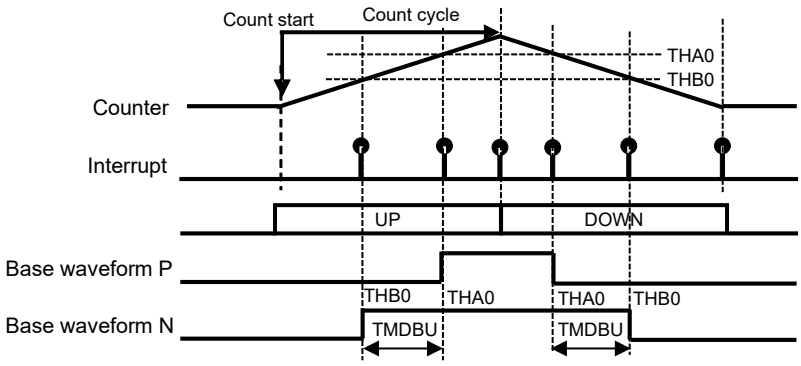


Figure 15-5 Waveforms and Threshold in the Triangle Wave Mode 3/4

15.2.7 NTMSn Timer Waveform Threshold Setting Buffer Register A0/A1 (TMTHA0B/ TMTHA1B)

Offset :0x30 (TMTHA0B), 0x34 (TMTHA1B)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THA0B/THA1B															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMTHA0B/TMTHA1B is SFR to operate as buffer for TMTHA0/TMTHA1 respectively.

15.2.8 NTMSn Timer Waveform Dead-time Setting Register U/D (TMDBU/TMDBD)

Offset :0x68 (TMDBU), 0x6C (TMDBD)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDBU/TDBD															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMDBU/TMDBD is SFR to set the dead-time.

Table 15-6 Description of TMDBU/TMDBD

Register	Sawtooth wave mode	Triangle wave mode 3/4
TDBU	Dead-time for THA0	Dead-time at Up-count
TDBD	Dead-time for THA1	-

Table 15-7 Restrictions of TMDBU/TMDBD

Register	Sawtooth wave mode	Triangle wave mode 3/4
TDBU	TDBU < THA1 (TDBU + TMDBD) < CRLN	TDBU < CRLN – 1
TDBD	THA0 < (CRLN – TDBD) (TDBU + TDBD) < CRLN	-

15.2.9 NTMSn Timer Waveform Dead-time Setting Buffer Register U/D (TMDBUB/TMDBDB)

Offset :0x70 (TMDBUB), 0x74 (TMDBDB)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDBUB/TDBDB															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMDBUB/TMDBDB is SFR to operate as buffer for TMDBU/TMDBD respectively.

15.2.10 NTMSn Timer Input Trigger Setting Register (TMTG)

Offset :0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	rsvd	rsvd	rsvd	rsvd	rsvd	EVS_EMG_SEL	EVS_EMG		
R/W	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1

TMTG is SFR to set input trigger.

Bit No	Bit name	Description
31 – 25	-	Reserved bits
24 – 20	rsvd	Reserved bits. Do not change from Initial value.
19	EVS_EMG_SEL	Emergency stop external input format selection 0 : When trigger output block and timer are operating with different clocks Trigger signal is delayed by two cycles of timer clock 1 : When trigger output block and timer are operating with synchronized clock
18 – 16	EVS_EMG	Emergency stop external input format selection Sets input condition to perform the emergency stop. 100 : “High” level 111 : Disabled Others : Setting prohibited
15 – 0	rsvd	Reserved bits. Do not change from Initial value.

15.2.11 NTMSn Timer Emergency Stop Setting Register (TMGM)

Offset :0x38

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UEDV		UEDT		UEPS		UENS		–	–	–	–	–	UEST	UERT	UEEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMGM is SFR to set the emergency stop setting.

Bit No	Bit name	Description
31 – 16	-	Reserved bits
15 – 14	UEDV	PWM output at dead-time when the emergency stop 0 : "L" level output 1 : "H" level output [15] PWM output N site [14] PWM output P site
13 – 12	UEDT	Dead-time setting when the emergency stop 0 : Dead-time off at emergency stop start, dead-time off at emergency stop end 1 : Dead-time on at emergency stop start, dead-time off at emergency stop end 2 : Dead-time off at emergency stop start, dead-time on at emergency stop end 3 : Dead-time on at emergency stop start, dead-time on at emergency stop end Waveform is controlled by TDBU during the dead-time start. When buffer of dead-time enabled, it is controlled by TDBU value that is latched to internal. Waveform is controlled by TDBD during the dead-time end. When buffer of dead-time enabled, it is controlled by TDBD value that is latched to internal.
11 – 10	UEPS	Setting of PWM output P level during the emergency stop 0 : Fixed to "0" 1 : Fixed to "1" 2 : Fixed to "High-impedance" 3 : Fixed to "High-impedance"
9 – 8	UENS	Setting of PWM output N level during the emergency stop 0 : Fixed to "0" 1 : Fixed to "1" 2 : Fixed to "High-impedance" 3 : Fixed to "High-impedance"
7 – 3	-	Reserved bits
2	UEST	Setting of emergency stop start timing when emergency stop is enabled. 0 : Start immediately 1 : Start at expiration of counter in the sawtooth wave mode. Start at expiration of down-count in the triangle wave mode.
1	UERT	Setting of recovery (emergency stop end) timing when emergency stop is disabled. 0 : Recovery immediately 1 : Recovery at expiration of counter in the sawtooth wave mode. Recovery at expiration of down-count in the triangle wave mode.
0	UEEN	Emergency stop trigger enable 0 : Disabled 1 : Enabled

15.2.12 NTMSn Timer Interrupt Mask Register (TMIMSK)

Offset :0x40

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	IMSK_TDB1	IMSK_TDB0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	IMSK_TB1	IMSK_TB0	IMSK_TA1	IMSK_TA0	IMSK_MOU	–	IMSK_CMP
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMIMSK is SFR to set an interrupt mask.

[Description of each bit]
This bit is used to set mask for the corresponding interrupt.
0 : No masked. Interrupt request is enabled.
1 : Masked. Interrupt request is disabled.

Bit No	Bit name	Description
31 – 18	-	Reserved bits
17	IMSK_DTB1	THB1 abnormal status
16	IMSK_DTB0	THB0 abnormal status
15	-	Reserved bits
14– 7	rsvd	Reserved bits. Set each bit to “1” before operating.
6	IMSK_TB1	THB1 value matched interrupt
5	IMSK_TB0	THB0 value matched interrupt
4	IMSK_TA1	THA1 value matched interrupt
3	IMSK_TA0	THA0 value matched interrupt
2	IMSK_MOU	Up-count expiration interrupt in the triangle wave mode
1	-	Reserved bit
0	IMSK_CMP	Timer expiration interrupt

15.2.13 NTMSn Timer Interrupt Source Storing Register (TMISTS)

Offset :0x44

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ISTS_DTB1	ISTS_DTB0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	ISTS_TB1	ISTS_TB0	ISTS_TA1	ISTS_TA0	ISTS_MOU	–	ISTS_CMP
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMISTS is SFR that store interrupt.

[Description of each bit]
This bit stores the corresponding status after TMIMSK is masking. This bit is not to “1” when the masking is enabled.
At reading
0 : No interrupt request
1 : Has interrupt request
At writing
0 : Invalid
1 : Clear interrupt request. The corresponding bit of TMISRT is cleared too.

Bit No	Bit name	Description
31 – 18	-	Reserved bits
17	ISTS_DTB1	THB1 abnormal status
16	ISTS_DTB0	THB0 abnormal status
15	-	Reserved bit
14– 7	rsvd	Reserved bits
6	ISTS_TB1	THB1 value matched interrupt
5	ISTS_TB0	THB0 value matched interrupt
4	ISTS_TA1	THA1 value matched interrupt
3	ISTS_TA0	THA0 value matched interrupt
2	ISTS_MOU	Up-count expiration interrupt in the triangle wave mode This bit does not become to “1” while emergency stop processing.
1	-	Reserved bits
0	ISTS_CMP	Timer expiration interrupt

15.2.14 NTMSn Timer RAW Interrupt Source Storing Register (TMISTR)

Offset :0x48

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ISTR_DTB1	ISTR_DTB0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	ISTR_TB1	ISTR_TB0	ISTR_TA1	ISTR_TA0	ISTR_MOU	–	ISTR_CMP
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMISTR is SFR that stores raw interrupt.

[Description of each bit]
This bit stores the corresponding raw status before TMIMSK is masking.
At reading
0 : No interrupt request
1 : Has interrupt request
At writing
0 : Invalid
1 : Clear interrupt request. The corresponding bit of TMISTS is cleared too.

Bit No	Bit name	Description
31 – 18	-	Reserved bits
17	ISTR_DTB1	THB1 abnormal raw status
16	ISTR_DTB0	THB0 abnormal raw status
15	-	Reserved bit
14– 7	rsvd	Reserved bits
6	ISTR_TB1	THB1 value matched interrupt raw status
5	ISTR_TB0	THB0 value matched interrupt raw status
4	ISTR_TA1	THA1 value matched interrupt raw status
3	ISTR_TA0	THA0 value matched interrupt raw status
2	ISTR_MOU	Raw status of the up-count expiration interrupt in the triangle wave mode
1	-	Reserved bits
0	ISTR_CMP	Timer expiration interrupt raw status

15.2.15 NTMSn Timer Count Monitor Register (TMMON)

Offset :0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MON															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMMON is SFR that stores timer count value.

Bit No	Bit name	Description
31 – 0	MON	Timer count monitor At reading Timer count value is indicated. At writing “0x0000_0000” Timer count is initialized to “0x0000”. Writing other than “0x0000_0000” is prohibited.

[Note]

- **Writing to this register is prohibited during timer operation. Initialize the timer count value while the timer is stopped.**
- **The count starts from 0, if counter value is larger than or equal to CRLN value of TMLEN at the start of counting.**

15.2.16 NTMSn Timer Status Register (TMEST)

Offset :0x5C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	ISTS_UP	ADCNT		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMEST is SFR to indicate status.

Bit No	Bit name	Description
31 – 4	-	Reserved bits
3	ISTS_UP	Up-count/Down-count status 0 : Up-count 1 : Down-count
2 – 0	ADCNT	Indicates count to reduce the number of AD conversion

15.2.17 NTMSn Timer AD Convertion Control Register (TMADCT)

Offset :0x78

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	rsvd	rsvd	ADEB	ADEA	–	–	–	–	rsvd	rsvd	HDEB	HDEA
R/W	–	–	–	–	R/W	R/W	R/W	R/W	–	–	–	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	THFR		THFU			rsvd	rsvd	LTOB	LTOA
R/W	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMADCT is SFR to set for AD conversion start request.

Bit No	Bit name	Description
31 – 28	-	Reserved bits
27, 26	rsvd	Reserved bits. Do not change from Initial value.
25	ADEB	AD conversion start request B direct control selection*1 (See the description of “ADEA” for detail.)
24	ADEA	AD conversion start request A direct control enable*1 Sets “1” to enable the trigger for ADC. 0 : Disabled 1 : Enabled
23 – 20	-	Reserved bits
19, 18	rsvd	Reserved bits. Do not change from Initial value.
17	HDEB	AD conversion start request B threshold TMADB enable *1(See the description of “HDEA” for detail.)
16	HDEA	AD conversion start request A threshold TMADA enable *1 Sets “1” to enable the trigger for ADC. 0 : Disabled 1 : Enabled
15 – 9	-	Reserved bits
8 – 7	THFR	Sets timing for count to reduce the number of AD conversion. 0: No reduced / No reduced 1: At expiration / At expiration of up-count 2: At expiration / At expiration of down-count 3: At expiration / At both expiration of up/down-count
6 – 4	THFU	Set the number of count to reduce the number of AD conversion. Setting value = number of a reduction (0 to 7 times)
3, 2	rsvd	Reserved bits. Do not change from Initial value.
1	LTOB	AD conversion start request B reduction function enable (See the description of “LTOA” for detail.)
0	LTOA	AD conversion start request A reduction function enable 0 : Disabled 1 : Enabled

*1 : See section “15.3.5 AD Conversion Request“ for detail.

15.2.18 NTMSn Timer AD Conversion Start Request Timing Register A/B (TMADA/TMADB)

Offset :0x7C (TMADA), 0x84 (TMADB)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TADA/TADB															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMADA/TMADB are SFRs to set the threshold for AD conversion start request.

Bit No	Bit name	Description
31 – 16	-	Reserved bits
15 – 0	TADA	Threshold setting for the AD conversion start request A
	TADB	Threshold setting for the AD conversion start request B

[Note]
- **Setting value should be less than the CRLN of TMLEN register.**

15.2.19 NTMSn Timer AD Conversion Start Request Timing Buffer Register A/B (TMADAB/ TMADBB)

Offset :0x80 (TMADAB), 0x88 (TMADBB)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TADAB/TADBB															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMADAB/TMADBB is SFR to operate as buffer for TMADA/TMADB respectively.

15.2.20 COMMON Timer Start Register (TMON)

Offset :0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	MCON_NS2	MCON_NS1	MCON_NS0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMON is SFR to start operation for each timer.

[Description of each bit]
This is used to start and indicate status the corresponding timer
At writing,
0: Invalid
1: Starts operation
At reading,
0: Stopped
1: Runing

Bit No	Bit name	Description
31– 19	-	Reserved bits
18	MCON_NS2	NTMS2 timer
17	MCON_NS1	NTMS1 timer
16	MCON_NS0	NTMS0 timer
15– 0	-	Reserved bits

[Note]
- **If writing “1” to this bit again while the timer is started, the counter is initialized and continues operation.**

15.2.21 COMMON Timer Stop Register (TMOFF)

Offset :0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	MCOFF_NS2	MCOFF_NS1	MCOFF_NS0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMOFF is SFR to stop operation for each timer.

[Description of each bit]
This is used to stop the corresponding timer
At writing,
0: Invalid
1: Stop operation

Bit No	Bit name	Description
31– 19	-	Reserved bits
18	MCOFF_NS2	NTMS2 timer
17	MCOFF_NS1	NTMS1 timer
16	MCOFF_NS0	NTMS0 timer
15– 0	-	Reserved bits

15.2.22 COMMON Timer CPU Break Setting Register (TMBRK)

Offset :0x44

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	BRK_NS	BRK_NS1	BRK_NS0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRDMSK	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMBRK is SFR to set each timer state when CPU is stopped by a such as software debugging break.

[Description of bit 18 to 16]
The corresponding timer state select at CPU break (debugging)
0: Count stop
1: Count not stopped

Bit No	Bit name	Description
31– 19	-	Reserved bits
18	BRK_NS2	NTMS2 timer
17	BRK_NS1	NTMS1 timer
16	BRK_NS0	NTMS0 timer
15	CRDMSK	Set this bit to “1” before operating.
14 – 0	-	Reserved bits

[Note]
- **It is recommended to set BRK_NS2 to 0 to “1”, since the PWM outputs are fixed while debugging break.**

15.2.23 COMMON Timer Buffer Control Register (TMBE)

Offset :0x48

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	N2_ADC	N2_TDE	N2_TAB	N2_LEN	N1_ADC	N1_TDE	N1_TAB	N1_LEN	N0_ADC	N0_TDE	N0_TAB	N0_LEN
R/W	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMBE is SFR to control buffer operation for each timer.

[Description of each bit]
This is used to contol the corresponding buffer operation
0: Buffer operatable
1: Buffer operation paused

Bit No	Bit name	Description
15-12	-	Reserved bits
11	N2_ADC	The buffer operation with combination TMADA/TMADB and TMADAB/TMADBB of NTMS2 It is controlled by this bit and NTMS2.TMD.BADA/B bit.
10	N2_TDE	The buffer operation with combination TMDBU/TMDBD and TMDBUB/TMDBDB of NTMS2 It is controlled by this bit and NTMS2.TMD.BTDE bit. Set N2_TAB bit to “0” too when buffer operation will be enabled.
9	N2_TAB	The buffer operation with combination TMTHA0/TMTHA1 and TMTHA0B/TMTHA1B of NTMS2 It is controlled by this bit and NTMS2.TMD.BTHA.
8	N2_LEN	The buffer operation with combination TMLEN and TMLENB of NTMS2 It is controlled by this bit and NTMS2.TMD.BLEN.
7	N1_ADC	The buffer operation with combination TMADA/TMADB and TMADAB/TMADBB of NTMS1 It is controlled by this bit and NTMS1.TMD.BADA/B bit.
6	N1_TDE	The buffer operation with combination TMDBU/TMDBD and TMDBUB/TMDBDB of NTMS1 It is controlled by this bit and NTMS1.TMD.BTDE bit. Set N1_TAB bit to “0” too when buffer operation will be enabled.
5	N1_TAB	The buffer operation with combination TMTHA0/TMTHA1 and TMTHA0B/TMTHA1B of NTMS1 It is controlled by this bit and NTMS1.TMD.BTHA.
4	N1_LEN	The buffer operation with combination TMLEN and TMLENB of NTMS1 It is controlled by this bit and NTMS1.TMD.BLEN.
3	N0_ADC	The buffer operation with combination TMADA/TMADB and TMADAB/TMADBB of NTMS0 It is controlled by this bit and NTMS0.TMD.BADA/B bit.
2	N0_TDE	The buffer operation with combination TMDBU/TMDBD and TMDBUB/TMDBDB of NTMS0 It is controlled by this bit and NTMS0.TMD.BTDE bit. Set N0_TAB bit to “0” too when buffer operation will be enabled.
1	N0_TAB	The buffer operation with combination TMTHA0/TMTHA1 and TMTHA0B/TMTHA1B of NTMS0 It is controlled by this bit and NTMS0.TMD.BTHA.
0	N0_LEN	The buffer operation with combination TMLEN and TMLENB of NTMS0 It is controlled by this bit and NTMS0.TMD.BLEN.

15.2.24 COMMON Timer Emergency Proccessing Start Register (TMES)

Offset :0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	MCES_NS2	MCES_NS1	MCES_NS0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMES is SFR to start the emergency processing (that is an emergency stop by software) for each timer.

[Description of each bit]
This is used to start the emergency processing and indicate state of the emergency processing, for the corresponding timer
At writing,
0: Invalid
1: Start the emergency processing
At reading,
0: No emergency processing
1: In the emergency processing

Bit No	Bit name	Description
31 – 19	-	Reserved bits
18	MCES_NS2	NTMS2 timer
17	MCES_NS1	NTMS1 timer
16	MCES_NS0	NTMS0 timer
15 – 0	-	Reserved bits

15.2.25 COMMON Timer Emergency Proccessing Stop Register (TMEE)

Offset :0x54

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	MCEE_NS02	MCEE_NS01	MCEE_NS00
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMES is SFR to stop the emergency processing by TMES register for each timer.

[Description of each bit]
This is used to stop the emergency processing for the corresponding timer.
At writing,
0: Invalid
1: Stop the emergency processing by TMES

Bit No	Bit name	Description
31 – 19	-	Reserved bits
18	MCEE_NS2	NTMS2 timer
17	MCEE_NS1	NTMS1 timer
16	MCEE_NS0	NTMS0 timer
15 – 0	-	Reserved bits

15.2.26 COMMON Timer Emergency Stop Trigger Select Register (TMGMSEL)

Offset :0xC0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	HLD1	POL1	EN1	SEL1			
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	HLD0	POL0	EN0	SEL0			
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMGMSEL is SFR to set trigger for the emergency stop.

Bit No	Bit name	Description
31 – 23	-	Reserved bits
22	HLD1	This is used to select input control for the emergency stop trigger 1 0 : Level control (Stopped at H level, and released at L level) 1 : Edge control (Stopped at rising-edge, and released by software clearing) The emergency stop is maintained until it is cleared by the software. The emergency status indication and clearing process should be performed in the TMGMC register.
21	POL1	This is used to invert the polarity of source for the emergency stop trigger 1. This bit is set to “1” to perform the emergency stop while trigger source is Low level.
20	EN1	This is used to enable the emergency stop trigger 1. 0 : Disabled 1 : Enabled
19-16	SEL1	This is used to select input source for the mergency stop trigger 1 0 : EXI0TGO, 1 : EXI1TGO, 2 : EXI2TGO, 3 : EXI3TGO, 4 : EXI4TGO, 5 : EXI5TGO, 6 : EXI6TGO, 7 : EXI7TGO, 8 : CMP0TRG, 9 : CMP1TRG, 10 : CMP2TRG
15 – 7	-	Reserved bits
6	HLD0	This is used to select input control for the emergency stop trigger 0 0 : Level control (Stopped at H level, and released at L level) Edge control (Stopped at rising-edge, and released by software clearing) The emergency stop is maintained until it is cleared by the software. The emergency status indication and clearing process should be performed in the TMGMC register.
5	POL0	This is used to invert the polarity of source for the emergency stop trigger 0. This bit is set to “1” to perform the emergency stop while trigger source is Low level.
4	EN0	This is used to enable the emergency stop trigger 0. 0 : Disabled 1 : Enabled
3– 0	SEL0	This is used to select input source for the emergency stop trigger 0 0 : EXI0TGO, 1 : EXI1TGO, 2 : EXI2TGO, 3 : EXI3TGO, 4 : EXI4TGO, 5 : EXI5TGO, 6 : EXI6TGO, 7 : EXI7TGO, 8 : CMP0TRG, 9 : CMP1TRG, 10 : CMP2TRG

[Note]

- **When the EN bit is "0", the trigger output is fixed at the Low level. Therefore, select “100” (the High level) as EVS_EMG of TMTG register for NTMSn.**

15.2.27 COMMON Timer Emergency Stop Trigger Control Register (TMGMC)

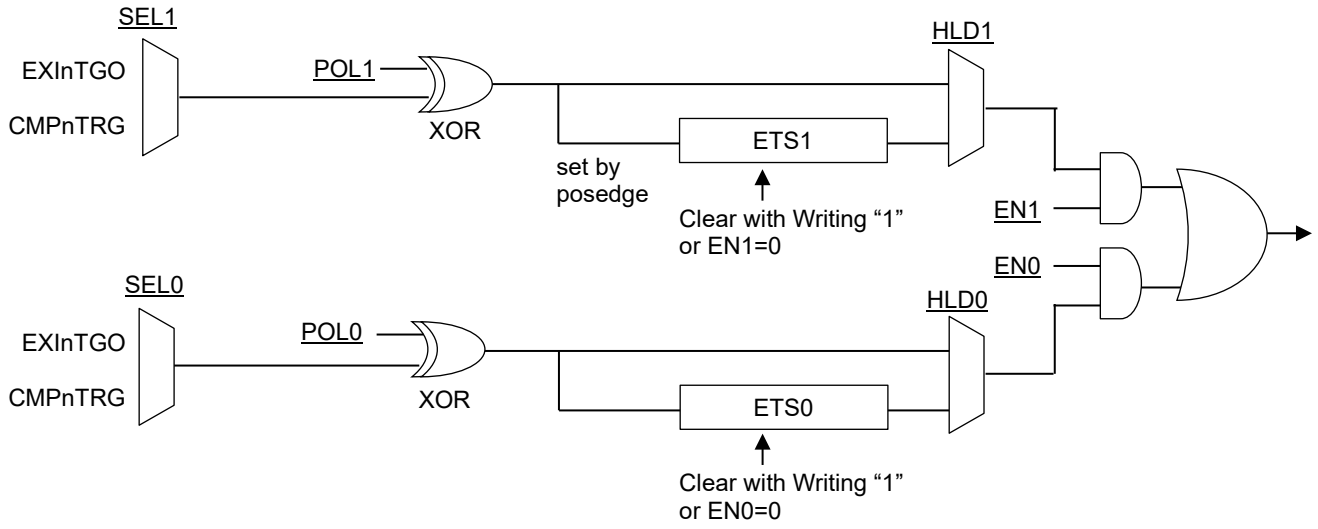
Offset :0xC4

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ETS1	ETS0
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMGMC is SFR to indicate an emergency trigger status and to release the status.

Bit No	Bit name	Description
31 – 2	-	Reserved bits
1	ETS1	ETS1 indicates the emergency stop request status by the emergency trigger 1. 0 : No emergency stop request. 1 : Has emergency stop request When HLD1 of TMGSEL register is “0”, writing to this bit is invalid. When HLD1 of TMGSEL register is “1”, this bit is cleared by writing “1” to this bit.
0	ETS0	ETS0 indicates the emergency stop request status by the emergency trigger 0. 0 : No emergency stop request. 1 : Has emergency stop request When HLD0 of TMGSEL register is “0”, writing to this bit is invalid. When HLD0 of TMGSEL register is “1”, this bit is cleared by writing “1” to this bit.



15.3 Description of Operation

15.3.1 Setting and Control

15.3.1.1 Setting Flow

Three channels timers with 6 phase outputs is operated at same time to control three-phase motor.
When starting multiple timers at the same time, set the system clock to the high-speed clock. If the system clock is the low-speed clock, the start of each of the multiple timers may be off by one cycle in the count clock. It is same at stop control.

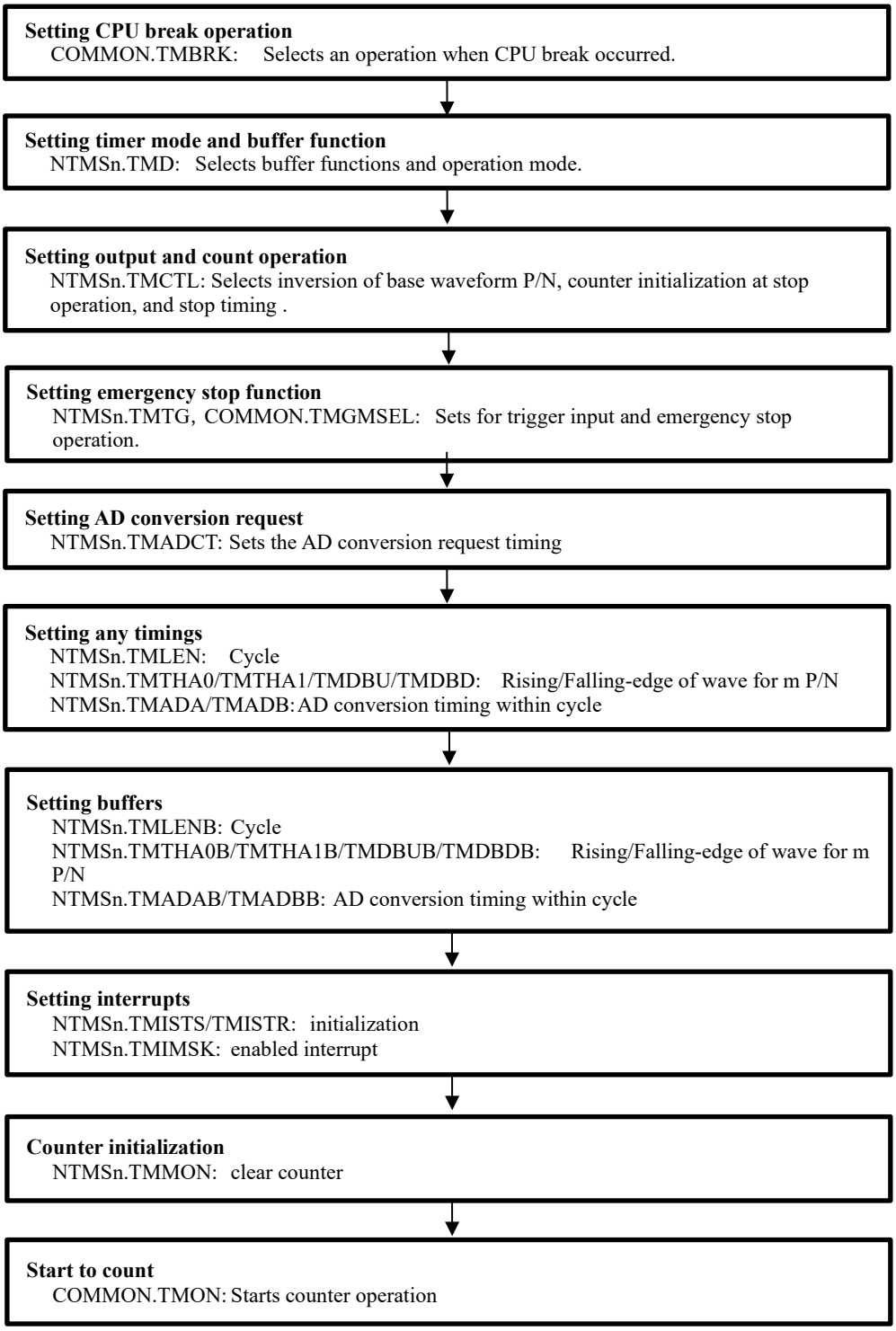


Figure 15-6 Setting Flow

15.3.1.2 Control Flow

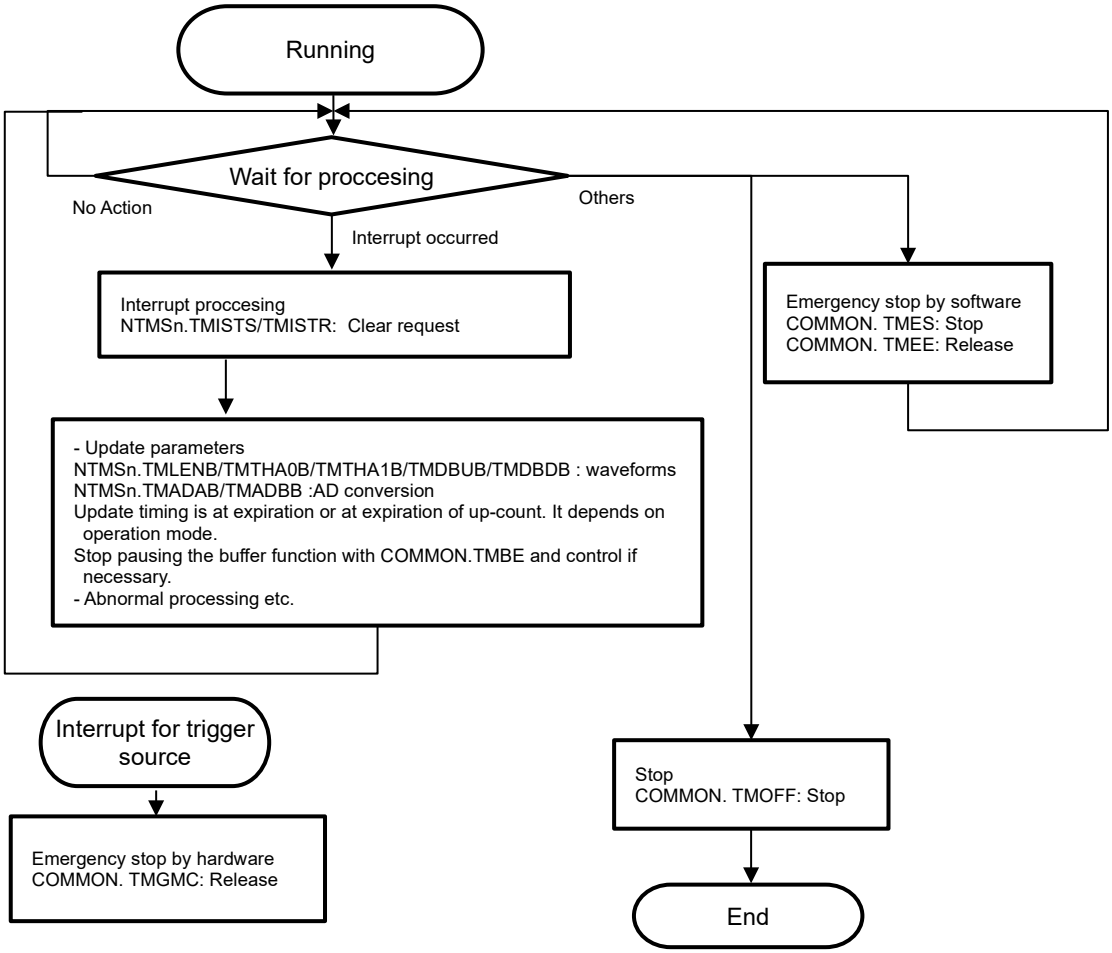


Figure 15-7 Proccesing Flow Overview

15.3.1.3 Stop and Resume

After stopping by TMOFF, the counter value changes depending on the setting of CSCS in the TMCTL register.
[When CSCS = 0]
The counter value is retained after the timer is stopped.
If the timer is started again, the count will continue from the retained value.
Write 0x0 to the TMMON register during the counter stopping to initialize the counter, if it is necessary.

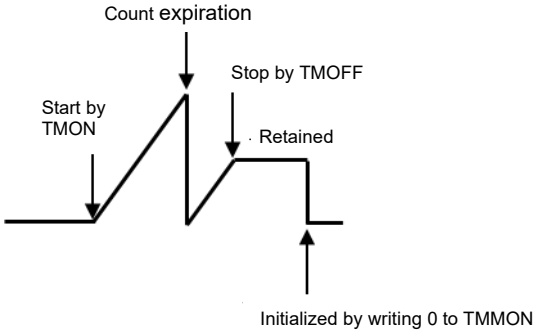


Figure 15-8 Stop Operation (CSCS=0)

[When CSCS = 1]
After the timer is stopped, the counter is initialized in the next clock.
Then the timer starts counting from the initialized state, if the timer starts again.

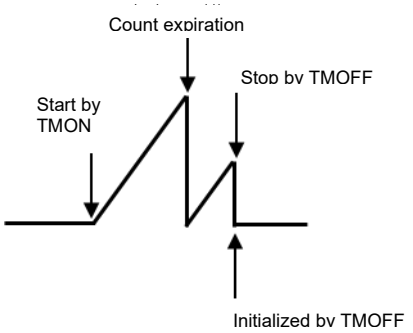


Figure 15-9 Stop Operation (CSCS=1)

15.3.2 Operation Mode

There are two modes of operation: sawtooth wave and triangle wave.
In the sawtooth wave mode, an up-count is repeated.
In the triangle wave mode, an up-count and down-count is repeated. This mode has 2 types.
The triangle wave mode 3 performs a buffer transfer after down-count (every 2 cycles).
The triangle wave mode 4 performs a buffer transfer after up-count and down-count (every 1 cycles).

The PWM waveforms are output by compare matching the TMTHA0, TMTHA1, TMTHB0 and TMTHB1 registers.
TMTHB0 and TMTHB1 store the results of dead-time calculations by setting the TMDBU and TMDBD registers.

See Section 15.3.3 for buffer operation and Section 15.3.4 for dead-time operations and restrictions.

15.3.2.1 Sawtooth wave mode

The sawtooth mode is a function that outputs PWM waveforms with edge alignment.
The following is the corresponding parameters and an example of waveform output (one channel).

Table15-8 Waveform parameters

Waveform	Parameter register	Restrictions
Cycle	TMLEN	-
Base waveform P Rising-edge	TMTHA0	0 ≤ TMTHA0 < TMTHA1
Base waveform P Falling-edge	TMTHA1	TMTHA0 < TMTHA1 ≤ TMLEN
Base waveform N Rising-edge	TMTHB0 = TMTHA0 – TMDBU	TMDBU < TMTHA1 TMDBU + TMDBD < TMLEN
Base waveform N Falling-edge	TMTHB1 = TMTHA1 + TMDBD	TMTHA0 < (TMLEN – TMDBD) TMDBU+TMDBD < TMLEN

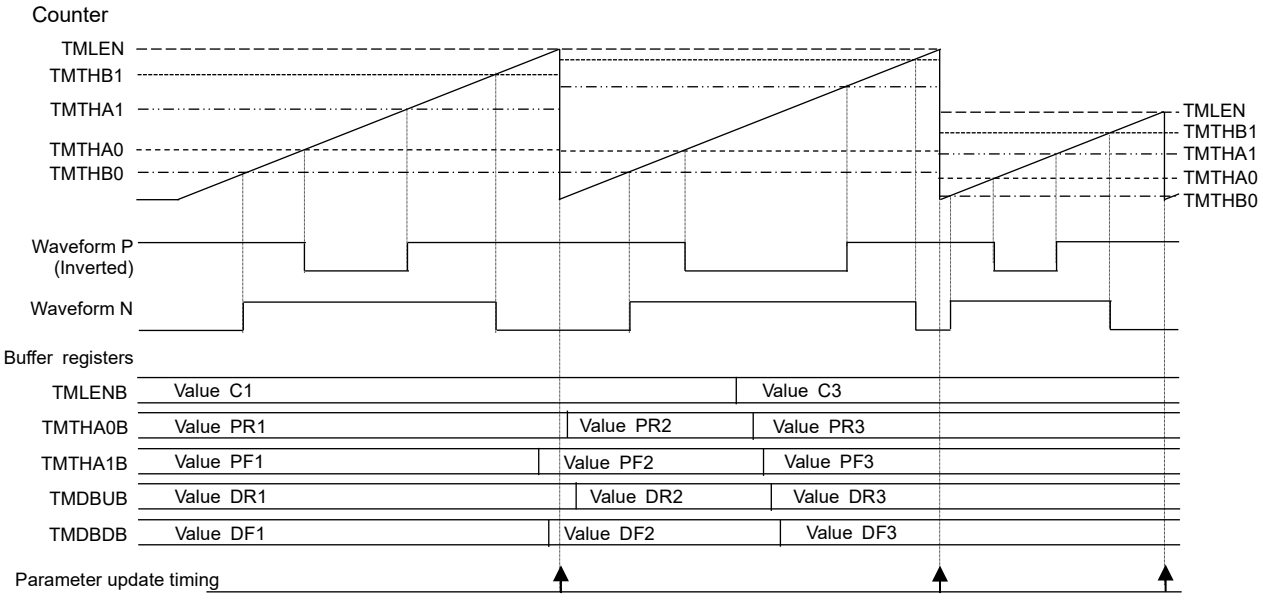


Figure 15-10 Example for Waveform output operation (When TMCTL UPNS= “01”)

Figure 15-11 shows example for three-phase waveform output in the sawtooth wave mode. The waveform P is set to “inverted” and the waveform N is set to “not inverted”. Therfore the waveform P output becomes “H” after the counter matched with the HTMTHA0 register then become “L” after the counter matched with the HTMTHA1 register. The waveform N output becomes “L” after the counter matched with the HTMTHB0 register then become “H” after the counter matched with the HTMTHB1 register.

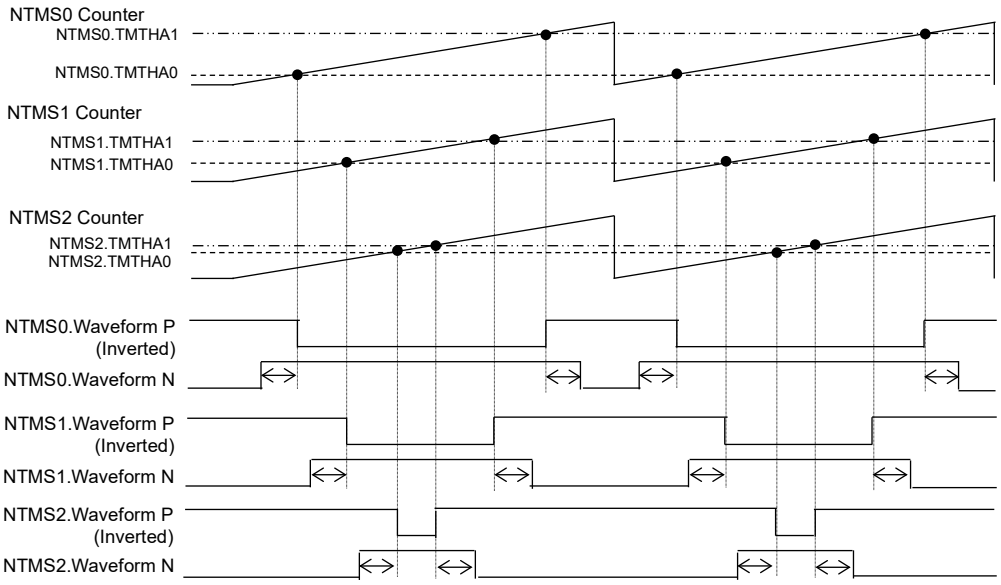


Figure 15-11 Three-phase wavefrom output in the sawtooth wave mode

15.3.2.2 Triangle wave mode

The triangle wave mode is a function that outputs PWM waveforms with center alignment.
The following is the corresponding parameters and an example of waveform output (one channel).

Table15-9 Waveform parameters

Waveform	Parameter register	Restrictions
Cycle	TMLEN × 2	-
Base waveform P Rising-edge / Falling-edge	TMTHA0	0 < TMTHA0 < TMLEN
Base waveform N Rising-edge / Falling-edge	TMTHB0 = TMTHA0 - TMDBU	TMDBU < TMLEN-1

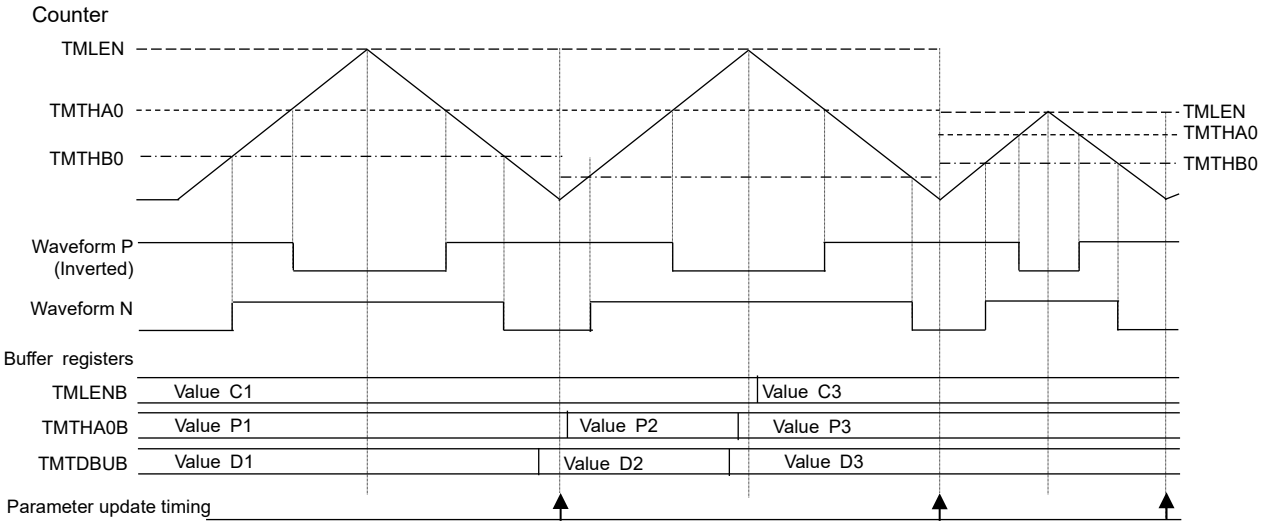


Figure 15-12 Example for Waveform output operation in the triangle wave mode 3 (When TMCTL UPNS= “01”)

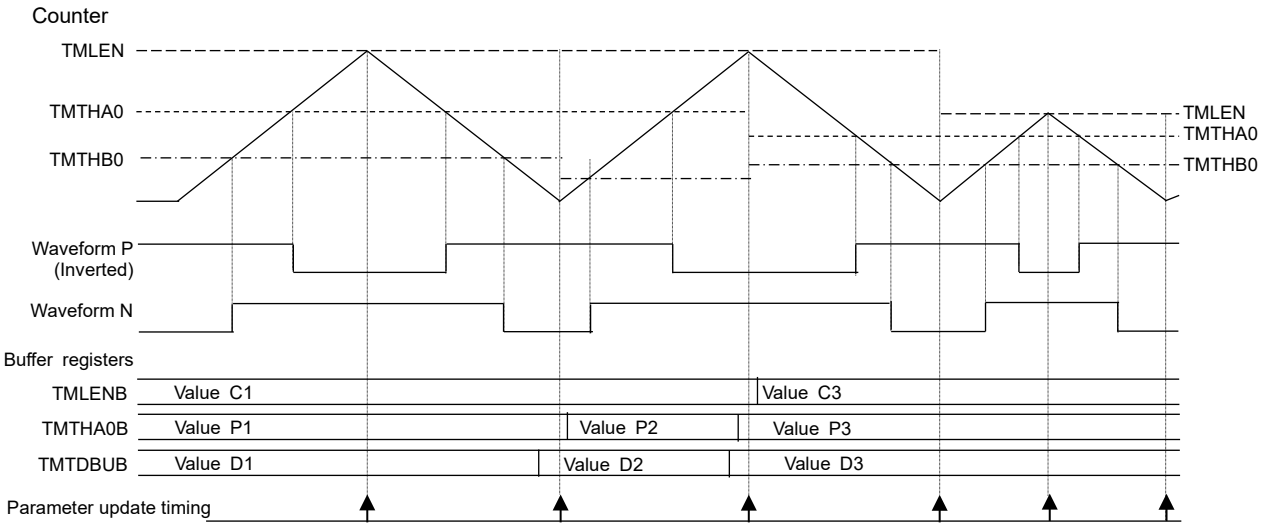


Figure 15-13 Example for Waveform output operation in the triangle wave mode 4 (When TMCTL UPNS= “01”)

Figure 15-14 shows example for three-phase waveform output in the triangle wave mode 3.

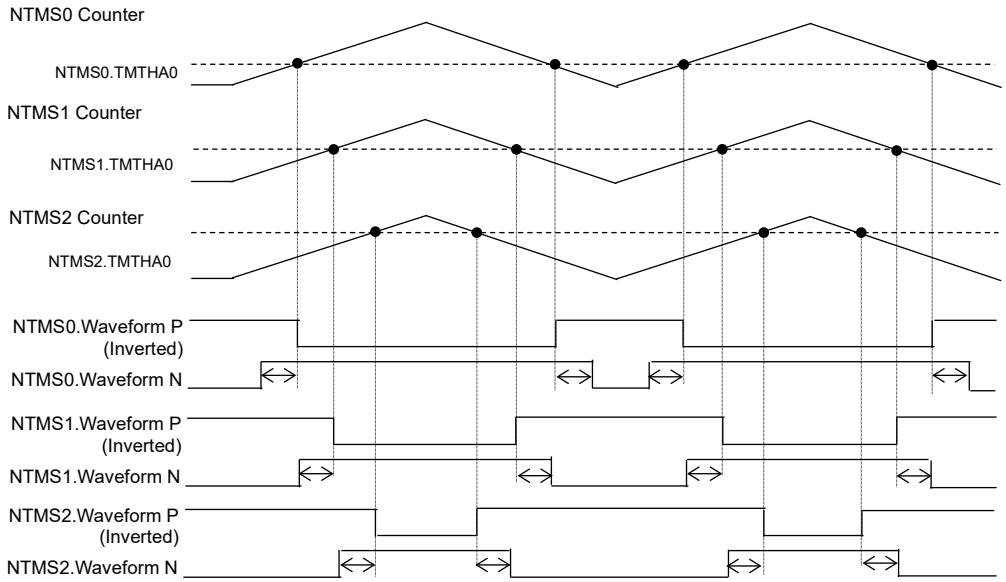


Figure 15-14 Three-phase waveform output in the triangle wave mode 3

15.3.3 Parameter Update Operation

During timer operation, the parameter registers that buffer function is enabled in the TMD register are periodically updated to the values written in the corresponding buffer register. Table 15-10 shows updatable parameters and update timings.

Table 15-10 List of updatable parameters

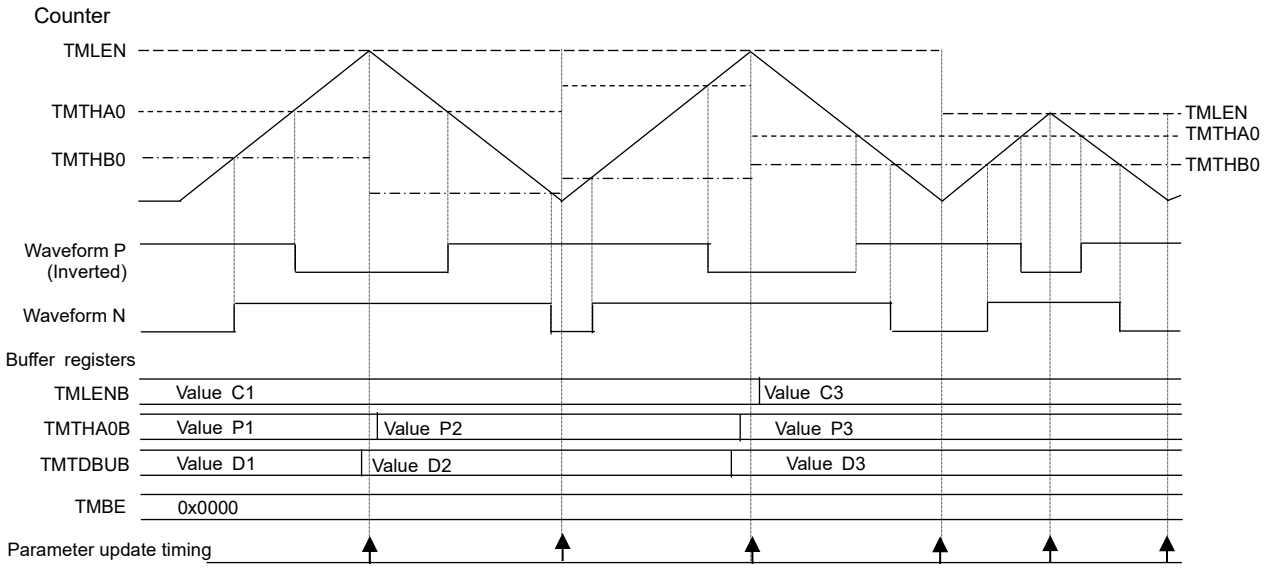
Function	Parameter register	Single buffer register	Update timing		
			Sawtooth wave mode	Triangle wave mode 3	Triangle wave mode 4
Cycle	TMLEN	TMLENB	At expiration	At expiration of down-count	At expiration of down-count
Base Waveform	TMTHA0	TMTHA0B	At expiration	At expiration of down-count	At both expiration of up/down-count
	TMTHA1	TMTHA1B			
(Dead-time)	TMDBU	TMDBUB			
	TMDBD	TMDBDB			
AD conversion start request	TMADA	TMADAB	At expiration	Select with BADA of TMD register At expiration up-count or down-count or up/down-count	
	TMADB	TMADBB		Select with BADB of TMD register At expiration up-count or down-count or up/down-count	

[Note]
- **Changing parameters other than the above during timer operation is prohibited.**
- **During timer operation, the access interval to the same address must be at least 6 clocks in system clock + 7 clocks in timer clock**

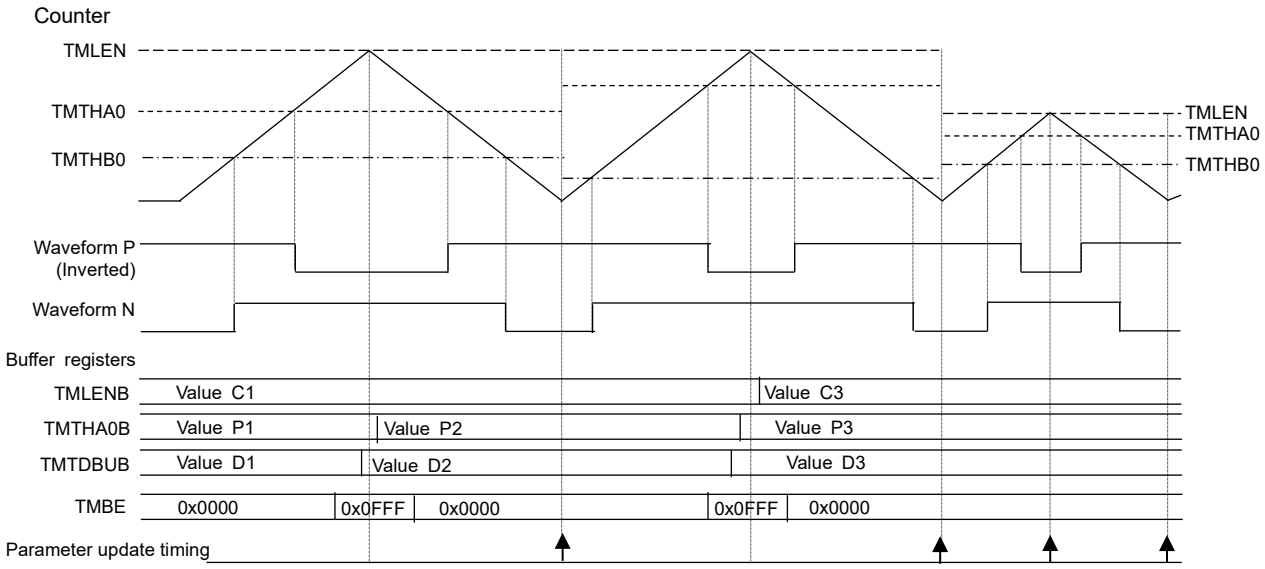
15.3.3.1 Pausing the Buffer function

There is function in case the writing to the buffer registers is not completed in time for the transfer timing.
The buffer operation can be paused by setting the corresponding bit of the TMBE register. Disable the buffer operation before the writing the buffer registers, and Enable the buffer operation after the writing the buffer registers.

Figure 15-15 shows example of using TMBE. When not using TMBE, there are unintentional waveforms such as only TMTDBU (value D2) beging updated at the expiration of the first down-count, and TMLEN not being updated at the expiration of the second up-count. Using TMBE is resolved this issue.



(a) Not using TMBE



(b) Using TMBE
Figure 15-15 Example of pausing the buffer operation

15.3.4 Dead-time Calcuration Safety Function

If an abnormality is detected in the dead-time calculation result, PWM DUTY error detection interrupt occurs and corrects calculation result.
The below shows conditions and what to do.
The interrupt source status ISTS_DTB0 becomes to “1”, when TMTHB0 calculation error occurs.
The interrupt source status ISTS_DTB1 becomes to “1”, when TMTHB1 calculation error occurs.

Table 15-11 Dead-time calculation error proccesing

Mode	TMTHA0, TMTHB0	TMTHA1, TMTHB1
Sawtooth wave mode	When TMTHA0 < TMDBU: TMTHB0 = 0 TMTHA0 = TMDBU	When TMLEN < (TMTHA1 + TMDBD): TMTHB1 = TMLEN TMTHA1 = (TMLEN – TMDBD)
Triangle wave mode	When TMTHA0 < TMDBU: TMTHB0 = 0 TMTHA0 =TMDBU	When TMTHA1 < TMDBD: TMTHB1 = 0 TMTHA1 = TMDBD

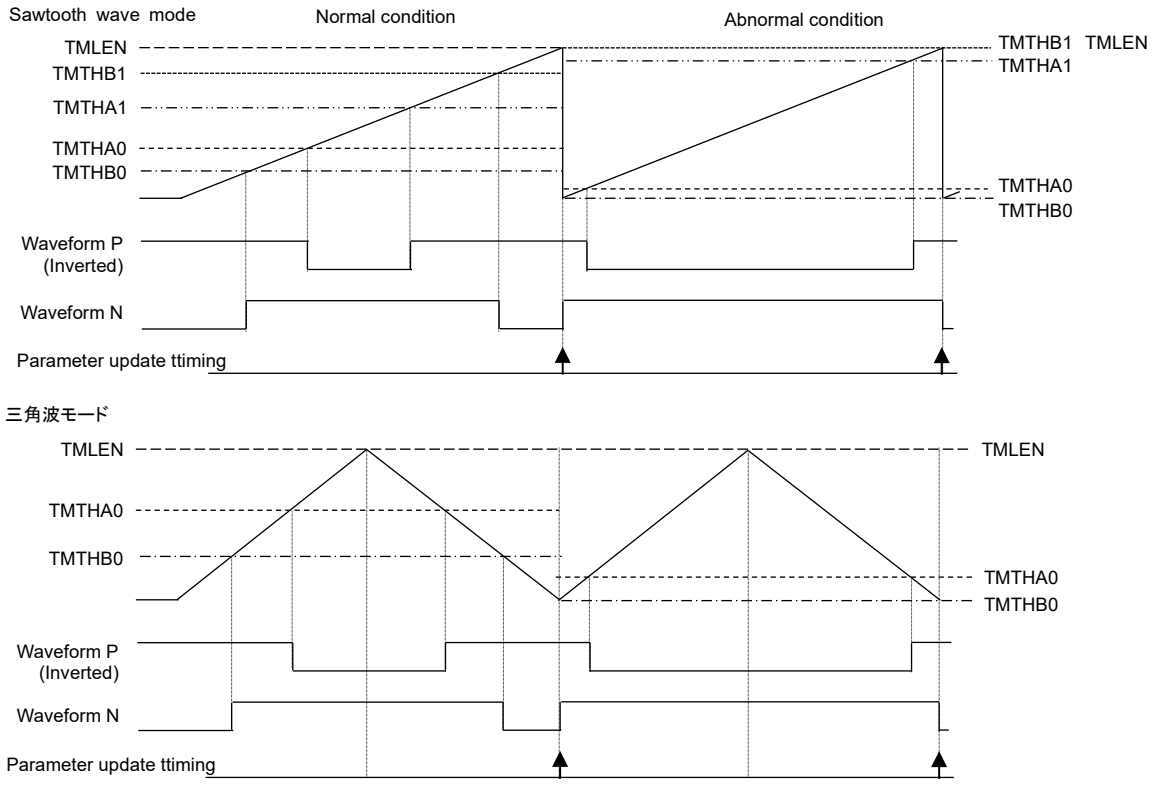


Figure 15-16 Dead-time calculation error proccesing

15.3.5 AD Conversion Request

The NTMS timer can generate an AD conversion request by compare matching the counter with the TMADA/TMADB registers. An occurrence timing is selectable from up-count, down-count, both up/down-count.
Select one of the following request to trigger on the ADC side. There are two ADCs, each of which can individually select trigger source.

- NTMS0_AD conversion start request A
- NTMS1_AD conversion start request A
- NTMS2_AD conversion start request A
- NTMS0_AD conversion start request B
- NTMS1_AD conversion start request B
- NTMS2_AD conversion start request B

It takes 1 cycle in the timer clock and 3 cycles in the system clock from the counter matching to the request output for ADC. In addition, it takes 3 cycles in the system clock and 3 cycles in the SAD clock to start AD conversion.

The TMADA and TMADB registers have buffer registers, and the buffer operation is selectable.
Table 15-12 shows the buffer register correspondence.

Table 15-12 AD conversion buffer register

AD conversion request register	Buffer register
TMADCT	-
TMADA	TMADAB
TMADB	TMADBB

The below shows example of the operation.
Where is, the AD conversion request A is set to 'the buffer transferred at both expiration of up/down-count', the AD conversion request B is set to 'the buffer transferred at both expiration of up-count'.

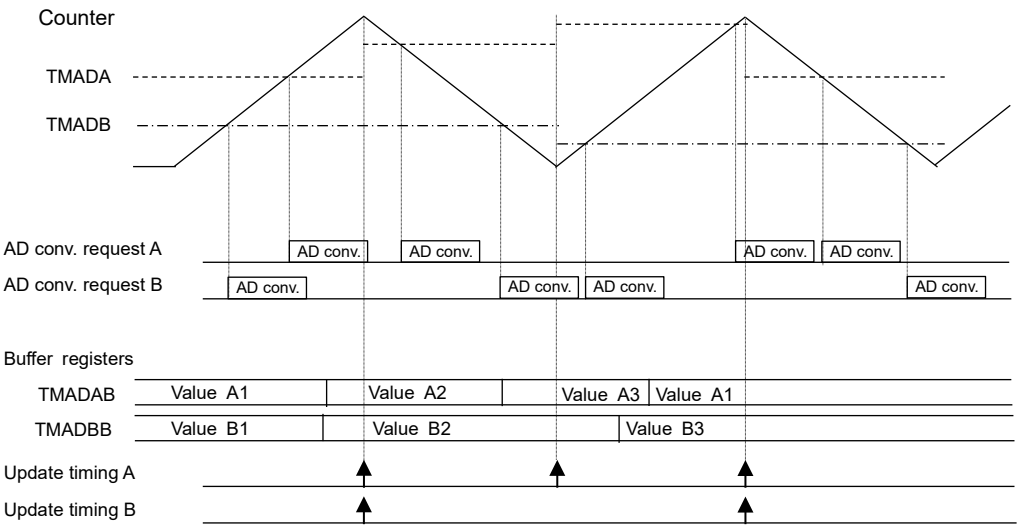


Figure 15-17 Example of the AD conversion request operation

15.3.5.1 Setting to Reduce the Number of AD Conversion

If the AD conversion is not completed in time, the number of the conversions can be reduced by the TMADC register setting.

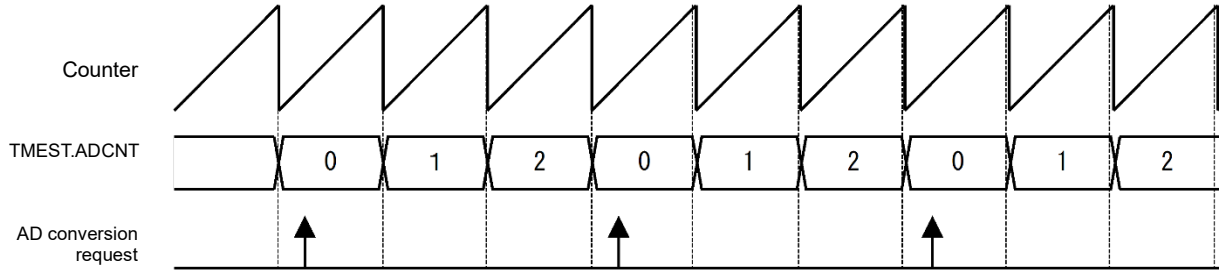


Figure 15-18 Number of reductions: 2 times in the sawtooth wave mode

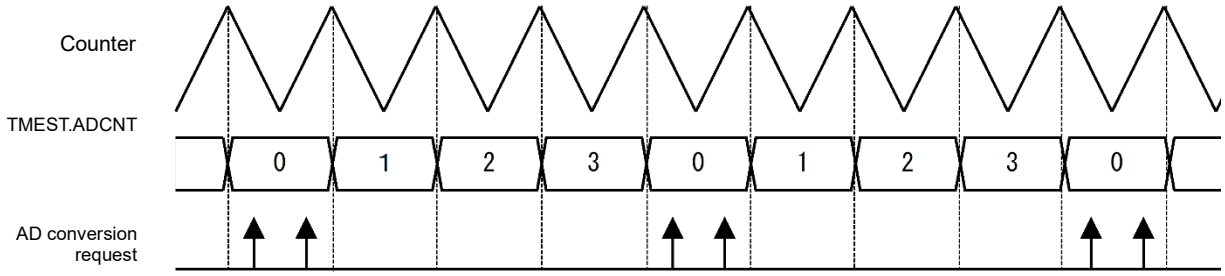


Figure 15-19 Number of up-count reductions: 3 times in the triangle wave mode

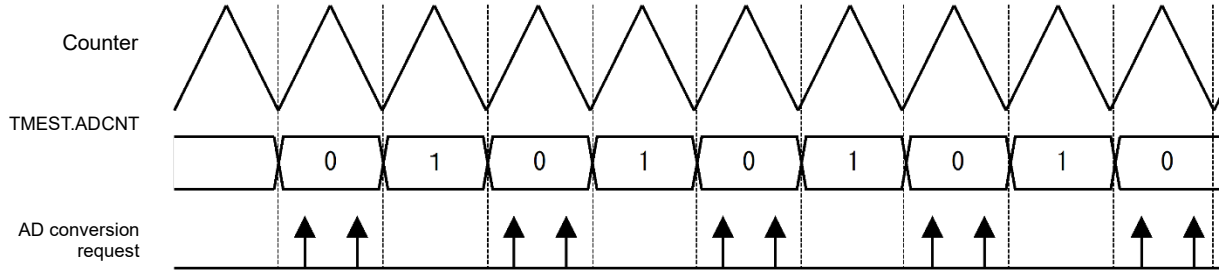


Figure 15-20 Number of down-count reductions: 1 time in the triangle wave mode

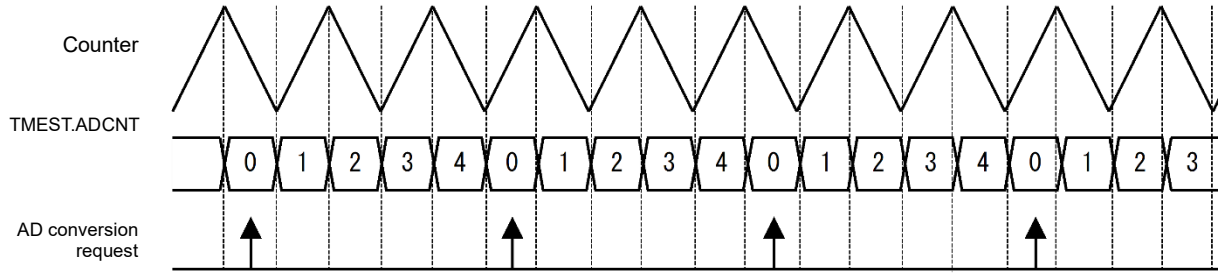


Figure 15-21 Number of up/down-count reductions: 4 times in the triangle wave mode

15.3.6 Emergency Stop Function

The emergency stop is a function that stops the PWM output by an event trigger input or by the TMES register.
When the emergency stop is requested, the PWM output P is fixed to the value set by the UEPS of the TMGM register, and the PWM output N is fixed to the value set by the UENS of the TMGM register.
The counter continues to operate during an emergency stop request.
When the emergency stop request is released, the output fixation is released according to the UERT of TMGM register.
The below interrupt source does not occur during an emergency stop request.
ISTS_TB1, ISTS_TB0, ISTS_TA1, ISTS_TA0, ISTS_MOU

If the timer is shifted to an emergency stop state while the counter is stopped, the following behavior occurs.

Table 15-13 Emergency stop request in stop state

Setting	Counter stop state	At counter starts
UEST=0 (Start immediately)	Emergency stop is OFF	Emergency stop turns on at the same time as start.
UEST=1 (Start at expiration)	Emergency stop is OFF	The first cycle operates normally. Emergency stop turns on from the second cycle in the case of a repeat setting.

When the timer is stopped in counter running state with the emergency stop, the emergency stop turns off as same time.

15.3.6.1 Emergency Stop by the Event trigger

The below show examples of an emergency stop operation by the event trigger with the following setting:
1）Stop immediately
TMGM.UEDT = “00” (Dead-time: disabled)
TMGM.UEPS = “00” (While emergency stop, PWM output P = fixed low)
TMGM.UEST / UERT = “0” / “0” (Start immediately / Recovery immediately)
TMGM.UEEN = “1” (Emergency stop function: enabled)

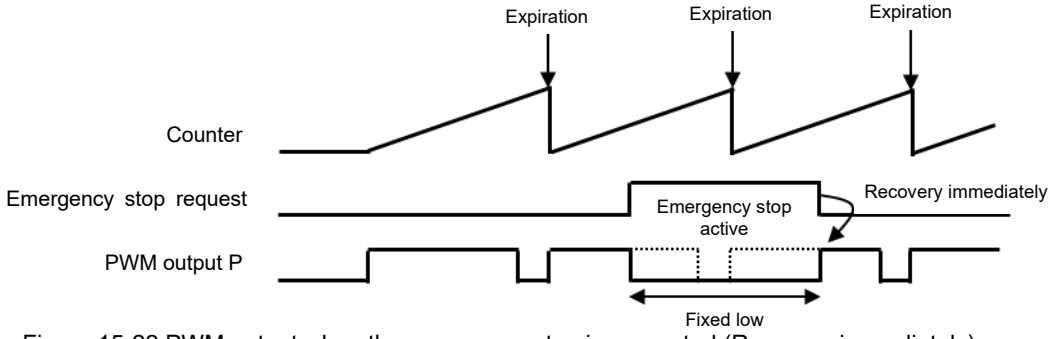


Figure 15-22 PWM output when the emergency stop is requested (Recovery immediately)

2）Recovery at expiration
TMGM.UEDT = “00” (Dead-time: disabled)
TMGM.UEPS / UENS = “0” / “0” (While emergency stop, PWM output P = fixed low / PWM output N = fixed low)
TMGM.UEST / UERT = “0” / “1” (Start immediately / Recovery at timer expiration)
TMGM.UEEN = “1” (Emergency stop function: enabled)

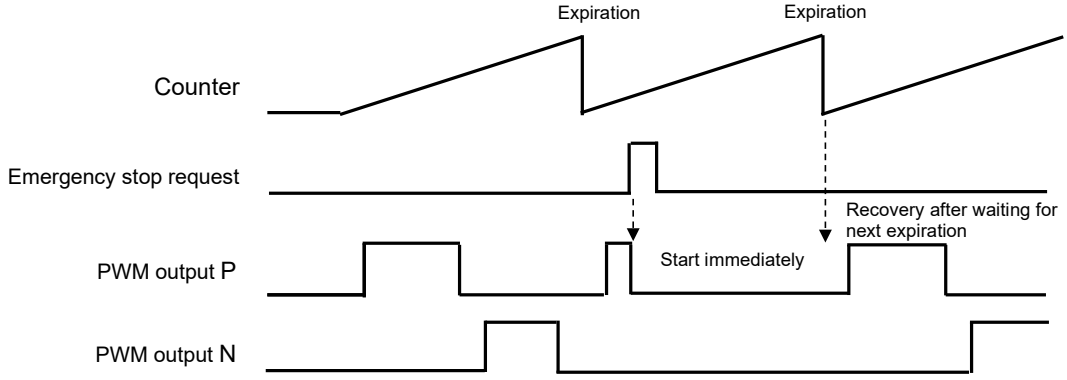


Figure 15-23 PWM output when the emergency stop is requested (Recovery at expiration)

3）With dead-time
TMGM.UEDT = “11” (Dead-time: enabled at emergency stop start, enabled at emergency stop end)
TMGM.UEDV = “00” (While dead-time, PWM output P = fixed low / PWM output N = fixed low)
TMGM.UEPS / UENS = “01” / “00” (While emergency stop, PWM output P = fixed high / PWM output N = fixed low)
TMGM.UEST / UERT = “0” / “0” (Start immediately / Recovery immediately)
TMGM.UEEN = “1” (Emergency stop function: enabled)

In the case of this setting, the following operations are performed.

1. While dead-time after the emergency stop start, the PWM output P and N are the “L” level.
2. After the end of the dead-time, the PWM output P is “H” level and PWM output N is the “L” level.
3. While dead-time after the emergency stop end, the PWM output P and N are the “L” level.
4. After the end of the dead-time, the timer recovers to normal operation.

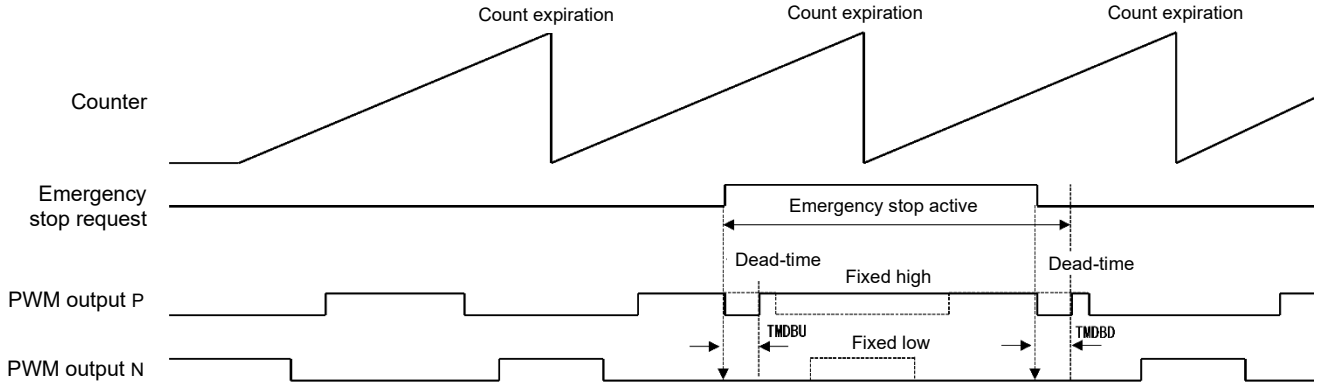


Figure 15-24 PWM output when the emergency stop is requested (the dead-time for start/end enabled)

15.3.6.2 Emergency Stop by TMES register

The below show examples of an emergency stop operation by software with the following setting:

1) Start immediately / Recovery immediately
   TMGM.UEDT = "00" (Dead-time: disabled)
   TMGM.UEPS / UENS = "00" / "01" (While emergency stop, PWM output P = fixed low / PWM output N = fixed high)
   TMGM.UEST / UERT = "0" / "0" (Start immediately / Recovery immediately)
2) Start immediately / Recovery at timer expiration
   TMGM.UEDT = "00" (Dead-time: disabled)
   TMGM.UEPS / UENS = "00" / "01" (While emergency stop, PWM output P = fixed low / PWM output N = fixed high)
   TMGM.UEST / UERT = "0" / "1" (Start immediately / Recovery at timer expiration)
3) Start at timer expiration / Recovery at timer expiration
   TMGM.UEDT = "00" (Dead-time: disabled)
   TMGM.UEPS / UENS = "00" / "01" (While emergency stop, PWM output P = fixed low / PWM output N = fixed high)
   TMGM.UEST / UERT = "1" / "1" (Start at timer expiration / Recovery at timer expiration)
4) Start immediately / Recovery at timer expiration with dead-time for start/end
   TMGM.UEDT = "11" (Dead-time: enabled at emergency stop start, enabled at emergency stop end)
   TMGM.UEDV= "00" (While dead-time, PWM output P = fixed low / PWM output N = fixed low)
   TMGM.UEPS / UENS = "01" / "00" (While emergency stop, PWM output P = fixed high / PWM output N = fixed low)
   TMGM.UEST / UERT = "0" / "1" (Start immediately / Recovery at timer expiration)

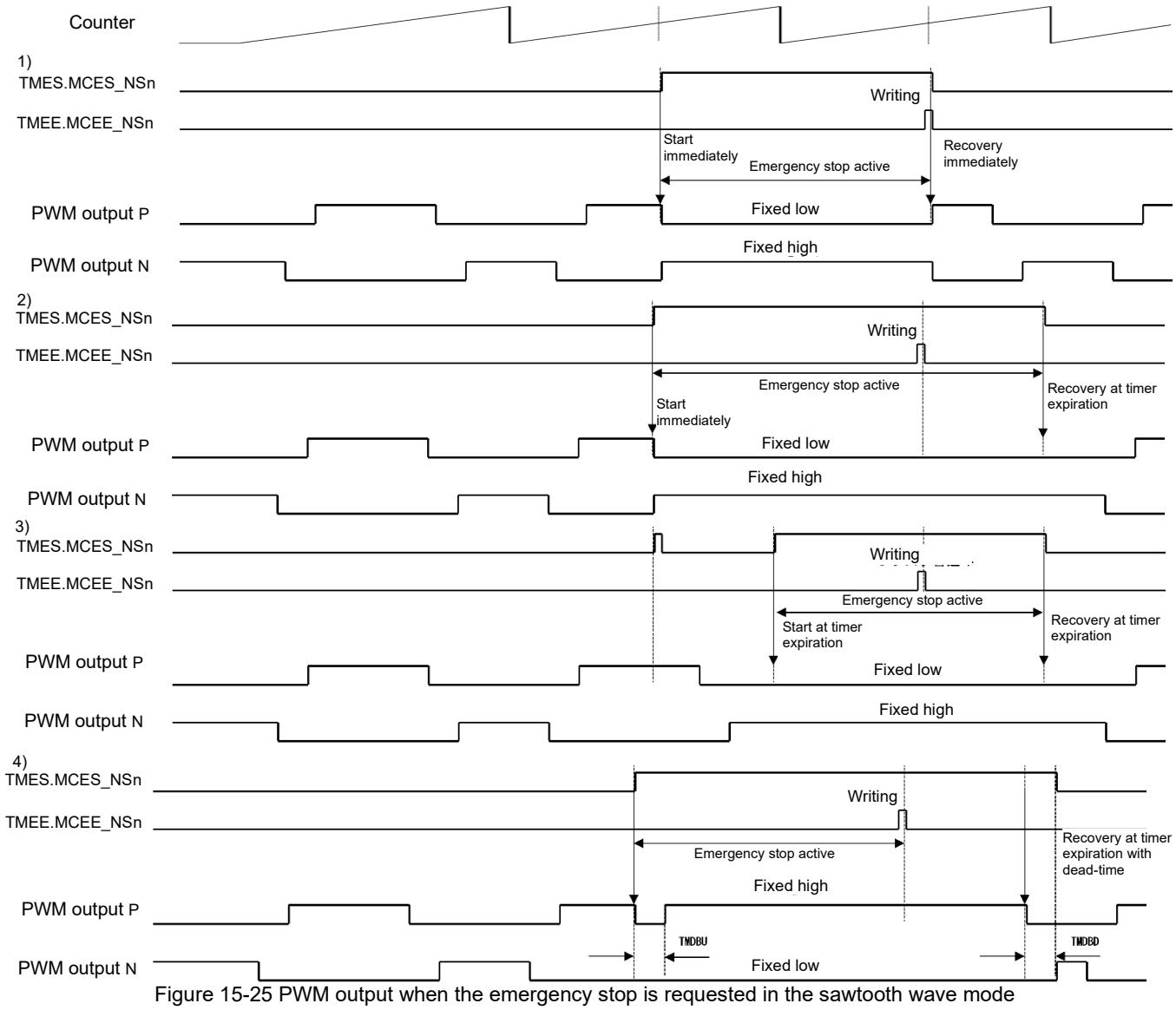


Figure 15-25 PWM output when the emergency stop is requested in the sawtooth wave mode

5) High-impedance output
TMGM.UEDT = “00” (Dead-time: disabled)
TMGM.UEPS / UENS = “01” / “00” (While emergency stop, PWM output P = fixed Hi-Z / PWM output N = fixed Hi-Z)
TMGM.UEST / UERT = “0” / “0” (Start immediately / Recovery immediately)

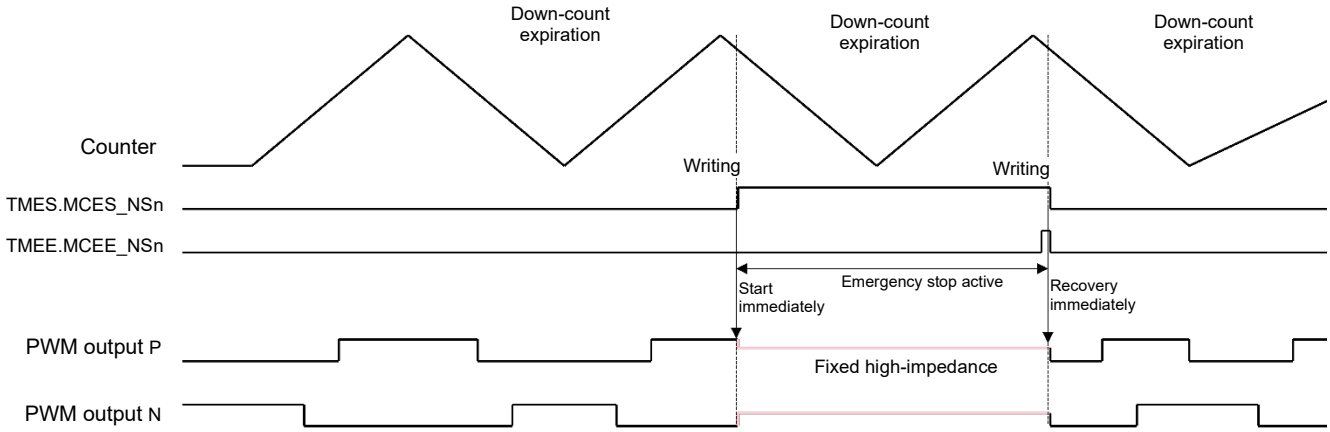


Figure 15-26 PWM output when the emergency stop is requested in the triangle wave mode

15.3.7 Interrupt

Table 15-14 shows interrupt sources. Three timers of NTMS0 to 2 have one common interrupt output.

Table 15-14 List of Interrupt Sources

Description
Up-count expiration interrupt in the triangle wave mode
Expiration interrupt in the sawtooth wave mode (Periodic) / Down-count expiration interrupt in the triangle wave mode
Compare-match interrupt (THA0/A1/B0/B1 value matched interrupt)
PWM DUTY error detection Interrupt (See “15.3.4 Dead-time Calcuration Safety Function“)

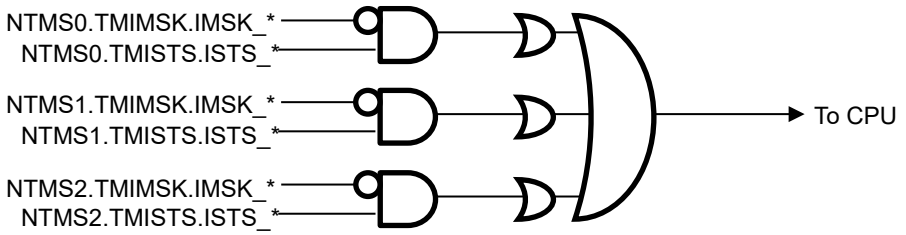


Figure 15-27 Configuration Overview of Interrupt output

15.3.7.1 Expiration Interrupt

1）Expiration interrupt (sawtooth wave mode): Expiration of cycle is detected and the interrupt is occurred.

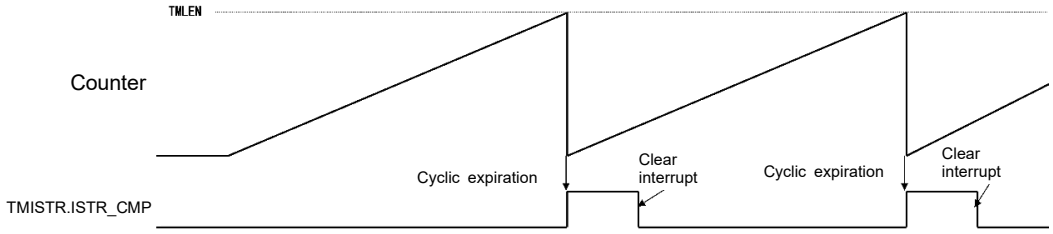


Figure 15-28 Example for expiration interrupt (sawtooth wave mode)

2）Down-count expiration interrupt (triangle wave mode): It is detected that count is 0, and the interrupt is occurred.

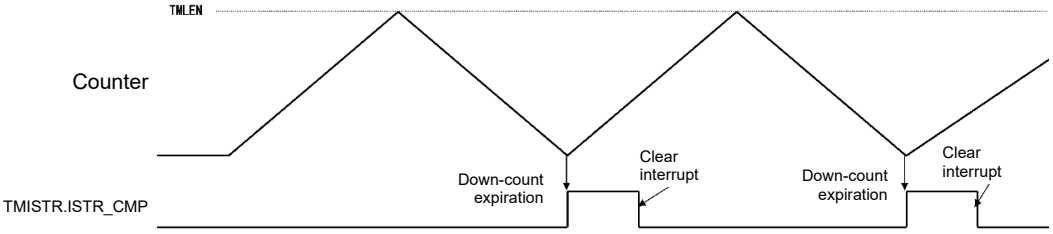


Figure 15-29 Example for down-count expiration interrupt (triangle wave mode)

3）Up-count expiration interrupt (triangle wave mode): It is detected that count is TMLEN value, and the interrupt is occurred.

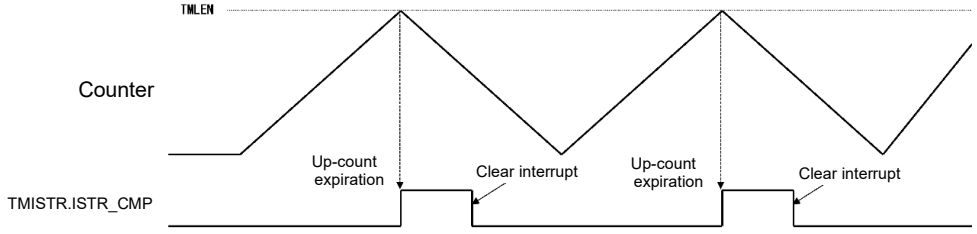


Figure 15-30 Example for up-count expiration interrupt (triangle wave mode)

15.3.7.2 Compare-match Interrupt

The interrupt is occurred at Rising/Falling of waveforms.

Table 15-15 Compare-match interrupt Generation Timing

Mode	Condition	Threshold register	Interrupt Generation Timing
Sawtooth wave mode	-	TH*(THA0,THB0,THA1,THB1)	At changing Counter value 'TH* – 1' to 'TH*'
Triangle wave mode	Up-count	TH*(THA0,THB0)	At changing Counter value 'TH* – 1' to 'TH*'
	Down-count	TH*(THA0,THB0)	At changing Counter value 'TH* + 1' to 'TH*'

Chapter 16

Synchronous Serial Port with FIFO (SSIOF)

16. Synchronous Serial Port with FIFO (SSIOF)

16.1 Overview

This LSI includes two channels of the 8 or 16 bit synchronous serial port (SSIO) with FIFO and can also be used to control the device incorporated with the SPI interface.
The use of SSIOF requires the function setting of the general purpose port. For the port function setting, see section 16.3.19 and Chapter 22 "General Purpose Port".

16.1.1 Features

- Supported the Full-duplex data transfer, Master / Slave mode
- For the transfer size, 8 bit (byte) or 16 bit (half word) selectable
- Built-in 4-stage FIFO on each of transmit- and receive-sides
- Interrupt by the number of data in received FIFO
- Interrupt by the number of untransmitted data in transmit FIFO.
- Either LSB first or MSB first can be selected
- The polarity and phase of the serial clock are selectable
- In Master mode, the HSCLK 2 to 2046-division clocks can be selected as the sync clock (1023 kinds).
- In Master mode, the interval before/after transfer can be controlled
- Detects a mode fault error to avoid multi-master bus contention
- Detects a write overflow error if any further writing is attempted when the transmit FIFO is in the full state
- Generates an interrupt when the transmit/receive FIFO is in a specific state or when a cause such as mode fault error occurs
- Generates transmission/receive DMA request

16.1.2 Configuration

Figure 16-1 shows the configuration of the SSIOF.

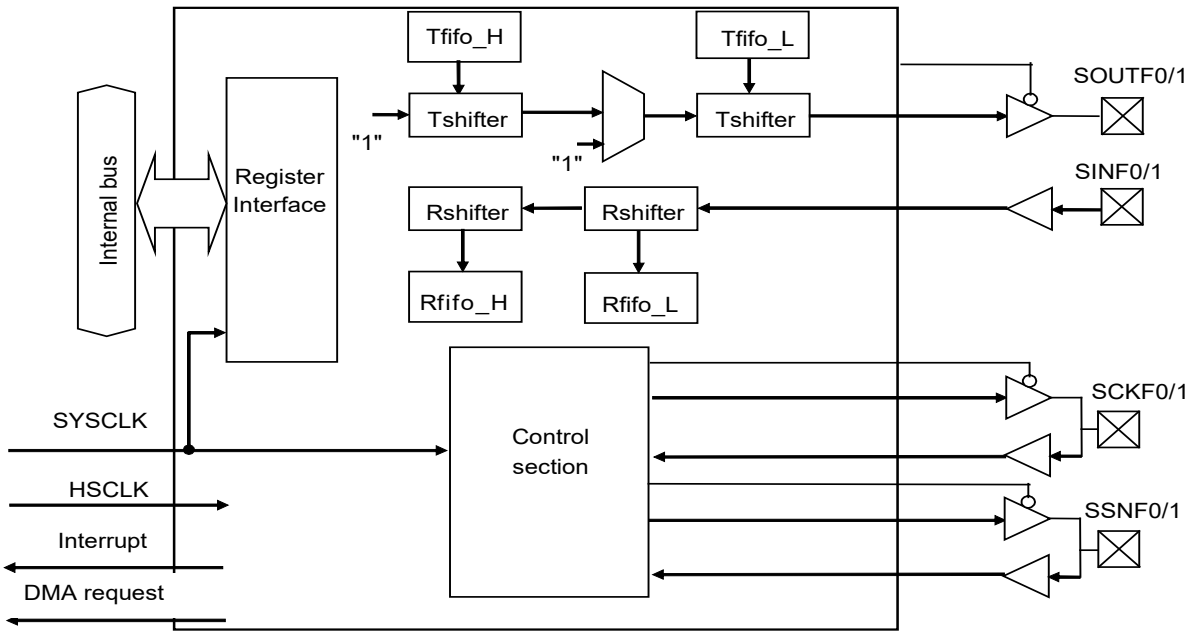


Figure 16-1 Configuration

16.1.3 List of Pins

Table 16-1 List of Pins

Pin name	I/O	Description
SOUTF0/1	O	Master serial output/slave serial output signal
SINF0/1	I	Master serial input/slave serial input signal
SCKF0/1	I/O	Baud rate clock
SSNF0/1	I/O	Slave selection signal

16.2 Description of Registers

16.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4100_0800	SSIOF channel 0 base address	SSIOF0	-	-	-
0x00	SIOF0 control register	SF0CTRL	R/W	32	0x0000_0000
0x04	SIOF0 interrupt control register	SF0INTC	R/W	32	0x0000_0000
0x08	SIOF0 transfer interval control register	SF0TRAC	R/W	32	0x0000_0002
0x0C	SIOF0 baud rate register	SF0BRR	R/W	32	0x0000_5002
0x10	SIOF0 status register	SF0SRR	R	32	0x0000_1400
0x14	SIOF0 status clear register	SF0SRC	W	32	0x0000_0000
0x18	SIOF0 FIFO status register	SF0FSR	R	32	0x0000_0000
0x1C	SIOF0 write data register	SF0DWR	R/W	32	0x0000_0000
0x20	SIOF0 read data register	SF0DRR	R	32	0x0000_0000
0x4100_0900	SSIOF channel 1 base address	SSIOF1	-	-	-
0x00	SIOF1 control register	SF1CTRL	R/W	32	0x0000_0000
0x04	SIOF1 interrupt control register	SF1INTC	R/W	32	0x0000_0000
0x08	SIOF1 transfer interval control register	SF1TRAC	R/W	32	0x0000_0002
0x0C	SIOF1 baud rate register	SF1BRR	R/W	32	0x0000_5002
0x10	SIOF1 status register	SF1SRR	R	32	0x0000_1400
0x14	SIOF1 status clear register	SF1SRC	W	32	0x0000_0000
0x18	SIOF1 FIFO status register	SF1FSR	R	32	0x0000_0000
0x1C	SIOF1 write data register	SF1DWR	R/W	32	0x0000_0000
0x20	SIOF1 read data register	SF1DRR	R	32	0x0000_0000

16.2.2 SIOFn Control Register (SFnCTRL: n = 0,1)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SF0MOZ	SF0SOZ	SF0SSZ	SF0FICL	-	SF0CPOL	SF0CPHA	SF0LSB	SF0MDFE	SF0SIZ	SF0MST	SF0SPE
R/W	-	-	-	-	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SFnCTRL is a special function register (SFR) used to control the operation of the SSIOF.

Bit No	Bit name	Description
11	SF0MOZ	SF0MOZ is used to set the SOUTFn, SCKFn output control of the IDLE state in the master mode. 0: 0/1 output (Initial value) 1: Hi-Z. The SCKFn pin should be pull-up or pull-down to avoid hi-impedance input state in this case.
10	SF0SOZ	SF0SOZ is used to set the SOUTFn output control when SSNFn = “1” in the slave mode. 0: 0/1 output (Initial value) 1: Hi-Z
9	SF0SSZ	SF0SSZ is used to set the SSNFn output control of the IDLE state in the master mode. 0: 0/1 output (Initial value) 1: Hi-Z
8	SF0FICL	SF0FICL is used to clear the FIFOs. 0: No action (Initial value) 1: Clear the received/transmitted count. After clearance, set this bit to “0”.
6	SF0CPOL	SF0CPOL is used to set the serial clock polarity. 0: Serial clock default is "0" ("0" during transmission/reception) (Initial value) 1: Serial clock default is "1" ("1" during transmission/reception)
5	SF0CPHA	SF0CPHA is used to set serial clock phase. 0: The data is sampled at the first edge and shifted at the second edge (Initial value) 1: The data is shifted at the first edge and sampled at the second edge
4	SF0LSB	SF0LSB is used to set the data transfer order. 0: LSB first (Initial value) 1: MSB first
3	SF0MDFE	SF0MDFE is used to enable the mode fault control. The mode fault is executed when SF0MST=1, SF0MDFE=1 and not transferring. 0: Disabled (Initial value) 1: Enabled
2	SF0SIZ	SF0SIZ Is used to set a transfer size. 0: 1B = 8Bit (Initial value) 1: 1HW = 16Bit
1	SF0MST	SF0MST is used to set master/slave selection. 0: Slave (Initial value) 1: Master
0	SF0SPE	SF0SPE is used to enable the communication. 0: Disabled (Initial value) 1: Enabled

16.2.3 SIOFn Interrupt Control Register (SFnINTC: n = 0,1)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	SF0RFIC		-	-	SF0TFIC		-	-	-	SF0MFIE	SF0ORIE	SF0FIE	SF0RFIE	SF0TFIE
R/W	-	-	R/W	R/W	-	-	R/W	R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SFnINTC is a special function register (SFR) used to control the interrupt operation of the SSIOF.

Bit No	Bit name	Description
13 to 12	SF0RFIC	SF0RFIC is used to set the number of received data in the FIFO to occur a reception interrupt. This is also used to generate the DMA request signal for reception. 00: 1 data (Initial value) 01: 2 data 10: 3 data 11: 4 data
9 to 8	SF0TFIC	SF0TFIC is used to set the number of untransmitted data in the FIFO to occur a transmission interrupt. This is also used to generate the DMA request signal for transmission. 00: 0 data (Initial value) 01: 1 data 10: 2 data 11: 3 data
4	SF0MFIE	SF0MFIE is used to enable the mode fault interrupt. 0: Disabled (Initial value) 1: Enabled
3	SF0ORIE	SF0ORIE is used to enable the overrun error interrupt. 0: Disabled (Initial value) 1: Enabled
2	SF0FIE	SF0FIE is used to enable the transfer completion interrupt. 0: Disabled (Initial value) 1: Enabled
1	SF0RFIE	SF0RFIE is used to enable the reception interrupt. 0: Disabled (Initial value) 1: Enabled
0	SF0TFIE	SF0TFIE is used to enable the transmission interrupt. 0: Disabled (Initial value) 1: Enabled

16.2.4 SIOFn Transfer Interval Control Register (SFnTRAC: n = 0,1)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	SF0DTL								
R/W	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SFnTRAC is a special function register (SFR) used to set the minimum interval of data transfer in the master mode.
For details, see 16.3.7, "Transfer Interval Setting".

16.2.5 SIOFn Baud Rate Register (SFnBRR: n = 0,1)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SF0LAG		SF0LEAD		-	-	SF0BR									
R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0

SFnBRR is a special function register (SFR) used to set the operation mode.
Do not change the setting of this register during transfer. Operation is not guaranteed if it is changed during transfer.

Bit No	Bit name	Description
15 to 14	SF0LAG	SF0LAG is used to set the delay inverval from SCKFn to SSNFn(H) in the master mode. 00: 0.5 X SCKFn 01: 0.5 X SCKFn (Initial value) 10: 1.0 X SCKFn 11: 1.5 X SCKFn
13 to 12	SF0LEAD	SF0LEAD is used to set the delay interval from SSNFn to SCKFn in the master mode. 00: 0.5 X SCKFn 01: 0.5 X SCKFn (Initial value) 10: 1.0 X SCKFn 11: 1.5 X SCKFn
9 to 0	SF0BR	This is used to set the baud rate (f_{SCK}) in the master mdoe. $f_{SCK}=f_{HSCLK}/ (2 X SF0BR)$, f_{HSCLK} : HSCLK frequency where it is calculated as SF0BR =1 when setting value is “0”.

[Note]
- **The maximum transfer frequency is 12MHz. Do not exceed this.**
- **Set the SF0LEAD bits and SF0LAG bits in the SFnBRR register to each 0x3 when a baud rate of SSIOFn is 12MHz. Another setting value are not supported.**

16.2.6 SIOFn Status Register (SFnSRR: n = 0,1)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SF0RFE	SF0RFF	SF0TFE	SF0TFF	SF0WOF	-	-	SF0SPIF	SF0MDF	SF0ORF	SF0FI	SF0RFI	SF0TFI
R/W	-	-	-	R	R	R	R	R	-	-	R	R	R	R	R	R
Initial value	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

SFnSRR is a special function register (SFR) used to indicate the data transfer state and error state of the SSIOF.

[Bit の Description]
It indicates the corresponding state.
0: Has not occurred (Initial value)
1: Has occurred

Bit No	Bit name	Description (corresponding state)
12	SF0RFE	The receive FIFO empty It is not generated interrupt.
11	SF0RFF	The receive FIFO full It is not generated interrupt.
10	SF0TFE	The transmit FIFO empty It is not generated interrupt.
9	SF0TFF	The transmit FIFO full It is not generated interrupt.
8	SF0WOF	The write overflow It is not generated interrupt.
5	SF0SPIF	The one data transfer completion
4	SF0MDF	The mode fault It is generated interrupt.
3	SF0ORF	The overrun error It is generated interrupt.
2	SF0FI	The transfer completion interrupt. This timing is at the transmit FIFO is empty and the transfer of the last one byte (one word) is finished.
1	SF0RFI	The reception interrupt This timing is when the number of data received in the receive FIFO is equal or more data count selected with SF0RFIC.
0	SF0TFI	The transmission interrupt This timing is when the remaining data in the transmit FIFO is equal data count selected with SF0TFIC.

16.2.7 SIOFn Status Clear Register (SFnSRC: n = 0,1)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SF0WOFC	-	-	SF0SPIFC	SF0MDFC	SF0ORFC	SF0FC	SF0RFC	SF0TFC
R/W	-	-	-	-	-	-	-	W	-	-	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SFnSRC is a special function register (SFR) used to clear the data transfer state and error state of the SSIOF.

[Description of each bit]
This is used to clear the corresponding state.
Writing “0”: Invalid
Writing “1”: Clear the corresponding status bit.

Bit No	Bit name	Description (corresponding state)
8	SF0WOFC	Write overflow (SF0WOF)
5	SF0SPIFC	One data transfer completion (SF0SPIF)
4	SF0MDFC	The mode fault (SF0MDF) with interrupt request
3	SF0ORFC	The overrun error (SF0ORF) with interrupt request
2	SF0FC	The transfer completion interrupt (SF0FI)
1	SF0RFC	The reception interrupt (SF0RFI)
0	SF0TFC	The transmission interrupt (SF0TFI)

16.2.8 SIOFn FIFO Status Register (SFnFSR: n = 0,1)

Offset : 0x18

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SF0RFD			-	-	-	-	-	SF0TFD		
R/W	-	-	-	-	-	R	R	R	-	-	-	-	-	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SF0FSR is a special function register (SFR) used to indicate the count transmitted and received by FIFO.

Bit No	Bit name	Description (corresponding state)
10 to 8	SF0RFD	SF0RFD indicates the number of received data in the receive FIFO. 000: null data / empty (Initial value) 001: 1 data 010: 2 data 011: 3 data 100: 4 data / full
2 to 0	SF0TFD	SF0TFD indicates the number of untransmitted data in the transmit FIFO. 000: null data / empty (Initial value) 001: 1 data 010: 2 data 011: 3 data 100: 4 data / full

16.2.9 SIOFn Write Data Register (SFnDWR: n = 0,1)

Offset : 0x1C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SF0WD															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SF0DWR is a special function register (SFR) used to write transmitted data.

Write according to the transfer size.
8 bit (SF0SIZ=0) : SF0WD[7:0]
16 bit (SF0SIZ=1) : SF0WD[15:0]

16.2.10 SIOFn Read Data Register (SFnDRR: n = 0,1)

Offset : 0x20

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SF0RD															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SF0DRR is a special function register (SFR) used to read received data.

Read according to the transfer size.
8 bit (SF0SIZ=0) : SF0RD[7:0]
16 bit (SF0SIZ=1) : SF0RD[15:0]

16.3 Description of Operation

The system clock needs to be the high-speed clock when operating the SSIOF0/1.
In slave mode, the baud rate that can be communicated is limited to the frequency of the HSCLK divided two, or less.
The SSIOF0 is described below. The SSIOF1 is same except for the communication speed in the master mode as the SSIOF0.

16.3.1 Master Mode and Slave Mode

The master mode and slave mode are provided as the transmission/reception mode. This is selected by the SF0MST bit of the SIOF0 control register.
SF0BR (baud rate) bit, SF0LEAD (SSNF0-SCKF0 delay interval) bit, and SF0LAG (SCKF0-SSNF0 delay interval) bit of the SIOF0 baud rate register and SF0DTL (minimum data transfer interval) bit of the SIOF0 transfer interval control register determine SCKF0 and SSNF0 operations. And these bits are also only valid during the master operation.
Each bit of SF0CPOL, SF0CPHA, SF0LSB, and SF0SIZ in the SF0CTRL register needs to have the same value for master and slave.

16.3.2 Transfer Size

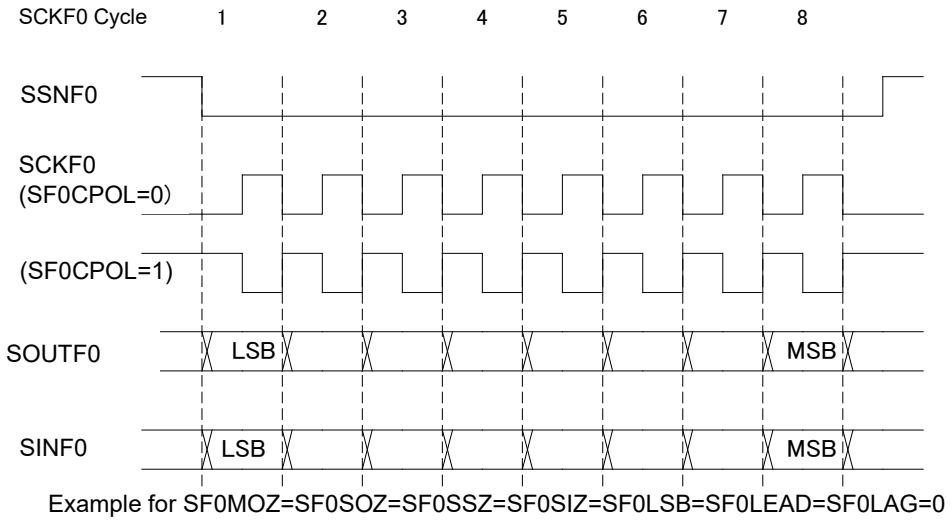
The transfer size can be selected in 8 bit (byte) or 16 bit (half word).
Transfer data read/write must be adjusted to the transfer size. As the number of FIFO stages is the same for both byte and half word, the number of transfers is the same.
The master and slaves which communicate with each other must have the same value for SF0SIZ.

16.3.3 Control of Polarity and Phase of Serial Clock

SF0CPOL bit of the SIOF0 control register controls the clock polarity. SF0CPHA bit of the SF0CTRL (SIOF0 control) register controls the clock phase and determines the shift timing of transmit data and the sampling timing of received data. The master and slave which communicate with each other must have the same setting values for SF0CPOL and SF0CPHA.

16.3.3.1 Data Transfer Timing When SF0CPHA Is "0"

Figure 16-2 shows the data transfer timing when SF0CPHA is "0". For the SCKF0, two cases are shown (SF0CPOL is "0" and "1"). SSNF0 is the slave selection input in Slave mode.
In Master mode, the transfer is started when data is written to the SF0DWR register. In Slave mode, the transfer is started at the SSNF0 falling edge. The received data is sampled at the rising-edge of SCKF0 in SF0CPOL is “0” and the falling-edge of SCKF0 in SF0CPOL is “1”. The transmitted data is shifted at the falling-edge of SCKF0 in SF0CPOL is “0” and the rising-edge of SCKF0 in SF0CPOL is “1”.



Example for SF0MOZ=SF0SOZ=SF0SSZ=SF0SIZ=SF0LSB=SF0LEAD=SF0LAG=0

Figure 16-2 Waveform When SF0CPHA = 0 (transfer size is 8 bit)

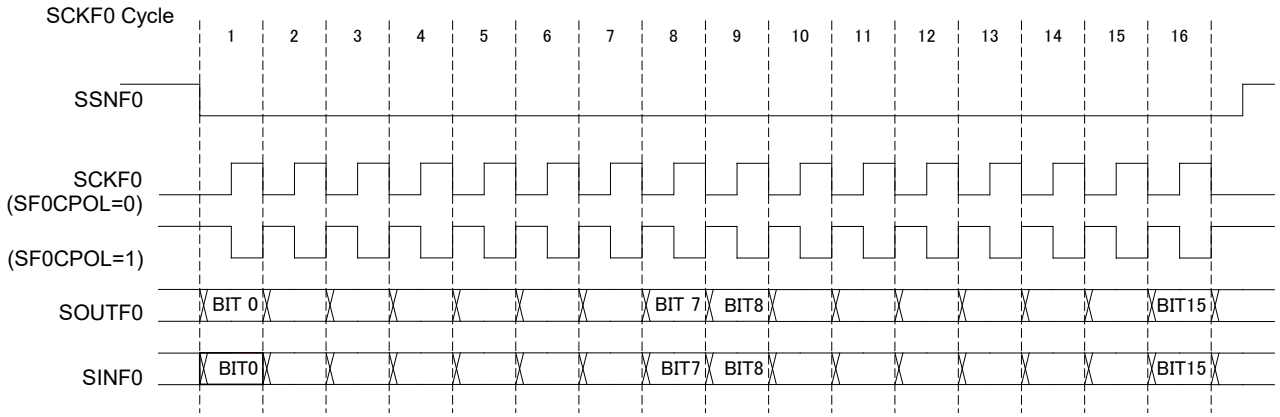
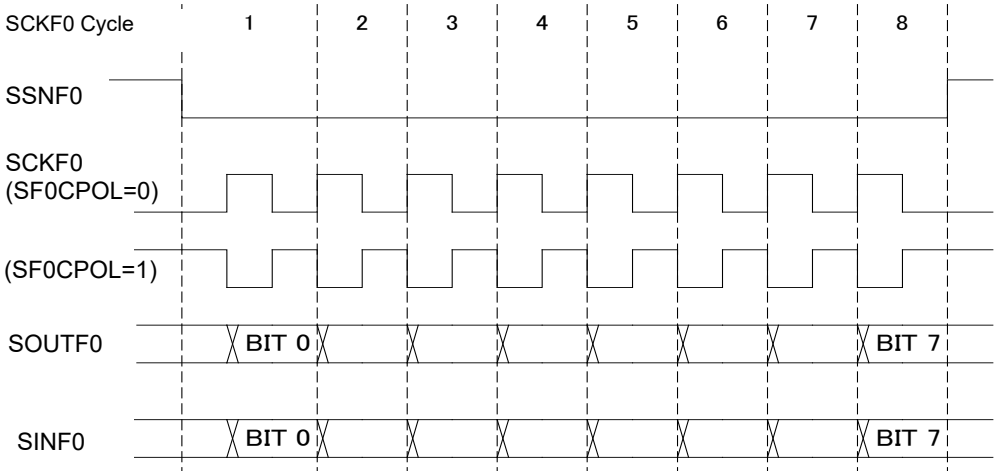


Figure 16-3 Waveform When SF0CPHA = 0 (transfer size is 16 bit)

16.3.3.2 Data Transfer Timing When SF0CPHA Is "1"

Figure 16-4 shows the data transfer timing when SF0CPHA is "1". For the SCKF0, two cases are shown (SF0CPOL is "0" and "1").
SSNF0 is the slave selection input in Slave mode.
In Master mode, the transfer is started when data is written to SF0DWR. In Slave mode, the transfer is started at the first edge of SCKF0. The received data is sampled at the falling-edge of SCKF0 in SF0CPOL is “0” and the rising-edge of SCKF0 in SF0CPOL is “1”. The transmitted data is shifted at the rising-edge of SCKF0 in SF0CPOL is “0” and the falling-edge of SCKF0 in SF0CPOL is “1”.



Example for SF0MOZ=SF0SOZ=SF0SSZ=SF0SIZ=SF0LSB=SF0LEAD=SF0LAG=0

Figure 16-4 Waveform When SF0CPHA = 1 (transfer size is 8 bit)

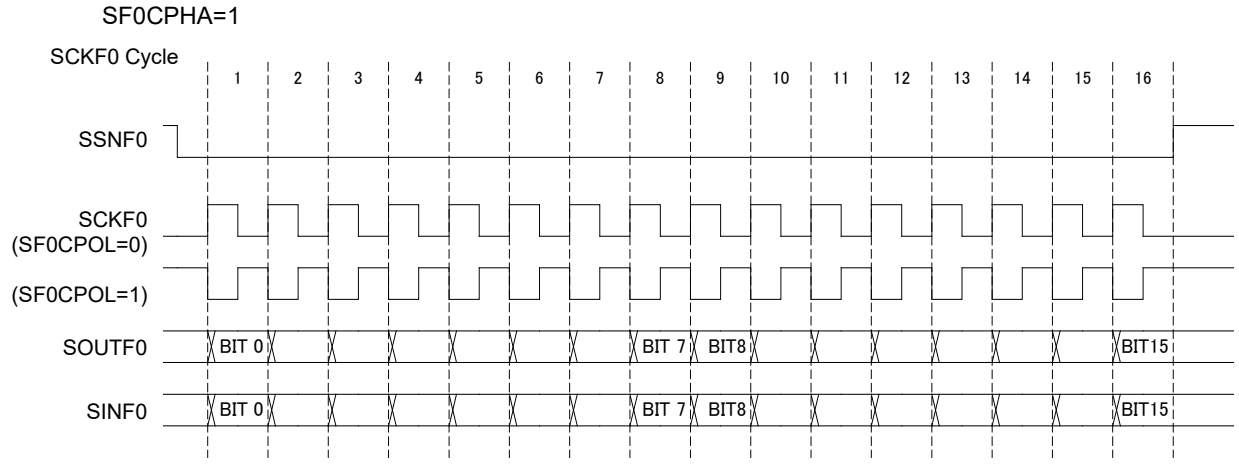


Figure 16-5 Waveform When SF0CPHA = 1 (transfer size is 16 bit)

16.3.4 Serial Clock Baud Rate

The baud rate is selected by the SF0BR of SF0BRR register. This is only valid in Master mode. The baud-rate clock SCKF0 is generated by dividing HSCLK.
The baud rate (f_{SCK}) is calculated as follows.

$f_{SCK} = f_{HSCLK} / (2 \times SF0BR[9:0])$

f_{SCK} : Frequency of baud-rate clock
f_{HSCLK} : Frequency of HSCLK
SF0BR : Value set in SF0BR[9:0] of the SF0BRR register (1 to 1023)
If 0 is set the SF0BR register, it is processed as 1.

For SF0BR, it can be selected from 1023 dividing types (2 to 2046).
In slave mode, the baud rate that can be communicated is limited to the frequency of the HSCLK divided two, or less.

16.3.5 Transfer Interval Setting

LEAD (SSNF0-SCKF0 time), LAG (SCKF0-SSNF0(H) time), and TDTL (SSNF0(H)-SSNF0(H)) can be set to adjust the speed to the slave. This setting is only valid in Master mode. It is ignored in Slave mode.
Setting during transferring is invalid.

(1) LEAD
A value from 0.5 to 1.5 SCKF0 can be set.
(2) LAG
A value from 0.5 to 1.5 SCKF0 can be set.
(3) TDTL
The minimum transfer interval can be controlled in SCKF0 clocks by setting SF0DTL bit of the SF0TRAC register.
If there is any transfer data in FIFO, the time set by this setting (SSNF0) changes to "1" during byte/word transfer.
If there is no transfer data in FIFO, this is "1" until any transmitted data is written.
If SF0DTL bit of the SF0TRAC register is set to “0”, the interval after transfer (TDTL) disappears and a continuous transfer is performed. SSNF0 is held to “0” and returns to “1” after the transfer is finished.

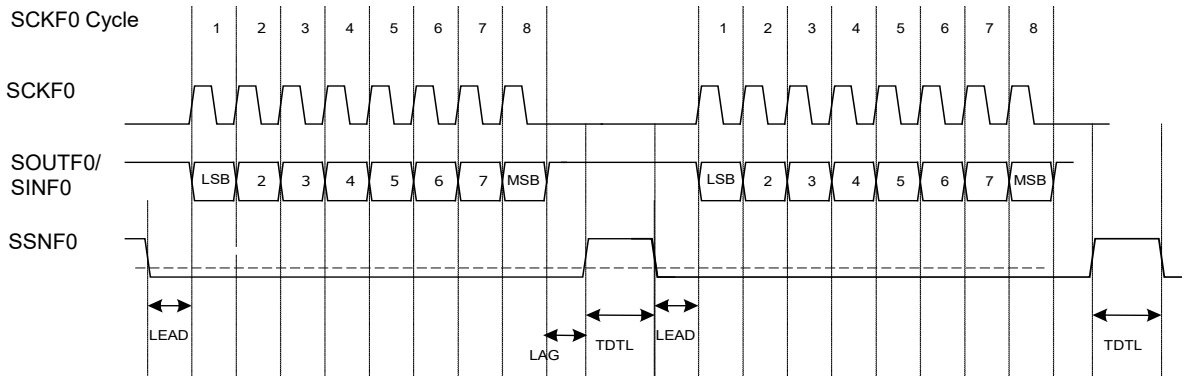


Figure 16-6 Transfer Interval (When SF0DTL Is Not "0")

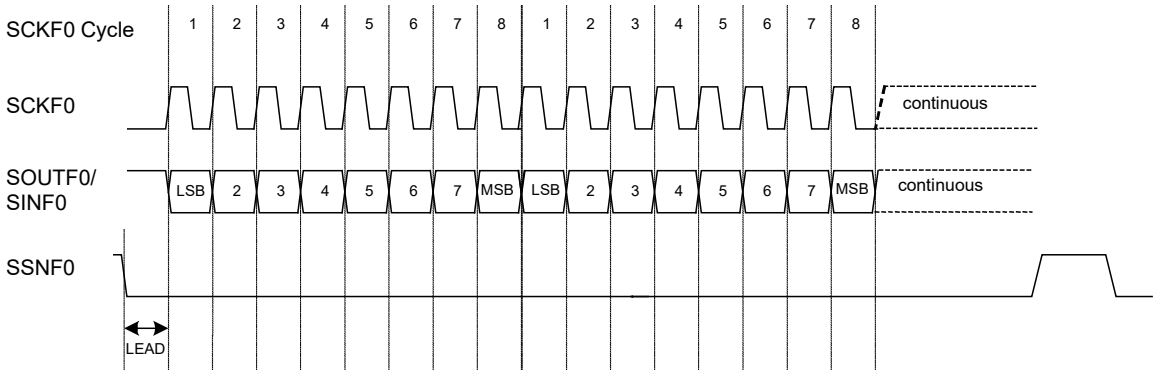


Figure 16-7 Transfer Interval (When SF0DTL Is "0")

16.3.6 Transmit Operation (Master Mode)

(1) Write the necessary values to SF0CTRL, SF0INTC, SF0BRR, and SF0TRAC, set the SF0MST bit to Master mode, and set the SF0SPE bit to enable the SSIOF transfer.
(2) When the transmitted data is written to SF0DWR, the transmit FIFO Empty flag changes to "0" (SF0TFE = "0"). SSIOF starts the automatic transmission and outputs the transmitted data from LSB or MSB on the SOUTF0 pin according to the SF0LSB setting.
(3) The synchronous clock, which was set by the SF0CPOL, SF0CPHA, and SF0BRR registers, is output from the SCKF0 pin.
(4) Transmitted data can be written to SF0DWR successively. However, if further writing is performed when the transmit FIFO is in Full status (SF0TFF = "1"), a write overflow occurs. (SF0WOF = "1", No interrupt is generated.)
(5) The SF0SPIF bit is set each time the transfer of 1 byte is completed. (SF0SPIF = "1")
(6) A transmission interrupt occurs if the remaining data in the transmit FIFO matches the data count selected with SF0TFIC. (SF0TFI = "1")
(7) If the transmit FIFO becomes empty and the transfer of the last byte is completed, a transfer completion interrupt is generated. (SF0FI = "1")

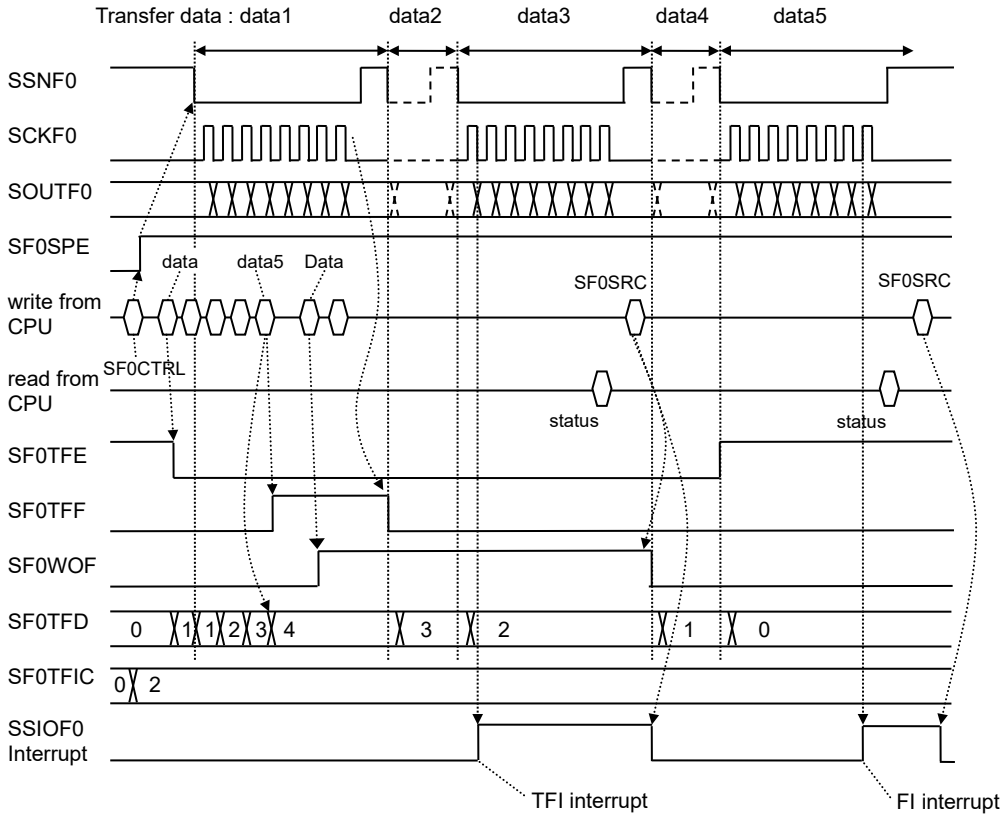


Figure 16-8 Transmit Operation in the master mode

16.3.7 Receive Operation (Master Mode)

The master mode of the synchronous serial with FIFO starts by setting data in a transmission buffer.
Data needs to be set into a transmission buffer even master mode reception only.

(1) Write the necessary values to SF0CTRL, SF0INTC, SF0BRR, and SF0TRAC, set the SF0MST bit to master mode, and set the SF0SPE bit to enable the SSIOF transfer.
(2) When the data is written to SF0DWR, the SSIOF transfer is started.
(3) The sync clock, which was set by the SF0CPOL, SF0CPHA, and SF0BRR registers, is output from the SCKF0 pin.
(4) On the SINF0 pin, the received data is sampled from LSB or MSB according to the SF0LSB setting and stored in the receive FIFO. The receive FIFO Empty flag changes to "0" (SF0RFE = "0").
(5) The SF0SPIF bit is set each time the transfer of 1 byte is completed. (SF0SPIF = "1")
(6) If the number of data received in the receive FIFO is equal to or more than matches following the data count selected with SF0RFIC of SF0CR, SF0RFI of SF0SRR is set to generate a reception interrupt.
(7) When the receive FIFO becomes Full, the subsequent reception is disabled. If the reception is performed in this state, an overrun error interrupt is generated. (SF0ORF = "1")
(8) If the temporary data of transmit FIFO becomes empty and the transfer of the last byte is completed, a transfer completion interrupt is generated. (SF0FI = "1")

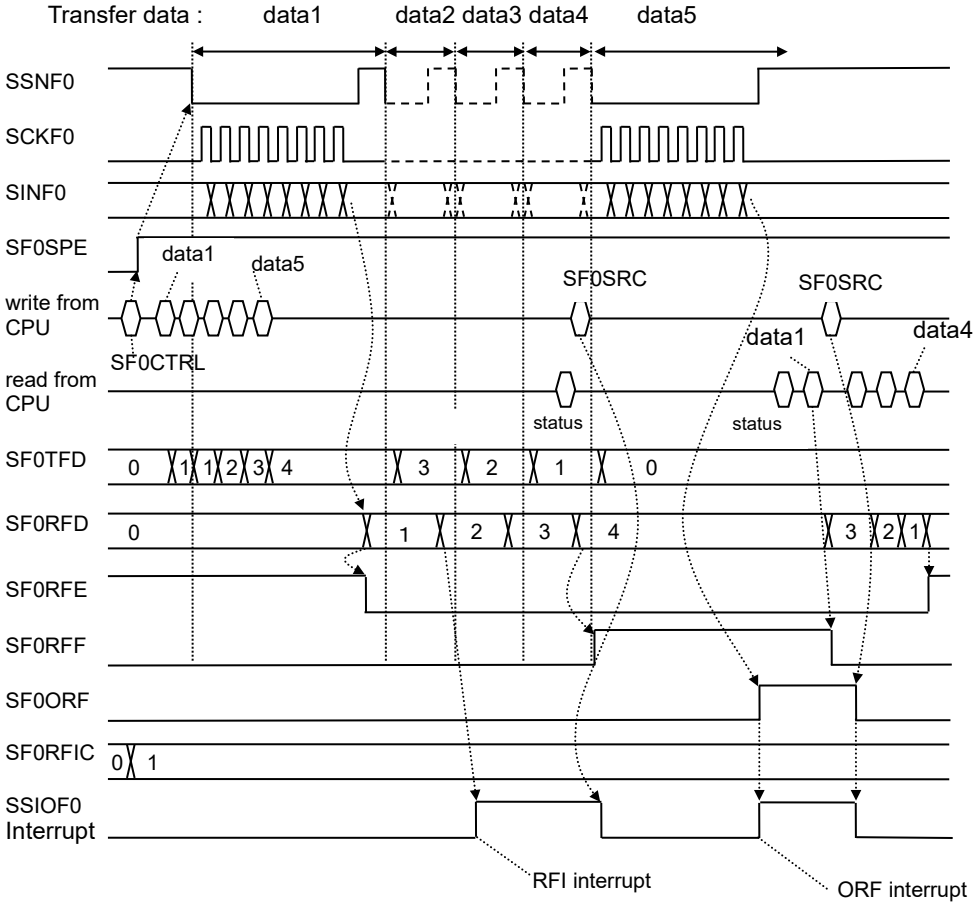


Figure 16-9 Receive Operation in the master mode

16.3.8 FIFO Operation

SSIOF includes the receive FIFO of 4-stages and the transmit FIFO of 4 stages. The FIFO state is indicated in the SF0TFF, SF0TFE, SF0RFF, and SF0RFE bits of SF0SRR, and the SF0TFD and SF0RFD bit of SF0FSR.
There are three FIFO states, Full (SF0TFF and SF0RFF), Empty (SF0TFE and SF0RFE), and Depth (SF0TFD and SF0RFD).

16.3.9 Write Overflow

If further writing is performed when the transmit FIFO is in Full status (SF0TFF = "1"), a write overflow is set. (SF0WOF = "1")
However, interrupt is not generated even when a write overflow occurs.
SF0WOF is cleared when write "1" in SF0WOFC bit of SF0SRC.

16.3.10 Overrun Error

If further reception is performed when the receive FIFO is in Full status (SF0RFF = "1"), an overrun error occurs. (SF0ORF = "1")
If an overrun error occurs, the SF0ORF bit of SF0SRR is set, and an overrun error interrupt is generated. The newly received data is not held.
Read the content of the receive FIFO, clear the SF0RFF bit, then write "1" in the SF0ORFC bit to clear the SF0ORF bit.

16.3.11 FIFO Clear

The transmit/receive counter control of FIFO can be initialized to the initial setting state (SF0TFF = "0", SF0TFE = "1", SF0RFF = "0", and SF0RFE = "1" in the SF0SRR register and SF0TFD = "000" and SF0RFD = "000" in the SF0FSR register) by setting the SF0FICL bit of the SF0CTRL register to "1".
The SF0FICL bit of the SF0CTRL register needs to be "0", before next transfer operation.
Even if SF0FICL bit of SF0CTRL register is set to "1", the interrupt is not changed for SF0RFIC, SF0TFIC, SF0ORIE, SF0FIE, SF0RFIE, and SF0TFIE of the SF0INTC register, and SF0ORF, SF0FI, SF0RFI, and SF0TFI of the SF0SRR register.
This bit can be used to discard the data of FIFO when the communication is aborted.

16.3.12 Transfer When Slave Has Different Number of FIFO Transfer Bytes/Half Words

(1) The master sends data only when the transmitted data is already written in FIFO.
(2) As the slave's transmit data count is determined by the master, data is transferred as follows if the number of FIFO transfer bytes/half words of slave is different from that of the master.
If the transmitted data is not written in the slave's FIFO, a 0xFF ((0xFFFF) for half word) is sent, including the state after a reset.

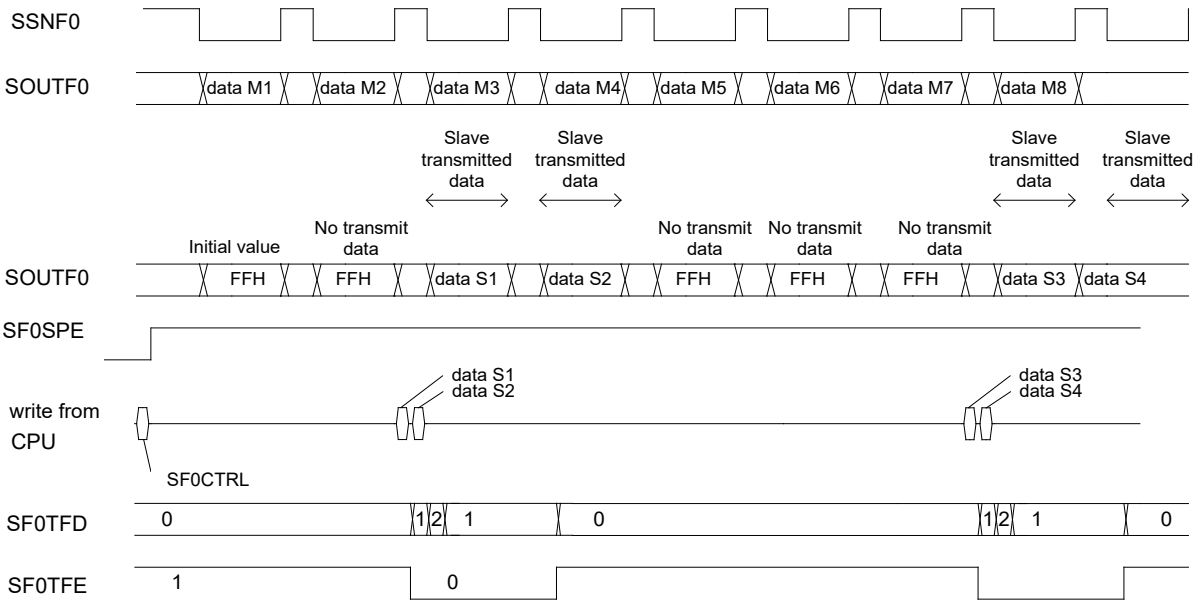


Figure 16-10 Transfer When Slave Has Different Number of FIFO Transfer Bytes/Half Words

16.3.13 Mode Fault (MDF)

When the SF0MDFE bit of SF0CTRL is “1”, a mode fault error occurs if the SSNF0 signal becomes low level in Master mode. (SF0SRR's SF0MDF is set.) If this bit becomes “1”, it indicates that there is risk of two or more masters competing for the bus. To monitor the SSNF0 signal, allow input of the SSNF0 pin with the PnmIE bit of the port mode register, and set the SF0SSZ bit of the SF0CTRL register to “1”.

When a mode fault error occurs, SSIOF performs the following operations since there is a risk of bus latch-up:
1. Automatically sets the SF0MST bit of SF0CTRL to “0” (slave).
2. Automatically sets the SF0SPE bit of SF0CTRL to “0” (disabled) to make the SSIOF unable to transfer.
3. Set SF0MDF of SF0SRR, and generates an interrupt if the SF0MFIE bit of SF0INTC is “1” (interrupt permitted).

The system should resolve the causes of the mode fault, and then clear SF0MDF according to the following steps:
1. Write “1” in SF0MDFC to clear it.
2. Set SF0CTRL again.

Figure 16-11 shows the timing that allows a mode fault operation.

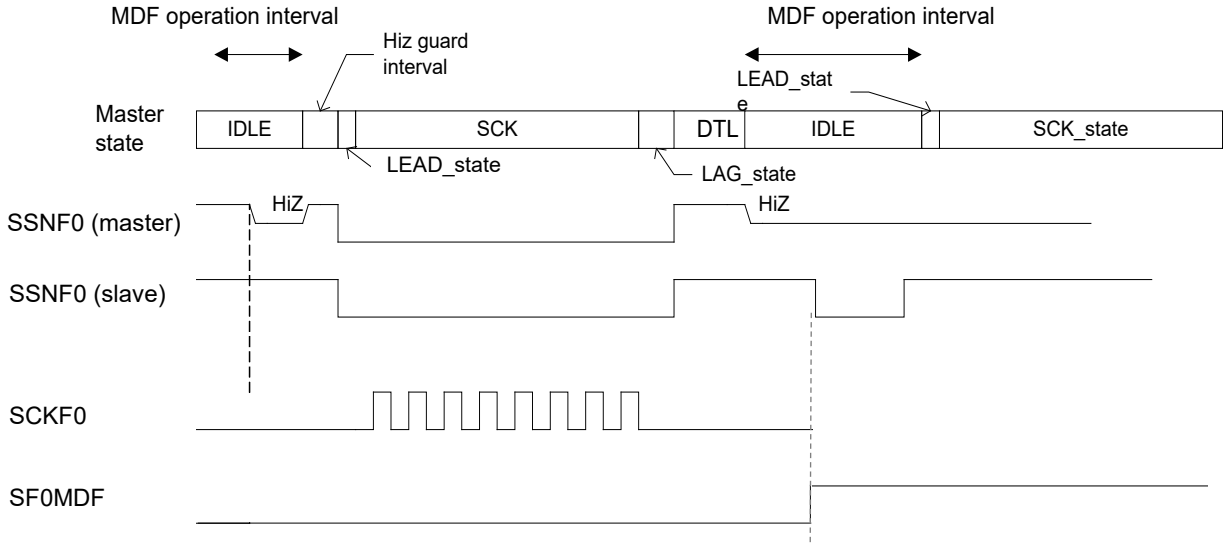


Figure 16-11 Timing That Allows Mode Fault Operation

16.3.14 Interrupt Source

16.3.14.1 SSIOF Interrupt Source

There are the following five types.

- Mode fault
  If a mode fault (multi-master bus contention) occurs, SF0MDF of SF0SRR is set and a mode fault interrupt is generated.
- Overrun
  If an overrun occurs, SF0ORF of SF0SRR is set, and an overrun error interrupt is generated.
- Transmit FIFO threshold
  If the remaining data of the transmit FIFO matches the byte count selected with SF0TFIC, SF0TFI of SF0SRR is set to generate a transmission interrupt.
- Receive FIFO threshold
  If the number of data received in the receive FIFO is equal to or more than following the byte count selected with SF0RFIC of SF0INTC, SF0RFI of SF0SRR is set to generate a reception interrupt.
- End of transfer
  If the transmit FIFO becomes empty and the transfer of the last byte is finished, SF0FI of SF0SRR is set to generate a transfer end interrupt.

16.3.14.2 Clear SSIOF Interrupt

An interrupt request is cleared by writing "1" to each interrupt bit (SF0TFC, SF0RFC, SF0FC, SF0ORFC, SF0MDFC, SF0SPIFC, and SF0WOFC) of the SF0SRC.

16.3.14.3 SSIOF Interrupt Timing

Figure 16-12 shows the interrupt timing. The remaining transmit byte count interrupt (TFI) generates an interrupt in 1 to 2 SYSCLK after the shift clock of the second bit.
For receive byte count interrupt (RFI), transfer completion interrupt (FI), and overrun (ORF), an interrupt is generated in 1 to 2 SYSCLK after the sampling clock at the MSB.
For MDF, an interrupt is generated at a mode fault occurrence.

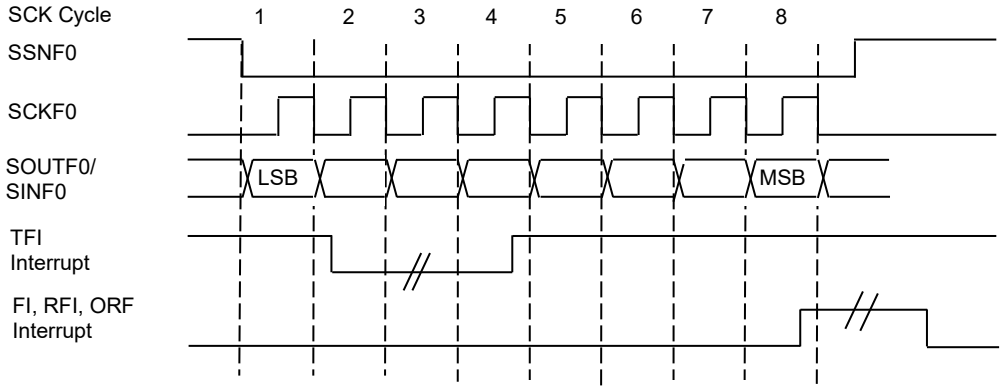


Figure 16-12 Interrupt Timing

16.3.14.4 Interrupt processing flow

Figure 16-13 shows the processing flowchart in the receiving operation of the slave mode.

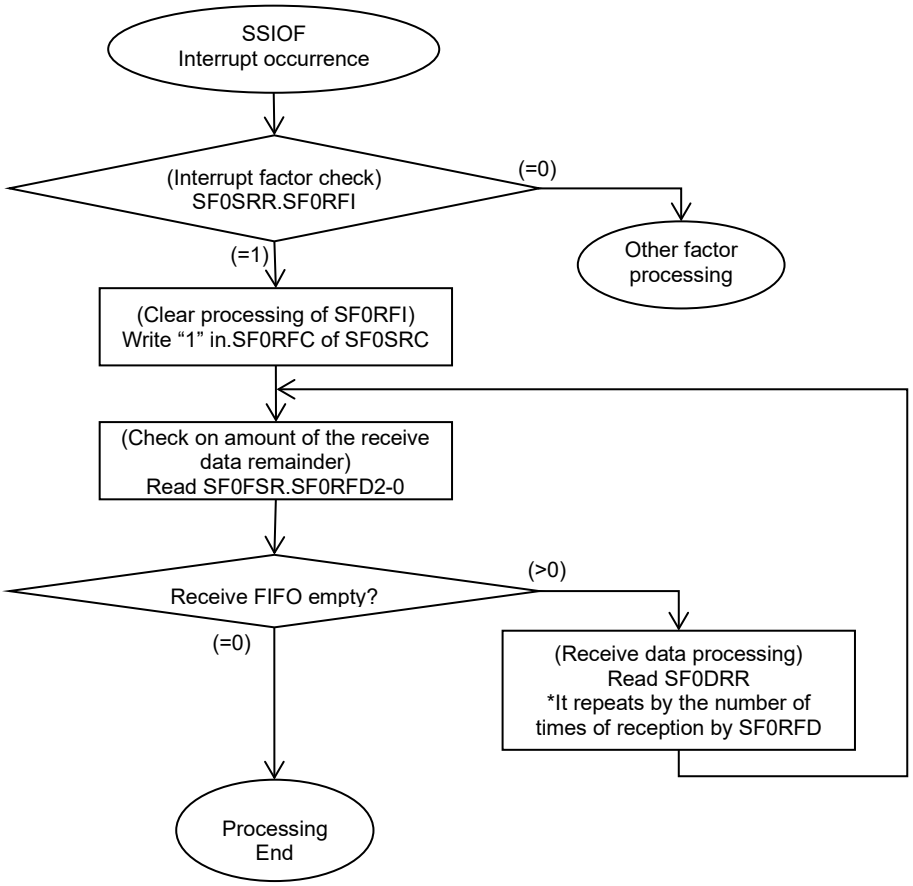


Figure 16-13 Example of the interrupt control flowchart

16.3.15 Hi-Z Operation

Figure 16-14 shows an example of using Hi-Z (SF0MOZ, SF0SOZ, and SF0SSZ).
The Hi-Z transmit interval of the master is limited to the IDLE time shown below.
To reduce the effect of noise in the Hi-Z state, "1"/"0" is fixed 1SCKF0 before the transmission starts (Hi-Z guard interval), and "1"/"0" is fixed during the DTL time of the transfer interval. If any of the SF0MOZ, SF0SOZ, and SF0SSZ bit is set to 1, the Hi-Z guard interval is inserted before the transmission starts.
If SF0MOZ = "1", the SCKF0 should be pull-up or pull-down to avoid hi-impedance input state.

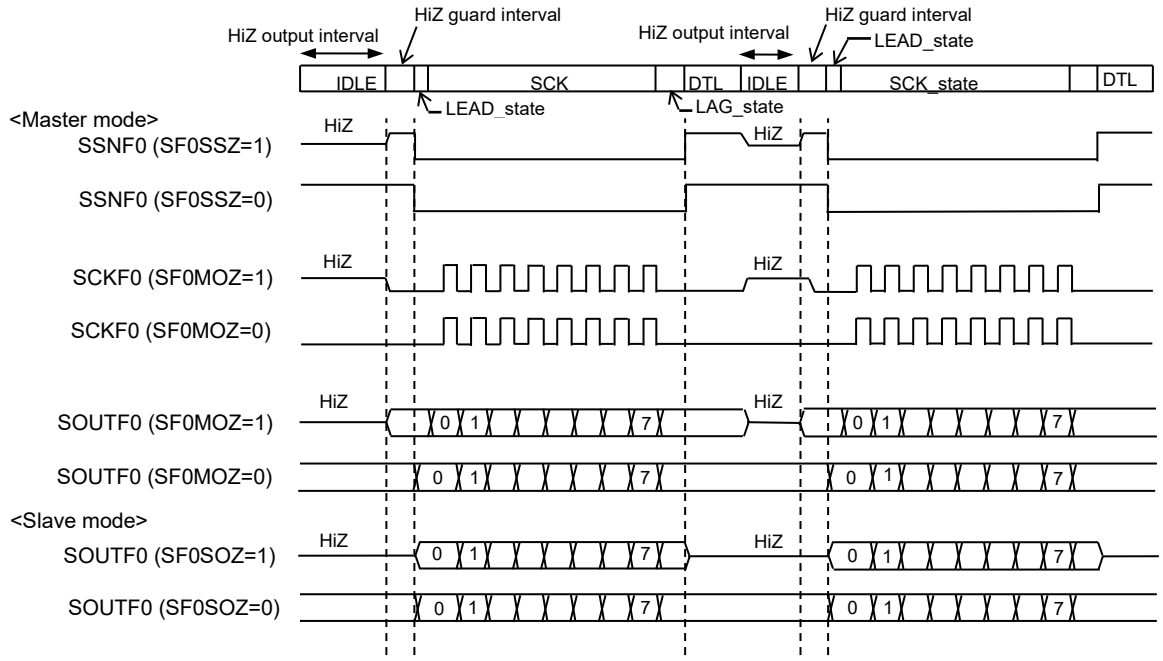


Figure 16-14 Hi-Z Operation

16.3.16 Interval from SF0MST Setting to Transfer Start

The SSIOF bus (SOUTF0, SCKF0, and SSNF0) remains high impedance until Master mode is set.
After setting SF0MSTR, wait for at least 100ns before starting the transmission (SF0SPE = "1", or transfer started by data write).

16.3.17 Pin Settings

To enable the SSIOF function, the applicable bit of each related port register needs to be set. See Chapter 22, "General Purpose Port (GPIO)" for details about the port registers.
For SOUTFn, SINFn, SCKFn and SSNFn, the ports can be selected from several possibilities.
Be sure to select one of the following combinations of ports for SOUTFn/SINFn/SCKFn/SSNFn.
Note that only one port can be selected as port.

Table 16-2 Pin Combination

	SSIOF pins	Combination 1	Combination 2
SSIOF0	SOUTF0, SINF0, SCKF0, SSNF0	P33, P34, P32, P35	P41, P42 P40, P43
SSIOF1	SOUTF1, SINF1, SCKF1, SSNF1	P61, P62, P60, P63	-

In the master mode, set the SOUTFn, SSNFn pins to output enable, the SINFn pin to input enable, and the SCKFn to input/output enable. However, set the SSNFn pin to input/output enable when the SF0SSZ in the SFnCTRL register is "1".
In the slave mode, set the SOUTFn pin to output enable, the SSNFn, SCKFn, SINFn pins to input enable.

Chapter 17

(Skipped)

Chapter 18

UART with FIFO (UARTF)

18. UART with FIFO (UARTF)

18.1 Overview

This LSI has 4 channels of UART with FIFO (UARTF). The UART with FIFO (UARTF) functions as the input/output interface, carries out serial-to-parallel conversion of the data sent from the peripheral devices, and converts the parallel data sent from the CPU into serial data. The UART has a 4-stages FIFO for transmission and reception, capable of storing up to 4 characters of data during transmission/reception in the FIFO mode.
Further, the receive FIFO generates 3 bits of error data for every character of received data. The CPU can read out the UARTF state at any time. The information that can be read out consists of the type and status of the transfer operation under execution, and the statuses of errors such as parity, overrun, framing errors, and break interrupt, etc.

The I/O pins of the UARTF are assigned as the secondary function of port 2, 3, and 4. For the ports 2, 3, and 4, see Chapter 22 "General Purpose Port (GPIO)".

18.1.1 Features

- Full duplex buffer system
- All status reporting function
- 4-stages transmission and reception FIFOs.
- Independent control of transmit, receive, line status data set interrupt and FIFO
- Programmable serial interface
  - 5, 6, 7, and 8 bit characters
  - Odd parity, even parity, no parity generation and verification
  - 1, 1.5, or 2 stop bit
- Communication speed: Settable within the range of 2400bps to 115200bps.
- Built-in baud rate generator.
- Generates transmission/receive DMA request

18.1.2 Configuration

Figure 18-1 shows the configuration of the UARTF.

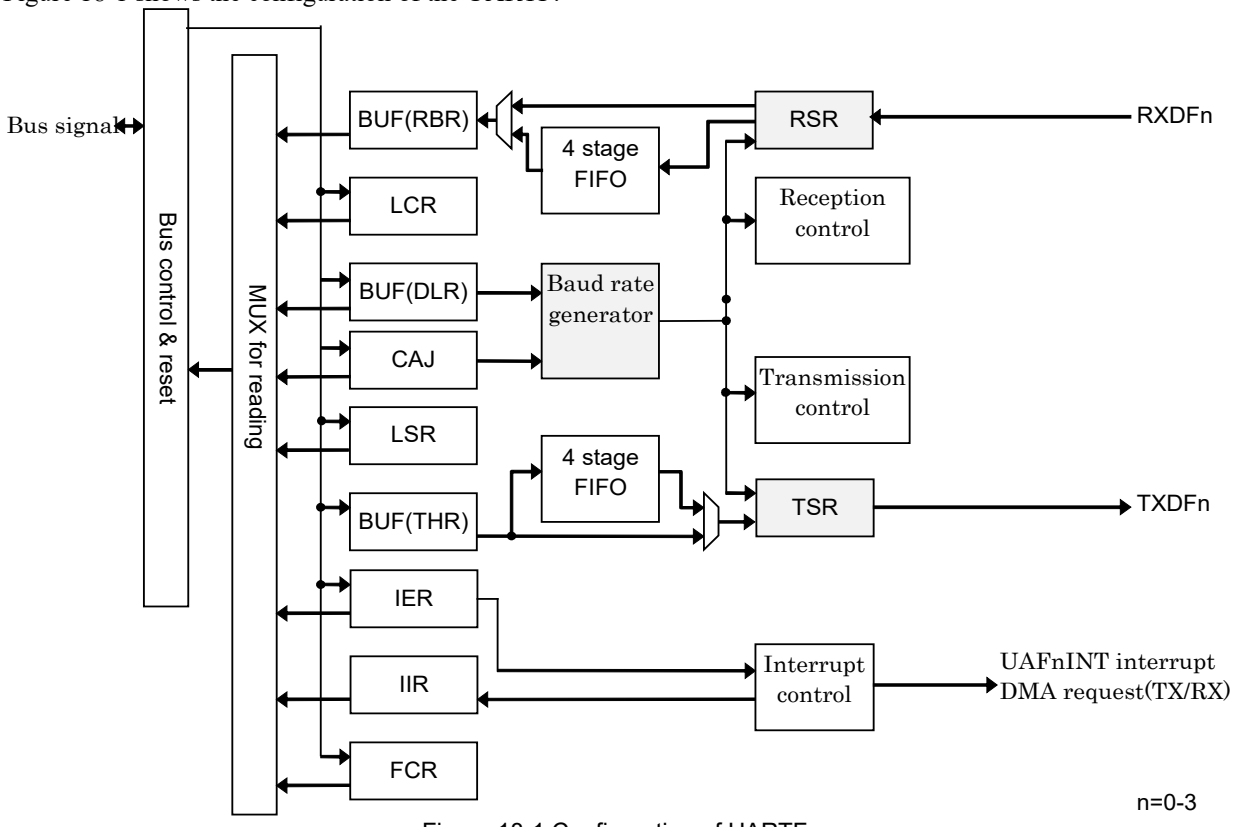


Figure 18-1 Configuration of UARTF

18.1.3 List of Pins

Table 18-1 Pins

Functional pin name	I/O	Description	Corresponding LSI pin name
RXDF0	I	UARTF0 data input	P20/P32/P60/P84
TXDF0	O	UARTF0 data output	P21/P33/P61/P85
RXDF1	I	UARTF1 data input	P46/P52/P64/P70
TXDF1	O	UARTF1 data output	P47/P51/P65/P71
RXDF2	I	UARTF2 data input	P22/P34/P44/P62
TXDF2	O	UARTF2 data output	P23/P35/P45/P63
RXDF3	I	UARTF3 data input	P40/P54/P56/P82
TXDF3	O	UARTF3 data output	P41/P53/P55/P83

18.2 Description of Registers

18.2.1 List of Registers

Address/Offset	Name	Symbol (Word)	R/W	Size	Initial value
0x4100_1800	UARTF channel 0 base address	UARTF0	-	-	-
0x4100_1820	UARTF channel 1 base address	UARTF1	-	-	-
0x4100_1840	UARTF channel 2 base address	UARTF2	-	-	-
0x4100_1860	UARTF channel 3 base address	UARTF3	-	-	-
0x00	UARTF transmission / reception buffer	UAF0BUF	R/W	32	0x0000_00xx
0x04	UARTF interrupt enable register	UAF0IER	R/W	32	0x0000_0000
0x08	UARTF interrupt status register	UAF0IIR	R	32	0x0000_0001
0x0C	UARTF mode register	UAF0MOD	R/W	32	0x0000_0000
0x10	UARTF line status register	UAF0LSR	R	32	0x0000_0160
0x14	UARTF clock adjustment register	UAF0CAJ	R/W	32	0x0000_000D

18.2.2 UARTF Transmission / Reception Buffer (UAF0BUF)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UF0B															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

UAF0BUF is a special function register (SFR) that provides the following three functions.

(1) Read-only register for buffering received data: Receiver Buffer Register (RBR)
RBR is the register that holds received data from 5 to 8 bit depending on the character length.
The bit 0 of a data word is always the first serial data bit received. If data less than 8 bit is received, the data is entered in the right justified manner towards the LSB.
When the UART carries out parallel-to-serial or serial-to-parallel conversion operation, the register has the double buffer configuration so that read operations can be made.
RBR can be read by software when UF0DLAB of UAF0MOD is “0”. When UAF0BUF is read, RBR can be read from UF0B[7:0]. For UF0B[15:8], 0x00 can be read.
The reset value is undefined.

(2) Write-only register for setting transmitted data: Transmitter Holding Register (THR)
THR is the register that holds transmitted data from 5 to 8 bit depending on the character length.
The bit 0 of the data word is always the first serial data bit that is transmitted.
When the UART carries out parallel-to-serial or serial-to-parallel conversion operation, the register has the double buffer configuration so that write operations can be made.
THR can be written by software when UF0DLAB of UAF0MOD is “0”. When UAF0BUF is written, the UF0B[7:0] data is written to THR. The UF0B[15:8] data are invalid.

(3) 16 bit divisor latch for baud rate generator: Divisor Latch Resister (DLR)
DLR can be read/written by software when UF0DLAB of UAF0MOD is “1”. For details, see "18.3.4 Baud rate clock generation".

Table 18-2 Functions

Action	UF0DLAB = 0		UF0DLAB = 1	
	UAF0B[15:8]	UAF0B[7:0]	UAF0B[15:8]	UAF0B[7:0]
Read	0x00	RBR	DLR[15:8]	DLR[7:0]
Write	Disabled	THR	DLR[15:8]	DLR[7:0]

18.2.3 UARTF Interrupt Enable Register (UAF0IER)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	UF0TEMTI	-	-	-	UF0ELSI	UF0ETBEI	UF0ERBFI
R/W	–	–	–	–	–	–	–	–	–	R/W	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UAF0IER is a special function register (SFR) used to set whether to enable/disable the UART interrupt source.

Bit No	Bit name	Description
6	UF0TEMTI	This is used to enable the transmission idle interrupt. 0: Disabled (Initial value) 1: Enabled
2	UF0ELSI	This is used to enable the received data error interrupt. 0: Disabled (Initial value) 1: Enabled
1	UF0ETBEI	This is used to enable the transmitted data write request interrupt. 0: Disabled (Initial value) 1: Enabled
0	UF0ERBFI	This is used to enable the received data read request interrupt. This interrupt includes the character timeout interrupt in the FIFO mode. 0: Disabled (Initial value) 1: Enabled

18.2.4 UARTF Interrupt Status Register (UAF0IIR)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	UF0FMD		-	-	UF0IRID			UF0IRP
R/W	–	–	–	–	–	–	–	–	R	R	-	-	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UAF0IIR is a special function register (SFR) used to indicate the UART state in interrupt transmit/receive operations. UAF0IIR stores information indicating that an interrupt with a certain priority level is pending, along with the type of that interrupt. UAF0IIR indicates the interrupt with the highest priority level that is pending.

Bit No	Bit name	Description
7 to 6	UF0FMD	UF0FMD indicates FIFO mode. 00: Non-FIFO mode (Initial value) 01: Reserved 10: Reserved 11: FIFO mode
3 to 1	UF0IRID	UF0IRID indicates the source of the UARTF0 interrupt. LVL=1 is the highest priority, and the interrupt source with the highest priority is notified. See Table 18-3 for detail.
0	UF0IRP	UF0IRP indicates whether a UARTF0 interrupt has occurred. 0: Occurred 1: Not occurred (Initial value)

Table 18-3 UF0IRID Status

UF0IRID	LVL	Flag	Source	Reset Process
000	-	-	No interrupts (Initial value)	-
011	1	Received data error	Overrun error, parity error, framing error, break interrupt	Read UAF0LSR
010	2	Received data read request	FIFO disabled: The received data is available. FIFO enabled: Reached the Trigger level	Read RBR, or when FIFO drops below trigger level
110	2	Character timeout	At least one character is present in the receive FIFO, and no other character was placed into or read out within 4 character time.	Read RBR
001	3	Transmitted data write request	Non-FIFOmode: THR has been writable. FIFO mode: the transmit FIFO has been emptied.	Read UAF0IIR or write THR
101	4	Transmission idle	THR and TSR are empty and not during the UART transmission.	Write THR

18.2.5 UARTF Mode Register (UAF0MOD)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	UF0FTL		-	UF0TFR	UF0RFR	UF0FEN	UF0DLAB	UF0BC	UF0PT			UF0STP	UF0LG	
R/W	–	–	R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UAF0MOD is a special function register (SFR) used to set the mode of the UART.

Bit No	Bit name	Description
13 to 12	UF0FTL	UF0FTL selects the trigger level for the receive FIFO interrupt. It is used to generate the reception DMA request too. 00: 1character (Initial value) 01: 2 characters 10: 3 characters 11: 4 characters
10	UF0TFR	UF0TFR instructs to reset the transmit FIFO. 0: Normal operation (Initial value) 1: Clears the transmit FIFO. A data being transmitted when transmit FIFO clearance is instructed is not cleared.
9	UF0RFR	UF0RFR instructs to reset the receive FIFO. 0: Normal operation (Initial value) 1: Clears the receive FIFO. A data being received when receive FIFO clearance is instructed is not cleared.
8	UF0FEN	UF0FEN selects to use FIFO. FIFO will be cleared when switching between FIFO enabled/disabled. 0: Disabled; Non-FIFO mode (Initial value) 1: Enabled; FIFO mode
7	UF0DLAB	UF0DLAB selects the access register of UAF0BUF. 0: RBR and THR (Initial value) 1: DLR
6	UF0BC	UF0BC selects the break control. Turning this to "1" brings the TXDFn pin to the spacing state (logical 0). The control by this bit is valid only on the TXDFn pin. This means that TXDF0 is masked but the transmit operation continues internally. The use of the break control allows the CPU to send an alarm to the terminal of the computer communication system. 0: Break control not implemented (Initial value) 1: Break control implemented
5 to 3	UF0PT	UF0PT selects the parity bit. xx0: No parity bit (Initial value) 001: Odd parity 011: Even parity 101: Fixed “1” 111: Fixed “0”
2	UF0STP	UF0STP selects the stop bit count of the character transmitted. 0: 1 stop bit (Initial value) 1: 1.5 stop bit (when character length is 5 bit) 2 stop bit (when character length is 6, 7, 8 bit)
1 to 0	UF0LG	UF0LG specify the character length of data. 00: 5 bit length (Initial value) 01: 6 bit length 10: 7 bit length 11: 8 bit length

18.2.6 UARTF Line Status Register (UAF0LSR)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	UF0TIDL	UF0RFE	UF0TEMT	UF0THRE	UF0BI	UF0FER	UF0PER	UF0OER	UF0DR
R/W	–	–	–	–	–	–	–	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0

UAF0LSR is a special function register (SFR) used to display the status.
UAF0LSR is normally the first register read out by the CPU for determining the interrupt cause or for polling the status of the serial communication channel.
UF0OER, UF0PER, UF0FER, and UF0BI are error conditions, which generate a received data error interrupt (a LVL = "1" interrupt in the UAF0IIR) if any of the states is detected. This interrupt is enabled by setting UF0ELSI of UARTF0IER to "1".

Bit No	Bit name	Description
8	UF0TIDL	UF0TIDL indicates a transmission idle state. The transmission idle shows THR and TSR are empty and UART is not transmitting. This bit is cleared by "0" by writing in THR. This bit is not cleared even if read UAF0LSR register. In the UART transmission, send the last data and become "1" after the transmission in stop bit. 0: Not transmission idle 1: Transmission idle (Initial value)
7	UF0RFE	UF0RFE indicates to have been occurred error in the FIFO mode. UF0RFE is always "0" when FIFO is disabled. This bit will be cleared when the data causing the error is read out from the RBR, or when the data causing the error is cleared by the FIFO clear and then reading out the UAF0LSR. 0: No data error in the FIFO mode (Initial value) 1: A parity error, framing error, or break interrupt occurred in the FIFO mode
6	UF0TEMT	UF0TEMT indicates a transmission data empty state. When a character is loaded into THR, this bit is cleared to "0" and remains "0" until the character is transferred from TXDF0. This bit is not cleared to "0" by reading UAF0LSR. 0: Transmitted data remains in either THR or TSR 1: Both THR and TSR are empty (Initial value) Both the transmit FIFO and the shift register are empty in the FIFO mode.
5	UF0THRE	UF0THRE indicates that preparations have been made for calling a new character to be transmitted by the UART. This bit is set to "1" when the character is transferred from THR to the shift register for transmission (TSR). This bit is cleared to "0" by writing to THR. This bit will not be cleared by reading out the UAF0LSR register. In FIFO mode, this bit is set when the transmit FIFO is empty. This bit is cleared when one character is written to the transmit FIFO. If the THRE interrupt is enabled by UF0ETBEI of UAF0IER, UF0THRE initiates the third-order priority interrupt (LVL=3) to UAF0IIR. 0: Transmit data still present in the THR (Initial value) 1: THR ready for transmission
4	UF0BI	UF0BI indicates that a break interrupt occurred. This bit is set to "1" when the input data is maintained in the spacing ("0") state during the transmission of one frame (start bit + data bit + parity bit + stop bit). This bit will be cleared when the CPU reads UAF0LSR. In FIFO mode, this is related to a specific character in the FIFO. This bit reflects the break interrupt state when the break character comes to the beginning of the FIFO. The CPU erases the error if the related character comes to the beginning of the FIFO before the first reading of UAF0LSR. When a break interrupt occurs, only one zero character will be loaded into the FIFO. 0: No break interrupt (Initial value) 1: Break interrupt occurred

Bit No	Bit name	Description
3	UF0FER	UF0FER indicates that a framing error occurred. A framing error indicates that there is no valid stop bit in the received character. This bit is set to "1" when the stop bit after the last data bit or after the parity bit is "0" (spacing level). This bit is cleared when UAF0LSR is read. In FIFO mode, the framing error is related to a specific character in the FIFO. This bit indicates that an error is present when that character comes to the beginning of the FIFO. 0: No framing error (Initial value) 1: Framing error occurred
2	UF0PER	UF0PER indicates that a parity error occurred. This is enabled only when parity is enabled. This bit is cleared when UAF0LSR is read. In FIFO mode, this bit indicates that an error exists for the leading data. If a parity error occurs in the data that is not the leading data in the FIFO, it is not reflected to this bit. 0: No parity error (Initial value) 1: Parity error occurred
1	UF0OER	UF0ER indicates that an overrun error occurred. The overrun error indicates that the CPU did not read the data in RBR before the next character was sent to RBR to overwrite the previous character. In FIFO mode, an overrun error occurs after the FIFO has become full when the next character is completely received. Reading UAF0LSR after an overrun error will clear the overrun error. The character during reception is overwritten instead of being transferred to FIFO. This bit is cleared when UAF0LSR is read. 0: No overrun error (Initial value) 1: Overrun error occurred
0	UF0DR	UF0DR indicates the RBR state. UF0DR is set to “1” when the input character has been received and transmitted to RBR. This bit is cleared when the RBR data is read. 0: No valid data in RBR (Initial value) 1: There is valid data in RBR

[Note]

- **After transmission data were sent by TXDFn pin, UF0TEMT bit becomes "1", but the transmission of parity bit and the stop bit is not completed at that point.**

18.2.7 UARTF Clock Adjustment Register (UAF0CAJ)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	UF0RMV	-	UF0CAJ				
R/W	–	–	–	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

UAF0CAJ is a special function register (SFR) used to adjust the base clock for the baud rate clock.

Bit No	Bit name	Description
6	UF0RMV	UF0RMV is used to enable a function to adjust the base clock for the baud rate clock. 0: Enabled (Initial value) 1: Disabled
4 to 0	UF0CAJ	UF0CAJ is used to set adjustment value the base clock for the baud rate clock. For details, see 18.3.4 "Baud Rate Clock Generation".

18.3 Description of Operation

The UARTF is programmed with UAF0IER, UAF0MOD, DLR(UAF0BUF), and UAF0CAJ. These registers define the character length, number of stop bit, parity, baud rate, etc.
Though the registers can be written in any order, UAF0IER needs to be written to last because it controls the interrupt enable. Once the UARTF is programmed to be operable, these registers can be updated any time when the UARTF is not transmitting or receiving data.

18.3.1 Data Transmission

Figure 18-2 shows the transmission timing.
Writing data to THR will transfer the contents through the transmit FIFO to the transmit shift register. Within 16 baud rate clocks after the THRE bit rise is detected, the start bit is sent, followed by the data one bit at a time from the least significant bit. When the data to be transmitted is 7 bit, the most significant bit will not be sent.
If parity is enabled by UF0PT of UAF0MOD, then the parity bit is sent. This is followed by the stop bit which indicates the end of transmitting one frame of data.
After the data is transmitted, the UF0THRE bit of UAF0LSR is set to "1" to indicate that it is ready for the next transmission. This bit is cleared when one character is written to the transmit FIFO. Also, if the THRE interrupt is enabled by UF0ETBEI of UAF0IER, THRE initiates a LVL=3 interrupt to UAF0IIR. If UF0THRE is the interrupt source indicated in UF0RID, this bit is cleared by reading the UAF0IIR register.

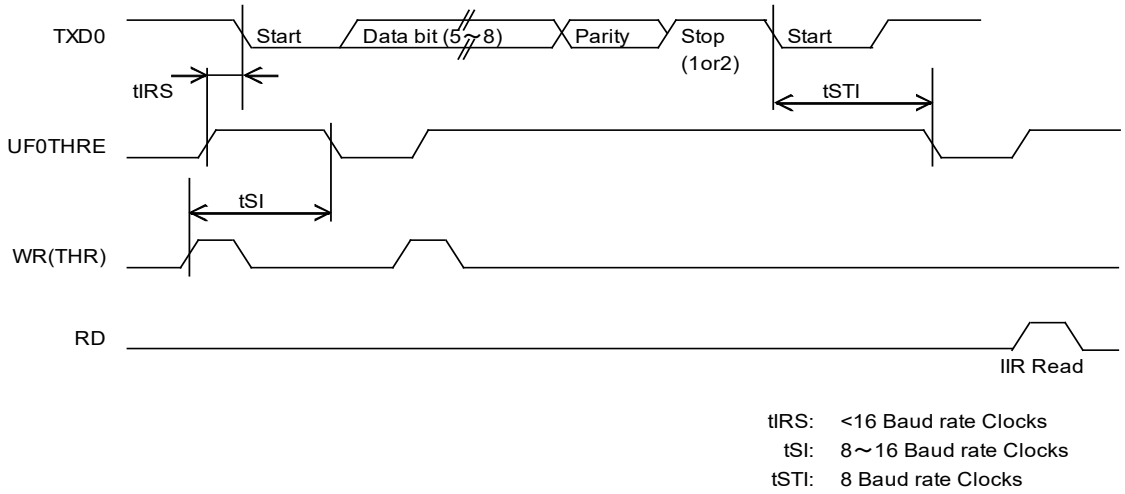


Figure 18-2 Transmission Timing

[Note]

- **Even if transmit FIFO is an empty state, there are some cases that all transmit processing doesn't complete. Confirm that it became the transmission idle state in UF0TIDL bit of UAF0LSR register before stopping high-speed clock (Transition to mode such as STOP). Or confirm that a transmission idle interrupt occurred.**

18.3.2 Transmission Flow

Figure18-3 shows the transmission flowchart.

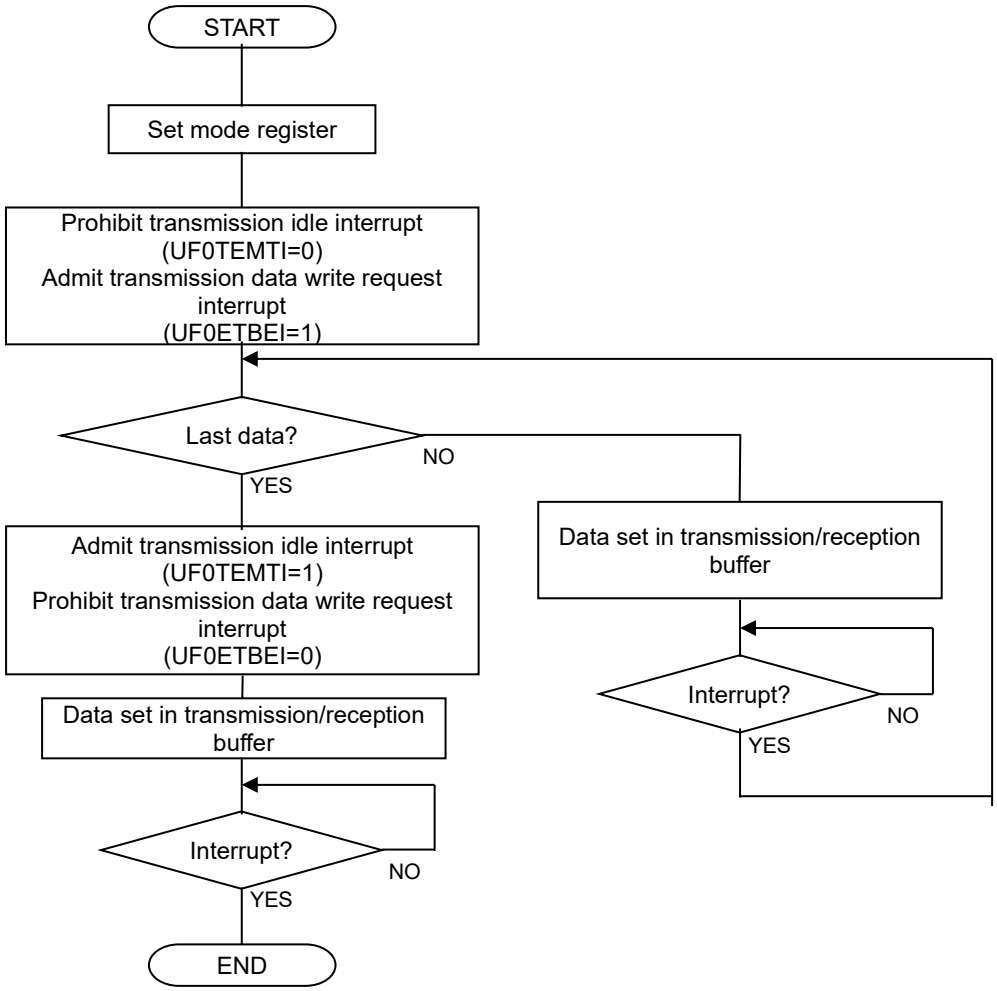


Figure18-3 Transmission Flowchart

18.3.3 Data Reception

Figure18-5 shows the reception timing. Figure18-6 shows the timing when the first byte in the receive FIFO is read, and Figure18-7 the reception timing when the remaining bytes in the receive FIFO are read.

The sampling clock is obtained by dividing the baud rate clock by 8.
First, when the start bit is detected from RXDF0, subsequent data is obtained and transferred to the receive shift register. The data in the receive shift register is transferred to RBR through the receive FIFO.
When the data reaches RBR, UF0DR of UAF0LSR is set to "1" to indicate there is valid data in RBR. This bit is cleared by reading the data in RBR.

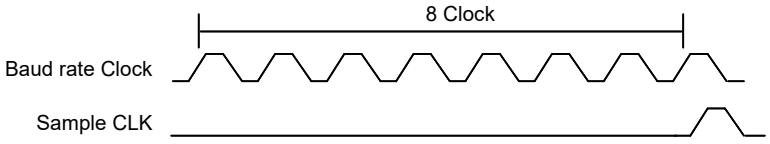


Figure18-4 Relation between Baud Rate Clock and Sample CLK

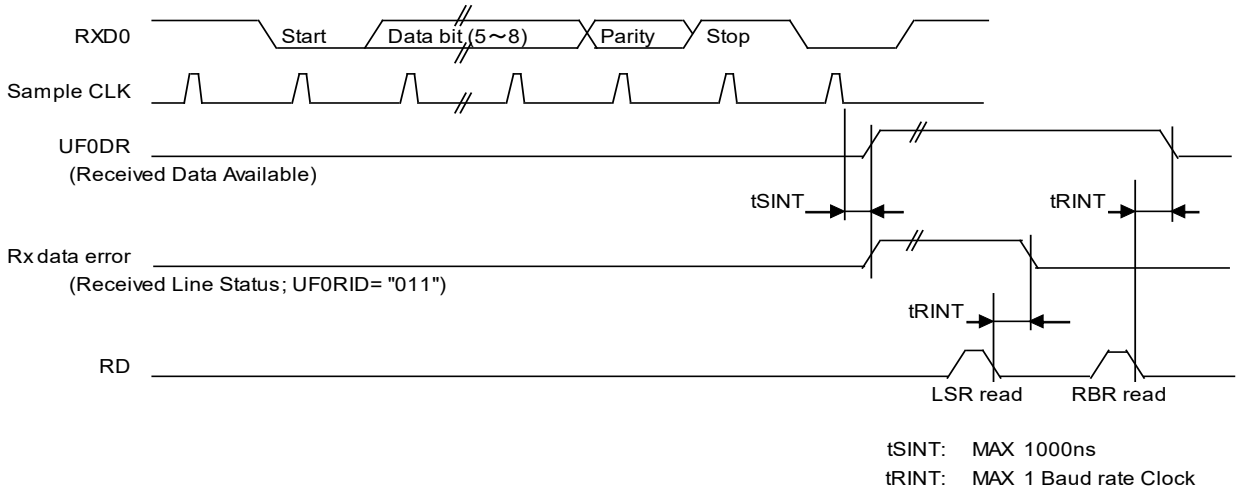


Figure18-5 Reception Timing

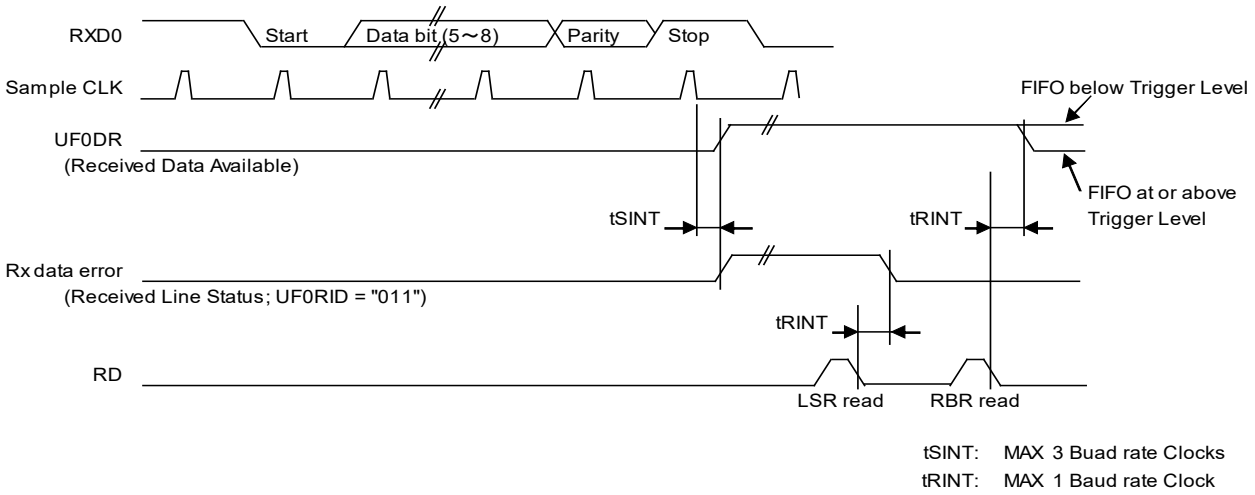


Figure18-6 First Data of Receive FIFO (Set RBR)

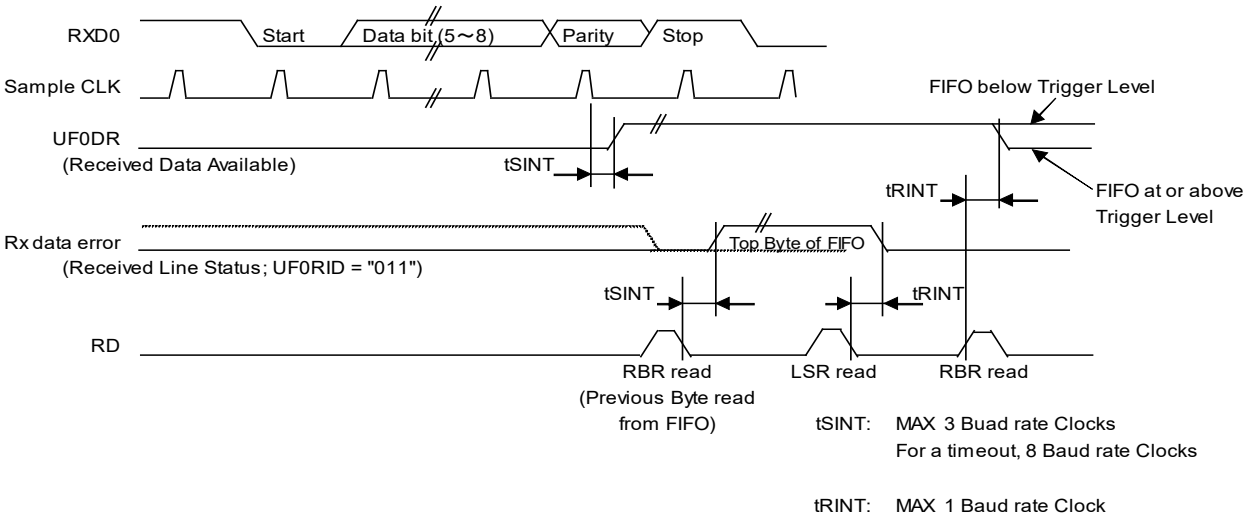


Figure18-7 Remaining Data in Receive FIFO

18.3.4 Baud Rate Clock Generation

A baud rate is obtained by the following expression with adjustment mode (UF0RMV=0):

$$\text{Baud rate frequency} = \frac{\text{HSCLK}}{\text{DLR}[15:0] \times 16} \times \frac{(\text{UAF0CAJ} - 1)}{\text{UAF0CAJ}}$$

The expression without adjustment mode (UF0RMV=1) is

$$\text{Baud rate frequency} = \frac{\text{HSCLK}}{\text{DLR}[15:0] \times 16}$$

[Note]

- **It is prohibited to set the divisor (DLR[15:0]) to 0x0 or 0x1.**

The following table shows the relation among HSCLK, DLR, and baud rates.

Table18-4 Baud Rate

Baud rate (bps)	HSCLK	UF0CAJ	DLR	Error *1 (%)
2400	20MHz	0x0D	0x01E1	-0.048
4800			0x00F0	+0.16
9600			0x0078	+0.16
19200			0x003C	+0.16
38400			0x001E	+0.16
57600			0x0014	+0.16
115200			0x000A	+0.16
2400	40MHz	0x1F	0x03F0	+0.006
4800			0x01F8	+0.006
9600			0x00FC	+0.006
19200			0x007E	+0.006
38400			0x003F	+0.006
57600			0x002A	+0.006
115200			0x0015	+0.006
2400	24.00256MHz	0x00	0x0271	+0.011
4800			0x0139	-0.149
9600			0x009C	+0.171
19200			0x004E	+0.171
38400			0x0027	+0.171
57600			0x001A	+0.171
115200			0x000D	+0.171
2400	48.00512MHz	0x19	0x04B0	+0.011
4800			0x0258	+0.011
9600			0x012C	+0.011
19200			0x0096	+0.011
38400			0x004B	+0.011
57600			0x0032	+0.011
115200			0x0019	+0.011

*1: The error does not include the clock error. User this in consideration of an error of HSCLK.

18.3.5 FIFO Mode

When the receive FIFO and reception interrupt are both enabled, reception interrupts are generated as follows:

- If the number of characters present within the FIFO exceeds the programmed trigger level, a received data read request interrupt is generated. This interrupt is immediately cleared to “0” when the number of characters present within the FIFO drops below the trigger level.
- As with the received data read request interrupt, the UAF0IIR received data read request display is set to “1” if the number of characters present within the FIFO exceeds the trigger level, and cleared to “0” if it drops below the trigger level.
- The received data error interrupt has a higher priority than the received data read request interrupt.
- The received data read request flag is set to “1” as soon as the data is transferred from the receive shift register to the FIFO, and cleared to "0" when the FIFO becomes empty.

When the receive FIFO and the reception interrupt are both enabled, a character timeout interrupt is generated as follows.

- A character timeout interrupt is generated when the following conditions are met.
  - There is at least one character present in the FIFO.
  - An amount of time required to transfer at least 4 characters has elapsed since a character was last received (if 2 stop bit are specified, the time after the first stop bit is calculated).
  - An amount of time required to transfer at least 4 characters has elapsed since the receive FIFO was last read.

  For example, if 1 start bit + 8 character bit + 1 parity bit + 2 stop bit is specified, and the transfer speed is 2400bps, the said amount of time will be approximately 20 ms.
- HSCLK is used to calculate the character time.
- When a character is read out from the FIFO, the character timeout interrupt and the timer used for timeout detection will be cleared.
- When no character timeout interrupt is generated, the timeout detection timer will be cleared when a character is read out from the FIFO or a new character is received.

The transmission interrupt is generated as follows when the transmitter section and the transmit FIFO interrupts have been enabled.

- If the transmit FIFO is empty, a transmitted data write request interrupt occurs. This interrupt is cleared when a character is written to the transmit FIFO or when UAF0IIR is read out.
- When the following conditions are met, the transmitted data write request interrupt will be delayed for an amount of time equivalent to “time required to transmit one character – time when last stop bit occurred”.
  - There was a period when only one character was present in the FIFO after UF0THRE (transmitted data write request) was last set to “1”.
  - UF0THRE was set to “1”.

[Note]

- **Even if transmit FIFO is an empty state, there are some cases that all transmit processing doesn’t complete. Confirm that it became the transmission idle state in UF0TIDL bit of UAF0LSR register before stopping high-speed clock (Transition to mode such as STOP). Or confirm that a transmission idle interrupt occurred.**

18.3.6 FIFO Polled Mode

If FIFO is enabled and UF0ELSI, UF0ETBEI, UF0TEMTI and UF0ERBFI of UAF0IER are “0”, the UART operates in the FIFO polled mode. Since the receiver section and transmitter section can be controlled separately, either one (or both) can be set to FIFO polled mode. In FIFO polled mode, the states of the receiver and transmitter sections must be checked by reading out the UAF0LSR (since no interrupt is generated).

- A state in which at least one character is present in the receive FIFO can be confirmed by the value “1” set to UF0DR.
- When UF0PER is cleared to “0”, an interrupt will not be generated even if an error is detected while receiving a character. The error state will not be indicated on the UAF0IIR value. Therefore, the error type must be checked with the UF0BI, UF0FER, UF0PER, and UF0OER values.
- It can be known that the transmit FIFO is empty by the fact that UF0THRE has been set to “1”.
- A state in which the transmit FIFO and transmit shift register are both empty can be confirmed by the value "1" set to UF0TEMT.
- A state in which the character associated with an error at the time of reception is present in the receive FIFO can be confirmed by the value “1” set to UF0RFE.
- UF0TIDL is set to “1”, can know that it is a transmission idle.

In FIFO polled mode, FIFO will operate; however, trigger level and timeout detection will not be performed (since they are only notified by interrupts).

18.3.7 Error Status

1. Overrun error
   An overrun error indicates that the data in RBR was not read before the next character was sent to RBR to overwrite the previous character.
   At this time, UF0OER of UAF0LSR is set to “1”.

2. Parity error
   A parity error indicates that the parity of the received data and the received parity bit did not match. At this time, UF0PER of UAF0LSR is set to “1”.
   Note that, this error will only occur when parity is enabled.
   In FIFO mode, this bit indicates that an error exists for the leading data. If a parity error occurs in the data that is not the leading data in the FIFO, it is not reflected to UF0PER of UAF0LSR.

3. Framing error
   A framing error indicates that there is no valid stop bit in the received character. This error will occur when the stop bit after the last data bit or after the parity bit is “0” (spacing level).
   At this time, UF0FER of UAF0LSR is set to “1”.
   In FIFO mode, this is related to a specific character in the FIFO. UF0FER indicates that an error is present when that character comes to the beginning of the FIFO.

4. Break interrupt
   A break interrupt indicates that the input data received was maintained in the spacing (“0”) state during the transmission of one frame (start bit + data bit + parity bit + stop bit).
   At this time, UF0BI of the UAF0LSR register is set to “1”.
   In FIFO mode, this is related to a specific character in the FIFO. UF0BI indicates that the break character is present at the beginning of the FIFO.

Chapter 19

CAN FD Controller

19. CAN FD Controller

19.1 Overview

CAN FD controller communicates according to ISO 11898-1:2015. CAN FD and classic CAN communication can be supported. For connection to the CAN bus, an external CAN transceiver is required.
CAN FD controller has the following features:

- Conform with ISO 11898-1:2015
- CAN FD with up to 64 data bytes supported
- CAN error logging
- SAE J1939 support
- Improved acceptance filtering
- Two configurable receive FIFOs (Rx FIFOs)
- Interrupt output when a high-priority message is received
- Up to 64 specific ID receive buffers (Rx buffers)
- Up to 32 specific ID transmit buffers (Tx buffers)
- Configurable transmit FIFO (Tx FIFO)
- Configurable transmit queue (Tx queue)
- Configurable transmit event FIFO (Tx event FIFO)
- Direct access to message RAM from CPU
- Programmable loopback test mode
- Maskable interrupt output
- Power down support

19.1.1 Configuration

19.1.1.1 Block Configuration

Figure19-1 shows the block configuration of the CAN FD Controller.

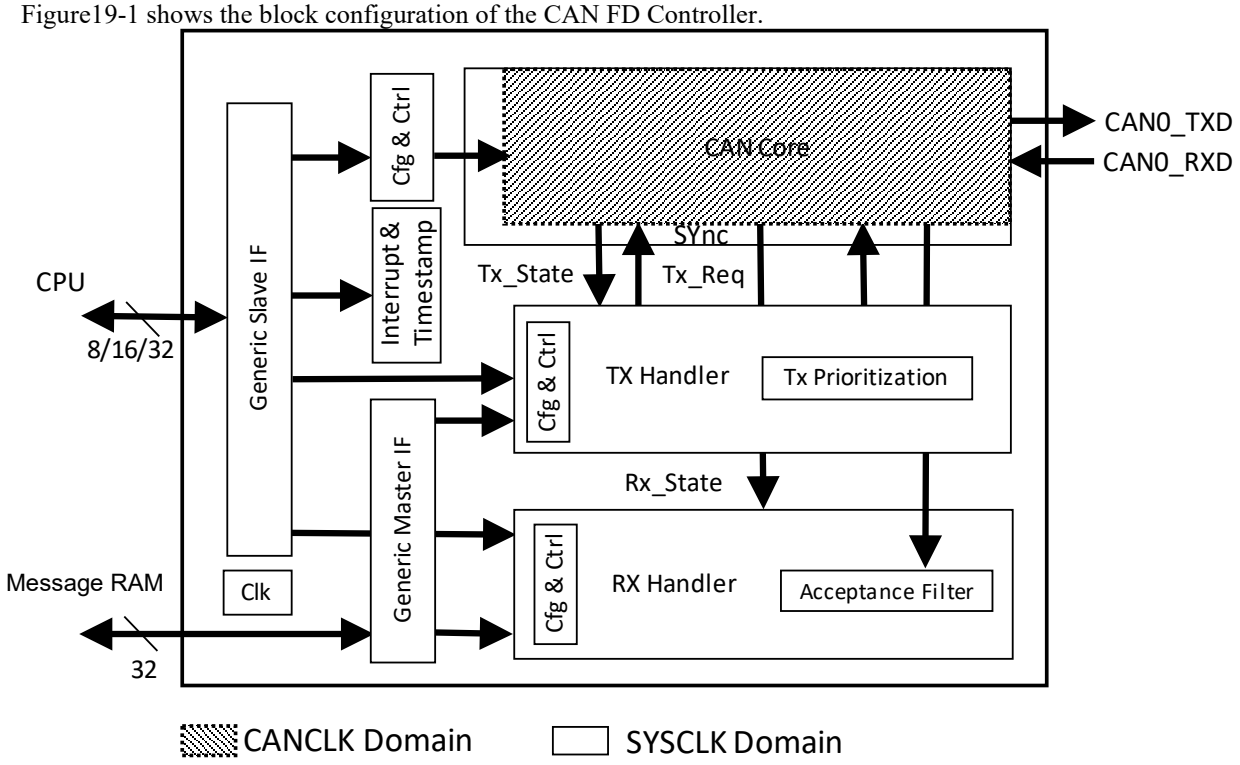


Figure19-1 Block Configuration of the CAN FD Controller

Table19-1 Pins

Fuctional pin name	I/O	LSI pin name
CAN_RXD0	I	P20
CAN_TXD0	O	P21

The functions of each block that constitutes CAN FD controller are shown below.

- CAN Core: Consists of CAN protocol controller and Rx/Tx shift registers. Handles all ISO 11898-1:2015 protocol functions. Supports 11-bit and 29-bit identifiers.
- Synchronization (Sync): Synchronizes signals from SYSCLK domain to CANCLK domain and vice versa.
- Clock (Clk): Synchronizes the reset signal to the SYSCLK and CANCLK domains.
- Configuration & Control (Cfg & Ctrl): Consists of configuration and control registers related to CAN core.
- Interrupt & Timestamp: Consists of interrupt control and a 16-bit CAN bit time counter for receive and transmit timestamp generation.
- Tx Handler: Controls the message transfer from message RAM to the CAN core. Up to 32 Tx buffers can be configured for transfer. A Tx buffer can be used as a dedicated Tx buffer, as a Tx FIFO, as part of a Tx queue, or as a combination of these. The Tx event FIFO stores the Tx timestamp along with the corresponding message ID. Also supports transmit cancellation.
- Rx Handler: Controls the transfer of incoming messages from CAN core to message RAM. The Rx handler supports two Rx FIFOs of configurable size and up to 64 dedicated Rx buffers to store all messages that pass the acceptance filtering. The dedicated Rx buffer is different from the Rx FIFO and is used to store only messages with a specific identifier. The Rx timestamp is stored with each message. Up to 128 filters can be defined for 11-bit IDs, and up to 64 filters can be defined for 29-bit IDs.
- Generic Slave IF (interface): Connects CAN FD controller to the CPU. The generic slave interface can support 8/16/32-bit access.
- Generic Master IF (interface): Connects CAN FD controller access to message RAM. 32-bit access is available.

19.1.1.2 Clock

CAN FD controller operates with two clocks, SYSCLK, which is a operation clock for the CPU and bus, and CANCLK, which is the communication clock for CAN.
The frequency profile of these two clocks must satisfy the frequency of SYSCLK ≥ the frequency of CANCLK.
A synchronization mechanism is in CAN FD controller to ensure data transfer between the two clock domains.

[Note]

- **In order to achieve stable functioning of CAN FD Controller, the frequency of SYSCLK must be faster or equal to the frequency of CANCLK at all times. Table19-2 shows an example of recommended settings for SYSCLK and CANCLK.**

Table19-2 Example of Recommended Settings for SYSCLK and CANCLK

CAN communication speed [bps]	SYSCLK			CANCLK		
	Frequency [MHz]	FCON01 register		Frequency [MHz]	SYSCANMOD register	
		OSCM setting value	SYSC setting value		{HXTSEL, HSSEL} setting value	DIV setting value
500K	12	00 (PLL)	010 (1/4 frequency)	12	01 (HSCLK)	001 (1/1 frequency)
500K	24	00 (PLL)	001 (1/2 frequency)	12	01 (HSCLK)	010 (1/2 frequency)
1M	24	00 (PLL)	001 (1/2 frequency)	24	01 (HSCLK)	001 (1/1 frequency)
1M	48	00 (PLL)	000 (1/1 frequency)	24	01 (HSCLK)	010 (1/2 frequency)
2M	20	01 (HXT)	000 (1/1 frequency)	20	01 (HSCLK)	001 (1/1 frequency)
2M	24	00 (PLL)	001 (1/2 frequency)	20	10 (HXTCLK)	001 (1/1 frequency)
2M	48	00 (PLL)	000 (1/1 frequency)	20	10 (HXTCLK)	001 (1/1 frequency)
5M	40	01 (HXT)	000 (1/1 frequency)	40	01 (HSCLK)	001 (1/1 frequency)
5M	48	00 (PLL)	000 (1/1 frequency)	40	10 (HXTCLK)	001 (1/1 frequency)

[Note]

- **CANCLK, due to ISO 11898-1:2015, has required accuracy conditions**

The tolerance of the reference frequency ($fnom$) for the CANCLK frequency ($fosc$) is given by the following equation.

$fnom$: Reference frequency $fosc$: CANCLK frequency df: CANCLK frequency tolerance

$$(1 - df) \times fnom \leq fosc \leq (1 + df) \times fnom$$

$$df < \frac{\mathrm{SJW}(N)}{2 \times 10 \times \mathrm{bit\ time}(N)} \quad (1)$$

$$df < \frac{\min[\mathrm{Phase\_Seg1}(N), \mathrm{Phase\_Seg2}(N)]}{2 \times [13 \times \mathrm{bit\ time}(N) - \mathrm{Phase\_Seg2}(N)]} \quad (2)$$

$$df < \frac{\mathrm{SJW}(D)}{2 \times 10 \times \mathrm{bit\ time}(D)} \quad (3)$$

$$df < \frac{\min[\mathrm{Phase\_Seg1}(N), \mathrm{Phase\_Seg2}(N)]}{2 \times \left([6 \times \mathrm{bit\ time}(D) - \mathrm{Phase\_Seg2}(D)] \times \frac{m(D)}{m(N)} + 7 \times \mathrm{bit\ time}(N) \right)} \quad (4)$$

$$df < \frac{\mathrm{SJW(D)} - \max\left(0, \frac{m(N)}{m(D)} - 1\right)}{2 \times \left([2 \times \mathrm{bit\ time}(N) - \mathrm{Phase\_Seg2}(N)] \times \frac{m(N)}{m(D)} + \mathrm{Phase\_Seg2}(D) + 4 \times \mathrm{bit\ time}(D) \right)} \quad (5)$$

m: prescaler
SJW: synchronous jump width
Phase_Seg1: Phasebuffer segment 1
Phase_Seg2: Phasebuffer segment 2
bit time: 1 bit time

(N): Indicates the value of arbitration phase. (D): Indicates the value of the data phase.

For Classical CAN: (1) and (2) must be satisfied.
For CAN FD: (1), (2), (3), (4) and (5) must be satisfied.

19.1.1.3 Interrupt

CAN FD controller has two interrupts. Each interrupt source generated by CAN FD controller can be assigned to either one of two interrupts (CAN0_INT0 or CAN0_INT1). In the initialization after reset, all interrupt factors are assigned to CAN0_INT0. By setting EINT0 bit of the CANILE register and EINT1 bit of the CANILE register, each interrupt of CAN0_INT0 and CAN0_INT1 can be enabled or disabled individually.

19.2 Description of Registers

19.2.1 Descriptions of Resets

After reset, each register of CAN FD controller retains its initial value as shown in Table19-3. Additionally, the Bus_Off state is released and the CAN0_TXD pin is set to recessive (High). The value 0x0001 in the CANCCCR register (INIT bit in the CANCCCR register = 1) enables software initialization. CAN FD controller does not affect the CAN bus until the software resets the INIT bit in the CANCCCR register to 0.

19.2.2 List of Registers

CAN FD controller allocates 256-byte address space. All registers are configured as 32-bit registers. CAN FD controller can be accessed by software via the generic slave interface of CAN FD Controller using a data width of 8 bits (byte access), 16 bits (half-word access) or 32 bits (word access). Software write access to the registers/bits marked "P = Protected Write" is only possible with CCE bit in the CANCCCR register = 1 and INIT bit in the CANCCCR register = 1. Due to the crossover between the SYSCLK and CANCLK clock domains, there is a delay between writing on the command register and updating of the associated status register bits.

Table19-3 List of Registers Register List of CAN FD controller

Address/Offset	Name	Symbol name	R/W	Initial value
0x4701000	CAN FD controller base address	-	-	-
0x0	Version number register	CANCREL	R	0x33081114
0x4	Endian test register	CANENDN	R	0x87654321
0x8	Reserved	-	R	0x00000000
0xC	FD bit timing register	CANDBTP	RP	0x00000A33
0x10	Test register	CANTEST	RP	0x00000080
0x14	RAM watchdog register	CANRWD	RP	0x00000000
0x18	CAN FD controller control register	CANCCCR	R/WPp	0x00000001
0x1C	Bit timing register	CANNBTP	RP	0x06000A03
0x20	Time stamp counter setting register	CANTSCC	RP	0x00000000
0x24	Time stamp counter value register	CANTSCV	RC	0x00000000
0x28	Timeout sounter setting register	CANTOCC	RP	0xFFFF0000
0x2C	Timeout counter value register	CANTOCV	RC	0x0000FFFF
0x30	Reserved	-	R	0x00000000
0x34	Reserved	-	R	0x00000000
0x38	Reserved	-	R	0x00000000
0x3C	Reserved	-	R	0x00000000
0x40	Error counter register	CANECR	RX	0x00000000
0x44	Protocol status register	CANPSR	RXS	0x00000707
0x48	Transmitter delay correction register	CANTDCR	RP	0x00000000
0x4C	Reserved	-	R	0x00000000
0x50	Interrupt register	CANIR	R/W	0x00000000
0x54	Interrupt enable register	CANIE	R/W	0x00000000
0x58	Interrupt signal selection register	CANILS	R/W	0x00000000
0x5C	Interrupt signal enable register	CANILE	R/W	0x00000000
0x60	Reserved	-	R	0x00000000
0x64	Reserved	-	R	0x00000000
0x68	Reserved	-	R	0x00000000
0x6C	Reserved	-	R	0x00000000
0x70	Reserved	-	R	0x00000000
0x74	Reserved	-	R	0x00000000
0x78	Reserved	-	R	0x00000000
0x7C	Reserved	-	R	0x00000000
0x80	Global filter configuration register	CANGFC	RP	0x00000000
0x84	Standard ID filter configuration register	CANSIDFC	RP	0x00000000
0x88	Extended ID filter configuration register	CANXIDFC	RP	0x00000000
0x8C	Reserved	-	R	0x00000000
0x90	Extended ID mask register	CANXIDAM	RP	0x1FFFFFFF
0x94	High priority message status register	CANHPMS	R	0x00000000
0x98	Specific ID receive message 1 register	CANNDAT1	R/W	0x00000000

Address/Offset	Name	Symbol name	R/W	Initial value
0x9C	Specific ID receive message 2 register	CANNDAT2	R/W	0x00000000
0xA0	Rx FIFO 0 setting register	CANRXF0C	RP	0x00000000
0xA4	Rx FIFO 0 status register	CANRXF0S	R	0x00000000
0xA8	Rx FIFO 0 acknowledge register	CANRXF0A	R/W	0x00000000
0xAC	Rx buffer configuration register	CANRXBC	RP	0x00000000
0xB0	Rx FIFO 1 configuration register	CANRXF1C	RP	0x00000000
0xB4	Rx FIFO 1 status register	CANRXF1S	R	0x00000000
0xB8	Rx FIFO 1 acknowledge register	CANRXF1A	R/W	0x00000000
0xBC	Rx buffer/FIFO element size setting register	CANRXESC	RP	0x00000000
0xC0	Tx buffer setting register	CANTXBC	RP	0x00000000
0xC4	Tx FIFO/Cue status register	CANTXFQS	R	0x00000000
0xC8	Tx buffer element size setting register	CANTXESC	RP	0x00000000
0xCC	Tx buffer request pending register	CANTXBRP	R	0x00000000
0xD0	Tx buffer additional request register	CANTXBAR	R/W	0x00000000
0xD4	Tx buffer cancellation request register	CANTXBCR	R/W	0x00000000
0xD8	Tx buffer transmission generation register	CANTXBTO	R	0x00000000
0xDC	Tx buffer cancellation completion register	CANTXBCF	R	0x00000000
0xE0	Tx buffer transmit interrupt enable register	CANTXBTIE	R/W	0x00000000
0xE4	Tx buffer cancellation completion interrupt register	CANTXBCIE	R/W	0x00000000
0xE8	Reserved	-	R	0x00000000
0xEC	Reserved	-	R	0x00000000
0xF0	Tx event FIFO setting register	CANTXEFC	RP	0x00000000
0xF4	Tx event FIFO status register	CANTXEFS	R	0x00000000
0xF8	Tx event FIFO acknowledge register	CANTXEFA	R/W	0x00000000
0xFC	Reserved	-	R	0x00000000

[Note]

- **R = Read, S = Set on read, X = Reset on read, W = Write, P = Protected write, p = Protected set, C = Clear/preset on write, r = release, d = date**

19.2.3 Access to reserved register addresses

If the software accesses a reserved address in CAN FD Controller register map, the write is ignored and the read is “0”.

19.2.4 Version Number Register (CANCREL)

Offset: 0x0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	REL				STEP				SUBSTEP				YEAR			
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MON								DAY							
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0

Bit No	Bit name	Description
31 - 28	REL[3:0]	Core Release A single digit number represented by BCD coding.
27 - 24	STEP[3:0]	Step of Core Release A single digit number represented by BCD coding.
23 - 20	SUBSTEP[3:0]	Sub-step of Core Release A single digit number represented by BCD coding.
19 - 16	YEAR[3:0]	Time Stamp Year A single digit number represented by BCD coding.
16 - 8	MON[7:0]	Time Stamp Month A two-digit number represented by BCD coding.
7 - 0	DAY[7:0]	Time Stamp Day A two-digit number represented by BCD coding.

19.2.5 Endian Test Register (CANENDN)

Offset: 0x4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETV															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETV															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1

Bit No	Bit name	Description
31 - 0	ETV[31:0]	Endianness Test Value The test value for endian is 0x87654321.

19.2.6 FD Bit Timing Register (CANDBTP)

Offset: 0xC

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	TDC	-	-	DBRP				
R/W	R	R	R	R	R	R	R	R	RP	R	R	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	DTSEG1					DTSEG2				DSJW			
R/W	R	R	R	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	1

This register can be written only when the CCE bit of the CANCCCR register and the INIT bit of the CANCCCR register have been set. The CAN bit time can be programmed within the range of 4 - 49 time quanta. The CAN time quantum can be programmed within the range 1 - 32 CANCLK periods.

tq = (DBRP + 1) mtq.

Where is, mtq: minimum time quantum = CANCLK, tq: time quantum.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.

Therefore, the bit time length is (program value) [DTSEG1 + DTSEG2 + 3] tq or (function value) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.

Information Processing Time (IPT) is zero. This means that the next bit of data is available at the first clock edge after the sample point.

Bit No	Bit name	Description
31 - 24	-	Reserved bits
23	TDC	Transmitter Delay Compensation 0: Transmitter delay compensation is disabled 1: Transmitter delay compensation is enabled
22 - 21	-	Reserved bits
20 - 16	DBRP[4:0]	Data Bit Rate Prescaler The value whereby the oscillator frequency is divided to generate the bit-time quanta. Bit times are constructed from multiples of this quanta. Valid values for the bitrate prescaler are 0 - 31. If TDC = 1, the range is restricted to 0, 1. The actual hardware value used is this setting value + 1.
15 - 13	-	Reserved bits
12 - 8	DTSEG1[4:0]	Data time segment before sample point Valid values are 0 - 31. The actual hardware value used is this setting value + 1.
7 - 4	DTSEG2[3:0]	Data time segment after sample point Valid values are 0 - 15. The actual hardware value used is this setting value + 1.
3 - 0	DSJW[3:0]	Data (Re) Synchronization Jump Width Valid values are 0 - 15. The actual hardware value used is this setting value + 1.

[Note]

- **With CAN clock of 8 MHz (CANCLK), a reset value of 0x0000_0A33 configures CAN FD controller for a data phase bit rate of 500 kBit/s.**
- **The bit rate set for the CAN FD data phase via CANDBTP must be greater than or equal to the bit rate set for the arbitration phase via CANNBTP.**

19.2.7 Test Register (CANTEST)

Offset: 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PVAL	TXBNP					RX	TX		LBCK	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	RP	RP	RP	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

The TEST bit in the CANCCCR register must be set to 1 to enable write access to the test register. When the TEST bit in the CANCCCR register is reset, all functions of the test register are set to their reset values.
The loopback mode and the software control of the CAN0_TXD pin are hardware test modes. A setting of TX≠00 may interfere with message transfer on the CAN bus.

Bit No	Bit name	Description
31 - 22	-	Reserved bits
21	SVAL	Start Valid 0: TXBNS value is invalid 1: TXBNS value is valid
20 - 16	TXBNS[4:0]	Tx Buffer Number Started Tx buffer number of the message for which transmission was initiated last. This is valid when SVAL is set. Valid values are 0 - 31.
15 - 14	-	Reserved bits
13	PVAL	Prepared Valid 0: TXBNP value is invalid 1: TXBNP value is valid
12 - 8	TXBNP[4:0]	Tx Buffer Number Prepared Tx buffer number of the message that is ready to be transmitted. This is valid when PVAL is set. Valid values are 0 - 31.
7	RX	Receive Pin The actual value of the CAN0_RXD pin is monitored. Before the CAN0_RXD pin is set to the port n-order function, “1” is read; after the pin is set to the port n-order function, the pin status is read. For details of the port n-order function settings, see “1.3.2 List of Pins” and Chapter 22 “”General Purpose Port (GPIO) “. 0: CAN bus is dominant level (CAN0_RXD = “0”) 1: CAN bus is at recessive level (CAN0_RXD = “1”)
6 - 5	TX[1:0]	Control of Transmit Pin Controls the CAN0_TXD pin. 00: Reset value, the CAN0_TXD pin is controlled by the CAN core and updated at the end of the CAN bit time 01: Sample points can be monitored on the CAN0_TXD pin 10: Dominant level is output on the CAN0_TXD pin (CAN0_TXD = “0”). 11: Recessive level is output on the CAN0_TXD pin (CAN0_TXD = “1”)
4	LBCK	Loop Back Mode 0: Reset value, and Loop Back Mode is disabled 1: Loop Back Mode is enabled (see section “19.3.2.9”)
3 - 0	-	Reserved bits

19.2.8 RAM Watchdog Register (CANRWD)

Offset: 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDV								WDC							
R/W	R	R	R	R	R	R	R	R	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RAM watchdog monitors the output of the message RAM. Message RAM access via the generic master interface of CAN FD Controller starts the message RAM watchdog counter with the value set by the WDC bit in the CANRWD register. When the message RAM notifies that it has successfully completed the access, the counter is reloaded with the WDC bit in the CANRWD register. In case there is no response from the message RAM until the counter counts down to zero, the counter stops, and the WDI bit in the CANIR register is set. RAM watchdog counter is synchronized with SYSCLK.

Bit No	Bit name	Description
31 - 16	-	Reserved bits
15 - 8	WDV[7:0]	Watchdog Value Actual message RAM watchdog counter value
7 - 0	WDC[7:0]	Watchdog Configuration Start value of the message RAM watchdog counter. When the value is “0”, the counter is disabled.

19.2.9 CAN FD Controller Control Register (CANCCCR)

Offset: 0x18

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TXP	EFBI	PXHD	WMM	-	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W	R	RP	RP	RP	RP	R	RP	RP	Rp	RP	Rp	R/W	R	Rp	RP	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

For more details on setting and resetting the INIT and CCE bits, see section “19.3.2.1”

Bit No	Bit name	Description
31 - 15	-	Reserved bits
14	TXP	Transmit Pause When this bit is set, CAN FD controller will pause for 2 CAN bits time after successfully transmitting a frame, before starting the next transmission (see section “19.3.6”). 0: Transmission pause is disabled 1: Transmission pause is enabled
13	EFBI	Edge Filtering during Bus Integration 0: Edge filtering is disabled 1: Two consecutive dominant tq required to detect the edge of hardware synchronization
12	PXHD	Protocol Exception Handling Disable 0: Protocol exception handling is enabled 1: Protocol exception handling is disabled In case protocol exception handling is disabled, CAN FD controller will transmit an error frame when it detects a protocol exception condition.
11	WMM	Wide Message Marker Enables the use of 16-bit wide message marker In case the 16-bit wide message marker is used (WMM = 1), the 16-bit internal timestamp in the Tx event FIFO is disabled. 0: Uses 8-bit message marker 1: Uses 16-bit message marker, replaces 16-bit timestamp in Tx event FIFO
10	-	Reserved bit
9	BRSE	Bit Rate Switch Enable 0: Bit rate switching for transmission is disabled 1: Bit rate switching for transmission is enabled When the CAN FD operation has been disabled by FDOE = “0”, BRSE is not evaluated.
8	FDOE	FD Operation Enable 0: FD operation is disabled 1: FD operation is enabled
7	TEST	Test Mode Enable 0: Normal operation, CANTEST register retains the reset value 1: Test mode, write access to the CANTEST register is enabled
6	DAR	Disable Automatic Retransmission 0: Automatic retransmission of messages that were not transmitted successfully is enabled. 1: Automatic retransmission is disabled
5	MON	Bus Monitoring Mode The MON bit can be set by the software only when both the CCE and INIT bits are set to “1”. This bit can be reset by the software at any time. 0: Bus monitoring mode is disabled 1: Bus monitoring mode is enabled
4	CSR	Clock Stop Request 0: Clock stop will not requested 1: Clock stop has been requested. When clock stop is requested, the INIT bit is set first, then the CSA bit, after all pending transfer requests have been completed and the CAN bus is idle.

Bit No	Bit name	Description
3	CSA	Clock Stop Acknowledge 0: Clock stop will not be acknowledged 1: CAN FD controller can be set to Power Down by stopping SYSCLK and CANCLK
2	ASM	Restricted Operation Mode The ASM bit can be set by the software only when both the CCE and INIT bits are set to “1”. This bit can be reset by the software at any time. For more details of Restricted Operation Mode, see section “19.3.2.5”. 0: Normal CAN operation 1: Restricted operation mode is active
1	CCE	Configuration Change Enable 0: The software does not have the write access to the protected configuration registers 1: The software has the write access to the protected configuration registers (in case INIT bit in the CANCCCR register = “1”)
0	INIT	Initialization 0 = Normal operation 1 = Initialization has started Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT is read back. Therefore, the software must verify that the previous value written to INIT has been accepted by reading INIT before setting it to the new value.

19.2.10 Bit Timing Register (CANNBTP)

Offset: 0x1C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NSJW							NBRP								
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NTSEG1								-	NTSEG2						
R/W	RP	RP	RP	RP	RP	RP	RP	RP	R	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1

This register can be written only when the CCE bit of the CANCCCR register and the INIT bit of the CANCCCR register have been set. CAN bit time can be programmed within the range of 4 - 385 time quanta. CAN time quantum can be programmed within the range of 1 - 512 CANCLK periods. tq = (NBRP + 1) mtq.
NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.
Therefore, the bit time length is (program value) [NTSEG1 + NTSEG2 + 3] tq or (function value) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.
Information Processing Time (IPT) is zero. This means that the next bit of data is available at the first clock edge after the sample point.

Bit No	Bit name	Description
31 - 16	NSJW[6:0]	Nominal (Re)Synchronization Jump Width Valid values are 0 - 127. The actual hardware value used is this setting value + 1.
24 - 16	NBRP[8:0]	Nominal Bit Rate Prescaler Value whereby the oscillator frequency is divided to generate the bit-time quantium. Bit times are constructed from multiples of this quantium. Valid values for the bit rate prescaler are 0 - 511. The actual hardware value used is this setting value + 1.
15 - 8	NTSEG1[7:0]	Nominal Time segment before sample point Valid values are 1 - 255. The actual hardware value used is this setting value + 1.
7	-	Reserved bit
6 - 0	NTSEG2[6:0]	Nominal Time segment after sample point Valid values are 1 - 127. The actual hardware value used is this setting value + 1. With CAN clock of 8 MHz (CANCLK), a reset value of 0x06000A03 sets CAN FD controller to a bit rate of 500 kBit/s.

19.2.11 Time Stamp Counter Setting Register (CANTSCC)

Offset: 0x20

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TSS	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is the setting register for the internal 16-bit time stamp counter. The handling of internal timestamp is provided in 19.3.3.

Bit No	Bit name	Description
31 – 20	-	Reserved bits
19 - 16	TCP[3:0]	Timestamp Counter Prescaler Sets the time unit for the timestamp and timeout counter in multiples of the CAN bit time [1...16]. The actual hardware value used is this setting value + 1.
15 - 2	-	Reserved bits
1 - 0	TSS[1:0]	Timestamp Select 00: Timestamp counter value is 0x0000 at all times 01: The timestamp counter value is incremented according to TCP [3:0] bit 10: This setting is prohibited, and there is no guarantee for the operation. 11: Same as “00”

19.2.12 Timestamp Counter Value Register (CANTSCV)

Offset: 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSC															
R/W	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 16	-	Reserved bits
15 - 0	TSC[15:0]	Timestamp Counter The internal timestamp counter value is captured at the start of the frame (both Rx and Tx). When TSS bit in the CANTSCC register = 01, the timestamp counter is incremented in multiples of the CAN bit time [1...16] according to the setting of the TCP bit in the CANTSCC register. A wrap around sets the TSW bit in the CANIR register. A write access resets the counter to 0. A "wrap around" is a change in the value of the timestamp counter from non-zero to zero, not due to a write access to CANTSCV. For byte access, writing one of the register bytes 3/2/1/0 resets the timestamp counter.

19.2.13 Timeout Counter Setting Register (CANTOCC)

Offset: 0x28

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOP															
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	TOS		ETOC
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 16	TOP[15:0]	Timeout Period Start value of the timeout counter (down counter). Sets the timeout period.
15 - 3	-	Reserved bits
2 - 1	TOS[1:0]	Timeout Select When operating in continuous mode, writing on the CANTOCV register presets the counter to the value set by the TOP bit of the CANTOCC register and continues down-counting. When the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value set by the TOP bit of the CANTOCC register. Once the first FIFO element is saved, the down-counting begins. 00: Continuous start 01: Timeout is start-controlled by Tx event FIFO 10: Timeout is start-controlled by Rx FIFO 0 11: Timeout is start-controlled by Rx FIFO 1
0	ETOC	Enable Timeout Counter 0: Timeout counter is disabled 1: Timeout counter is enabled

[Note]

- **For information on using the timeout function with CAN FD, see section “19.3.4”.**

19.2.14 Timeout Counter Value Register (CANTOCV)

Offset: 0x2C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOC															
R/W	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC	RC
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No	Bit name	Description
31 - 16	-	Reserved bits
15 - 0	TOC[15:0]	Timeout Counter The timeout counter decreases in multiples of the CAN bit time [1...16] according to the setting of the TCP bit in the CANTSCC register. When decremented to 0, the TOO bit of the CANIR register is set and the timeout counter stops. The start and reset/restart conditions are set via the TOS bit in the CANTOCC register. For byte accesses, when the TOS bit of the CANTOCC register = 00, writing one of the register bytes 3/2/1/0 will preset the timeout counter.

19.2.15 Error Counter register (CANECR)

Offset: 0x40

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RP	REC							TEC							
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 24	-	Reserved bits
23 - 16	CEL[7:0]	CAN Error Logging The counter increments every time the TEC [7:0] bits, which are 8-bit transmit error counters, or the REC [6:0] bits, which are 7-bit receive error counters, are incremented due to CAN protocol errors. When the Bus_Off limit is reached, the counter is also incremented. When only the RP bits are set without changing the REC [6:0] bits, they are not incremented. Incrementing the CEL [7:0] bits is followed by incrementing the REC [6:0] or TEC [7:0] bits. The counter is reset by a read access to the CEL [7:0] bits. The counter stops at 0xFF. The next increment of TEC [7:0] bits or REC [6:0] bits sets the ELO bit in the CANIR register. Byte Access: Reading byte 2 resets the CEL to zero and is not affected by reading byte 3/1/0.
15	RP	Receive Error Passive 0: The receive error counter is less than the error passive level of 128 1: The receive error counter has reached the error passive level of 128
14 - 8	REC[6:0]	Receive Error Counter The value of 0 - 127 in the actual state of the receive error counter
7 - 0	TEC[7:0]	Transmit Error Counter The value of 0 - 255 in the actual state of the transmit error counter In case the CANCCCR register ASM bit is set, the CAN protocol controller will not increment TEC [7:0] and REC [6:0] bits but will increment CEL [7:0] bits when a CAN protocol error is detected.

19.2.16 Protocol Status Register (CANPSR)

Offset: 0x44

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PXE	RFDF	RBRS	RESI	DLEC			BO	EW	EP	ACT		LEC		
R/W	R	X	X	X	X	S	S	S	R	R	R	R	R	S	S	S
Initial value	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit No	Bit name	Description
31 – 23	-	Reserved bits
22 - 16	TDCV[6:0]	Transmitter Delay Compensation Value The position of the secondary sample point Defined by the sum of the measured delays from the CAN0_TXD pin to the CAN0_RXD pin and to the TDCO bit in the CANTDCR register. In the data phase, the position of the SSP bit is the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 - 127 mtq.
15	-	Reserved bit
14	PXE	Protocol Exception Event 0: No protocol exception events have occurred since the last read access. 1: A protocol exception event has occurred. For byte accesses, reading byte 0 resets the PXE. There is no effect of reading byte 3/2/1.
13	RFDF	Received a CAN FD Message This bit is set independently of the acceptance filtering. 0: This bit has been reset by the software, and thus no CAN FD message has been received. 1: A message in CAN FD format with the FDF flag set was received. For byte accesses, reading byte 0 resets the RFDF. There is no effect of reading byte 3/2/1.
12	RBRS	BRS flag of last received CAN FD Message This bit is set in conjunction with the RFDF bit independently of the acceptance filtering. 0: The BRS flag of the last received CAN FD message is not set. 1: The BRS flag of the last received CAN FD message has been set. For byte access, reading byte 0 resets the RBRS. There is no effect of reading byte 3/2/1.
11	RESI	ESI flag of last received CAN FD Message This bit is set in conjunction with the RFDF bit independently of the acceptance filtering. 0: The ESI flag of the last received CAN FD message is not set. 1: The ESI flag of the last received CAN FD message has been set. For byte accesses, reading byte 0 resets the RESI. There is no effect of reading byte 3/2/1.
10 - 8	DLEC[2:0]	Data Phase Last Error Code Type of the last error that occurred in the data phase of a CAN FD format frame with the BRS flag set. The encoding is the same as for the LEC [2:0] bits. When a CAN FD format frame with the BRS flag set is transferred (received or transmitted) without errors, this field is cleared to 0. For byte accesses, reading byte 0 sets the DLEC to 111. There is no effect of reading byte 3/2/1.
7	BO	Bus_Off Status 0: CAN FD controller is not in Bus_Off state. 1: CAN FD controller is in Bus_Off state.
6	EW	Warning Status 0: Both error counters are below the Error_Warning limit of 96. 1: At least one error counter has reached the Error_Warning limit of 96.
5	EP	Error Passive 0: CAN FD controller is in Error_Active state Normally, participates in bus communication and transmits an active error flag when an error is detected 1: CAN FD controller is in Error_Passive state

Bit No	Bit name	Description
4 - 3	ACT[1:0]	Activity Monitors the CAN communication status of the module. 00: Indicates synchronization in progress, nodes being synchronized via CAN communication 01: Indicates idle, node being neither a receiver nor a transmitter 10: Indicates a receiver, the node being acting as a receiver 11: Transmitter node is operating as a transmitter ACT is set to 00 by a protocol exception event.
2 - 0	LEC[2:0]	Last Error Code LEC indicates the type of the last error that has occurred on the CAN bus. When a message is transferred (received or transmitted) without errors, this field is cleared to 0. 0: Indicates no error, while the LEC [2:0] bit has been reset by a normal receive or transmit, and thus no error has occurred. 1: Indicates a staff error, the occurrence of five or more equal bits in the sequence in some of the received messages not allowed. 2: Indicates a form error, an incorrect format of the fixed format portion of the received frame. 3: Indicates an Ack error, in which the message transmitted by CAN FD controller was not acknowledged by another node. 4: Indicates Bit1 error, in which, during the transmission of a message (except for the arbitration field), the device attempted to transmit a recessive level (bit with logical value 1), but the monitored bus value was dominant. 5: Indicates Bit0 error, in which, during the transmission of a message (or acknowledge bit, active error flag, or overload flag), the device attempted to transmit a dominant level (logic value 0 for data or identifier bit), but the monitored bus value was recessive. During Bus_Off recovery, this status is set each time the sequence of 11 recessive bits is monitored. This allows the software to monitor the progress of the Bus_Off recovery sequence (indicating that the bus is not dominant or continuously disturbed). 6: Indicates a CRC error, an incorrect CRC checksum of the received message. The CRC of the received message does not match the CRC calculated from the received data. 7: Indicates No Change, the read access to the protocol status register to be reinitialized to LEC [2:0] bits to 7. In case LEC [2:0] bit indicates the value of 7, no CAN bus events have been detected since the last software read access to the protocol status register. When a frame in CAN FD format reaches the data phase with the BRS flag set, the next CAN event (error or valid frame) is indicated by DLEC instead of LEC [2:0] bit. An error in the fixed staff bits of the CAN FD CRC sequence is displayed as a form error, not as a staff error. Bus_Off recovery sequence (See ISO 11898-1: 2015) cannot be shortened even by setting or resetting the INIT bit in the CANCCCR register. When the device is Bus_Off, the hardware sets its own INIT bit in the CANCCCR register and stops all bus activity. When the INIT bit in the CANCCCR register is cleared by the software, the device waits until 129 bus idles (129*11 consecutive recessive bits) have occurred before resuming normal operation. At the end of the Bus_Off recovery sequence, the error management counter is reset. During the wait time after resetting the INIT bit in the CANCCCR register, every time a sequence of 11 recessive bits is monitored, a Bit0 Error code is written on LEC [2:0] bit in the CANPSR register, and the software monitors the CAN bus for dominant or continuous disturbances and Bus_Off recovery sequences. The REC bit in the CANECR register is used to count these sequences. For byte access, reading byte 0 sets the LEC [2:0] bit to 111. There is no effect of reading byte 3/2/1.

19.2.17 Transmitter Delay Correction Register (CANTDCR)

Offset: 0x48

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TDCO							-	TDCF						
R/W	R	RP	RP	RP	RP	RP	RP	RP	R	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 15	-	Reserved bits
14 - 8	TDCO[6:0]	Transmitter Delay Compensation SSP Offset Offset value that defines the distance between the measured delay from the CAN0_TXD pin to the CAN0_RXD pin and the secondary sample point. Valid values are 0 - 127 mtq.
7	-	Reserved bit
6 - 0	TDCF[6:0]	Transmitter Delay Compensation Filter Window Length Defines the minimum value of the SSP position. The dominant edge on the CAN0_RXD pin that represents the previous SSP position is ignored in the transmitter delay measurement. This function is enabled when TDCF [6:0] bit is set to a value greater than TDCO [6:0] bit. Valid values are 0 - 127 mtq.

19.2.18 Interrupt Register (CANIR)

Offset: 0x50

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	ARA	PED	PEA	WDI	BO	EW	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When one of the listed conditions is detected, a flag is set (edge dependent). The flag remains set until it is cleared by the software. Writing 1 on the corresponding bit position clears the flag. Writing 0 has no effect. A hardware reset clears the register. The setting of the CANIE register controls whether or not an interrupt is generated. The setting of the CANILS register controls the interrupt signal which notifies an interrupt.

Bit No	Bit name	Description
31 - 30	-	Reserved bits
29	ARA	Access to Reserved Address 0: Access to the reserved address did not occur 1: Access to a reserved address has occurred
28	PED	Protocol Error in Data Phase (Data Bit Time is used) 0: No protocol errors in the data phase 1: A protocol error was detected in the data phase (DLEC bit in the CANPSR register ≠ 0, 7).
27	PEA	Protocol Error in Arbitration Phase (Nominal Bit Time is used) 0: No protocol errors in the arbitration phase 1: A protocol error was detected in the arbitration phase (LEC bit in the CANPSR register ≠ 0, 7).
26	WDI	Watchdog Interrupt 0: Message RAM watchdog event has not occurred 1: Message RAM watchdog event has occurred
25	BO	Bus_Off Status 0: Bus_Off status has not been changed 1: Bus_Off status has been changed
24	EW	Warning Status 0: Error_Warning status has not been changed 1: Error_Warning status has been changed
23	EP	Error Passive 0: Error_Passive status has not been changed 1: Error_Passive status has been changed
22	ELO	Error Logging Overflow 0: The CAN error log counter did not overflow 1: The CAN error log counter experienced overflow
21	BEU	Bit Error Uncorrected A message RAM bit error has been detected but not corrected. An uncorrected message RAM bit error sets the INIT bit in the CANCCCR register to “1”. This is executed in order to avoid transmitting corrupted data. 0: No bit errors were detected when reading from the message RAM. 1: A bit error has been detected but not corrected.
20	BEC	Bit Error Corrected A message RAM bit error has been detected and corrected. 0: No bit errors were detected when reading from message RAM 1: A bit error has been detected and corrected
19	DRX	Message stored to Dedicated Rx Buffer A flag is set each time a received message is stored in the dedicated Rx buffer. 0: Rx buffer is not being updated 1: At least one received message has been stored in the Rx buffer

Bit No	Bit name	Description
18	TOO	Timeout Occurred 0: No timeout 1: Timeout has occurred
17	MRAF	Message RAM Access Failure - A flag is set in case the Rx handler has not completed the acceptance filtering or storage of accepted messages until the arbitration field of the next message is received. In this case, the acceptance filtering or message storage is aborted, and the Rx handler starts processing the next message. - A flag is set in case the Rx handler could not write a message on Message RAM. In this case, message storage is aborted. In both cases, the FIFO put index is not updated. In case a new data flag in the dedicated Rx buffer is not set, the storing of the next message in this location will overwrite the partially stored message. This flag is set also in case the Tx handler fails to read a message from the message RAM in time. In this case, message transmission is aborted. In case an access to the Tx handler fails, CAN FD controller switches to limited operation mode (see Section 19.3.2.5). To exit the limited operation mode, the software must reset the ASM bit in the CANCCCR register. 0: Message RAM access failure has not occurred 1: Message RAM access failure has occurred
16	TSW	Timestamp Wraparound 0: No timestamp counter wrap-around 1: Timestamp counter wrapped around
15	TEFL	Tx Event FIFO Element Lost 0: Tx event FIFO elements have not been lost 1: Tx event FIFO elements have been lost (and this status is set also after an attempt to write on a Tx event FIFO of size 0).
14	TEFF	Tx Event FIFO Full 0: Tx event FIFO is not full 1: Tx event FIFO is full
13	TEFW	Tx Event FIFO Watermark Reached 0: The filling level of the Tx event FIFO is below the watermark 1: The filling level of the Tx event FIFO has reached the watermark
12	TEFN	Tx Event FIFO New Entry 0: No changes in Tx event FIFO 1: Tx handler has written an Tx event FIFO element
11	TFE	Tx FIFO Empty 0: Tx FIFO is not empty 1: Tx FIFO is empty
10	TCF	Transmission Cancellation Finished 0: Transmission cancellation has not finished 1: Transmission cancellation has finished
9	TC	Transmission Completed 0: Transmission has not been completed 1: Transmission has been completed
8	HPM	High Priority Message 0: No high-priority messages have been received 1: A high-priority message has been received
7	RF1L	Rx FIFO 1 Message Lost 0: Rx FIFO 1 message has not been lost 1: Rx FIFO 1 message has been lost (and this status is set also after an attempt to write on Rx FIFO 1 of size 0)
6	RF1F	Rx FIFO 1 Full 0: Rx FIFO 1 is not full 1: Rx FIFO 1 is full
5	RF1W	Rx FIFO 1 Watermark Reached 0: Rx FIFO 1 filling level is below the watermark 1: Rx FIFO 1 filling level has reached the watermark
4	RF1N	Rx FIFO 1 New Message 0: No new messages are written to Rx FIFO 1 1: A new message has been written to Rx FIFO 1
3	RF0L	Rx FIFO 0 Message Lost 0: Rx FIFO 0 message has not been lost 1: Rx FIFO 0 message has been lost (and this status is set also after an attempt to write to Rx FIFO 0 of size 0)
2	RF0F	Rx FIFO 0 Full 0: Rx FIFO 0 is not full 1: Rx FIFO 0 is full

Bit No	Bit name	Description
1	RF0W	Rx FIFO 0 Watermark Reached 0: Rx FIFO 0 filling level is below the watermark 1: Rx FIFO 0 filling level has reached the watermark
0	RF0N	Rx FIFO 0 New Message 0: No new messages are written to Rx FIFO 0 1: A new messages has been written to Rx FIFO 0

19.2.19 Interrupt Enable Register (CANIE)

Offset: 0x54

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The setting of Interrupt Enable Register determines which status changes in the interrupt register are notified on the interrupt line.

[Description of each bit]
This is used to enable the corresponding interrupt.
0: Disabled (Initial value)
1: Enabled

Bit No	Bit name	Description (corresponding interrupt)
31 - 30	-	Reserved bits
29	ARAE	Access to Reserved Address Enable Interrupt enable by access to the reserved address
28	PEDE	Protocol Error in Data Phase Enable Interrupt enable by protocol error in data phase
27	PEAE	Protocol Error in Arbitration Phase Enable Interrupt enable by protocol error in arbitration phase
26	WDIE	Watchdog Interrupt Enable Interrupt enable by message RAM watchdog event
25	BOE	Bus_Off Status Interrupt Enable Interrupt enable by Bus_Off status
24	EWE	Warning Status Interrupt Enable Interrupt enable by Error_Warning status
23	EPE	Error Passive Interrupt Enable Interrupt enable by Error_Passive status
22	ELOE	Error Logging Overflow Interrupt Enable Interrupt enable by error log counter overflow
21	BEUE	Bit Error Uncorrected Interrupt Enable Interrupt enable by uncorrected bit error
20	BECE	Bit Error Corrected Interrupt Enable Interrupt enable by corrected bit error
19	DRXE	Message stored to Dedicated Rx Buffer Interrupt Enable Interrupt enable by receive messages stored to Rx buffer
18	TOOE	Timeout Occurred Interrupt Enable Interrupt enable by timeout
17	MRAFE	Message RAM Access Failure Interrupt Enable Interrupt enable by message RAM access failure
16	TSWE	Timestamp Wraparound Interrupt Enable Interrupt enable by timestamp counter
15	TEFLE	Tx Event FIFO Event Lost Interrupt Enable Interrupt enable by Tx event FIFO events loss
14	TEFFE	Tx Event FIFO Full Interrupt Enable Interrupt enable by Tx event FIFO full
13	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable Interrupt enable by reaching the watermark of Tx event FIFO filling level

Bit No	Bit name	Description (corresponding interrupt)
12	TEFNE	Tx Event FIFO New Entry Interrupt Enable Interrupt enable by Tx event FIFO new entry
11	TFEE	Tx FIFO Empty Interrupt Enable Interrupt enable by Tx FIFO empty
10	TCFE	Transmission Cancellation Finished Interrupt Enable Interrupt enable by transmission cancellation finished
9	TCE	Transmission Completed Interrupt Enable Interrupt enable by transmission completion
8	HPME	High Priority Message Interrupt Enable Interrupt enable by receiving high priority messages
7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable Interrupt enable by Rx FIFO 1 messages loss
6	RF1FE	Rx FIFO 1 Full Interrupt Enable Interrupt enable by Rx FIFO 1 full
5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable Interrupt enable by reaching the watermark of Rx FIFO 1 filling level
4	RF1NE	Rx FIFO 1 New Message Interrupt Enable Interrupt enable by new messages written to Rx FIFO 1
3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable Interrupt enable by Rx FIFO 0 messages loss
2	RF0FE	Rx FIFO 0 Full Interrupt Enable Interrupt enable by Rx FIFO 0 full
1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable Interrupt enable by reaching the watermark of Rx FIFO 0 filling level
0	RF0NE	Rx FIFO 0 New Message Interrupt Enable Interrupt enable by new messages written to Rx FIFO 0

19.2.20 Interrupt Signal Select Register (CANILS)

Offset: 0x58

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The interrupt signal selection register assigns an interrupt generated by a particular interrupt flag from the interrupt register to one of the two module interrupt lines. To generate an interrupt, the respective interrupt signals must be enabled through bit EINT0 of the CANILE register and bit EINT1 of the CANILE register.
The setting values for each bit are as follows.

[Description of each bit]
This is used to assign the corresponding interrupt.
0: Assigns the interrupt to interrupt signal CAN0_INT0
1: Assigns the interrupt to interrupt signal CAN0_INT1

Bit No	Bit name	Description (corresponding interrupt)
31 – 30	-	Reserved bits
29	ARAL	Access to Reserved Address Line Interrupt by access to reserved address line
28	PEDL	Protocol Error in Data Phase Line Interrupt by protocol error in data phase line
27	PEAL	Protocol Error in Arbitration Phase Line Interrupt by protocol error in arbitration phase line
26	WDIL	Watchdog Interrupt Line Interrupt line by message RAM watchdog events
25	BOL	Bus_Off Status Interrupt Line Interrupt line by Bus_Off status
24	EWL	Warning Status Interrupt Line Interrupt line by Error_Warning status
23	EPL	Error Passive Interrupt Line Interrupt line by Error_Passive status
22	ELOL	Error Logging Overflow Interrupt Line Interrupt line by error log counter overflow
21	BEUL	Bit Error Uncorrected Interrupt Line Interrupt line by uncorrected bit error
20	BECL	Bit Error Corrected Interrupt Line Interrupt line by corrected bit error
19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line Interrupt line by messages stored to Rx buffer
18	TOOL	Timeout Occurred Interrupt Line Interrupt line by timeout
17	MRAFL	Message RAM Access Failure Interrupt Line Interrupt line by message RAM access failure
16	TSWL	Timestamp Wraparound Interrupt Line Interrupt line by timestamp counter wraparound
15	TEFLL	Tx Event FIFO Event Lost Interrupt Line Interrupt line by Tx event FIFO elements loss
14	TEFFL	Tx Event FIFO Full Interrupt Line Interrupt line by Tx event FIFO full

Bit No	Bit name	Description (corresponding interrupt)
13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line Interrupt line by reaching the watermark of Tx event FIFO filling level
12	TEFNL	Tx Event FIFO New Entry Interrupt Line Interrupt line by Tx event FIFO new entry
11	TFEL	Tx FIFO Empty Interrupt Line Interrupt line by Tx FIFO empty
10	TCFL	Transmission Cancellation Finished Interrupt Line Interrupt line by transmission cancellation finished
9	TCL	Transmission Completed Interrupt Line Interrupt line by transmission completion
8	HPML	High Priority Message Interrupt Line Interrupt line by receiving high priority messages
7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line Interrupt line by Rx FIFO 1 messages loss
6	RF1FL	Rx FIFO 1 Full Interrupt Line Interrupt line by Rx FIFO 1 full
5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line Interrupt line by reaching the watermark of Rx FIFO 1 filling level
4	RF1NL	Rx FIFO 1 New Message Interrupt Line Interrupt line by Rx FIFO 1 new messages
3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line Interrupt line by Rx FIFO 0 messages loss
2	RF0FL	Rx FIFO 0 Full Interrupt Line Interrupt line by Rx FIFO 0 full
1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line Interrupt line by reaching the watermark of Rx FIFO 0 filling level
0	RF0NL	Rx FIFO 0 New Message Interrupt Line Interrupt line by Rx FIFO 0 new messages

19.2.21 Interrupt Signal Enable Register (CANILE)

Offset: 0x5C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EINT1	EINT0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Each of the two interrupt signals to the CPU can be individually enabled or disabled by setting the EINT0 and EINT1 bits.

Bit No	Bit name	Description
31 - 2	-	Reserved bits
1	EINT1	Enable Interrupt Line 1 0: Interrupt signal CAN0_INT1 is disabled 1: Interrupt signal CAN0_INT1 is enabled
0	EINT0	Enable Interrupt Line 0 0: Interrupt signal CAN0_INT0 is disabled 1: Interrupt signal CAN0_INT0 is enabled

19.2.22 Global Filter Configuration Register (CANGFC)

Offset: 0x80

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	ANFS		ANFE		RRFS	RRFE
R/W	R	R	R	R	R	R	R	R	R	R	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Global settings for message ID filtering. The global filter settings control the filter path for standard and extended messages, as described at Figure19-11 and Figure19-12.

Bit No	Bit name	Description
31 – 6	-	Reserved bits
5 - 4	ANFS[1:0]	Accept Non-matching Frames Standard Defines the handling of received messages with 11-bit IDs that do not match any element in the filter list. 00: Accepted at Rx FIFO 0 01: Accepted at Rx FIFO 1 10: To be discarded 11: To be discarded
3 - 2	ANFE[1:0]	Accept Non-matching Frames Extended Defines the handling of received messages with 29-bit IDs that do not match any element in the filter list. 00: Accepted at Rx FIFO 0 01: Accepted at Rx FIFO 1 10: To be discarded 11: To be discarded
1	RRFS	Reject Remote Frames Standard 0: Filters remote frames with 11-bit standard ID 1: Discards all remote frames with 11-bit standard ID
0	RRFE	Reject Remote Frames Extended 0: Filters remote frames with 29-bit extended ID 1: Discards all remote frames with 29-bit extended ID

19.2.23 Standard ID Filter Configuration Register (CANSIDFC)

Offset: 0x84

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLSSA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Configuration of 11-bit standard message ID filtering. Standard ID filter configuration controls the filter path for standard messages, as described at Figure19-11.

Bit No	Bit name	Description
31 - 24	-	Reserved bits
23 - 16	LSS[7:0]	List Size Standard 0: No standard message ID filter 1-128: Number of standard message ID filter elements > 128: Values greater than 128 are interpreted as 128
15 - 2	FLSSA[15:2]	Filter List Standard Start Address Start address of the standard message ID filter list (32-bit word address, see Figure19-2).
1 - 0	-	Reserved bits

19.2.24 Extended ID Filter Configuration Register (CANXIDFC)

Offset: 0x88

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLESA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 23	-	Reserved bits
22 - 16	LSE[6:0]	List Size Extended 0: No extended message ID filter 1-64: Number of extended message ID filter elements > 64: Values greater than 64 are interpreted as 64
15 - 2	FLESA[15:2]	Filter List Extended Start Address Start address of the extended message ID filter list (32-bit word address, see Figure19-2).
1 - 0	-	Reserved bits

19.2.25 Extended ID Mask Register (CANXIDAM)

Offset: 0x90

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	EIDM												
R/W	R	R	R	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EIDM															
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit No	Bit name	Description
31 - 29	-	Reserved bits
28 - 0	EIDM[28:0]	Extended ID Mask In the acceptance filtering of extended frames, the extended ID mask is logically multiplied by the message ID of the receive frame. The register covers the masking of SAE J1939 29-bit IDs. When all bits are set to “1”, the reset value, the mask will not be active.

19.2.26 High Priority Message Status Register (CANHPMS)

Offset: 0x94

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLST	FIDX							MSI		BIDX					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is updated each time a message ID filter element configured to generate a priority event is matched. This can be used to monitor the status of reception of high-priority messages and to allow fast access to these messages.

Bit No	Bit name	Description
31 - 16	-	Reserved bits
15	FLST	Filter List Indicates a filter list with matching filter elements. 0: Standard filter list 1: Extended filter list
14 - 8	FIDX[6:0]	Filter Index Index of the matching filter elements, within the range 0 - CANSIDFC.LSS-1 or CANXIDFC.LSE-1.
7 - 6	MSI[1:0]	Message Storage Indicator 00: FIFO is not selected 01: FIFO messages have been lost 10: Messages are stored in FIFO 0 11: Messages are stored in FIFO 1
5 - 0	BIDX[5:0]	Buffer Index Index of the Rx FIFO element where messages have been stored. Valid only when MSI [1] bit = “1”.

19.2.27 Specific ID Receive Message 1 Register (CANNDAT1)

Offset: 0x98

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	ND31 to ND0	New Data This register retains new data flags for Rx buffers 0 - 31. A flag is set when each Rx buffer is updated from a receive frame. The flag will remain set until cleared by the software. Writing “1” on the corresponding bit clears the flag. Writing 0 has no effect. A hardware reset clears the register. 0: Rx buffer has not been updated 1: Rx buffer has been updated with a new message

19.2.28 Specific ID Receive Message 2 Register (CANNDAT2)

Offset: 0x9C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	ND63 to ND32	New Data This register retains new data flags for Rx buffers 32 - 63. A flag is set when each Rx buffer is updated from a receive frame. The flag will remain set until cleared by the software. Writing “1” on the corresponding bit clears the flag. Writing 0 has no effect. A hardware reset clears the register. 0: Rx buffer has not been updated 1: Rx buffer has been updated with a new message

19.2.29 Rx FIFO 0 Configuration Register (CANRXF0C)

Offset: 0xA0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	F0OM	F0WM							-	F0S						
R/W	RP	RP	RP	RP	RP	RP	RP	RP	R	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F0SA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31	F0OM	FIFO 0 Operation Mode FIFO 0 can operate in blocking or overwriting mode (see section “19.3.5.2”). 0: FIFO 0 blocking mode 1: FIFO 0 overwrite mode
30 - 24	F0WM[6:0]	Rx FIFO 0 Watermark 0: Watermark interrupt is disabled 1-64: Rx FIFO 0 watermark interrupt level (RF0W bit in the CANIR register) > 64: Watermark interrupt is disabled
23	-	Reserved bit
22 - 16	F0S[6:0]	Rx FIFO 0 Size 0: Rx FIFO 0 none 1-64: Number of Rx FIFO 0 elements > 64: Values greater than 64 are interpreted as 64 Rx FIFO 0 element is given an index from 0 to F0S-1
15 – 2	F0SA[15:2]	Rx FIFO 0 Start Address Start address of Rx FIFO 0 in message RAM (32-bit word address, see Figure19-2).
1 – 0	-	Reserved bits

19.2.30 Rx FIFO 0 Status Register (CANRXF0S)

Offset: 0xA4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	RF0L	F0F	-	-	F0PI					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	F0GI						-	F0FL						
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 26	-	Reserved bits
25	RF0L	Rx FIFO 0 Message Lost This bit is a copy of the RF0L bit in the CANIR register. When the RF0L bit in the CANIR register is reset, this bit is also reset. 0: Rx FIFO 0 message is not lost 1: This status is also set after an attempt to write to Rx FIFO 0 of size 0 for which Rx FIFO 0 message is lost Even when the oldest message is overwritten while the F0OM bit in the CANRXF0C register = “1”, this flag is not set.
24	F0F	Rx FIFO 0 Full 0: Rx FIFO 0 is not full 1: Rx FIFO 0 is full
23 - 22	-	Reserved bits
21 - 16	F0PI[5:0]	Rx FIFO 0 Put Index Rx FIFO 0 write index pointer with a range of 0 - 63.
15 - 14	-	Reserved bits
13 - 8	F0GI[5:0]	Rx FIFO 0 Get Index Rx FIFO 0 read index pointer with a range of 0 - 63.
7	-	Reserved bit
6 - 0	F0FL[6:0]	Rx FIFO 0 Fill Level The number of elements stored in Rx FIFO 0, in the range 0 - 64.

19.2.31 Rx FIFO 0 Acknowledge Register (CANRXF0A)

Offset: 0xA8

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	F0AI					
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 6	-	Reserved bits
5 - 0	F0AI[5:0]	Rx FIFO 0 Acknowledge Index After the software reads a message or series of messages from Rx FIFO 0, it must write the buffer index of the last element read from Rx FIFO 0 to bits F0AI [5:0] bit. This sets the F0GI bit in the CANRXF0S register to F0AI + 1 and updates the F0FL bit in the CANRXF0S register.

19.2.32 Rx Buffer Configuration Register (CANRXBC)

Offset: 0xAC

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBSA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 16	-	Reserved bits
15 - 2	RBSA[15:2]	Rx Buffer Start Address Sets the start address of the Rx buffer section of message RAM (32-bit word address). Also used to reference debug messages A, B, and C. Debugging function is not supported
1 - 0	-	Reserved bits

19.2.33 Rx FIFO 1 Configuration Register (CANRXF1C)

Offset: 0xB0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	F1OM	F1WM							-	F1S						
R/W	RP	RP	RP	RP	RP	RP	RP	RP	R	RP	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F1SA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31	F1OM	FIFO 1 Operation Mode FIFO 1 can operate in blocking or overwriting mode (see section “19.3.5.2”). 0 = FIFO 1 blocking mode 1 = FIFO 1 overwrite mode
30 - 24	F1WM[6:0]	Rx FIFO 1 Watermark 0: Watermark interrupt is disabled 1-64: Rx FIFO 1 watermark interrupt level (RF1W bit in the CANIR register) > 64: Watermark interrupt is disabled
23	-	Reserved bit
22 - 16	F1S[6:0]	Rx FIFO 1 Size 0: Rx FIFO 1 none 1-64: Number of Rx FIFO 1 elements > 64: Values greater than 64 are interpreted as 64 Rx FIFO 1 element is given an index from 0 to F1S-1
15 - 2	F1SA[15:2]	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in message RAM (32-bit word address, see Figure19-2).
1 - 0	-	Reserved bits

19.2.34 Rx FIFO 1 Status Register (CANRXF1S)

Offset: 0xB4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMS		-	-	-	-	RF1L	F1F	-	-	F1PI					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	F1GI						-	F1FL						
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 30	DMS[1:0]	Debug Message Status 00: Idle state, waiting to receive debug messages, DMA requests are cleared 01: Received debug message A 10: Received debug messages A and B 11: Received debug messages A, B, and C, and the DMA request is set Debugging function is not supported
29 - 26	-	Reserved bits
25	RF1L	Rx FIFO 1 Message Lost This bit is a copy of the CANIR register RF1L bit. When the RF1L bit in the CANIR register is reset, this bit is also reset. 0: Rx FIFO 1 message has not been lost 1: Rx FIFO 1 message is lost, and this status is also set after an attempt to write to Rx FIFO 0 of size 0 Even when the oldest message is overwritten while the F1OM bit in the CANRXF1C register = 1, this flag is not set.
24	F1F	Rx FIFO 1 Full 0: Rx FIFO 1 is not full 1: Rx FIFO 1 is full
23 - 22	-	Reserved bits
21 - 16	F1PI[5:0]	Rx FIFO 1 Put Index Rx FIFO 1 write index pointer with a range of 0 - 63.
15 - 14	-	Reserved bits
13 - 8	F1GI[5:0]	Rx FIFO 1 Get Index Rx FIFO 1 read index pointer with a range of 0 - 63.
7	-	Reserved bit
6 - 0	F1FL[6:0]	Rx FIFO 1 Fill Level The number of elements stored in Rx FIFO 1, in the range 0 - 64.

19.2.35 Rx FIFO 1 Acknowledge Register (CANRXF1A)

Offset: 0xB8

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	F1AI					
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 6	-	Reserved bits
5 - 0	F1AI[5:0]	Rx FIFO 1 Acknowledge Index After the software reads a message or series of messages from Rx FIFO 1, it must write the buffer index of the last element read from Rx FIFO 1 to F1AI. This sets the F1GI bit in the CANRXF1S register to F1AI + 1 and updates the F1FL bit in the CANRXF1S register.

19.2.36 Rx Buffer/FIFO Element Size Setting Register (CANRXESC)

Offset: 0xBC

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RBDS			-	F1DS			-	F0DS		
R/W	R	R	R	R	R	RP	RP	RP	R	RP	RP	RP	R	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 11	-	Reserved bits
10 - 8	RBDS[2:0]	Rx Buffer Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field
7	-	Reserved bit
6 - 4	F1DS[2:0]	Rx FIFO 1 Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field
3	-	Reserved bit
2 - 0	F0DS[2:0]	Rx FIFO 0 Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field

[Note]

- **In case the data field size of the received CAN frame exceeds the data field size set in the matching Rx buffer or Rx FIFO, only the number of bytes set in the CANRXESC register will be stored in the Rx buffer. The rest of the data fields in the frame will be ignored.**

19.2.37 Tx Buffer Configuration Register (CANTXBC)

Offset: 0xC0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	TFQM	TFQS						-	-	NDTB					
R/W	R	RP	RP	RP	RP	RP	RP	RP	R	R	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBSA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31	-	Reserved bits
30	TFQM	Tx FIFO/Queue Mode 0 = Tx FIFO operation 1 = Tx queue operation
29 - 24	TFQS[5:0]	Transmit FIFO/Queue Size 0: Tx FIFO/queue none 1-32: Number of Tx buffers used for Tx FIFO/Queue > 32: Values greater than 32 are interpreted as 32
23 - 22	-	Reserved bits
21 - 16	NDTB[5:0]	Number of Dedicated Transmit Buffers 0: Dedicated Tx buffer none 1-32: Number of dedicated Tx buffers > 32: Values greater than 32 are interpreted as 32
15 - 2	TBSA[15:2]	Tx Buffers Start Address Start address of the Tx buffer section in message RAM (32-bit word address, see Figure19-2).
1 - 0	-	Reserved bits

[Note]

- **Note that the sum of TFQS and NDTB is less than or equal to 32. There is no check for incorrect settings. The Tx buffer section of message RAM starts with a dedicated Tx buffer.**

19.2.38 Tx FIFO/Queue Status Register (CANTXFQS)

Offset: 0xC4

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	TFGI					-	-	TFFL					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 22	-	Reserved bits
21	TFQF	Tx FIFO/Queue Full 0: Tx FIFO/Queue is not full 1: Tx FIFO/Queue is full
20 - 16	TFQPI[4:0]	Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer with a range of 0 - 31.
15 - 13	-	Reserved bits
12 - 8	TFGI[4:0]	Tx FIFO Get Index Tx FIFO read index pointer with a range of 0 - 31. To be read as 0 when Tx queue operation is set (TFQM bit in the CANTXBC register = “1”).
7 - 6	-	Reserved bits
5 - 0	TFFL[5:0]	Tx FIFO Free Level Number of consecutive free Tx FIFO elements, starting with TFGI, in the range 0 - 32. To be read as 0 when Tx queue operation is set (TFQM bit in the CANTXBC register = “1”).

[Note]

- **In a mixed configuration where dedicated Tx buffers are combined with Tx FIFO or Tx queue, the Put and Get indices indicate the number of Tx buffers starting with the first dedicated Tx buffer.**
  Example: In a Tx FIFO configuration of 12 dedicated Tx buffers and 20 buffers, Put index of 15 points to the fourth buffer in the Tx FIFO.

19.2.39 Tx Buffer Element Size Setting Register (CANTXESC)

Offset: 0xC8

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	TBDS		
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sets the number of data bytes belonging to the Tx buffer element. Data field sizes larger than 8 bytes are dedicated to CAN FD operation.

Bit No	Bit name	Description
31 - 3	-	Reserved bits
2 - 0	TBDS[2:0]	Tx Buffer Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field

[Note]

- **In case the data length code DLC of the Tx buffer element is set to a value greater than TBDS [2:0] bit in the Tx buffer data field size CANTXESC register, bytes, which are not defined in the Tx buffer, are transmitted as 0xCC (padding bytes).**

19.2.40 Tx Buffer Request Pending Register (CANTXBRP)

Offset: 0xCC

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	TRP31 to TRP0	Transmission Request Pending Bits to be used to indicate the transmit request pending status of each Tx buffer. These bits are set through the CANTXBAR register. These bits are reset after the requested transmission has been completed or canceled through the CANTXBCR register. These bits are set only for Tx buffers configured through the CANTXBC register. After a CANTXBRP bit is set, Tx scan (see section “19.3.6”) is initiated to check for the highest priority (the Tx buffer with the lowest message ID). A cancellation request resets the corresponding transmit request pending bit in the CANTXBRP register. In case the transmission has already started when the cancellation is requested, this will be executed at the end of the transmission, regardless of whether the transmission was successful or not. The cancellation request bit is reset immediately after the corresponding CANTXBRP bit is reset. After the cancellation is requested, the completed cancellation is notified through the CANTXBCF register -after a successful transmission along with the corresponding CANTXBTO bit -in case the transmission has not yet started at the time of cancellation -In case the transmission is discontinued due to loss of arbitration -In case an error occurs during the frame transmission In DAR mode, in case all transmissions are unsuccessful, all transmissions are automatically canceled, and the corresponding CANTXBCF bit is set for all unsuccessful transmissions. 0: No pending transmission request 1: Pending transmission request A CANTXBRP bit, which is set during the execution of Tx scan, is not considered during this particular Tx scan. In case the cancellation is requested for such a Tx buffer, this transmit request is immediately canceled, and the corresponding CANTXBRP bit is reset.

19.2.41 Tx Buffer Additional Request Register (CANTXBAR)

Offset: 0xD0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	AR31 to AR0	Add Request Transmit request bits for each Tx buffer. Writing “1” sets a corresponding transmit request bit. Writing 0 has no effect. This allows the software to set multiple Tx buffer transmit requests with a single write to the CANTXBAR register. A bit in the CANTXBAR register is set only for Tx buffers that are set through the CANTXBC register. In case no Tx scan is being performed, this bit is reset immediately. Otherwise, this bit will remain set until the Tx scan process is complete. 0: No transmit request 1: Transmit to be requested In case a transmit request is applied to a Tx buffer containing a pending transmit request (with the corresponding CANTXBRP bit already set), this transmit request is ignored.

19.2.42 Tx Buffer Cancellation Request Register (CANTXBCR)

Offset: 0xD4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	CR31 to CR0	Cancellation Request Cancellation request bits for each Tx buffer. Writing “1” sets a corresponding cancellation request bit. Writing “0” has no effect. This allows the software to set multiple Tx buffer cancellation requests with a single write to the CANTXBCR register. Bits in the CANTXBCR register are set only for Tx buffers set through the CANTXBC register and remain set until the corresponding bit in the CANTXBRP register is reset. 0: No waiting for cancellation 1: Waiting for cancellation

19.2.43 Tx Buffer Transmit Occurrence Register (CANTXBTO)

Offset: 0xD8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	TO31 to TO0	Transmission Occurred Transmit completion bits for each Tx buffer. These bits are set when a bit in the corresponding CANTXBRP register is cleared after a successful transmission. These bits are reset when a new transmission is requested by writing “1” to the corresponding bit in the CANTXBAR register. 0: Transmission not completed 1: Transmission completed

19.2.44 Tx Buffer Cancellation Finish Register (CANTXBCF)

Offset: 0xDC

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	CF31 to CF0	Cancellation Finished Cancellation completion bits for each Tx buffer. These bits are set when the corresponding CANTXBRP bit is cleared after a cancellation is requested through the CANTXBCR register. In case the corresponding CANTXBRP bit is not set at the time of the cancellation, a CFn (n = 0 to 31) bit is set immediately. These bits are reset when a new transmission is requested by writing “1” to the corresponding bit in the CANTXBAR register. 0: Tx buffer cancellation not completed 1: Tx buffer cancellation completed

19.2.45 Tx Buffer Transmit Interrupt Enable Register (CANTXBTIE)

Offset: 0xE0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	TIE31 to TIE0	Transmission Interrupt Enable Transmission interrupt enable bits for each Tx buffer. 0: Disables transmission interrupt 1: Enables transmission interrupt

19.2.46 Tx Buffer Cancellation Complete Interrupt Enable Register (CANTXBCIE)

Offset: 0xE4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 0	CFIE31 to CFIE0	Cancellation Finished Interrupt Enable Cancellation finished interrupt enable bits for each Tx buffer. 0: Disables cancellation finish interrupt 1: Enables cancellation finish interrupt

19.2.47 Tx Event FIFO Configuration Register (CANTXEFC)

Offset: 0xF0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	EFWM						-	-	EFS					
R/W	R	R	RP	RP	RP	RP	RP	RP	R	R	RP	RP	RP	RP	RP	RP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EFSA														-	-
R/W	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	RP	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 – 30	-	Reserved bits
29 - 24	EFWM[5:0]	Event FIFO Watermark 0: Disables watermark interrupt 1-32: Level of Tx event FIFO watermark interrupt (TEFW bit in CANIR register) > 32: Watermark interrupt is disabled
23 - 22	-	Reserved bits
21 - 16	EFS[5:0]	Event FIFO Size 0: Tx event FIFO is disabled 1-32: Number of Tx event FIFO elements > 32: Values greater than 32 are interpreted as 32 Tx Event FIFO element is given an index from 0 to EFS-1
15 - 2	EFSA[15:2]	Event FIFO Start Address Start address of the Tx event FIFO in message RAM (32-bit word address, see Figure19-2).
1 - 0	-	Reserved bits

19.2.48 Tx Event FIFO Status Register (CANTXEFS)

Offset: 0xF4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	TEFL	EFF	-	-	-	EFPI				
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	EFGI					-	-	EFFL					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 26	-	Reserved bits
25	TEFL	Tx Event FIFO Element Lost This bit is a copy of the TEFL bit in the CANIR register. When the TEFL bit in the CANIR register is reset, this bit is also reset. 0: Tx event FIFO elements have not been lost 1: Tx event FIFO elements have been lost, or this status is set also after writing to a Tx event FIFO of size 0.
24	EFF	Event FIFO Full 0: Tx event FIFO is not full 1: Tx event FIFO is full
23 - 21	-	Reserved bits
20 - 16	EFPI[4:0]	Event FIFO Put Index Tx event FIFO write index pointer with a range of 0 - 31.
15 - 13	-	Reserved bits
12 - 8	EFGI[4:0]	Event FIFO Get Index Tx event FIFO read index pointer with a range of 0 - 31.
7 - 6	-	Reserved bits
5 - 0	EFFL[5:0]	Event FIFO Fill Level Number of Tx event FIFO elements, in the range 0 - 32.

19.2.49 Tx Event FIFO Acknowledge Register (CANTXEFA)

Offset: 0xF8

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	EFAI				
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit No	Bit name	Description
31 - 5	-	Reserved bits
4 - 0	EFAI[4:0]	Event FIFO Acknowledge Index After the software reads an element or a sequence of elements from the Tx Event FIFO, the index of the last element read from the Tx event FIFO must be written to the EFAI [4:0] bit. This sets the EFGI bit in the CANTXEFS register to EFAI + 1 and updates the EFFL bit in the CANTXEFS register.

19.3 Description of Operation

19.3.1 Description of Message RAM

Message RAM is used to store Rx/Tx messages and to store filter settings.

[Note]

- **Message RAM has a parity function to detect corruption of the retained data. To create valid parity bits, it is recommended that message RAM be initialized after a hardware reset by writing a 0x0000_0000 or similar to each message RAM word. This prevents interrupts caused by BEU bit of the CANIR register when reading from an uninitialized message RAM words.**

19.3.1.1 Configuration of Message RAM

The width of message RAM is 32 bits. CAN FD controller can be configured to allocate up to 1600 words for Message RAM. It is not necessary to configure each of the sections shown in Figure19-2, and there are no restrictions on the order of the sections.

When operating in CAN FD mode, the required message RAM size is highly dependent on the Rx FIFO 0, Rx FIFO 1, Rx buffer, and Tx buffer element sizes, which are set through the F0DS bit in the CANRXESC register, the F1DS bit in the CANRXESC register, the RBDS bit in the CANRXESC register, and the TBDS bit in the CANTXESC register.

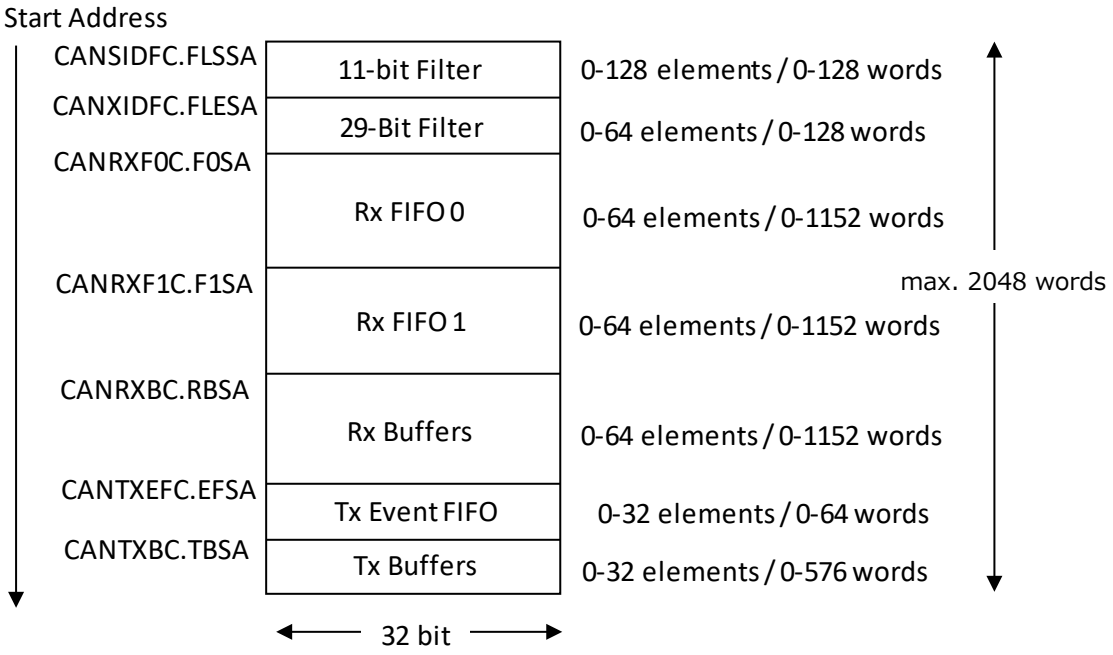


Figure19-2 Configuration of Message RAM

When CAN FD controller designates the address of message RAM, it specifies a 32-bit word address instead of a single byte. The configurable start address is a 32-bit word address, namely, only 31 - 2 bits are evaluated and the least significant 2 bits are ignored.

[Note]

- **CAN FD controller does not check message RAM for incorrect settings. In particular, the setting of the start addresses of the different sections and the number of elements in each section should be done carefully to avoid data tampering or loss.**

19.3.1.2 Rx Buffer and FIFO Element

Message RAM can be configured with up to 64 Rx buffers and two Rx FIFOs. Each Rx FIFO section can be configured to store up to 64 receive messages. The structure of the Rx buffer/FIFO element is shown below in Figure19-3. The element size can be configured through the CANRXESC register to store CAN FD messages with a data field of up to 64 bytes.

R1: RXTS [15:0] retains the 16-bit timestamp generated by the internal timestamp logic of CAN FD controller.

R2 - Rn: Depending on the setting of the CANRXESC register, 2 - 16 32-bit words (Rn = 2 - 17) are used to store the data fields of the CAN messages.

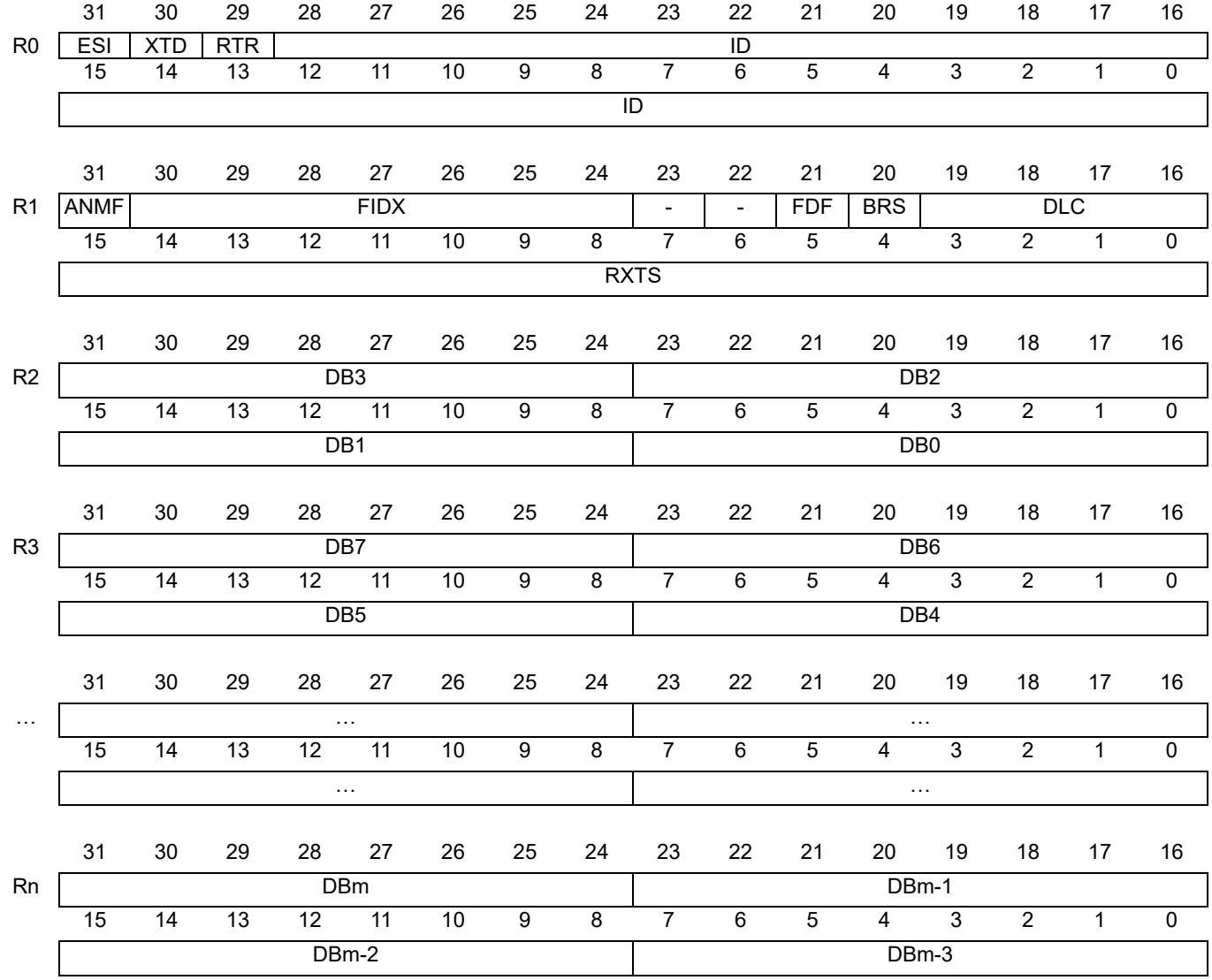


Figure19-3 Rx Buffer and FIFO Elements

Table19-4 R0 Fields

Bit No	Bit name	Description
31	ESI	Error State Indicator 0: Indicates that the transmission node is error active 1: Indicates that the transmission node is error passive
30	XTD	Extend Identifier Indicates whether the received frame is a standard identifier or an extended identifier. 0: 11-bit standard identifier 1: 29-bit extended identifier
29	RTR	Remote Transmission Request Indicates whether the received frame is a data frame or a remote frame. 0: Received frame is a data frame 1: Received frame is a remote frame There is no remote frame in CAN FD format. In the CAN FD frame (FDF = “1”), the dominant RRS (Remote Request Substitution) bit is replaced with the RTR (Remote Transmission Request) bit.
28 - 0	ID[28:0]	Identifier Check the XTD to see whether ID [28:0] is a standard identifier or an extended identifier. In case of a standard identifier, it is stored in ID [28:18].

Table19-5 R1 Fields

Bit No	Bit name	Description
31	ANMF	Accepted Non-matching Frame 0: Indicates that the received frame matched FIDX [6:0] 1: Indicates that the received frame did not match any Rx filter element The acceptance of non-matching frames can be enabled by the ANFS bit in the CANGFC register and the ANFE bit in the CANGFC register.
30 - 24	FIDX[6:0]	Filter Index 0 - 127: Indicates the index number that matched the Rx acceptance filter element (not valid in case ANMF = 1) The value ranges from 0 to (LSS bit in the CANSIDFC register - 1) or (LSE bit in the CANXIDFC register - 1)
23 - 22	-	Reserved
21	FDF	FD Format 0: Indicates that the frame is in classic CAN frame format 1: Indicates that the frame is in CAN FD frame format (new DLC encoding, CRC)
20	BRS	Bit Rate Switch 0: Indicates that the frame was received without bit rate switching 1: Indicates that the frame was received using bit rate switching
19 - 16	DLC[3:0]	Data Length Recode 0 - 8 (Classic CAN or CAN FD): Indicates that the number of data bytes in the received frame is 0 - 8 bytes 9 - 15 (Classic CAN): Indicates that the number of data bytes in the received frame is 8 bytes 9 - 15 (CAN FD): Indicates that the number of data bytes in the received frame is 12/16/20/24/32/48/64 bytes.
15 - 0	RXTS[15:0]	Rx Timestamp Indicates the value of the timestamp counter captured at the start of frame reception. The resolution depends on the setting of the timestamp counter prescaler (TCP bit in the CANTSCC register).

Table19-6 R2 Fields

Bit No	Bit name	Description
31 - 24	DB3[7:0]	Data Byte 3
13 - 16	DB2[7:0]	Data Byte 2
15 - 8	DB1[7:0]	Data Byte 1
7 - 0	DB0[7:0]	Data Byte 0

Table19-7 R3 Fields

Bit No	Bit name	Description
31 - 24	DB7[7:0]	Data Byte 7
13 - 16	DB6[7:0]	Data Byte 6
15 - 8	DB5[7:0]	Data Byte 5
7 - 0	DB4[7:0]	Data Byte 4

Table19-8 Rn Fields

Bit No	Bit name	Description
31 - 24	DBm[7:0]	Data Byte m
13 - 16	DBm-1[7:0]	Data Byte m-1
15 - 8	DBm-2[7:0]	Data Byte m-2
7 - 0	DBm-3[7:0]	Data Byte m-3

19.3.1.3 Tx Buffer Element

A Tx buffer section can be configured to have a mix of dedicated Tx buffers and Tx FIFO/Tx queues. When a Tx buffer section is shared by a dedicated Tx buffer and a Tx FIFO/Tx queue, the dedicated Tx buffer starts at the beginning of the Tx buffer section and is followed by buffers allocated to the Tx FIFO or Tx queue. The Tx handler identifies the dedicated Tx buffer and the Tx FIFO/Tx queue by the setting of the TFQS bit in the TXBC register and the NDTB bit in the TXBC register in the Tx buffer configuration register. The element size can be set via the CANTXESC register for the storage of CAN FD messages with a data field of up to 64 bytes.

T2 - Tn: Depending on the setting of the CANTXESC register, 2 - 16 32-bit words (Tn = 2 - 17) are used to store the data fields of the CAN messages.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T0	ESI	XTD	RTR	ID												
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ID															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T1	MM[7:0]								EFC	TSCE	FDF	BRS	DLC			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MM[15:8]								-	-	-	-	-	-	-	-
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T2	DB3								DB2							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DB1								DB0							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T3	DB7								DB6							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DB5								DB4							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
…	…								…							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	…								…							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Tn	DBm								DBm-1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DBm-2								DBm-3							

Figure19-4 Tx Buffer Element

Table19-9 T0 Fields

Bit No	Bit name	Description
31	ESI	Error Status Indicator 0: Sets the ESI bit of CAN FD format to depend only on the error passive flag 1: The ESI bit in CAN FD format sets the recessive to be transmitted The ESI bit in the Tx buffer is a logical OR with the error passive flag to determine the value of the ESI bit in the transmission FD frame. As required by the CAN FD protocol specification, error active nodes can optionally transmit the ESI bit to the recessive, but error passive nodes always transmit the ESI bit to the recessive.
30	XTD	Extended Identifier Sets whether the frame to be transmitted is a standard identifier or an extended identifier. 0: 11-bit standard identifier 1: 29-bit extended identifier
29	RTR	Remote Transmission Request Indicates whether the frame to be transmitted is a data frame or a remote frame. 0: Transmits a data frame 1: Transmits a remote frame When RTR = “1”, CAN FD controller transmits remote frames according to ISO 11898-1:2015, even if transmission in CAN FD format is enabled by the FDOE bit in the CANCCCR register.
28 - 0	ID[28:0]	Identifier Whether it is a standard or extended identifier depends on the XTD setting. In case of a standard identifier, write to ID [28:18].

Table19-10 T1 Fields

Bit No	Bit name	Description
31 - 24	MM[7:0]	Message Marker To identify the Tx message status, the value written at the time of setting Tx buffer is copied to the Tx event FIFO element when the transmission is completed.
23	EFC	Event FIFO Control 0: Not to be stored in Tx event FIFO element 1: To be stored in Tx event FIFO element
22	TSCE	Timestamp Caputure Enable for TSU Set the value of this bit to 0 at all times. Operation is not guaranteed in case “1” is written.
21	FDF	FD Format 0: Transmits frames in classic CAN format 1: Transmits frames in CAN FD format
20	BRS	Bit Rate Switch 0: Transmits CAN FD frames without bit rate switching 1: Transmits CAN FD frames with bit rate switching ESI of T0 as well as FDF and BRS of T1 are evaluated only when CAN FD operation is enabled. BRS is evaluated only when the BRSE bit in the CANCCCR register = “1”.
19 - 16	DLC[3:0]	Data Length Recode 0 - 8 (Classic CAN or CAN FD): Sets the number of data bytes in the transmission frame to 0 - 8 bytes 9 - 15 (Classic CAN): Set the number of data bytes in the transmission frame to 8 bytes 9 - 15 (CAN FD): Set the number of data bytes in the transmission frame to 12/16/20/24/32/48/64 bytes
15 - 8	MM[15:8]	Message Marker Upper byte of wide message marker. Available only when WMM bit in the CANCCCR register = 1. To identify the Tx message status, the value written at the time of setting Tx buffer is copied to the Tx event FIFO element when the transmission is completed.
7 - 0	-	Reserved

Table19-11 T2 Fields

Bit No	Bit name	Description
31 - 24	DB3[7:0]	Data Byte 3
13 - 16	DB2[7:0]	Data Byte 2
15 - 8	DB1[7:0]	Data Byte 1
7 - 0	DB0[7:0]	Data Byte 0

Table19-12 T3 Fields

Bit No	Bit name	Description
31 - 24	DB7[7:0]	Data Byte 7
13 - 16	DB6[7:0]	Data Byte 6
15 - 8	DB5[7:0]	Data Byte 5
7 - 0	DB4[7:0]	Data Byte 4

Table19-13 Tn Fields

Bit No	Bit name	Description
31 - 24	DBm[7:0]	Data Byte m
13 - 16	DBm-1[7:0]	Data Byte m-1
15 - 8	DBm-2[7:0]	Data Byte m-2
7 - 0	DBm-3[7:0]	Data Byte m-3

19.3.1.4 Tx Event FIFO Element

Each element contains information about the messages that have been transmitted. By reading the Tx event FIFO, the software retrieves this information in the order in which the messages were transmitted. Tx event FIFO status information can be obtained from the CANTXEFS register.

E1A: In case the WMM bit in the CANCCCR register = 0, TXTS [15:0] retains the 16-bit timestamp generated by the internal timestamp logic of CAN FD controller.

E1B: When the 16-bit message marker is enabled (WMM bit in the CANCCCR register = 1), MM [15:8] retains the upper 8 bits of the wide message marker.
TSC = 0 at all times, and TXTSP [3:0] is not enabled.

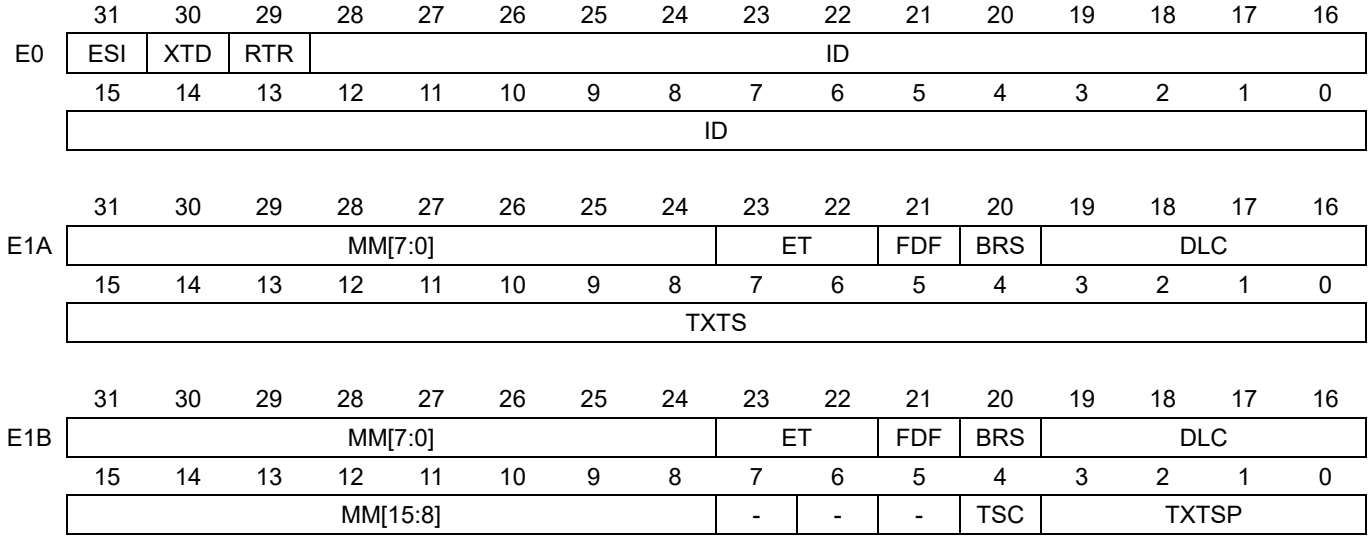


Figure19-5 Tx Event FIFO Element

Table19-14 E0 Fields

Bit No	Bit name	Description
31	ESI	Error Status Indicator 0: Indicates that the transmission node is error active 1: Indicates that the transmission node is error passive
30	XTD	Extended Identifier 0: 11-bit standard identifier 1: 29-bit extended identifier
29	RTR	Remote Transmission Request 0: Indicates that a data frame has been transmitted 1: Indicates that a remote frame has been transmitted
28 - 0	ID[28:0]	Identifier Check the XTD to see whether ID [28:0] is a standard identifier or an extended identifier. In case of a standard identifier, it is stored in ID [28:18].

Table19-15 E1A Fields

Bit No	Bit name	Description
31 - 24	MM[7:0]	Message Marker To identify the Tx message status, to be copied from the Tx buffer to the Tx event FIFO element.
23 - 22	ET[1:0]	Event Type 00: Reserved 01: Indicates a valid Tx event 10: Indicates that the message was transmitted despite cancellation (this value is set at all times when transmitting in DAR mode). 11: Reserved
21	FDF	FD Format 0: Classic CAN frame format 1: CAN FD frame format (new DLC coding, CRC)
20	BRS	Bit Rate Switch 0: Indicates that the frame was transmitted without bit rate switching 1: Indicates that the frame was transmitted using bit rate switching
19 - 16	DLC[3:0]	Data Length Recode 0 - 8 (Classic CAN or CAN FD): Indicates that the number of data bytes in the transmitted frame is 0 - 8 bytes 9 - 15 (Classic CAN): Indicates that the number of data bytes in the transmitted frame is 8 bytes 9 - 15 (CAN FD): Indicates that the number of data bytes in the transmitted frame is 12/16/20/24/32/48/64 bytes
15 - 0	TXTS[15:0]	Tx Timestamp Indicates the value of the timestamp counter taken in at the start of frame transmission. The resolution depends on the setting of the timestamp counter prescaler (TCP bit in the CANTSCC register).

Table19-16 E1B Fields

Bit No	Bit name	Description
31 - 24	MM[7:0]	Message Marker To identify the status of a Tx message, to be copied from the Tx buffer to the Tx event FIFO element.
23 - 22	ET[1:0]	Event Type 00: Reserved 01: Indicates a valid Tx event 10: Indicates that the message was transmitted despite cancellation (this value is set at all times when transmitting in DAR mode). 11: Reserved
21	FDF	FD Format 0: Classic CAN frame format 1: CAN FD frame format (new DLC coding, CRC)
20	BRS	Bit Rate Switch 0: Indicates that the frame was transmitted without bit rate switching 1: Indicates that the frame was transmitted using bit rate switching
19 - 16	DLC[3:0]	Data Length Code 0 - 8 (Classic CAN or CAN FD): Indicates that the number of data bytes in the transmitted frame is 0 - 8 bytes 9 - 15 (Classic CAN): Indicates that the number of data bytes in the transmitted frame is 8 bytes 9 - 15 (CAN FD): Indicates that the number of data bytes in the transmitted frame is 12/16/20/24/32/48/64 bytes
15 - 8	MM[15:8]	Message Marker Upper byte of the wide message marker written by the software when transmitting the Tx buffer. To identify the status of a Tx message, to be copied to the Tx event FIFO element.
7 - 5	-	Reserved
4	TSC	Timestamp Captured Being 0 at all times, indicating that no timestamp has been captured.
3:0	TXTSP[3:0]	Tx Timestamp Pointer The retained value of this bit is invalid.

19.3.1.5 Standard Message ID Filter Element

A maximum of 128 filter elements with 11-bit IDs can be set. When accessing a standard message ID filter element, its address is the FLSSA bit in the CANSIDFC register plus the index of the filter element (0 to 127) is added.

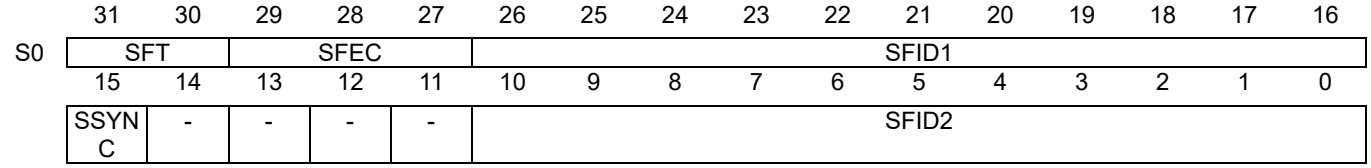


Figure19-6 Standard Message ID Filter Element

Table19-17 S0 Fields

Bit No	Bit name	Description
31 - 30	SFT[1:0]	Standard Filter Type 00: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) 01: Dual ID filter of SFID1 or SFID2 10: Classic filter (used as SFID1 = filter, SFID2 = mask) 11: Filter element is disabled With SFT= 11, the filter element is disabled and the acceptance filtering continues (same processing as with SFEC= “000”).
29 - 27	SFEC[2:0]	Standard Filter Configuration All enabled filter elements are used for the acceptance filtering of 11-bit ID frames. The acceptance filtering stops at the first-matched enabled filter element, or when the end of the filter list is reached. In case of SFEC= 100, 101, 110, a match sets the HPM bit in the CANIR register and an interrupt is generated if enabled. In this case, the register CANHPMS is updated in the status of priority match. 000: Disables the filter element 001: When the filter matches, to be stored in Rx FIFO0 010: When the filter matches, to be stored in Rx FIFO1 011: ID to be discarded when the filter matches 100: Sets the priority when the filter matches 101: Set the priority and store it in FIFO0 when the filter matches 110: Set the priority and store it in FIFO1 when the filter matches 111: To be stored in Rx buffer or as debug message, while SFT [1:0] setting is ignored Debugging function is not supported
26 - 16	SFID1[10:0]	Standard Filter ID 1 First ID of the standard ID filter element. In case of filtering Rx buffers, Sync messages, or debug messages, this field defines the ID of the messages to be saved. The received identifiers must match exactly and the masking mechanism is not used.
15	SSYNC	Standard Sync Message The retained value of this bit is invalid.
14 - 11	-	Reserved
10 - 0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the SFEC setting. 1) SFEC: “001”...“110” Second ID of the standard ID filter element 2) SFEC: “111” Filter for Rx buffers or debug messages SFID2 [10:9] determines whether the received message is stored in the Rx buffer or handled as message A, B, or C in the debug message sequence. 00: Stores the message in the Rx buffer 01: Debug message A 10: Debug message B 11: Debug message C SFID2 [8:6] is reserved. SFID2 [5:0] defines the offset to the Rx buffer start address CANRXBC.RBSA to store the matched messages. Debugging function is not supported

19.3.1.6 Extended Message ID Filter Element

A maximum of 64 filter elements with 29-bit IDs can be set. When accessing the extended message ID filter element, its address is the FLESA bit of the filter list extended start address CANXIDFC register plus a double of the index of the filter element (0 to 63).

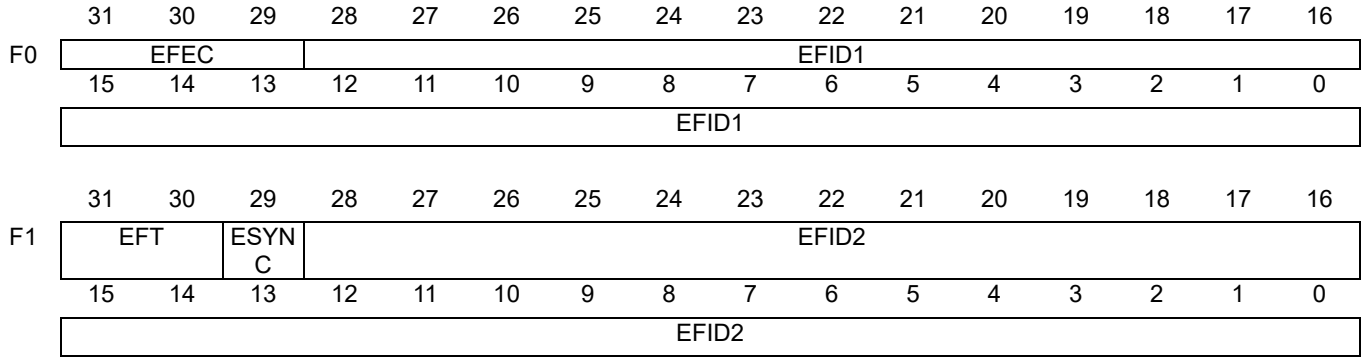


Figure19-7 Extended Message ID Filter Element

Table19-18 F0 Fields

Bit No	Bit name	Description
31 - 29	EFEC[2:0]	Extended Filter Element Configuration All valid filter elements are used for the acceptance filtering of 29-bit ID frames. The acceptance filtering stops at the first-matched enabled filter element or when the end of the filter list is reached. In case of EFEC= “100”, “101”, “110”, a match sets the HPM bit in the CANIR register, and an interrupt is generated if enabled. In this case, the register CANHPMS is updated in the status of priority match. 000: Disables the filter element 001: When the filter matches, to be stored in Rx FIFO0 010: When the filter matches, to be stored in Rx FIFO1 011: ID to be discarded when the filter matches 100: Sets the priority when the filter matches 101: Sets the priority and stores it in FIFO0 if the filter matches 110: Sets the priority and stores it in FIFO1 if the filter matches 111: To be stored in Rx buffer or as debug message, while EFT [1:0] setting is ignored Debugging function is not supported
28 - 0	EFID1[28:0]	Extended Filter ID 1 First ID of the extended ID filter element. In case of filtering Rx buffers, Sync messages, or debug messages, this field defines the ID of the extended messages.

Table19-19 F1 Fields

Bit No	Bit name	Description
31 - 30	EFT[1:0]	Extended Filter Type 00: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) 01: Dual ID filter for EFID1 or EFID2 10: Classic filter (EFID1 = filter, EFID2 = mask) 11: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), while CANXIDAM mask is not applied
29	ESYNC	Extended Sync Message The retained value of this bit is invalid.
28 - 0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the EFEC setting. 1) “001”...“110” Second ID of the extended ID filter element 2) EFEC = “111” Rx buffer or filter for debug messages EFID2 [10:9] determines whether the received message is stored in the Rx buffer or handled as message A, B, or C in the debug message sequence. 00: Stores the message in the Rx buffer 01: Debug message A 10: Debug message B 11: Debug message C EFID2 [8:6] is reserved. EFID2 [5:0] defines the offset to the Rx buffer start address CANRXBC.RBSA to store the matched messages. Debugging function is not supported

19.3.2 Operation Mode

19.3.2.1 Software initialization

Software initialization is initiated by setting the INIT bit in the bit CANCCCR register either by software or by a hardware reset, when an uncorrected bit error was detected in the message RAM, or by going Bus_Off. While the INIT bit of the CANCCCR register is set, message transfer to and from the CAN bus is stopped and the status of the CAN bus output CAN0_TXD pin becomes recessive (HIGH). The counter in the error management logic EML is not changed. Setting the INIT bit in the CANCCCR register does not change the configuration register. Resetting the INIT bit in the CANCCCR register completes the software initialization. Afterwards the bitstream processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of 11 consecutive recessive bits (≡Bus_Idle) before it joins the bus activity and starts the message transfer.

Access to CAN FD controller configuration register is enabled only when both the INIT bit in the CANCCCR register and the CCE bit in the CANCCCR register are set (Protected Write).

The CCE bit of the CANCCCR register can be set/reset only while the INIT bit of the CANCCCR register = 1. When the INIT bit of the CANCCCR register is reset, the CCE bit of the CANCCCR register is automatically reset.

When the CCE bit in the CANCCCR register is set, the following registers are reset:

- CANHPMS
- CANRXF0S
- CANRXF1S
- CANTXFQS
- CANTXBRP
- CANTXBTO
- CANTXBCF
- CANTXEFS

The timeout counter value TOC of the CANTOCV register is preset to the value set by the TOP bit of the CANTOCC register while the CCE bit of the CANCCCR register is set.

In addition, the state machine of the Tx and Rx handlers are held in idle state while the CCE bit in the CANCCCR register = 1.

The following registers can be written only while CCE bit in the CANCCCR register = “0”.

- CANTXBAR
- CANTXBCR

While the INIT bit of the CANCCCR register = “1” and the CCE bit of the CANCCCR register = “1”, the TEST bit of the CANCCCR register and the MON bit of the CANCCCR register can be set only by the software. Both bits can be reset at any time. The DAR bit of the CANCCCR register can be set/reset only while the INIT bit of the CANCCCR register = “1” and the CCE bit of the CANCCCR register = “1”.

[Note]

- **It is recommended that, after a hardware reset, a valid parity be created and initialized by writing 0x00000000 or similar to each word of message RAM. This prevents interrupting the BEU bit of the CANIR register by reading from an uninitialized message RAM section.**

19.3.2.2 Normal Operation

When CAN FD Controller is initialized and the INIT bit in the CANCCCR register is reset to zero, CAN FD controller is synchronized with the CAN bus and ready for communication.

After passing the acceptance filtering, the received message, including the message ID and DLC, is stored in a dedicated Rx buffer, or Rx FIFO0 or Rx FIFO1.

A dedicated Tx buffer and/or Tx FIFO or Tx queue can be initialized or updated for the messages to be transmitted. Automatic transmission when receiving a remote frame is not implemented.

19.3.2.3 Operation of CAN FD

There are two formats for the transmission of CAN FD frames, the first being a CAN FD frame without bit rate switching. The second is a CAN FD frame where the control field, data field and CRC field are transmitted at a higher bit rate than the beginning and ending of the frame.

Bits previously reserved in CAN frames with 11-bit identifiers and bits initially reserved in CAN frames with 29-bit identifiers will now be decoded as FDF bits. FDF = recessive indicates a CAN FD frame, while FDF = dominant indicates a classic CAN frame. In a CAN FD frame, the two bits following the FDF, res and BRS, determine whether the bit rate in this CAN FD frame is switched or not. Bit rate switching in a CAN FD is indicated by res=dominant and BRS=recessive. The res=recessive encoding is reserved for future protocol extensions. When CAN FD controller receives a frame with FDF=recessive and res=recessive, it sets the PXE bit in the CANPSR register to notify a protocol exception event. When protocol exception handling is enabled (PXHD bit in the CANCCCR register = “0”), the operating state changes from Receiver (ACT bit in the CANPSR register = “10”) to Integrating (ACT bit in the CANPSR register = “00”) at the next sample point. When protocol exception handling is disabled (PXHD bit in the CANCCCR register = “1”), CAN FD controller handles the recessive res bit as a form error and responds with an error frame.

CAN FD operation is enabled by setting the FDOE bit in the CANCCCR register. When the FDOE bit in the CANCCCR register = “1”, transmitting and receiving CAN FD frames are enabled. Transmitting and receiving classic CAN frames are possible at all times. Whether to transmit a CAN FD frame or a classic CAN frame can be set with the FDF bit of the respective Tx buffer element. When the FDOE bit in the CANCCCR register = “0”, the received frame is interpreted as a classic CAN frame and an error frame is transmitted in case a CAN FD frame is received. When the CAN FD operation is disabled, no CAN FD frame is transmitted even if the FDF bit of the Tx buffer element is set. Only while the INIT bit of the CANCCCR register and the CCE bit of the CANCCCR register are set, the FDOE bit of the CANCCCR register and the BRSE bit of the CANCCCR register can be changed.

When the FDOE bit in the CANCCCR register = “0”, the settings of the FDF and BRS bits are ignored and the frame is transmitted in classic CAN format. When the FDOE bit in the CANCCCR register = “1” and the BRSE bit in the CANCCCR register = “0”, only the bit FDF of the Tx buffer element is evaluated. When the FDOE bit of the CANCCCR register = “1” and the BRSE bit of the CANCCCR register = “1”, the CAN FD frame can be transmitted by bit rate switching. All Tx buffer elements with the FDF and BRS bits set are transmitted in CAN FD format with bit rate switching.

Mode change during CAN operation is recommended only under the following conditions:

- When the failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case the CAN FD bit rate switching option for transmission is disabled.
- In system start-up, all nodes transmit classic CAN messages until it is confirmed that they can communicate in CAN FD format. When it is confirmed, all nodes will switch to CAN FD. Operation is necessary in response to this.
- The wake-up message in CAN Partial Networking must be transmitted in classic CAN format.
- End-of-line programming performed when not all nodes support CAN FD. Nodes other than CAN FD will be held in silent mode until the setting configuration is complete. Afterwards, all nodes will switch to classic CAN communication.

In the CAN FD format, the DLC coding is different from the standard CAN format. DLC Codes 0 to 8 are coded the same as standard CAN, but Codes 9 to 15, which are all 8-byte data fields coded in standard CAN, are coded as shown below Table19-20.

Table19-20 DLC Coding in CAN FD

DLC	9	10	11	12	13	14	15
Data Byte	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing set in the CANNBTP register is used. The subsequent CAN FD data phase uses the bit timing of the data phase set in the CANDBTP register. The bit timing is switched either from the data phase timing by the CRC delimiter or when an error is detected, whichever comes first.

The maximum configurable bit rate of the CAN FD data phase depends on the CAN clock frequency (CANCLK). Example: With a CAN clock frequency of 20 MHz and a minimum configurable bit time of 4 tq, the bit rate of the data phase is 5 Mbit/s.

For both CAN FD and CAN FD data frame formats with bit rate switching, the value of the ESI bit is determined by the error status of the transmitter at the start of transmission. If the transmitter is error passive, the ESI is transmitted to the recessive, otherwise it is transmitted to the dominant.

19.3.2.4 Transmitter Delay Compensation

In the data phase of CAN FD transmission, only one node is transmitting and all other nodes are receiving.
The bus line length has no effect. When transmitting through the CAN0_TXD pin, CAN FD controller receives the transmitted data from the local CAN transceiver through the CAN0_RXD pin. The received data is delayed by the transmitter's delay. In case this delay is greater than TSEG1 (the time segment before the sample point), a bit error is detected. To allow even shorter data phase bit times than the transmitter delay, delay compensation is implemented. Without transmitter delay compensation, the bit rate of the data phase of the CAN FD frame is limited by the transmitter delay.

The protocol unit of CAN FD controller implements a delay compensation mechanism to compensate for the transmitter delay, thereby allowing transmission at higher bit rates during the CAN FD data phase independent of the delay of the specific CAN transceiver.

To check for bit errors during the data phase of the transmitting node, the delayed transmission data is compared with the received data at the secondary sample point (SSP). When a bit error is detected, the transmitter will respond to this bit error at the next normal sample point. The delay compensation is disabled at all times during the arbitration phase.

Transmitter delay compensation allows settings wherein the data bit time is less than the transmitter delay and its details are provided in ISO 11898-1:2015. This function is enabled by setting the TDC bit in the CANDBTP register.

The receive bits are compared to the transmit bits at the SSP. The SSP position is defined as the sum of the measured delay from the CAN0_TXD pin, the transmit output of CAN FD controller via the transceiver, to the CAN0_RXD pin, the receive input, plus the transmitter delay compensation offset set by the TDCO bit in the CANTDCR register is added. The transmitter delay compensation offset is used to adjust the SSP position in the receive bits (e.g., half the bit time of the data phase). The position of the second sample point is truncated to the next integer number mtq.

The TDCV bit in the CANPSR register indicates the actual transmitter delay compensation value. The TDCV bit in the CANPSR register is cleared when the INIT bit in the CANCCCR register is set, and is updated for each FD frame transmission while the TDC bit in the CANDBTP register is set.

For the transmitter delay compensation implemented in CAN FD controller, the following boundary conditions need to be considered:

- The sum of the measured delay from the CAN0_TXD pin to the CAN0_RXD pin and the TDCO bit in the configured transmitter delay compensation offset CANTDCR register must be less than or equal to 6 bit time in the data phase.
- The sum of the measured delay from the CAN0_TXD pin to the CAN0_RXD pin and the TDCO bit in the configured transmitter delay compensation offset CANTDCR register must be less than or equal to 127mtq. In case this sum is greater than 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter and stops checking the receive bits at the SSP.

○ Transmitter Delay Compensation Measurement
When transmitter delay compensation is enabled with TDC bit in the CANDBTP register = “1”, the measurement starts on the falling edge from the FDF bit to the res bit in the CAN FD frame for each transmitter. When this edge is confirmed on the CAN0_RXD pin, the receive input of the transmitter, the measurement stops. The resolution of this measurement is 1 mtq.

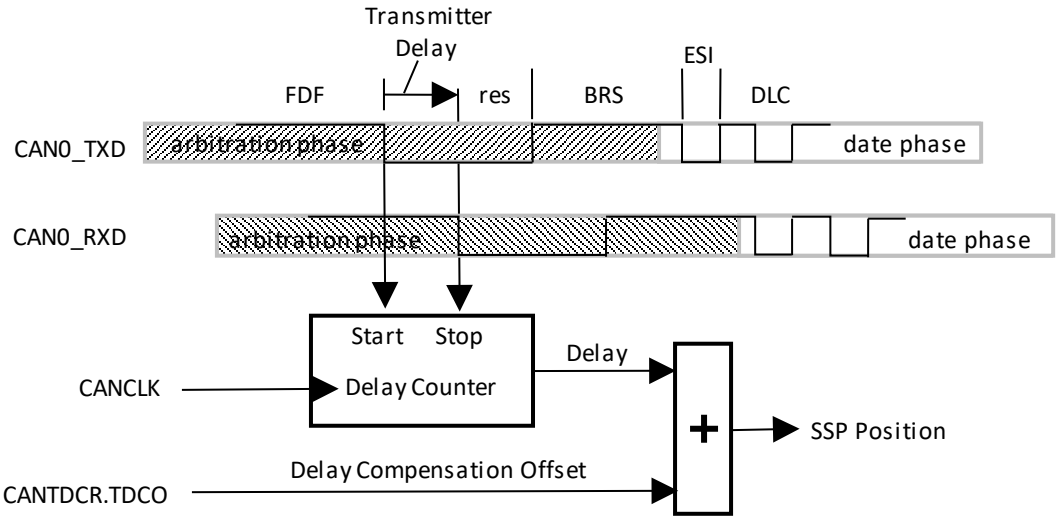


Figure19-8 Measurement of Transmitter Delay

To avoid a dominant glitch in the received FDF bit causing the delay compensation measurement to terminate before the falling edge of the received res bit, resulting in an earlier SSP position, the use of the transmitter delay correction filter window can be enabled by setting the TDCF bit in the CANTDCR register. This defines the minimum value of the SSP position. The edge of the dominant on the CAN0_RXD pin, which is an early SSP position, is ignored in the transmitter delay measurement. The measurement stops when the SSP position is at least the TDCF bit of the CANTDCR register and the CAN0_RXD pin is low.

[Note]

- **CAN FD controller performs transmitter delay measurement at all times, but does not support static (fixed) transmitter delay configurations.**

19.3.2.5 Restricted Operation Mode

In restricted operation mode, a node can receive data frames and remote frames and give acknowledge to valid frames, but it cannot transmit data frames, remote frames, active error frames, or overload frames. In case of error or overload condition, the dominant bit is not transmitted and the CAN communication is resynchronized after waiting for bus idle condition to occur. The error counter (REC bit in CANECR register and TEC bit in CANECR register) freezes while the error logging (CEL bit in CANECR register) is active. The software can set CAN FD controller to restricted operation mode by setting the ASM bit in the CANCCCR register. This bit can be set by the software only when both the CCE bit in the CANCCCR register and the INIT bit in the CANCCCR register are set to “1”. This bit can be reset by the software at any time.

The restricted operation mode is automatically entered when the Tx handler cannot read the data from message RAM in time. To release the restricted operation mode, the software must reset the ASM bit in the CANCCCR register.

The restricted operation mode can be used in such an application that adapts to different CAN bit rates. In this case, the application tests different bit rates and releases the restricted operation mode after receiving a valid frame.

[Note]

- **The restricted operation mode must not be combined with the loopback mode (internal or external).**

19.3.2.6 Bus Monitoring Mode

CAN FD controller is set to bus monitoring mode by setting the MON bit in the CANCCCR register to “1”. In bus monitoring mode (see ISO 11898-1:2015, 10.14 Bus Monitoring), CAN FD controller can receive valid data frames and valid remote frames, but cannot initiate transmission. In this mode, only the recessive bits on the CAN bus are transmitted. In case CAN FD controller needs to transmit a dominant bit (ACK bit, overload flag, active error flag), the CAN bus may remain in the recessive state, but the bit is rerouted internally so that CAN FD Controller monitors this dominant bit. In bus monitoring mode, the CANTXBRP register is held in the reset state.

Bus monitoring mode can be used to analyze the traffic on the CAN bus without affecting the CAN bus traffic by transferring dominant bits. Figure19-9 shows the connection of the CAN0_TXD pin and the CAN0_RXD pin to CAN FD controller in bus monitor mode.

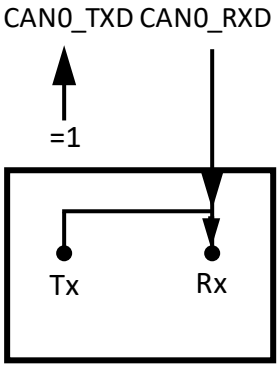


Figure19-9 Pin Control in Bus Monitor Mode

19.3.2.7 Disabled Automatic Retransfer

According to the CAN specifications (see ISO 11898-1:2015, 8.3.4 Recovery Management), CAN FD Controller provides a means for automatic retransfer of frames that have lost arbitration or that have been disturbed by errors during transmission. Automatic retransfer is enabled by default after a reset. To support time-triggered communication as described in ISO 11898-1:2015, 9.2, automatic retransfer can be disabled through the DAR bit in the CANCCCR register.

○ Frame Transmission in DAR Mode
In DAR mode, all transmissions are automatically canceled after they have been initiated on the CAN bus. The TRPx bit in the CANTXBRP register of the Tx buffer is reset after a successful transmission, in case the transmission has not yet started at the time of cancellation, in case it has been interrupted due to loss of arbitration, or in case an error has occurred during frame transmission.

- Transmission successful
  - The TOx bit of the CANTXBTO register of the corresponding Tx buffer is set
  - The CFx bit in the CANTXBCF register of the corresponding Tx buffer is not set

- Transmission successful despite cancellation
  - The TOx bit of the CANTXBTO register of the corresponding Tx buffer is set
  - The CFx bit in the CANTXBCF register of the corresponding Tx buffer is set.

- Arbitration has been lost or frame transmission has been interrupted
  - The TOx bit of the CANTXBTO register of the corresponding Tx buffer is not set
  - The CFx bit in the CANTXBCF register of the corresponding Tx buffer is set.

In case the frame transmission is successful and Tx event storage is enabled, the Tx Event FIFO element is written with ET = 10 (transmission despite cancellation).

19.3.2.8 Power Down (Sleep Mode)

CAN FD controller can be set to power down mode through the CSR bit in the CANCCCR register.

After all pending transmit requests are completed, CAN FD controller waits until bus idle condition is detected. CAN FD controller then sets the INIT bit of the CANCCCR register to “1” to prevent further CAN transfers. CAN FD controller confirms that it is ready for power-down by setting the CSA bit in the CANCCCR register to “1”. Register accesses other than the INIT bit of the CANCCCR register, which is retained at “1”, can be performed before SYSCLK and CANCLK are turned off.

[Note]

- **When the CAN bus is considerably disturbed, the idle state may not be reached and the INIT bit in the CANCCCR register may not be set by CAN FD controller. This situation can be detected by polling the ACT bit in the CANPSR register. In case CAN FD controller does not enter the idle state, the software can write INIT bit in the CANCCCR register = “1”, which immediately stops the CAN communication of CAN FD Controller, regardless of whether transmission/reception is in progress.**

- **To exit the power down mode, the software must turn on the clock before resetting the CSR bit in the CANCCCR register. CAN FD controller recognizes this by resetting the CSA bit in the CANCCCR register. The software can then resume CAN communication by resetting the INIT bit in the bit CANCCCR register.**

19.3.2.9 Test Mode

To enable write access to the CANTEST register (see section “19.2.7”), the TEST bit in the CANCCCR register must be set to 1. This allows to set the test mode and test functions.

Four output functions are available for the CAN0_TXD pin, which is CAN transmission, by setting the TX bit in the CANTEST register. In addition to the serial data output, which is a function by the initial settings after a reset, the CAN sample point signal can be driven to monitor the bit timing of CAN FD controller, which can drive a constant dominant or recessive value. The actual value of the CAN0_RXD pin can be read from the RX bit of the CANTEST register. Both functions can be used to check the physical layer of the CAN bus.

Due to the synchronization mechanism between the CANCLK and SYSCLK domains, the SYSCLK cycle may be delayed by a few cycles in the SYSCLK between the write to the TX bit of the CANTEST register and the display of the new configuration on the CAN0_TXD pin. This is also the case when reading the CAN0_RXD pin via the RX bit of the CANTEST register.

[Note]

- **Use the test mode only for production testing or self-testing. The software control of the CAN0_TXD pin interferes with all CAN protocol functions. It is not recommended to use the test mode for applications.**

○ External Loopback Mode
CAN FD controller can set to external loopback mode by setting the LBCK bit in the CANTEST register to “1”. In loopback mode, CAN FD controller handles its own transmit messages as receive messages and stores them in the Rx buffer or Rx FIFO (in case they pass the acceptance filtering). Figure19-10 shows the connection of the CAN0_TXD and CAN0_RXD pins to CAN FD controller in external loopback mode.

This mode is provided for hardware self-testing. To be independent from external stimulation, CAN FD controller ignores acknowledge errors (recessive bits sampled in the acknowledge slot of the data/remote frame) in loopback mode. In this mode, CAN FD controller performs internal feedback from the Tx output to the Rx input. The actual value of the CAN0_RXD pin is ignored by CAN FD controller. The transmitted messages can be monitored on the CAN0_TXD pin.

○ Internal Loopback Mode
Internal loopback mode is set by setting the LBCK bit in the CANTEST register and the MON bit in the CANCCCR register to “1”. This mode can be used for "hot cell testing". Namely, CAN FD controller can perform testing without affecting the running CAN system connected to the CAN0_TXD and CAN0_RXD pins. In this mode, the CAN0_RXD pin is disconnected from CAN FD controller and the CAN0_TXD pin is held recessive. Figure19-10 shows the connection of the CAN0_TXD and CAN0_RXD pins to CAN FD controller in internal loopback mode.

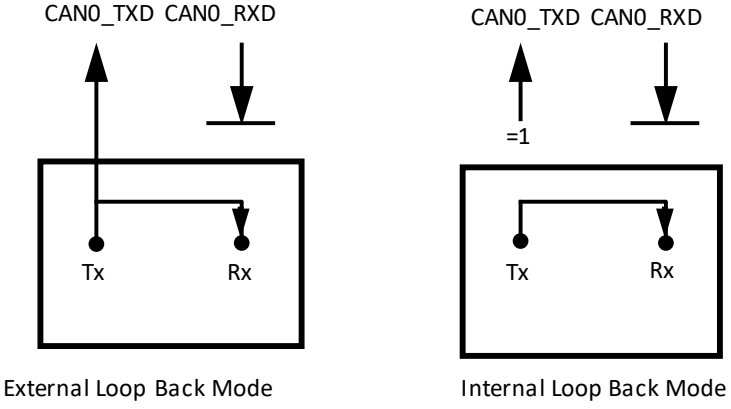


Figure19-10 Pin Control in Loopback Mode

19.3.3 Generation of Timestamp

19.3.3.1 Generation of Internal Timestamp

For internal timestamp generation, CAN FD controller provides a 16-bit wrap-around counter. The TCP in the prescaler CANTSCC register can be set to clock the counter in multiples of the CAN bit time (1 to 16). The counter can be read through the TSC bit in the CANTSCV register. A write access to the CANTSCV register resets the counter to zero. When the timestamp counter wraps around, the TSW bit of the CANIR register is set.

At the start of a frame reception/transmission (SOF), the counter value is taken in and stored in the timestamp section of the Rx buffer/Rx FIFO (RXTS [15:0]) or Tx event FIFO (TXTS [15:0]).

19.3.4 Timeout Counter

To signalize the timeout conditions of the Rx FIFO0, Rx FIFO1 and Tx event FIFOs, CAN FD controller supplies a 16-bit timeout counter. It operates as a down counter and uses a prescaler controlled by the TCP bit in the same CANTSCC register as the timestamp counter. The timeout counter is set through the CANTOCC register. The actual counter value can be read from the TOC bit in the CANTOCV register. The timeout counter can be activated only while the INIT bit in the CANCCCR register = “0”. The timeout counter stops when the INIT bit in the CANCCCR register = “1”, for example, when CAN FD controller enters the Bus_Off state.

The operating mode is selected by the TOS bit in the CANTOCC register. In continuous mode operation, the counter is activated when the INIT bit of the CANCCCR register is reset. By writing to CANTOCV, the counter is preset to the value set by the TOP bit of the CANTOCC register and continues to count down.

In case the timeout counter is controlled by one of the FIFOs, an empty FIFO will cause the counter to be preset to the value set by the TOP bit in the CANTOCC register. The down count starts when the first FIFO element is stored. Writing to CANTOCV has not effect.

When the counter reaches zero, the TOO bit in the CANIR register is set. In continuous mode, the TOP bit of the CANTOCC register restarts the counter immediately.

[Note]

- **The clock signal of the timeout counter is based on the sample point signal of the CAN core. Therefore, the point at which the timeout counter is decremented may vary depending on the synchronization/resynchronization mechanism of the CAN core. Also, when the bit rate switching function of CAN FD is used, the clock of the timeout counter is different between the arbitration and data fields.**

19.3.5 Reception Processing

The Rx handler controls the acceptance filtering, the transfer of received messages to either the Rx buffer or one of the two Rx FIFOs, and the Put and Get indexes of the Rx FIFO.

19.3.5.1 Acceptance Filtering

CAN FD controller can configure two acceptance filters, one for standard identifiers and the other for extended identifiers. These filters can be assigned to Rx buffers or Rx FIFO0, 1. In acceptance filtering, each list of filters runs from element #0 to the first matching element. The acceptance filtering stops at the first matched element. The following filter elements are not evaluated for this message:

The main functions are as follows:
- Each filter element can be configured as follows:
  - Range Filter
  - Dedicated ID Filter
  - Classic Bit Mask Filter
- Each filter element can be configured for acceptance or discard filtering
- Each filter element can be individually enabled or disabled
- Filters are checked sequentially, and the check stops at the first matching filter element

Associated registers are as follows:
- CANGFC
- CANSIDFC
- CANXIDFC
- CANXIDAM

Depending on the setting of the filter elements (SFEC/EFEC), a match will trigger one of the following actions:
- Stores the received frame in FIFO 0 or FIFO 1
- Saves the received frame to the Rx buffer
- Discards the received frame
- Sets the HPM bit in the CANIR register
- Sets the HPM bit in the CANIR register and stores the received frame in FIFO 0 or FIFO 1.

Acceptance filtering is initiated after a full identifier is received. When acceptance filtering is completed and a matching Rx buffer or Rx FIFO is found, the message handler begins writing the received message data to the matching Rx buffer or Rx FIFO in 32-bit portions. In case the CAN protocol controller detects an error (e.g. CRC error), this message will be discarded with the following effects on the Rx buffer or Rx FIFO to be affected.

- Rx buffer
  - A new data flag of the matching Rx buffer is not set, but the Rx buffer has been (partially) overwritten with the received data. For the error type, refer to the LEC bit in the CANPSR register and the DLEC bit in the CANPSR register.
- Rx FIFO
  - The Put index of the matching Rx FIFO will not be updated, but the associated Rx FIFO element will be (partially) overwritten with the received data. For the error type, refer to the LEC bit in the CANPSR register and the DLEC bit in the CANPSR register. In case the matching Rx FIFO operates in overwrite mode, the boundary conditions described at section “19.3.5.2” must be considered.

[Note]
- **When an accepted message is written to one of the two Rx FIFOs or to an Rx buffer, the unmodified receive identifier is stored regardless of the filter used. The result of the acceptance filtering process strongly depends on the sequence of filter elements that are set.**

○ Range Filter
The filter matches all received frames with message IDs within the range defined by EF1ID/EF2ID of SF1ID/SF2ID.

When range filtering is used in conjunction with extended frames, there are two possibilities:
- EFT= “00”
  The message ID of the received frame is logically multiplied by the extended ID and mask (CANXIDAM) before the range filter is applied
- EFT= “11”
  The logical product of the extended ID and mask (CANXIDAM) is not used for range filtering

○ Filtering for Specific IDs
A filter element can be configured to perform filtering for one or two specific message IDs. In order to filter one specific message ID, the filter element must be set to SF1ID=SF2ID, EF1ID=EF2ID.

○ Classic Bit Mask Filter
Classic bit mask filtering is used to filter out groups of message IDs by masking the single bits of the received message IDs. In classic bit mask filtering, SF1ID/EF1ID is used as the message ID filter and SF2ID/EF2ID is used as the filter mask.

When designating 0 bit for the filter mask, the corresponding bit position of the configured ID filter will be masked. Only the bits of the received message ID for which the corresponding mask bit is “1” are subject to acceptance filtering.

In case all mask bits are set to “1”, the received message ID and the message ID filter match only if they are identical. In case all mask bits are 0, all message IDs match.

○ Standard Message ID Filtering
Figure19-11 below shows the flow of standard message ID (11-bit identifier) filtering. The standard message ID filter element is described at section “19.3.1.5”.

The filtering, controlled by the global filter setting (CANGFC) and the standard ID filter setting (CANSIDFC), compares the message ID, remote transmit request bit (RTR), and identifier extension bit (IDE) of the received frame against the list of configured filter elements.

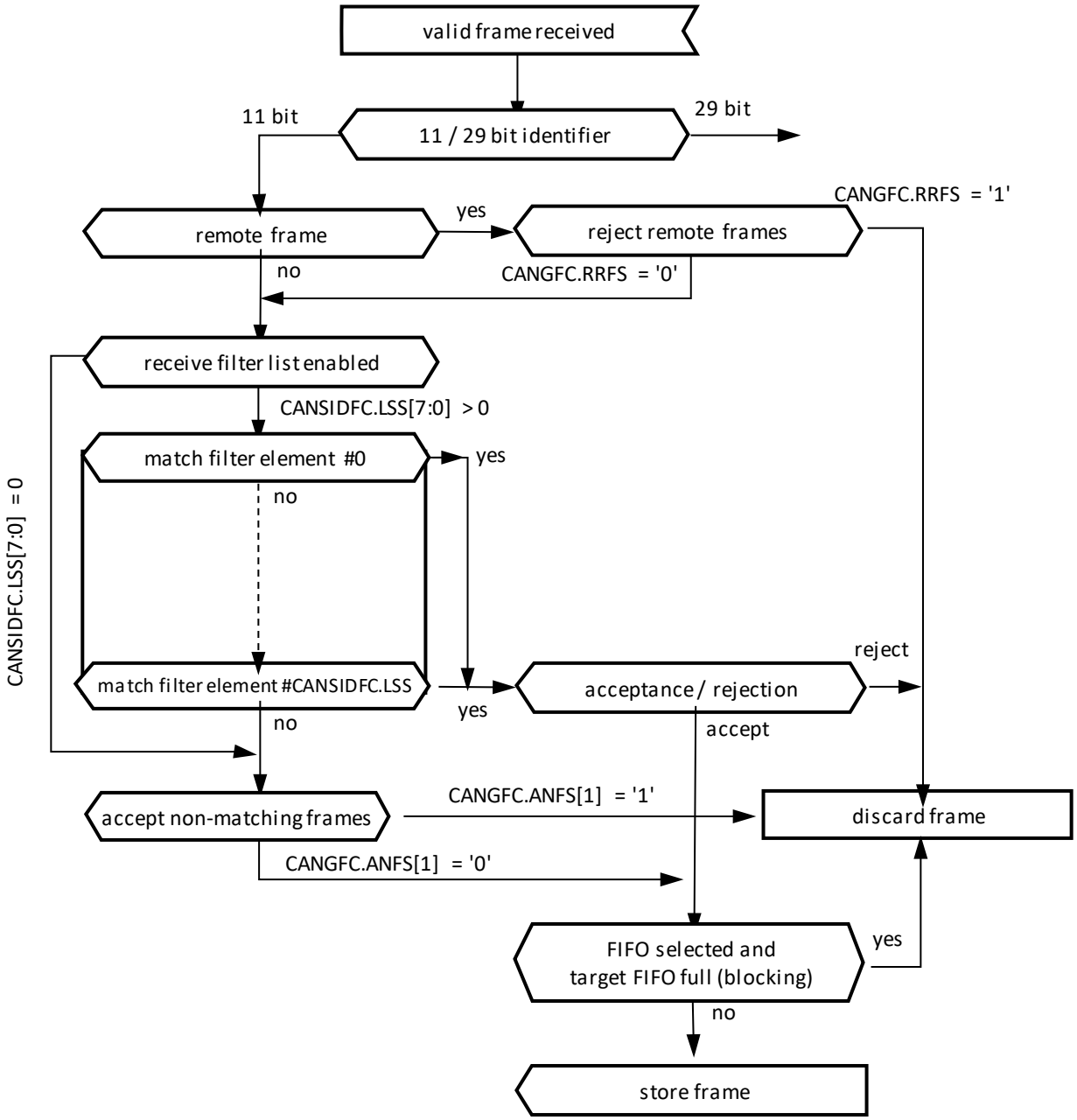


Figure19-11 Standard Message ID Filter Path

○ Extended Message ID Filtering

Figure19-12 below shows the flow for extended message ID (29-bit identifier) filtering. The extended message ID filter element is described at section “19.3.1.6”.

The filtering, controlled by the global filter setting (CANGFC) and the extended ID filter setting (CANXIDFC), compares the message ID, remote transmit request bit (RTR) and identifier extension bit (IDE) of the received frame against the list of configured filter elements.

The logical product of the extended ID and mask (CANXIDAM) is logically multiplied by the identifier received before the filter list is executed.

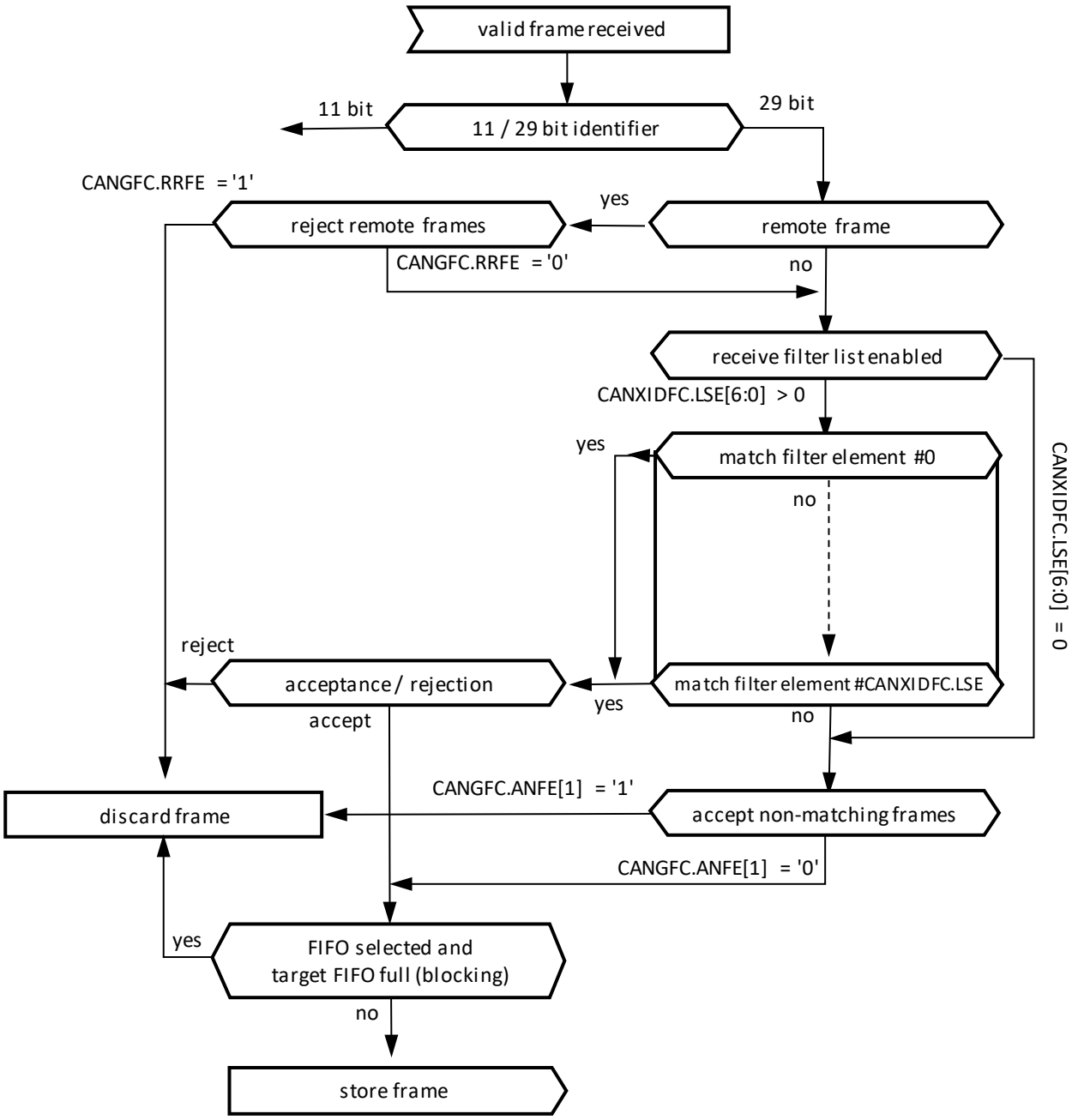


Figure19-12 Extended Message ID Filter Path

19.3.5.2 Rx FIFO

Rx FIFO 0 and Rx FIFO 1 can each be configured to retain a maximum of 64 elements. The configuration of the two Rx FIFOs is performed through the CANRXF0C and CANRXF1C registers.

Received messages that pass the acceptance filtering are forwarded to the Rx FIFO configured by the matching filter element. For information on the filter mechanism available for Rx FIFO 0 and Rx FIFO 1, see section “19.3.5.1”. For more information on the Rx FIFO element, see section “19.3.1.2”.

An Rx FIFO watermark can be used to avoid Rx FIFO overflow. When the Rx FIFO filling level reaches the Rx FIFO watermark set by the FnWM bit in the CANRXFnC register, the RFnW bit in the CANIR register is set. When the Rx FIFO Put index reaches the Rx FIFO Get index, the Rx FIFO full status is notified by the FnF bit in the CANRXFnS register. Also, the RFnF bit in the CANIR register is set.

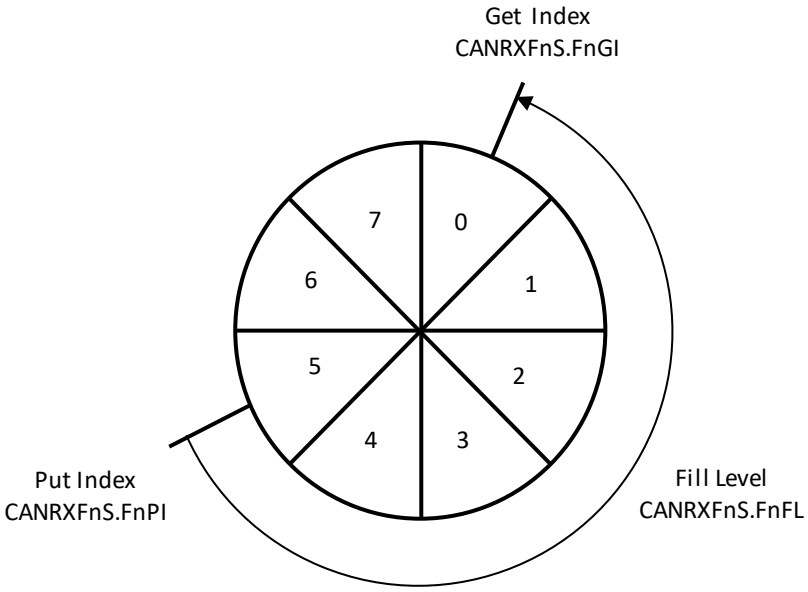


Figure19-13 Rx FIFO Status

When reading from the Rx FIFO, the FnGI bit in the CANRXFnS register must be added to the FnSA bit in the CANRXFnC register, the start address of the corresponding Rx FIFO.

Table19-21 Rx Buffer/FIFO Element Size

RBDS [2:0] bit in the CANRXESC register FnDS [2:0] bit in the CANRXESC register	Data field [byte]	FIFO element size [number of RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

○ Rx FIFO Blocking Mode
The blocking mode of the Rx FIFO is set by the FnOM bit in the CANRXFnC register = 0. This is an operation mode by default after resetting the Rx FIFO.

When the Rx FIFO reaches the full state (FnPI bit in CANRXFnS register = FnGI bit in CANRXFnS register), at least one message is read and no more messages are written to the corresponding Rx FIFO until the Rx FIFO Get index is incremented. The full state of the Rx FIFO is signaled by the FnF bit in the CANRXFnS register = “1”. Also, the RFnF bit in the CANIR register is set.

In case a message is received with the corresponding Rx FIFO being full, the message is discarded and the message lost status is reported with RFnL bit in the CANRXFnS register = “1”. Also, the RFnL bit in the CANIR register is set.

○ Rx FIFO Overwrite Mode
Rx FIFO overwrite mode is set by FnOM bit in the CANRXFnC register = “1”.

When the full state of the Rx FIFO (FnPI bit in CANRXFnS register = FnGI bit in CANRXFnS register) is signalled by the FnF bit in the CANRXFnS register = “1”, the next message accepted into the FIFO overwrites the oldest FIFO message. Both the Put and Get indexes are incremented by one.

When the Rx FIFO operates in overwrite mode and the full state of the Rx FIFO is signalled, then reading of the Rx FIFO element must start from at least Get index + 1. This is because a received message may be written to message RAM (Put index) while the software is reading from Message RAM (Get index). In this case, inconsistent data may be read from each Rx FIFO element. This problem can be avoided by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on the speed at which the software accesses the Rx FIFO. Figure19-14 shows two offsets for the Get index when reading the Rx FIFO. In this case, the two messages stored in Elements 1 and 2 are lost.

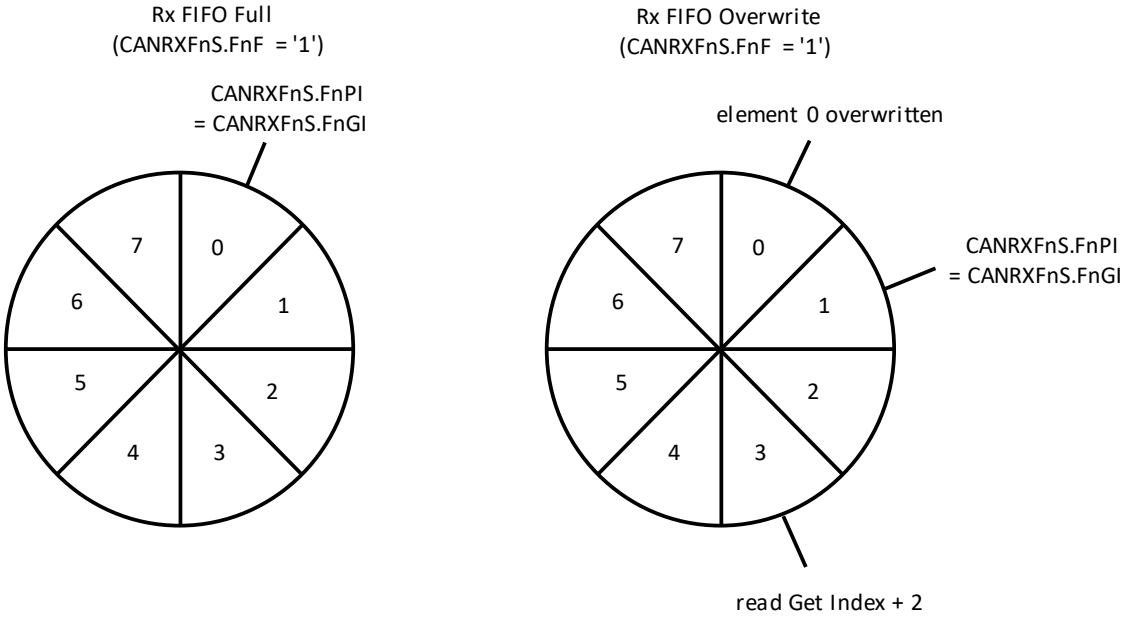


Figure19-14 Handling of Rx FIFO Overflow

After reading from the Rx FIFO, the number of the last element read must be written to the FnA bit of the CANRXFnA register. This increments the Get index to that element number. In case the Put index is not incremented to this Rx FIFO element, the full state of the Rx FIFO is reset (FnF bit in the CANRXFnS register = “0”).

19.3.5.3 Dedicated Rx Buffer

CAN FD controller supports up to 64 dedicated Rx buffers. The start address of the dedicated Rx buffer section is set by the RBSA bit in the CANRXBC register.

For each Rx buffer, a standard or extended message ID filter element with SFEC/EFEC= “111” and SFID2/EFID2 [10:9] = “00” must be set (see section “19.3.1.5” and “19.3.1.6”).

When a received message is accepted by the filter element, the message is stored in the Rx buffer of message RAM referenced by the filter element. The format is the same as for the Rx FIFO element. Also, the DRX bit of the CANIR register is set.

Table19-22 Examples of Rx Buffer Filter Settings

Filter element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

When the last word of the matching received message is written to message RAM, a respective new data flag in the CANNDAT1 and 2 registers is set. While the new data flag is being set, each Rx buffer is locked against updates from received matching frames. The new data flag must be reset by the software by writing 1 to each bit position.

While the Rx buffer's new data flag is being set, the message ID filter element referencing this particular Rx buffer will not match while the acceptance filtering continues. Following the message ID filter element may cause the received message to be stored in a different Rx buffer or Rx FIFO, or may cause the message to be discarded, depending on the filter setting.

○ Processing of Rx Buffer
- Resets the DRX bit in the CANIR register
- Reads the CANNDAT1 and 2 registers
- Reads a message from the message RAM
- Resets the new data flag of the processed message

19.3.5.4 Debug Function

The debug functions described in the following sections regarding debug messages A/B/C are not supported.

19.2.34 Rx FIFO 1 Status Register (CANRXF1S)
19.3.1.5 Standard Message ID Filter Element
19.3.1.6 Extended Message ID Filter Element

19.3.6 Transmission Processing

The Tx handler processes transmit requests for dedicated Tx buffers, Tx FIFOs, and Tx queues. It controls the transfer of transmit messages to the CAN core, Put and Get indexes, and the Tx event FIFO. Up to 32 Tx buffers can be configured for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be set individually for each Tx buffer element. For more information on the Tx buffer element, see Section “19.3.1.3 Tx Buffer Element”. Table19-23 below shows the possible settings for transmitting frames.

Table19-23 Possible Settings for Frame Transmission

CANCCCR		Tx buffer element		Frame transmission
BRSE	FDOE	FDF	BRS	
Ignored	0	Ignored	Ignored	Classic CAN
0	1	0	Ignored	Classic CAN
0	1	1	Ignored	CAN FD without bit rate switching
1	1	0	Ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the CANTXBRP register is updated, the Tx handler initiates a Tx scan of Message RAM to check for the highest priority pending Tx request (Tx buffer with the lowest message ID). The CANTXBRP register is updated when a transmission is completed, or after the software has requested a transmission by writing to the CANTXBAR register, or when the software cancels a pending transmission by writing to the CANTXBCR register.

In case there are new scan results, the temporary buffer retaining the first four preloaded RAM words of the two messages with the highest Tx priority is updated. The time required for Tx scan and preload depends on the SYSCLK clock frequency and the number of Tx buffers configured.

Since the Tx scan of message RAM is time consuming, the following scenarios are possible:

- Temporary buffers are preloaded before the completion of transmission/reception in progress
- The temporary buffer preload is not completed before the completion of transmission/reception in progress. Tx messages cannot be initiated at the first opportunity. In this case, other nodes can initiate transmission without arbitrating against the transmission messages of CAN FD controller. In case the other messages have a lower priority, it can be considered an inversion of the external priority.

19.3.6.1 Transmission Pause

The transmission pause function is intended for use in the CAN system where the CAN message identifiers are designated (permanently) to a specific value and cannot be easily changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, but for certain applications the relative arbitration priorities are reversed. Therefore, when one ECU transmits a burst of CAN messages, the CAN messages of the other ECUs may be delayed due to the low CAN arbitration priority of those messages.

For example, when CANECU-1 has the transmission pause function enabled and the application software requests the transmission of four messages, it will wait for two CAN bit bus idle times after the successful transmission of the first message and before it is allowed to start the next requested message. In case there are other ECUs with pending messages, those messages will be initiated during the idle time and do not need to be arbitrated with the next message in ECU-1. After receiving the message, ECU-1 can start the next transmission as soon as the received message is released from the CAN bus.

The transmission pause function is controlled by the TXP bit in the bit CANCCCR register. When this bit is set, each time CAN FD controller successfully transmits a message, it will pause for two CAN bits before starting the next transmission. This allows other CAN nodes in the network to transmit messages even when the identifier ahead of the message is low. In the default setting after a reset, transmission pause is disabled (TXP bit in the CANCCCR register = 0).

This function disables burst transmissions from a single node and protects against "Babbling Idiot" scenarios where an application program accidentally requests too many transmissions.

19.3.6.2 Dedicated Tx Buffer

The dedicated Tx buffer is intended for message transmission under the full control of the software. Each dedicated Tx buffer is configured with a specific message ID. When multiple Tx buffers are configured with the same message ID, the Tx buffer with the lowest buffer number will be transmitted first.

In case the data section has been updated, transmission is requested by Add Request through the ARn bit in the CANTXBAR register. The requested message is arbitrated internally with messages from the optional Tx FIFO or Tx queue and externally with messages on the CAN bus and transmitted according to the message ID.

The dedicated Tx buffer allocates an element-size 32-bit word in message RAM (see Table19-24). Therefore, the start address of the dedicated Tx buffer in message RAM is calculated by adding the Tx buffer index (0 to 31) = element size to the TBSA bit of the Tx buffer start address CANTXBC register.

Table19-24 Tx Buffer/FIFO/Queue Element Size

CANTXESC register TBDS	Data field [byte]	Element size [number of RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

19.3.6.3 Tx FIFO

Tx FIFO operation is configured by setting the TFQM bit in the CANTXBC register to 0. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the TFGI bit in the CANTXFQS register. After each transmission, the Get index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO allows messages with the same message ID from different Tx buffers to be transmitted in the order in which these messages were written to the Tx FIFO. CAN FD controller calculates the TFFL bit in the CANTXFQS register as the difference between the Get and Put indexes. This indicates the number of available (free) Tx FIFO elements.

The new transmit message must be written to the Tx FIFO starting from the Tx buffer referenced by the TFQPI bit in the CANTXFQS register. Add Request increments the Put index to the next empty Tx FIFO element. When the Put index reaches the Get index, Tx FIFO full (TFQF bit in the CANTXFQS register = “1”) is notified. In this case, no more messages can be written to the Tx FIFO until the next message has been transmitted and the Get index has been incremented.

When a single message is added to the Tx FIFO, transmission is requested by writing “1” to the CANTXBAR bit associated with the Tx buffer referenced by the Put index of the Tx FIFO.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx buffers starting at the Put index. Transmission is then requested through the CANTXBAR register. The Put index is then incremented by n periodically. The number of Tx buffers requested must not exceed the number of empty Tx buffers indicated by the Tx FIFO free level.

When a transmit request for the Tx buffer referenced by the Get index is canceled, the Get index is incremented to the next Tx buffer with a pending transmit request and the Tx FIFO free level is recalculated. In case transmit cancellation is applied to another Tx buffer, the Get index and FIFO free level are not changed.

The Tx FIFO element allocates an element-size 32-bit word in message RAM (see Table19-24). Therefore, the start address of the next available (free) Tx FIFO buffer is calculated by adding the TFQPI (0...31) bits of the CANTXFQS register = element size to the TBSA bits of the CANTXBC register.

19.3.6.4 Tx Queue

The operation of Tx queue is configured by setting the TFQM bit in the CANTXBC register to “1”. Messages stored in the Tx queue are transmitted starting with the message with the lowest message ID. In case multiple queue buffers are set for the same message ID, the queue buffer with the lowest buffer number is transmitted first.

The new message must be written to the Tx buffer referenced by the TFQPI bit in the CANTXFQS register. In case the Tx queue is full (TFQF bit in the CANTXFQS register = “1”), the Put index is not enabled and no more messages should be written to the Tx queue until at least one of the requested messages has been transmitted or a pending transmit request has been canceled.

The software can use the CANTXBRP register instead of the Put index, allowing messages to be placed in any Tx buffer without a pending transmit request.

The Tx queue buffer allocates an element size 32 bit word in message RAM (see Table19-24). Therefore, the start address of the next available (free) Tx queue buffer is calculated by adding the TBSA bit of the Tx buffer start address CANTXBC register to the TFQPI (0...31) bit of the CANTXFQS register = element size.

19.3.6.5 Mix of Dedicated Tx Buffers/Tx FIFO

In this case, the Tx buffers section of message RAM is subdivided into sets of dedicated Tx buffers and Tx FIFOs. The number of dedicated Tx buffers is set by the NDTB bit in the CANTXBC register. The number of Tx buffers allocated to the Tx FIFO is set by the TFQS bit in the CANTXBC register. When the TFQS bit in the CANTXBC register is set to "0", only the dedicated Tx buffers are used.

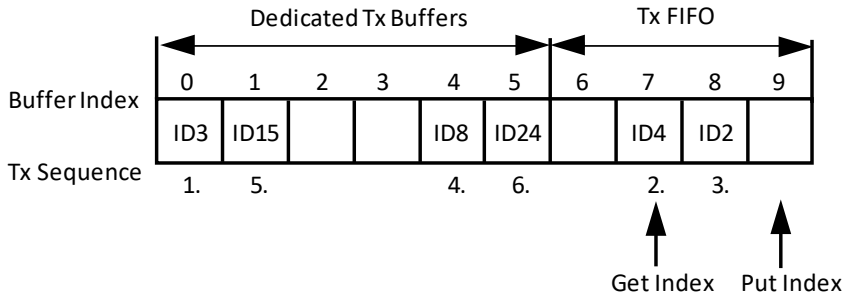


Figure19-15 Example of Mixed Configuration of Dedicated Tx Buffers/Tx FIFO

The prioritization of Tx is as follows:

- Scans the dedicated Tx buffers and the oldest pending Tx FIFO buffer (referenced by the TFGI bit in the TXFS register)
- The buffer with the lowest message ID has the highest priority and will be transmitted next

19.3.6.6 Mix of Dedicated Tx Buffers/Tx Queue

In this case, the Tx buffers section of message RAM is subdivided into sets of dedicated Tx buffers and Tx queues. The number of dedicated Tx buffers is set by the NDTB bit in the CANTXBC register. The number of Tx queue buffers is set by the TFQS bit in the CANTXBC register. When the TFQS bit in the CANTXBC register is set to "0", only the dedicated Tx buffers are used.

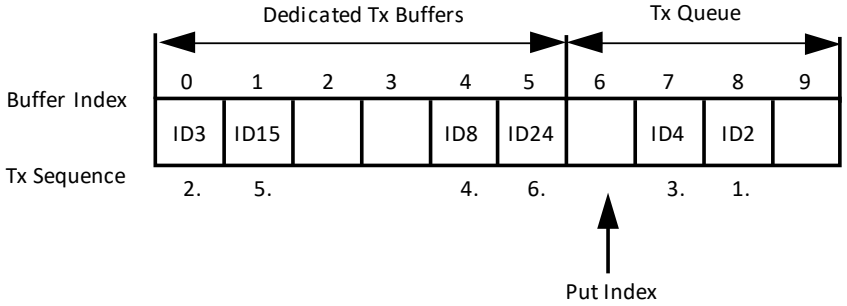


Figure19-16 Example of Mixed Configuration of Dedicated Tx Buffers/Tx Queue

The prioritization of Tx is as follows:

- Scans all Tx buffers with activated transmit request
- The Tx buffers with the lowest message ID has the highest priority and will be transmitted next

19.3.6.7 Transmission Cancellation

CAN FD controller supports transmit cancellation. This function is especially intended for gateway applications and AUTOSAR-based applications. To cancel a transmission requested from a dedicated Tx buffer or Tx queue buffer, the software must write 1 to the corresponding bit position (= number of Tx buffers) in the CANTXBCR register. Transmit cancellation is not intended for the operation of Tx FIFO.

A successful transmit cancellation is notified by setting the corresponding bit in the CANTXBCF register to "1".

When a transmit cancellation is requested during a transmission from the Tx buffer, the corresponding bit in the CANTXBRP register will remain set as long as the transmission is in progress. When the transmission is successful, the

corresponding bits in the CANTXBTO and CANTXBCF registers are set. When the transmission is not successful, the transmission is not repeated and only the corresponding bit in the CANTXBCF register is set.

[Note]

- **In case a pending transmission is canceled just before the transmission is initiated, a short-time window which does not initiate transmission follows, even if another message is pending on this node. This may allow another node to transmit a message with a lower priority than this node's second message.**

19.3.6.8 Tx Event Processing

CAN FD controller implements a Tx event FIFO in order to support Tx event processing. After CAN FD controller transmits a message on the CAN bus, the message ID and timestamp are stored in the Tx event FIFO element. In order to link a Tx event to a Tx event FIFO element, copy the message marker from the transmitted Tx buffer to the Tx event FIFO element.

Whether to use an 8-bit message marker or a 16-bit wide message marker can be set by the WMM bit in the CANCCCR register

[Note]

- **In case the WMM bit in the CANCCCR register = “1”, the internal timestamp is disabled (see Section “19.3.1.4”).**

The Tx event FIFO can be configured for up to 32 elements. For more information on the Tx event FIFO element, see Section “19.3.1.4”.

The purpose of the Tx event FIFO is to separate the processing of the transmit status information from the processing of the transmit message. This has the advantage, especially when operating a dynamically managed Tx queue, that the Tx buffer can be used for new messages immediately after a successful transmission. It is not necessary to save the transmit status information from the Tx buffer before overwriting the Tx buffer.

When the Tx event FIFO is filled up by the TEFF bits in the CANIR register, no more elements will be written to the Tx event FIFO until at least one element has been read and the Get index of the Tx event FIFO has been incremented. When a Tx event occurs while the Tx event FIFO is full, this event is discarded and the TEFL bit in the CANIR register is set.

To avoid overflow of the Tx event FIFO, a watermark value for the Tx event FIFO can be used. When the filling level of the Tx event FIFO reaches the watermark of the Tx event FIFO set by the EFWM bit in the CANTXEFC register, the TEFW bit in the CANIR register is set.

When reading from the Tx event FIFO, the EFSA bit of the CANTXEFC register must be added to the double the EFGI bits of the CANTXEFS register.

19.3.7 FIFO Acknowledge Processing

The Get indexes of the Rx FIFO0, Rx FIFO1, and Tx event FIFOs are controlled by writing to the corresponding FIFO Acknowledge indexes (see section “19.2.31”, “19.2.35”, “19.2.49”). Writing to the FIFO Acknowledge indexes sets the FIFO Get indexes to the FIFO Acknowledge indexes + 1, thereby updating the FIFO filling level. 2 use cases are available:

- In case only one element (the element pointed to by the Get index) is read from the FIFO, this Get index value is written to the FIFO Acknowledge index
- When a series of elements are read from a FIFO, it is sufficient to write the FIFO Acknowledge index only once at the end of the series of elements read, in order to update the Get index of the FIFO (the value is the index of the last element read).

Since the software has free access to message RAM of CAN FD controller, special care must be taken when reading the FIFO elements in any order (the Get index is not taken into account). This is useful when reading high priority messages from one of the two Rx FIFOs. In this case, the Acknowledge index of the FIFO should not be written. In this case, some of the old FIFO elements will be lost.

[Note]

- **The software must ensure that valid values are written to the FIFO Acknowledge index. CAN FD controller does not check for incorrect values.**

19.3.8 Access to Message RAM and Parity Errors

CAN message RAM is accessible from CAN FD controller, CPU or DMAC.
The circuit diagram is shown in Figure19-17.

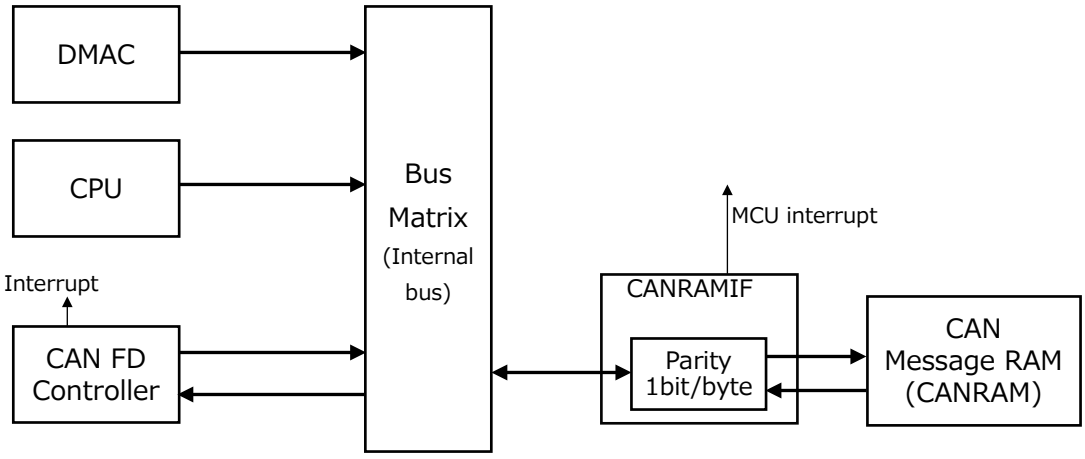


Figure19-17 CAN-related Memory Access Configuration

RAMIF is a functional block for CPU/DMAC access to RAM.
CANRAMIF is a functional block that provides access to the CANRAM from CPU/DMAC/CAN FD controller.

A data written to RAM from somewhere is added parity 1 bit per byte.
If the CAN_PAR_EN of the MCUINTEN register of the system control function is set to "1", the CAN_PAR of the MCUISTAT register is set to "1" and the MCU interrupt request is occurred when a parity error is detected when RAM is read by either CPU, DMAC, or CAN.

When writing to CANRAM, the hardware automatically performs read-modify-write operation when writing other than 32-bit width from the CPU because CANRAM is limited to 32-bit width fixed access due to the hardware configuration. Thus, a parity error may occur because a read operation is included in a write operation.

When performing 32-bit writing from CPU, the read operation is not included in the write operation.
The access from CAN FD controller is fixed to 32-bit width.

The access priority is given to CAN FD controller over CPU. When the access from CPU occurs at the same time as that from CAN FD controller, the access from CAN FD controller to CANRAM is executed first, and the access from CPU put on hold. The access from CPU is executed after the access from CAN FD controller is completed.

[Note]

- **Since the initial value of RAM is indefinite, a parity error may occur if it is read without writing it once.**

Chapter 20

I²C Bus Interface with FIFO (I2CF)

20. I²C Bus Interface with FIFO (I2CF)

20.1 Overview

This LSI includes one channel of I²C bus interface that conforms to the typical I²C bus specification.

20.1.1 Features

- Master / Slave function (multi-master non-correspondence)
- The communication speed is selectable with two kinds of standard-mode, fast-mode.
- Master / Slave function supports clock stretch.
- The 4-byte buffer function is provided.
- Generates transmission/receive DMA request

[Note]

- **The minimum system clock frequency required for operation is 3MHz in the standard mode and 5MHz in fast mode.**
- **The minimum system clock frequency to be satisfied AC characteristics is 5MHz in the standard mode and 12MHz in fast mode. If it is less than that, the system clock 1 cycle + 20ns is required as data setup time of reception.**

20.1.2 Configuration

Figure 20-1 shows the overview diagram of the I2CF.

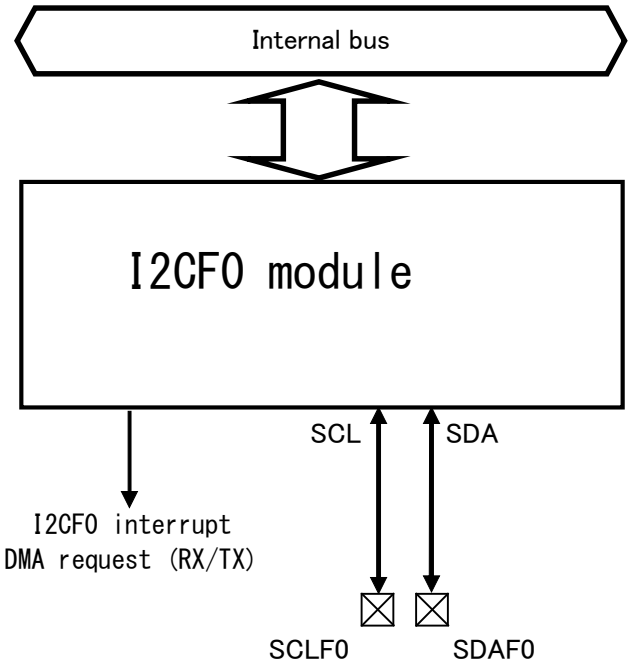


Figure 20-1 Overview diagram

20.1.3 List of Pins

Table 20-1 List of Pins

Pin name	I/O	Description
SDAF0	I/O	I2CF0 serial data input/output
SCLF0	I/O	I2CF0 serial transfer clock

20.2 Description of Registers

20.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4100_2800	I2CF0 base address	I2CF0	-	-	-
0x00	I2CF slave address register	I2F0SAD	R/W	32	0x0000_0000
0x04	I2CF control register	I2F0CTL	R/W	32	0x0000_0000
0x08	I2CF status register	I2F0SR	R/W	32	0x0000_0000
0x0C	I2CF data register	I2F0DR	R/W	32	0x0000_0000
0x10	I2CF bus monitor register	I2F0MON	R	32	0x0000_0003
0x14	I2CF transfer rate setup counter	I2F0BC	R/W	32	0x0000_0000
0x18	I2CF mode register	I2F0MOD	R/W	32	0x0000_0000
0x1C	I2CF buffer mode slave address register	I2F0BSV	R/W	32	0x0000_0000
0x20	I2CF buffer mode sub address register	I2F0BSB	R/W	32	0x0000_0000
0x24	I2CF buffer mode format register	I2F0BFR	R/W	32	0x0000_0000
0x28	I2CF buffer mode control register	I2F0BCT	R/W	32	0x0000_0000
0x2C	I2CF buffer mode interrupt mask register	I2F0BMK	R/W	32	0x0000_0000
0x30	I2CF buffer mode status register	I2F0BSR	R/W	32	0x0000_0000
0x34	I2CF buffer mode level register	I2F0BLV	R/W	32	0x0000_0000
0x48	I2CF timer register	I2F0TMR	R/W	32	0x0000_0000
0x50	I2CF input noise filter setting register	I2F0NF	R/W	32	0x0000_0001

[Note]

- **When a low-speed clock (LSCLK) is selected for the system clock (SYSCLK), do not write to these registers.**

20.2.2 I2CF Slave Address Register (I2F0SAD)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	I2F0SA[10:1]										–
R/W	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0SAD is a special function register (SFR) to set the slave address.

The I2F0SA[10:1] bits are used for the slave address in the I²C slave communication. I2F0SA[7:1] bits are needed to set the 7 bit slave address, and I2F0SA[10:8] bits are needed to set to “000”

20.2.3 I2CF Control Register (I2F0CTL)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	I2F0CS	I2F0DLEN	I2F0STPI	–	I2F0CFIE	–	I2F0MEN	I2F0ASIE	I2F0MSTA	I2F0MTX	I2F0TXAK	I2F0RSTA	I2F0MD	
R/W	–	–	R/W	R/W	R/W	–	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0CTL is a register controlling the transmission and reception of the I²C bus. Each interrupt permission register of I2F0ASIE, I2F0CFIE, I2F0STPI and I2F0DLEN, control only an interrupt and do not control bit that corresponding to the status register. Even in the case of interrupt disable, the status register changes.
Use only I2F0MD, I2F0MEN bit in the buffer mode setting.

Bit No	Bit name	Description
13	I2F0CS	Specifies to halt SCL in the master mode. SCL stops subsequently to an MCF interrupt that is generated after this bit is set to “1”. When inserting a repeated start after executing the transmit mode, set this bit to “1” during a 1-byte data transfer period immediately before the insertion. After the completion of the transfer, set the I2F0RSTA bit to “1” and, at the same time, clear this bit. 0: Continues SCL output. (Initial value) 1: Stops SCL output upon completion of the next transfer.
12	I2F0DLEN	Specifies to enable or disable a DR_LD interrupt. The I2F0DRLD status will change even if an interrupt is disabled by this bit. 0: Disabled (Initial value) 1: Enabled
11	I2F0STPI	Specifies to enable or disable an STP interrupt. The I2F0STP status will change even if an interrupt is disabled by this bit. 0: Disabled (Initial value) 1: Enabled
9	I2F0CFIE	Specifies to enable or disable an MCF interrupt. The I2C0MCF status will change even if an interrupt is disabled by this bit. 0: Disabled (Initial value) 1: Enabled
7	I2F0MEN	Specifies to initialize this I2CF module. Set this bit to “1” when using this I2CF module. 0: Initializes (Initial value) 1: Uninitialized
6	I2F0ASIE	Specifies to enable or disable an MAAS interrupt. The I2F0MAAS status will change even if an interrupt is disabled by this bit. 0: Disabled (Initial value) 1: Enabled
5	I2F0MSTA	Specifies to transmit a START condition or a STOP condition in the master mode. If this bit is rewritten from “0” to “1” in the master mode, a start sequence is sent to the bus. When this bit is cleared, a stop sequence is sent, and switch to slave mode. 0: Sends a STOP condition (Initial value) 1: Sends a START condition
4	I2F0MTX	Selects the data transfer direction in the master mode. 0: Reception (Initial value) 1: Transmission
3	I2F0TXAK	Specifies transmission of ACK or NACK in the receive mode. The acknowledge data that was set to this bit in advance is sent to the transmit device after data is received. But, the ACK reply after the header data reception in the slave mode does not depend on this setting. Return NACK at the time of the slave address disagreement, and return ACK at the time of slave address agreement or non-decision. 0: ACK (Initial value) 1: NACK

Bit No	Bit name	Description
2	I2F0RSTA	Specifies to transmit the repeated START condition. When the I2CF module writes in "1" during communication in masters, send out repeated START condition to the bus. This bit is automatically reset to "0" after sending a repeated START condition.
1 ~ 0	I2F0MD	Select the standard-mode or the fast-mode. 00: Standard-mode (100kHz) (Initial value) 01: Fast-mode (400kHz) 1x: Setting prohibited

[Note]

- **If the I2F0MEN bit is set to "0", the I2CF bus control section, I2CF status register (including the I2F0MBB bit), I2CF buffer level register, and I2CF buffer mode status register will be initialized. The I2CF buffer mode format register, I2CF control register, I2CF data register, I2CF bus monitor register, I2CF bus transfer rate setup counter, I2CF mode register, I2CF buffer mode slave address register, I2CF buffer mode sub address register, I2CF buffer mode control register, I2CF buffer mode interrupt mask register, I2CF timer register and I2CF input noise filter setting register will not be initialized.**
- **When sending a repeated START condition, overwrite I2F0MSTA as is with the setting of "1". Operation cannot be guaranteed if "1" is written to this bit I2F0RSTA and "0" to the I2F0MSTA bit. Because this bit is automatically reset to "0" after sending the repeated START condition, if another bit in the control register will be set after this bit is set to "1", set this bit to "0" or keep the previous value. If "1" is written again, the repeated START condition will be sent at that moment.**

20.2.4 I2CF Status Register (I2F0SR)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	I2F0STP	–	I2F0MCF	I2F0MAAS	I2F0MBB	–	I2F0DRLD	I2F0SRW	I2F0MIF	I2F0RXAK
R/W	–	–	–	–	–	–	R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0SR is a register indicating the status of the I²C bus.
About each bit except I2F0RXAK, I2F0SRW, I2F0MBB, cleared by writing in “0”. Also, use only I2F0MBB bit in the buffer mode setting.

Bit No	Bit name	Description
9	I2F0STP	This bit indicates that data reception is completed in the receive mode. The completion of data reception is fixed by receiving the STOP condition or the repeated START condition in the compound mode from the master device. This bit is cleared by writing “0” by software. This bit is set to “1” when STOP condition is detected, or when the slave address is mismatched after the repeated START condition. 0: Data reception is not completed in the slave mode. (Initial value) 1: Data reception is completed in the slave mode. This bit is valid only at data reception in the slave mode.
7	I2F0MCF	This bit indicates that data transfer has been completed. This bit is set to “1” when the transmission/reception of 1-byte data (*1) is completed. This bit is cleared by writing “0” by software. This bit is set to “1” at a rising edge of SCL when a 1-byte transfer is complete and an ACK response is started. (*1) The data signifies all of the 1-byte transfer, which includes the transfer of an address immediately after start/repeated start that the master transmits. 0: Before data transfer start or during data transfer (Initial value) 1: Data transfer is completed In the receive mode, clear this bit by writing “0” to it after reading the receive data from the I2F0DR register. If any data is left in the receive buffer, this bit cannot be cleared even if “0” is written to it.
6	I2F0MAAS	This bit indicates whether the external master device specifies this MCU as their slave device or not. This bit is set to “1” when the slave address in the I2F0SAD register is matched to the slave address from the external master device. To clear this bit, the software writes “0” in this bit. This bit becomes “1” with the MCF interrupt that the transfer of address data completed. It becomes “1” after 2Byte transfer in the 10bit address after 1Byte transfer in the 7bit address. 0: The external master device doesn’t specify as the slave device. (Initial value) 1: The external master device specifies as the slave device.
5	I2F0MBB	This bit indicates the status of the I²C bus. This bit is set to “1” when a START condition is detected; this bit is reset to “0” when a STOP condition is detected. By reading this bit, it is possible to check whether the bus is currently occupied or released. This bit is set to “1” after a START condition is detected (at a falling edge of SCL after SDA changes from “1” to “0” when SCL is “1”), and is set to “0” after a STOP condition is detected (at a rising edge of SDA when SCL is “1”). 0: The I²C bus is open (Initial value) 1: The I²C bus is occupied. Before starting a master transmit or master receive transfer, read this bit and make sure that the bus is open.

Bit No	Bit name	Description
3	I2F0DRLD	This bit indicates that the transmit buffer is emptied and transmit data can be loaded to the I2F0DR register. After this bit has been set to “1” in the transmit mode, the data to be sent next can be written into the data register without destroying the previously transmitted data. This bit is cleared by writing “0” by software. This bit is set to “1” when the transfer of 2 bit out of 8 bit transmit data is finished (at a falling edge of SCL). By using this bit (interrupt), transmit data can be written before an MCF interrupt. 0: Data load to the data register is not allowed (Initial value) 1: Data load to the data register is allowed
2	I2F0SRW	This bit indicates whether the access type from the master device is “read” or “write” during the slave mode. This bit is updated at getting R/W bit in the address data. (R/W bit is the last bit of the address data) 0: Write mode : master device transmits to slave device (Initial value) 1: Read mode : mater device receives from slave device
1	I2F0MIF	This bit indicates that an interrupt has been requested. This bit is cleared by writing “0” by software. This bit is an interrupt line monitoring bit. If interrupt is enabled for MCF interrupt, MAAS interrupt, DR_LD interrupt and STP interrupt, this bit is set to “1” at the same time that the bit (I2F0MCF, I2F0DRLD, I2F0STP, and I2F0MAAS) for each interrupt source is set to “1”. 0: No interrupt request (Initial value) 1: Interrupt request Whether or not to enable each interrupt is set by the I2F0CTL register. For details, refer to "20.2.3 I2CF Control Register (I2F0CTL)".
0	I2F0RXAK	This bit indicates the reception status of ACK/NACK. Acknowledge data to be replied by the receiving device in the transmit mode is stored. This bit is updated every time ACK/NACK is received, and this bit holds the acknowledge data received last even after the bus is released. In the transmit mode, check this bit at the time of the MCF interrupt. If a NACK is received, finish the transfer. 0: Received an ACK (Initial value) 1: Received a NACK

20.2.5 I2CF Data Register (I2F0DR)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	I2F0DA							
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0DR is a register setting the transmit data or indicating received data.

In the buffer mode, up to 4 bytes of transmit data can be stored in the buffer by writing this register. When receiving data, the received data stored in the buffer can be read by reading this register.

20.2.6 I2CF Bus Monitor Register (I2F0MON)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	I2F0SDA	I2F0SCL
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

I2F0MON is a register expressing a level of SDA, SCL of the I²C bus.

Bit No	Bit name	Description
1	I2F0SDA	Monitors the level of the SDA line.
0	I2F0SCL	Monitors the level of the SCL line.

20.2.7 I2CF Transfer Rate Setup Counter (I2F0BC)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	I2F0BSC						
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BC is a register setting a count level for a counter generating the transfer timing of the I²C bus from system clock.

Use it for SCL/SDA generation in a master mode, but please usually set it by all means because you use the verge of the slave transmission to find data set up time at the time of resumption from SCL clock stop. Therefore, perform this setting by all means.

The relationship between the setting value of I2F0BSC and the transfer rate of the I²C bus is as follows:

I²C bus transfer rate [bps] = (System clock frequency) / (I2F0BSC setting value × 8)

I2F0BSC = (System clock frequency) / (I²C bus transfer rate [bps] × 8)
However, set I2F0BSC to equal or more than 3.
If the I2F0BSC is set to "0", the counter for timing generation is stopped.

Calculate the system clock frequency with the value that added a frequency error. The following table gives examples of setting values.

Table 20-2 Example for setting values

System clock Frequency	I2F0BC			
	Standard mode (I2F0MD = "00")	(Ref. rate)	Fast mode (I2F0MD = "01")	(Ref. rate)
48MHz	0x3C	Approx. 99kbps	0x0F	Approx. 384kbps
40MHz	0x32	Approx. 99kbps	0x0D	Approx. 367kbps
24MHz	0x1E	Approx. 98kbps	0x08	Approx. 348kbps
20MHz	0x19	Approx. 98kbps	0x07	Approx. 328kbps
12MHz	0x0F	Approx. 96kbps	0x04	Approx. 324kbps
10MHz	0x0D	Approx. 92kbps	0x04 (*1)	Approx.270kbps
6MHz	0x08	Approx. 87kbps	0x03 (*1)	Approx. 207kbps
5MHz	0x07	Approx. 82kbps	0x03 (*1)	Approx. 172kbps
3MHz	0x04 (*1)	Approx. 81kbps	Out of guaranteed operation	
Less than 3MHz	Out of guaranteed operation		Out of guaranteed operation	

Also, this reference rate is a guide and will vary depending on external conditions.

[Note]

- **Set the I2F0BC register before setting the I2F0CTL register.**
- **Can't communicate the I²C master that does not confirm the input state, because SCL is low for satisfying the data setup time at transmitting of the slave mode.**
- **The minimum system clock frequency required for operation is 3MHz in the standard mode and 5MHz in fast mode.**
- **The minimum system clock frequency to be satisfied AC characteristics is 5MHz in the standard mode and 12MHz in fast mode. If it is less than that (condition *1 in Table 20-1), the system clock 1 cycle + 20ns is required as data setup time of reception.**

20.2.8 I2CF Mode Register (I2F0MOD)

Offset : 0x18

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	I2F0BMEN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0MOD is a register choosing buffer mode use / not use for master transfer (the master transmission and the master reception).

Bit No	Bit name	Description
0	I2F0BMEN	This is used to enable the buffer mode. 0: Disabled (Initial value) 1: Enabled

[Note]

- **Set this register in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this register is changed during transfer.**
- **During buffer mode setting, cannot use the slave function. Even if other master devices send out a slave address, this I2CF module returns NACK and is not appointed to a slave device.**

20.2.9 I2CF Buffer Mode Slave Address Register (I2F0BSV)

Offset : 0x1C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2F0BA															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BSV is a register setting the slave address of the forwarding address device.
This register is effective only when use a buffer mode.

Set the slave address of the transfer destination device.
Set I2F0BA[7:1] to 7 bit address. Set I2F0BA[15:8] and [0] to "0".

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	A7	A6	A5	A4	A3	A2	A1	0

[Note]

- **Set this bit in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this bit is changed during transfer.**

20.2.10 I2CF Buffer Mode Sub Address Register (I2F0BSB)

Offset : 0x20

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2F0BS1								I2F0BS0							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BSB is a register setting a sub address to transmit to a forwarding address device.
This register is effective only when use a buffer mode.

Bit No	Bit name	Description
15 ~ 8	I2F0BS1	This sets the sub address 1 sent to the transfer destination device. When the value set for I2F0BSL of the I2F0BFR register is “00”, the setting value of this register is disabled, and the sub address is not sent. When the setting value of I2F0BSL is “01”, the data in I2F0BS0 is sent to the I²C bus. When the setting value of I2F0BSL is “10”, the data in I2F0BS1 and I2F0BS0 is sent to the I²C bus in this order.
7 ~ 0	I2F0BS0	This sets the sub address 0 sent to the transfer destination device.

[Note]

- **Set this register in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this bit is changed during transfer.**

20.2.11 I2CF Buffer Mode Format Register (I2F0BFR)

Offset : 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	I2F0BDL			–	–	–	–	I2F0BMRW	–	I2F0BMEN	
R/W	–	–	–	–	–	R/W	R/W	R/W	–	–	-	–	R/W	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BFR is a register setting the communication format in the buffer mode.
Set the data head of the sub address to transmit, the transfer direction of data, the number of bytes of data to transfer.
This register is effective only when use a buffer mode.

Bit No	Bit name	Description
10 ~ 8	I2F0BDL	This is used to set the transferred byte count in the buffer mode. When starting transfer with I2F0BDL set to “0 byte”, I2F0BMDZ will be set to “1”, and transfer will not be started. 000: 0Byte (Initial value) 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte Other: 0Byte
3	I2F0BMRW	This bit is used to set the data transfer direction in the buffer mode. 0: Transmission (Initial value) 1: Reception
2 ~ 0	I2F0BSL	This is used to set the data length of the transferred sub address in the buffer mode. 00, 11: Not sent (Initial value) 01, 10: the sub address of the I2F0BSB register is sent.

[Note]

- **Set this register in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this bit is changed during transfer.**

20.2.12 I2CF Buffer Mode Control Register (I2F0BCT)

Offset : 0x28

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	I2F0BMST
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BCT is a register appointing a start of the I²C movement in a buffer mode.
This register is effective only when use a buffer mode.

Bit No	Bit name	Description
0	I2F0BMST	This bit indicates starting the transfer in the buffer mode. Before the transfer, it is necessary to set the I2F0BSV, I2F0BSB, I2F0BFR, I2F0BMK, and I2F0TMR registers, and to write the transmit data to the buffer. And then the transfer starts when setting this bit to “1”. This bit will be cleared to “0” after starting transfer when the transfer of the specified bytes is finished, or transfer is stopped due to errors such as NACK reception, unexpected STOP condition, and timeout. If the number of transferred bytes is different from the value of the buffer mode level register, the transmission does not start. This bit will be cleared to “0”, and an I2F0BMAG interrupt will occur. If the number of transferred bytes is “0”, the transmission and reception does not start. This bit will be cleared to ”0”, and an I2F0BMDZ interrupt will occur. 0: Transfer in the I²C buffer mode is stopped (Initial value) 1: Transfer in the I²C buffer mode is started

20.2.13 I2CF Buffer Mode Interrupt Mask Register (I2F0BMK)

Offset : 0x2C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	I2F0BDIE	I2F0BAIE	I2F0BSIE	I2F0BTIE	I2F0BNIE	–	I2F0BFIE
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BMSK is a register controlling each interrupt signal in the buffer mode.
Do not control the bit to support of the status register. (in the case of the interrupt suppression, the status register changes.)
This register is effective only when use a buffer mode.

[Description of each bit]
This is used to enable the corresponding interrupt request.
0: Disabled (Initial value)
1: Enabled

Bit No	Bit name	Description (corresponding interrupt)
6	I2F0BDIE	I2F0BMDZ interrupt. The I2F0BMDZ status will change even if an interrupt is disabled by this bit.
5	I2F0BAIE	I2F0BMAG interrupt. The I2F0BMAG status will change even if an interrupt is disabled by this bit.
4	I2F0BSIE	I2F0BMIS interrupt. The I2F0BMIS status will change even if an interrupt is disabled by this bi
3	I2F0BTIE	I2F0BMTO interrupt. The I2F0BMTO status will change even if an interrupt is disabled by this bit.
2	I2F0BNIE	I2F0BMNA interrupt. The I2F0BMNA status will change even if an interrupt is disabled by this bit.
0	I2F0BFIE	I2F0BMFI interrupt. The I2F0BMFI status will change even if an interrupt is disabled by this bit.

[Note]

- **Set this bit in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this register is changed during transfer.**

20.2.14 I2CF Buffer Mode Status Register (I2F0BSR)

Offset : 0x30

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	I2F0BMDZ	I2F0BMAG	I2F0BMIS	I2F0BMTO	I2F0BMNA	–	I2F0BMFI
R/W	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BSR register is a register indicating each status in the buffer mode.
About each bit, cleared by writing in "0".
This register is effective only when use a buffer mode.

Bit No	Bit name	Description
6	I2F0BMDZ	This bit is set to “1” if the number of transferred bytes set in I2F0BDL is 0 at the start of the transmission (when “1” is written to I2F0BMST). In this case, the transfer is not started. This bit is cleared by writing “0” by software.
5	I2F0BMAG	This bit is set to “1” if the number of transferred bytes set in I2F0BDL and the value of I2F0BML do not match at the start of the transmission (when I2F0BMRW is set to “0” and “1” is written to I2F0BMST). In this case, the transfer is not started. This bit is cleared by writing “0” by software.
4	I2F0BMIS	This bit is set to “1” if a STOP condition occurs at unexpected timing, and the transfer ends abnormally. This bit is set to “1” if a STOP condition is detected before this I²C device transmits a STOP condition during a transfer in the buffer mode. This bit is cleared by writing “0” by software.
3	I2F0BMTO	It is set to “1” when a timeout occurs before the end of the transfer, and the transfer ends abnormally. For example, when SCL prolonging by the slave device is not terminated in a certain time, this bit is set to “1”. This bit is cleared by writing “0” by software.
2	I2F0BMNA	It is set to “1” when it receives a NACK and the transfer is finished. This bit is cleared by writing “0” by software.
0	I2F0BMFI	It is set to “1” when this I2CF module finishes the transfer in the buffer mode. This bit is cleared by writing “0” by software.

20.2.15 I2CF Buffer Mode Level Register (I2F0BLV)

Offset : 0x34

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	I2F0BML		
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0BLV shows the residual quantity of the data which collected in the buffer.
This register is effective only when use a buffer mode.

Bit No	Bit name	Description
2 ~ 0	I2F0BML	This indicated the amount of data remaining in the buffer. The buffer size is 4 bytes. The value in this register increases one byte, when one byte is written to the I2F0DR register or one byte is received. Meanwhile, the value in this register decreases one byte when one byte is read from the I2F0DR register or one byte is sent. 000: 0 Bytes (Initial value) 001: 1 Byte 010: 2 Bytes 011: 3 Bytes 100: 4 Bytes When transmitting data, up to 4 bytes of data can be written to the I2F0DR register. When receiving data, the number of bytes indicated by the I2F0BML can be read from the I2F0DR register. The buffer is cleared by writing 0 to this register.

[Note]

- **When before data transmission start of the buffer transfer, writing in "0" at Low side I2F0BML of this register and buffer cleared. And write in transmission data at I2F0DR register. Can collect transmission data in the buffer for 4 bytes. When before data reception start of the buffer transfer, writing in "0" at Low side I2F0BML of this register and buffer cleared.**

20.2.16 I2CF Timer Register (I2F0TMR)

Offset : 0x48

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2F0T															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2F0TMR is a register setting distance of the time-out outbreak in the buffer mode.
This register is effective only when use a buffer mode.
Set the I2F0BMTO bit of the I2F0BSR register in “1” when it takes the time that transmission and reception transfer every 1 byte takes more than a set point of the time-out in the buffer mode. When time-out occurs, the transfer is stopped.

The timeout interval can be calculated from the setting value of I2F0T bit as follows:
Timeout interval = (I2F0T setting value × 8) / (System clock frequency)
Here are the examples of setting the timeout interval to 1ms and 8ms.

Table 20-3 setting value for timeout

System clock frequency	I2F0T	
	At 1ms	At 8ms
48 MHz	0x1770	0xBB80
40 MHz	0x1388	0x9C40
24 MHz	0x0BB8	0x5DC0
20 MHz	0x09C4	0x4E20
12 MHz	0x05DC	0x2EE0
6 MHz	0x02EE	0x1770
5 MHz	0x0271	0x1388
3 MHz	0x0177	0x0BB8

When I2F0T is set to ”0”, a timeout interrupt does not occur.

[Note]

- **Set this register in the initial setting flow or before the master transfer start. Operation cannot be guaranteed if the value of this register is changed during transfer.**
- **Setting I2F0T to less than the time required for the transfer always results in a timeout. One-byte transfer takes 90 µs in the standard mode (100 kbps) and 22.5 µs in the fast mode (400 kbps).**

20.2.17 I2CF Input Noise Filter Setting Register (I2F0NF)

Offset : 0x50

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	I2F0NFON
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

I2F0NF is a register to enable a noise filter for SCL and SDA inputs.

Bit No	Bit name	Description
0	I2F0NFON	This is used to enable input noise filter for I²C interface. 0: Disabled 1: Enabled (Initial value)

20.3 Description of Operation

20.3.1 Flow of Initial Setting

This subsection shows the initial setting flow. "bit name ← 1(0)" represents that software writes 1(0) to this bit. The shaded portions show interrupt sources.

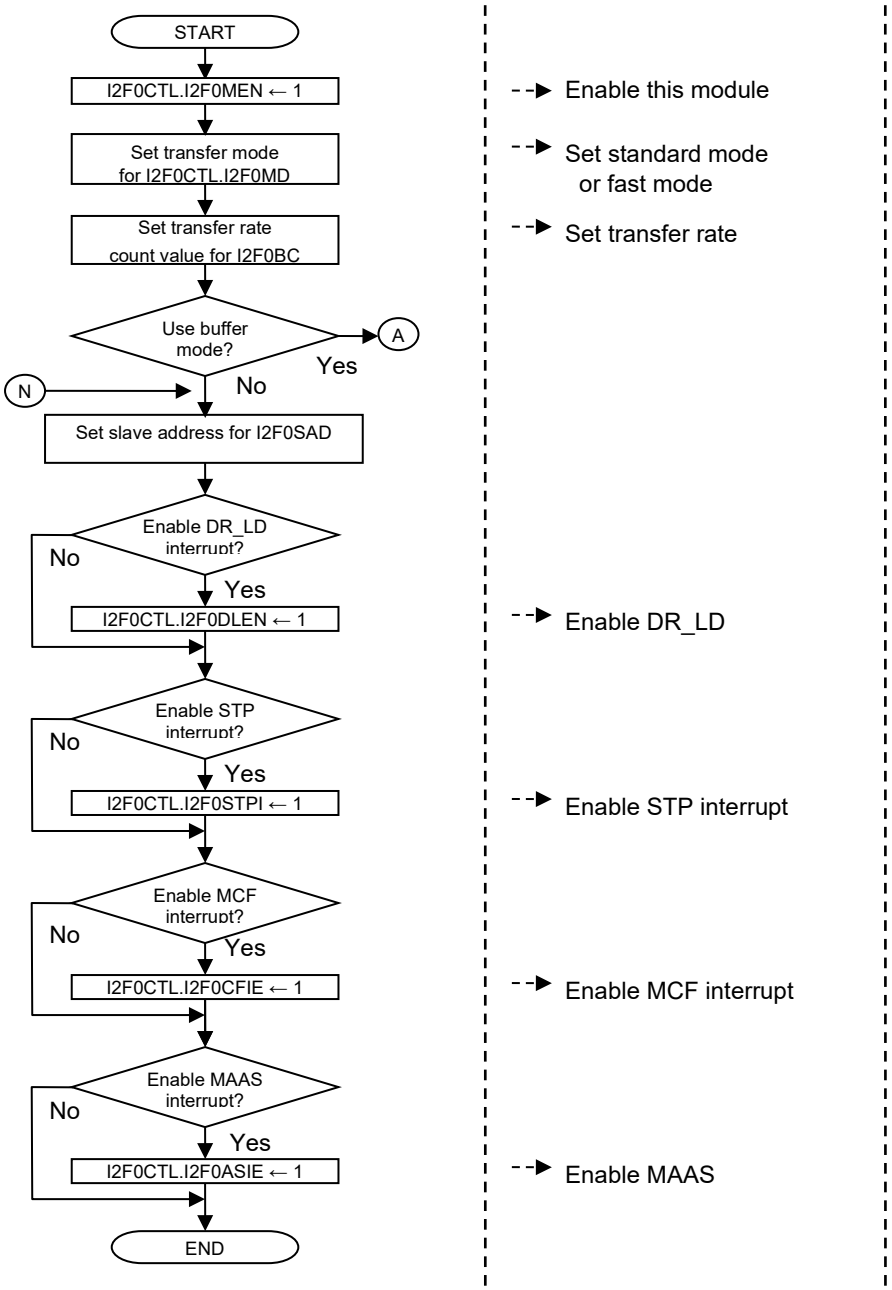


Figure 20-2 Initial Setting Flowchart

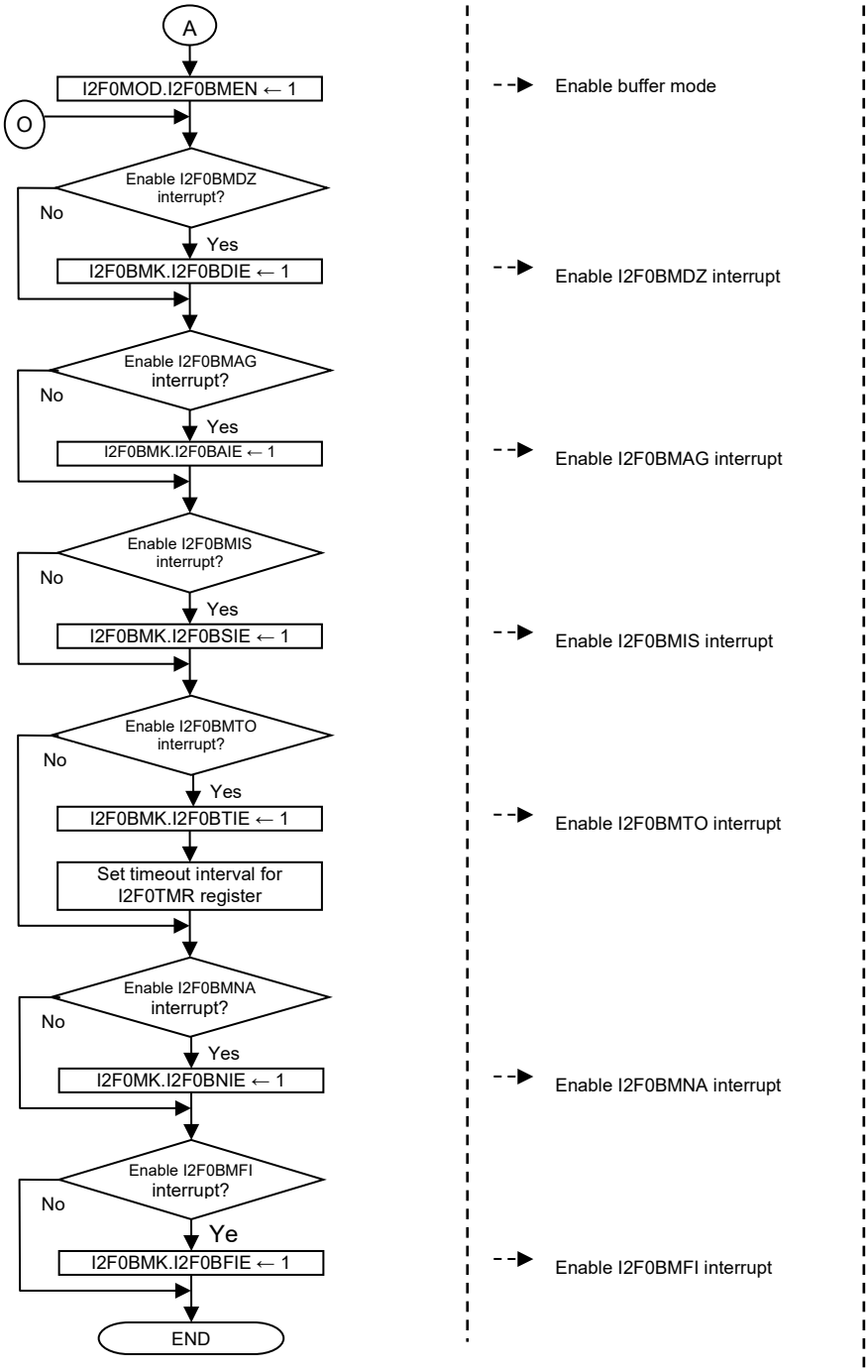


Figure 20-3 Initial Setting Flowchart (Buffer Mode)

20.3.2 Flow of Slave Reception

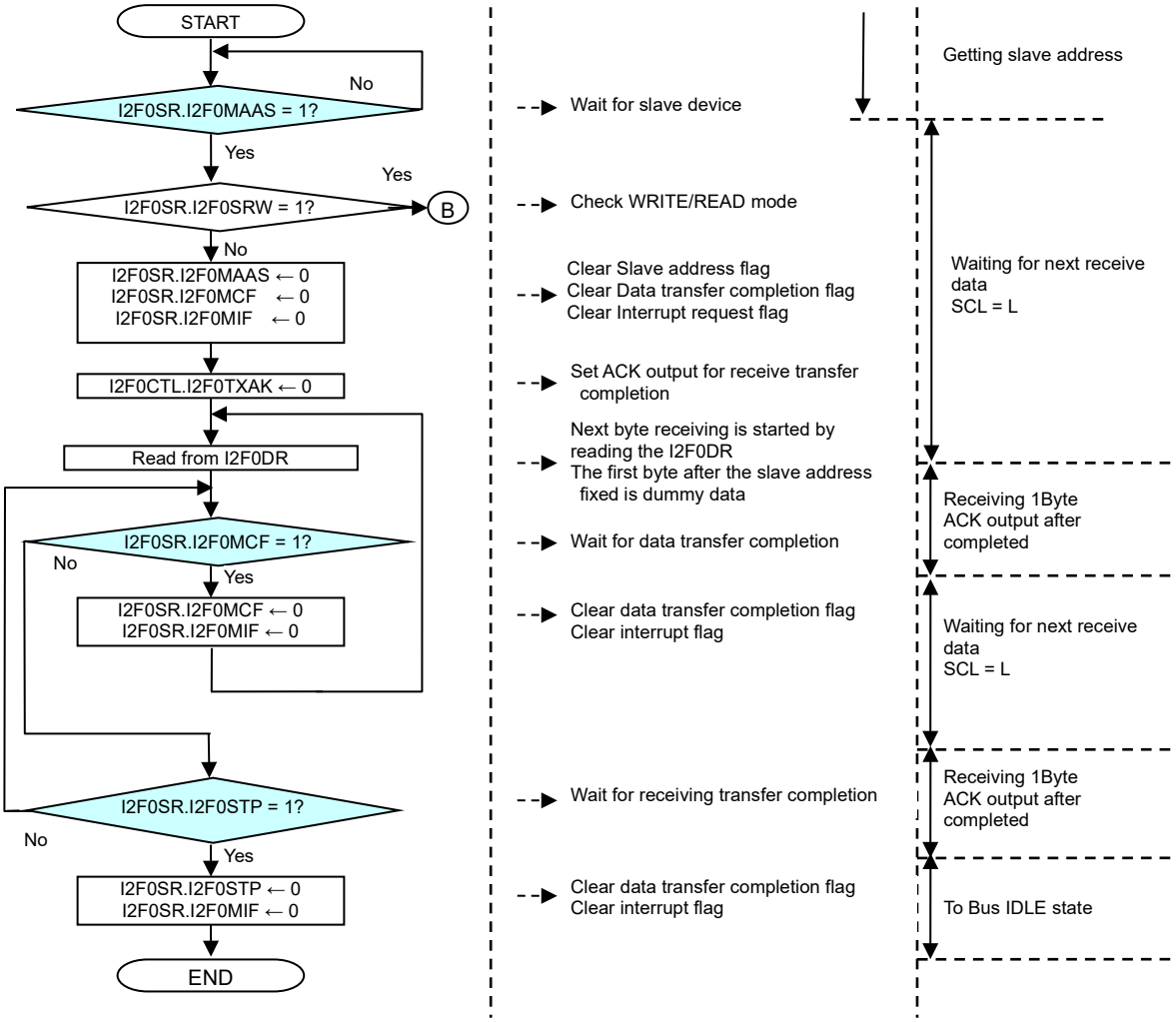


Figure 20-4 Slave Reception Flowchart

20.3.3 Flow of Slave Transmission

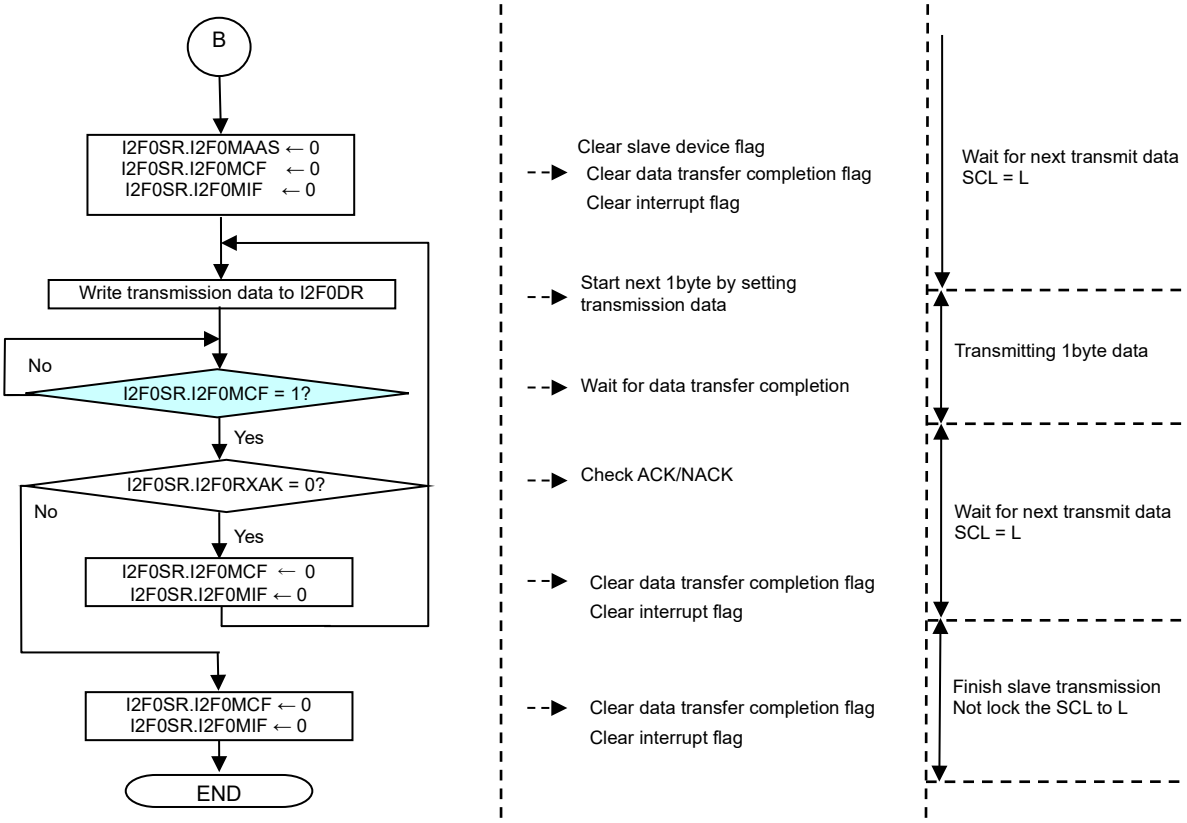


Figure 20-5 Slave Transmission Flowchart

20.3.4 Flow of Master Transmission

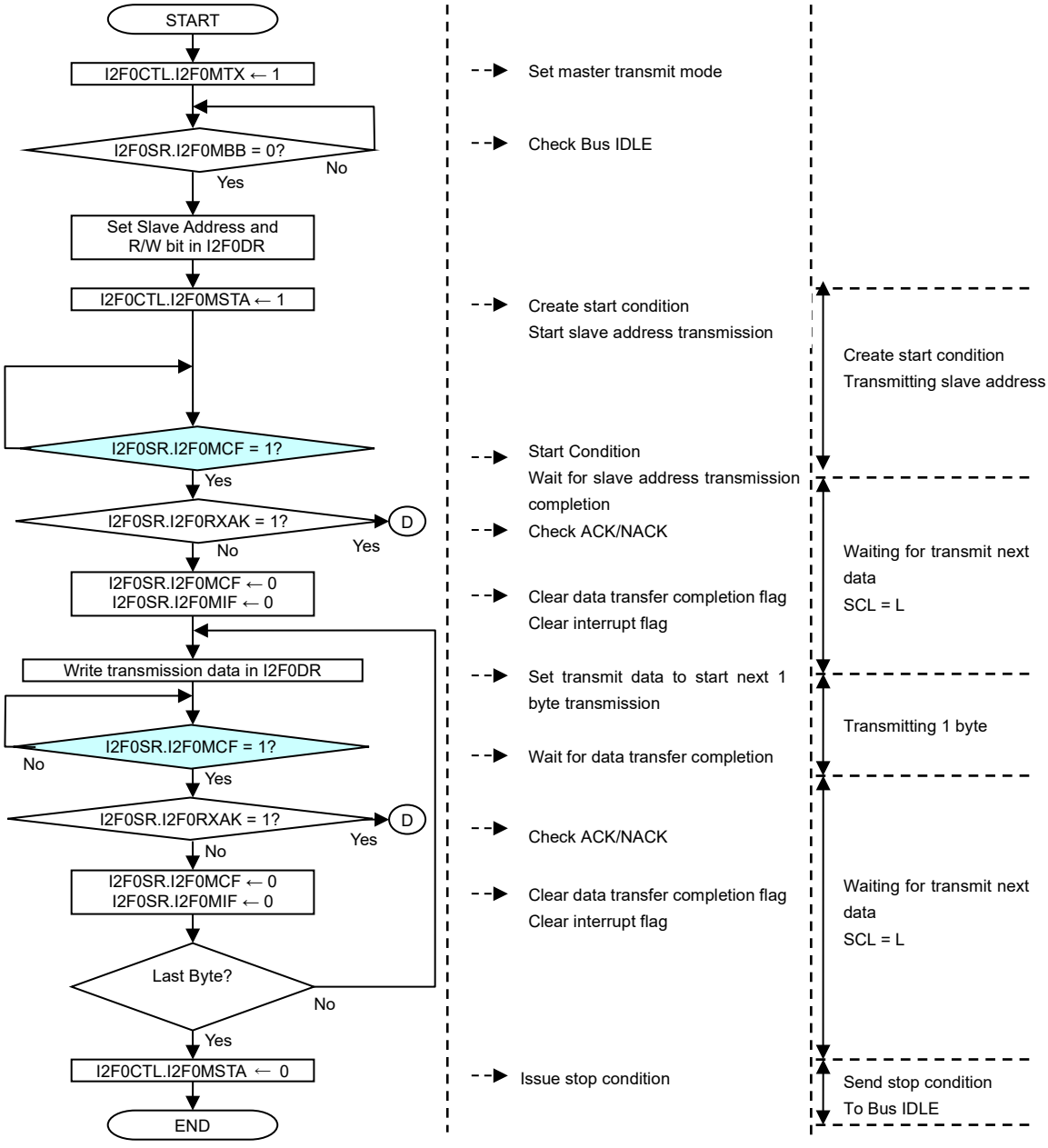


Figure 20-6 Master Transmission Flowchart

20.3.5 Flow of Master Reception

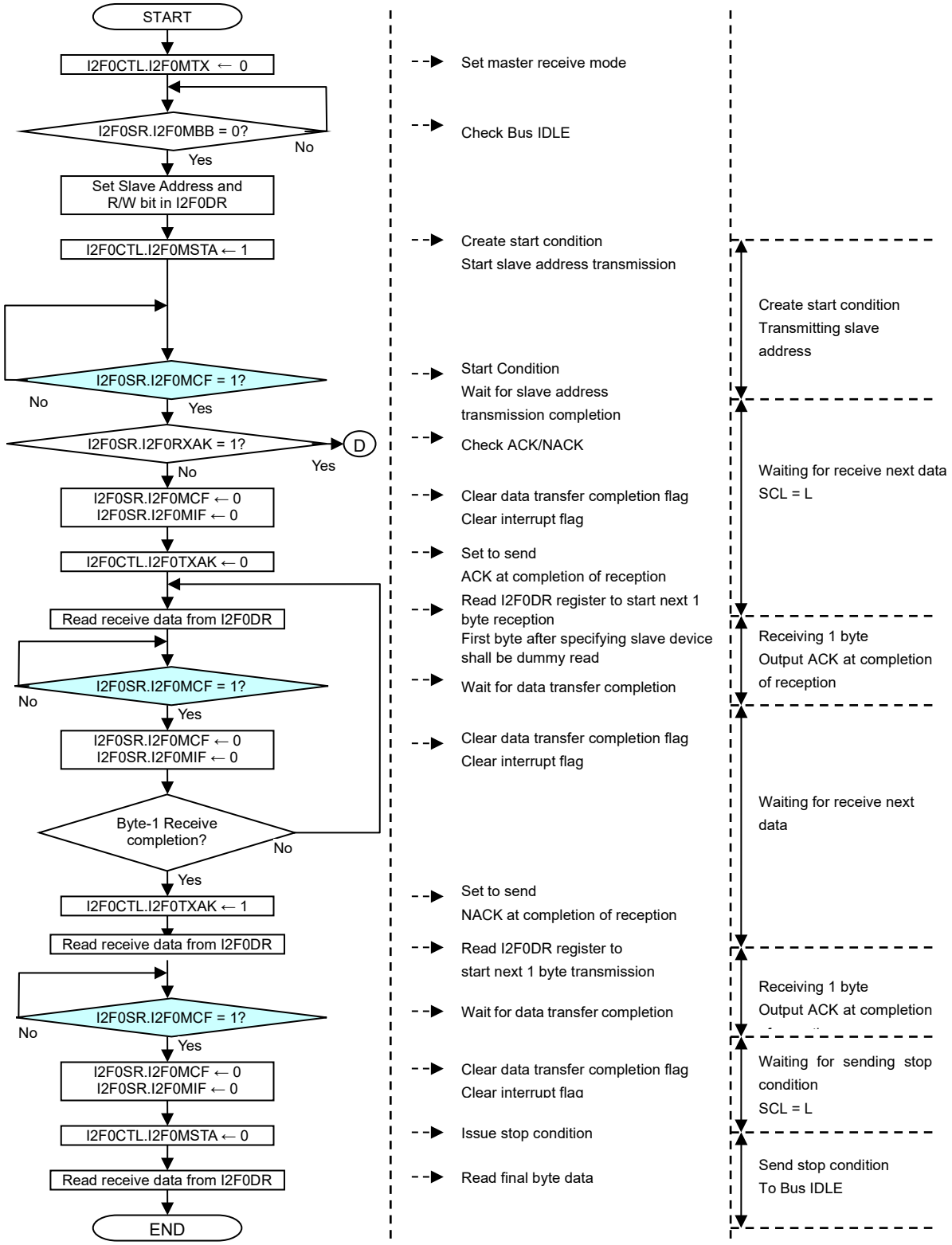
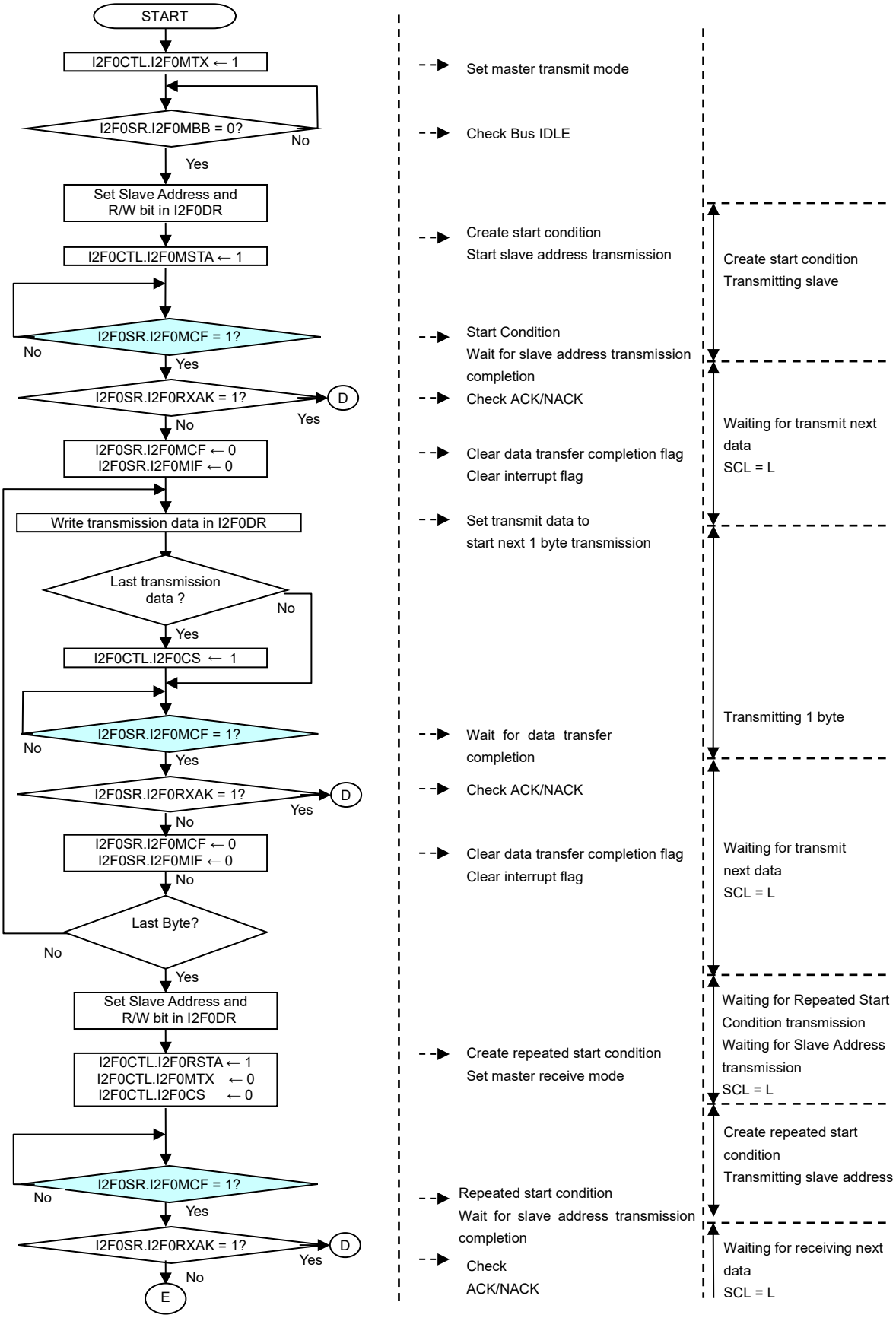


Figure 20-7 Master Reception Flowchart

20.3.6 Compound Flow in The Master Mode (Receiving after Transmitting)



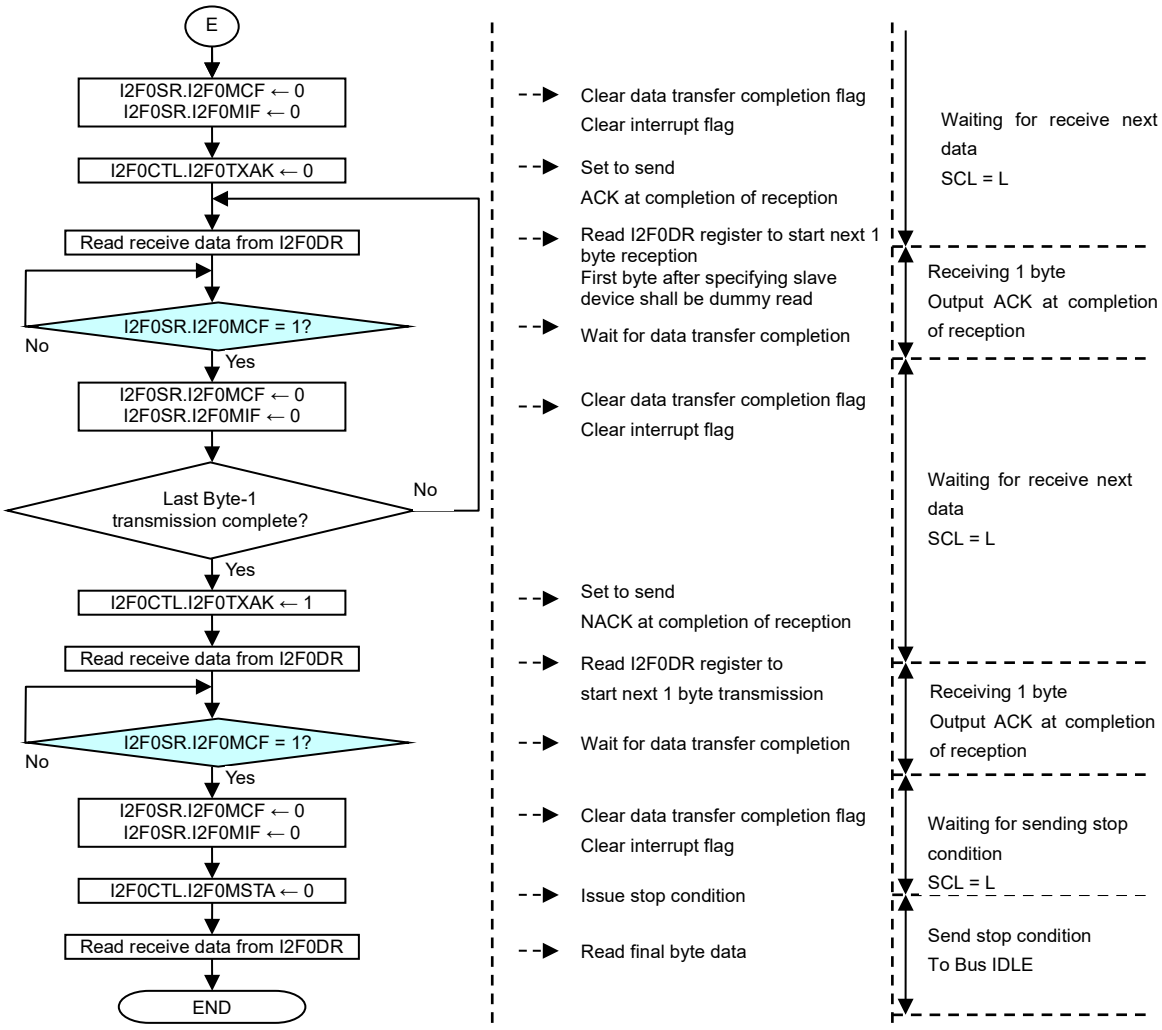
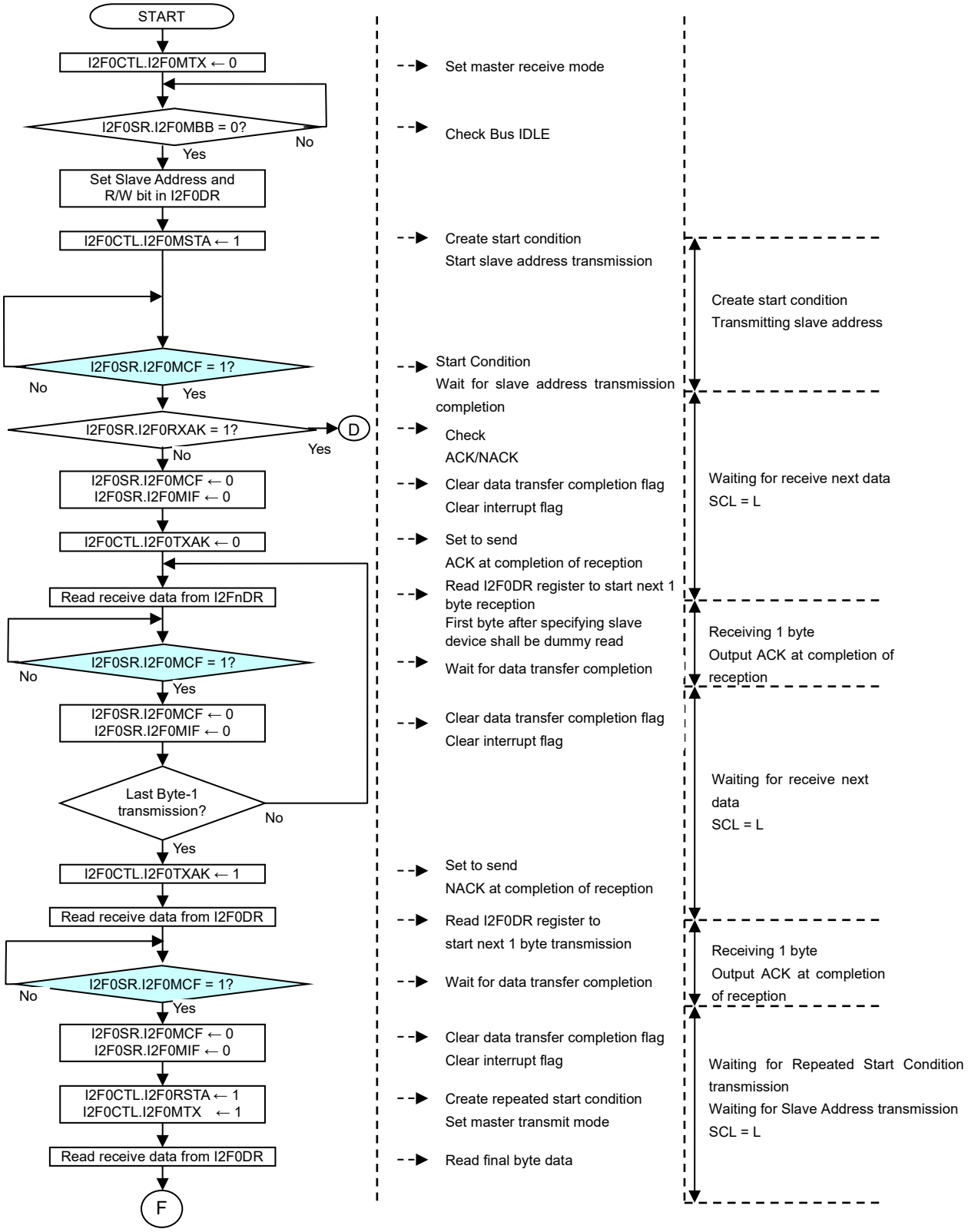


Figure 20-8 Compound Flowchart in the master mode (Receiving after Transmitting)

20.3.7 Compound Flow in the master mode (Transmitting after Receiving)



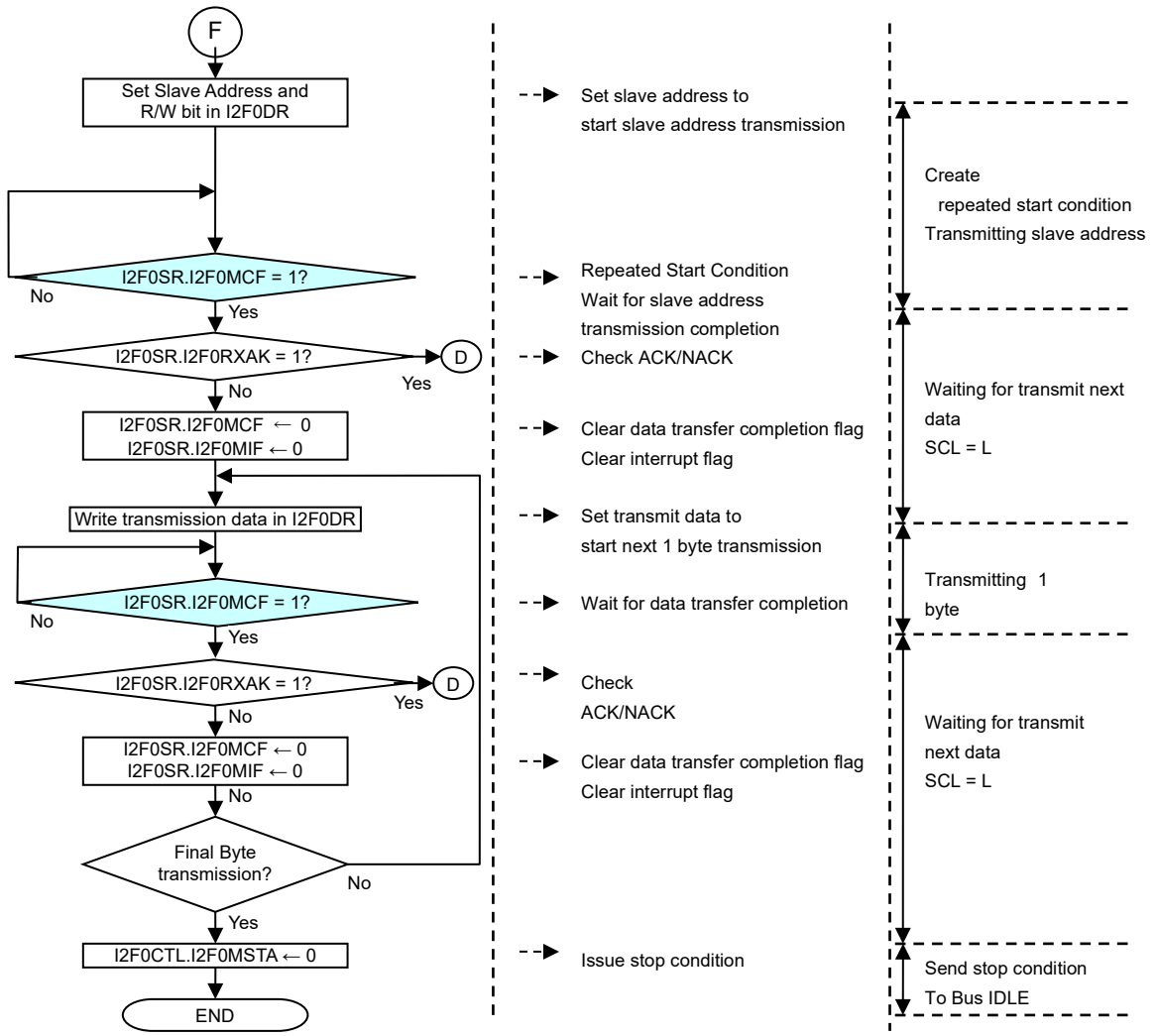


Figure 20-9 Compound Flowchart in the master mode (Transmitting after Receiving)

20.3.8 Flow of NACK Receiving

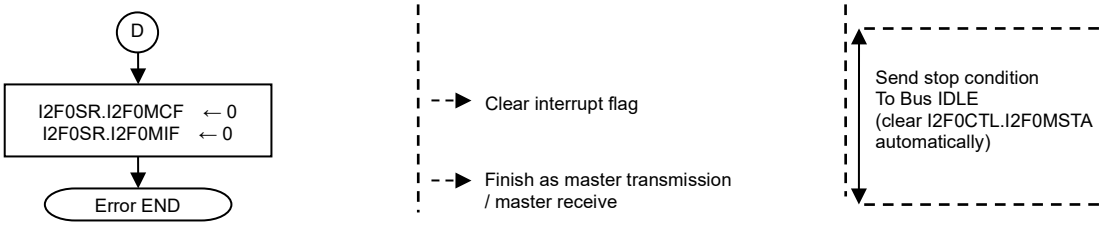


Figure 20-10 NACK Receiving Flowchart

20.3.9 Flow when Using Buffer Mode

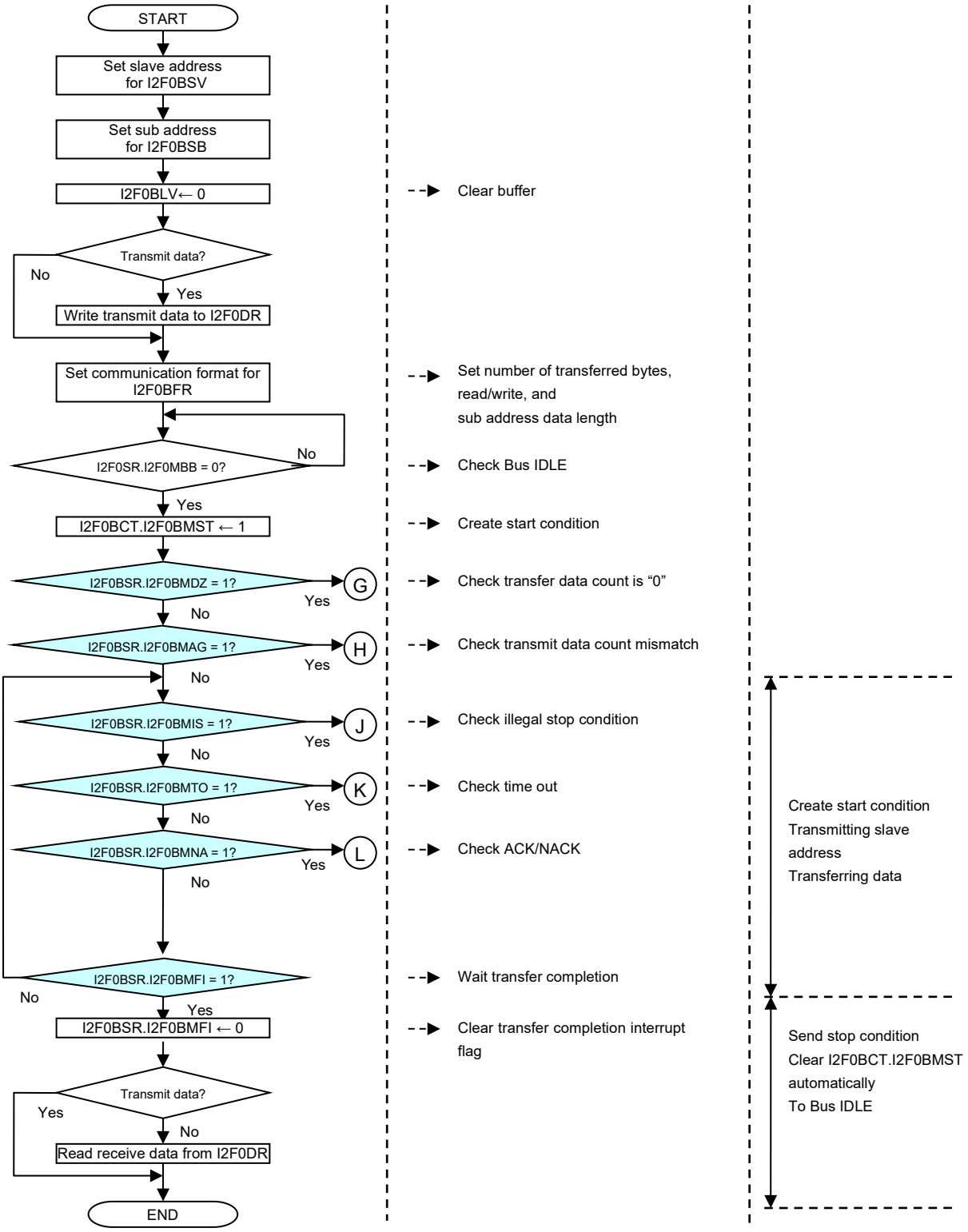


Figure 20-11 Buffer Mode Flowchart (1)

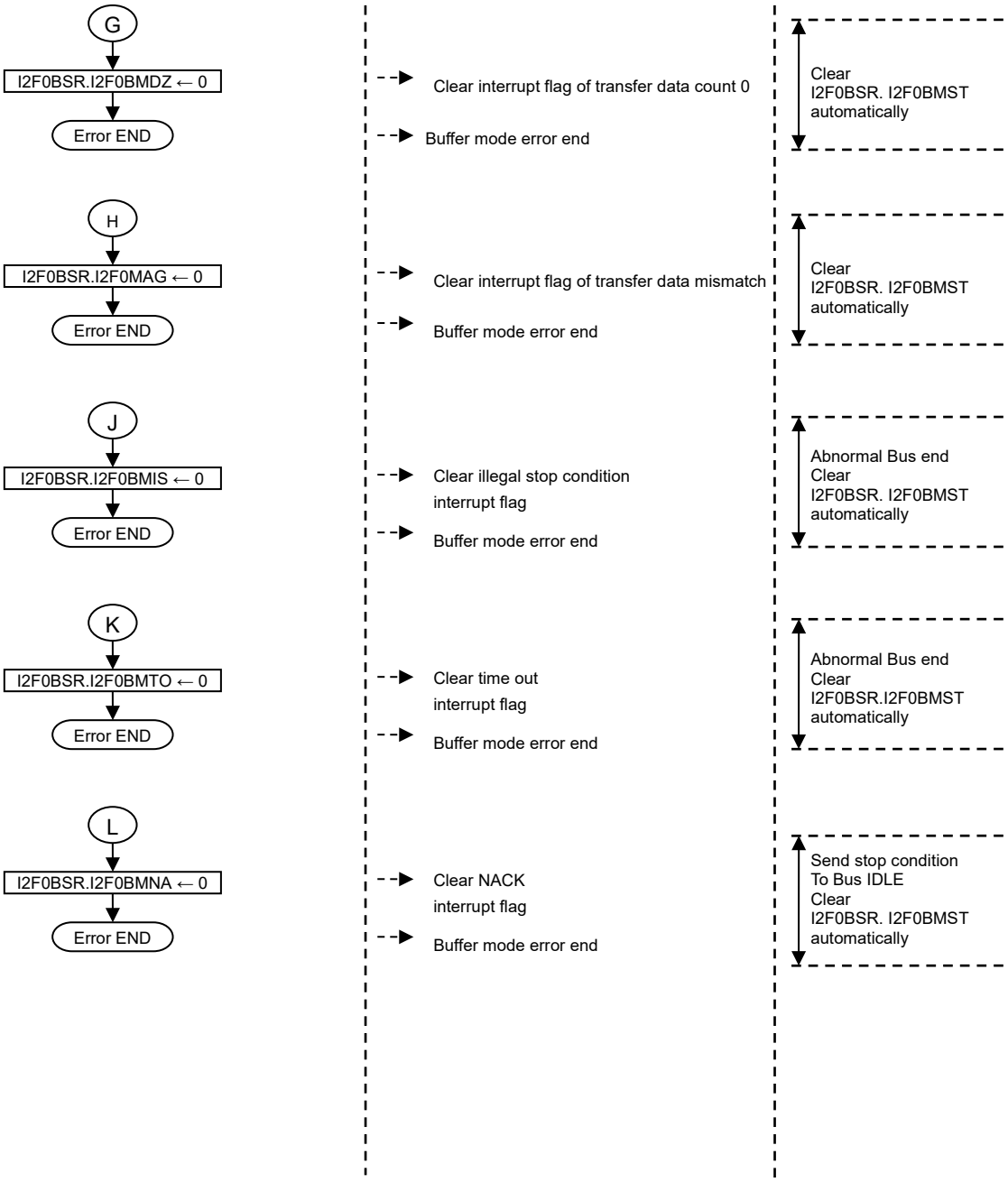


Figure 20-12 Buffer Mode Flowchart (2)

20.3.10 Flow of Switching Mode

20.3.10.1 Flow of Switching to Normal Mode

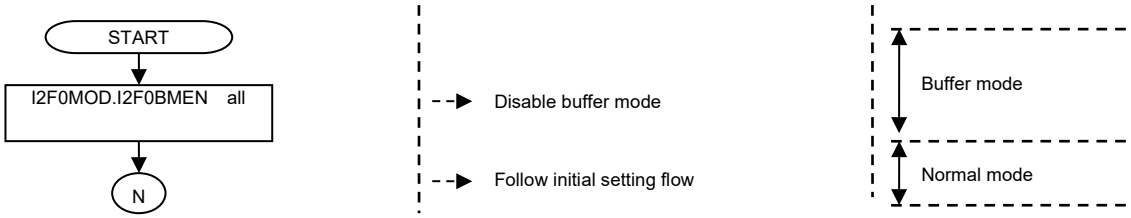


Figure 20-13 Switching to Normal Mode

20.3.10.2 Flow of Switching to Buffer Mode

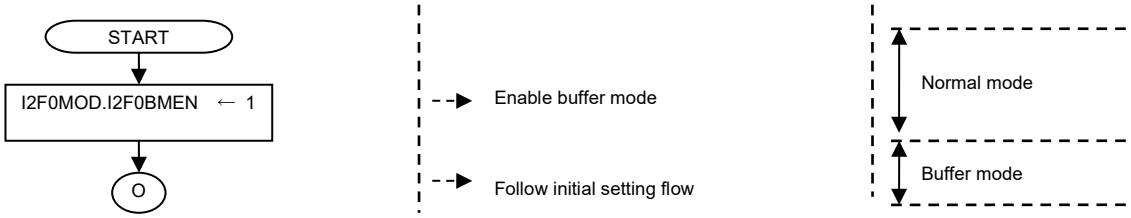


Figure 20-14 Switching to Buffer Mode

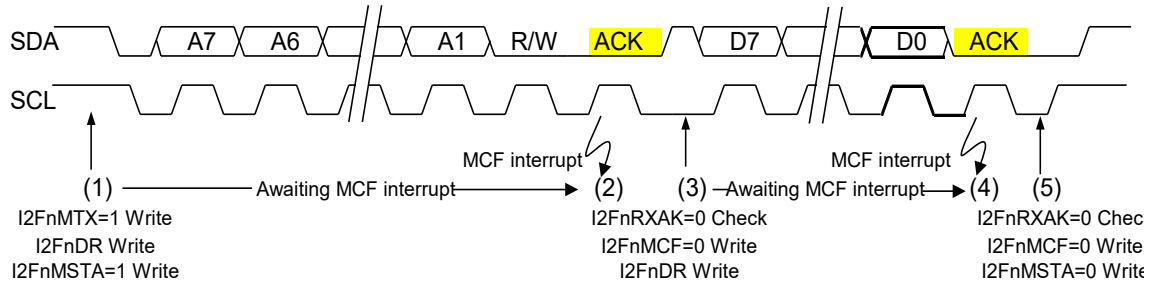
[Note]

- **Be sure to switch modes before starting I^2C transfer. When change the buffer mode during slave movement, an internal buffer mode signal is replaced after slave movement.**

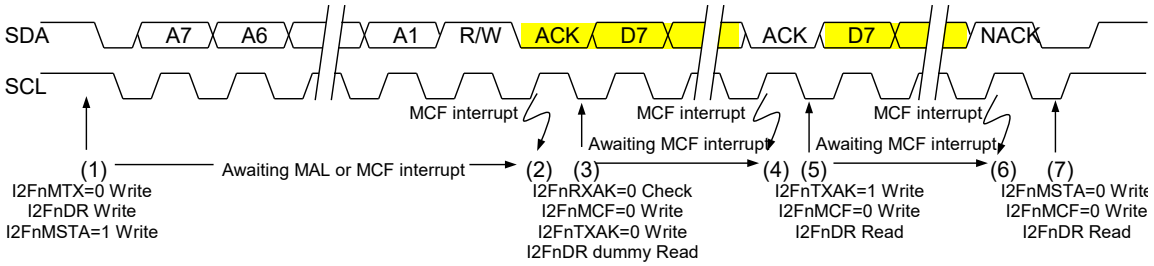
20.4 Waveform in Each Mode

In the figures below, the hatched portions are the segments that are driven by the transfer destination, and n=0.

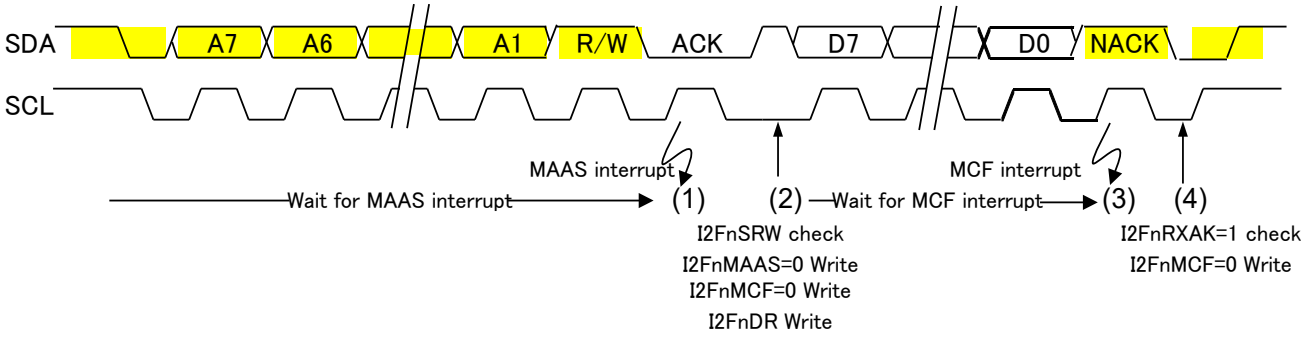
20.4.1 Waveform Transmitted by Master



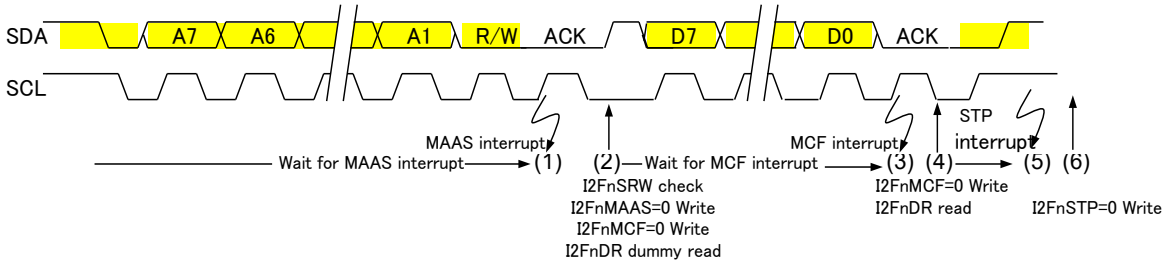
20.4.2 Waveform Received by Master



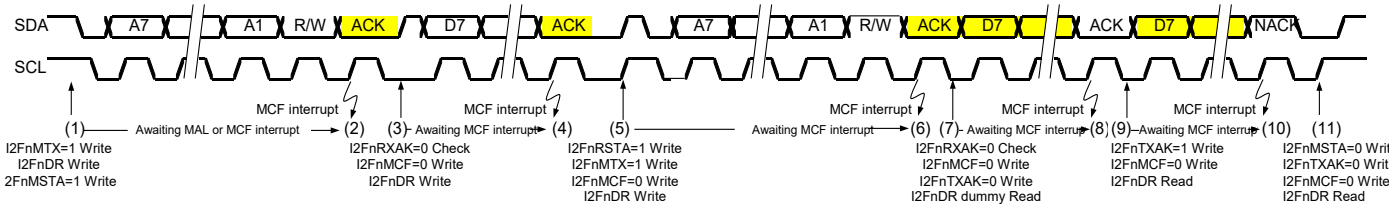
20.4.3 Waveform Transmitted by Slave



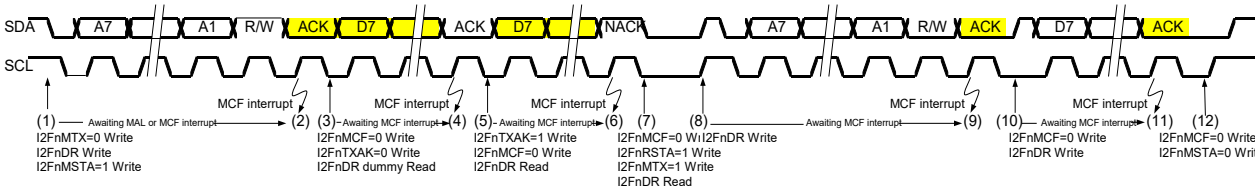
20.4.4 Waveform Received by Slave



20.4.5 Waveform of Compound Format (Master Transmission + Master Reception)

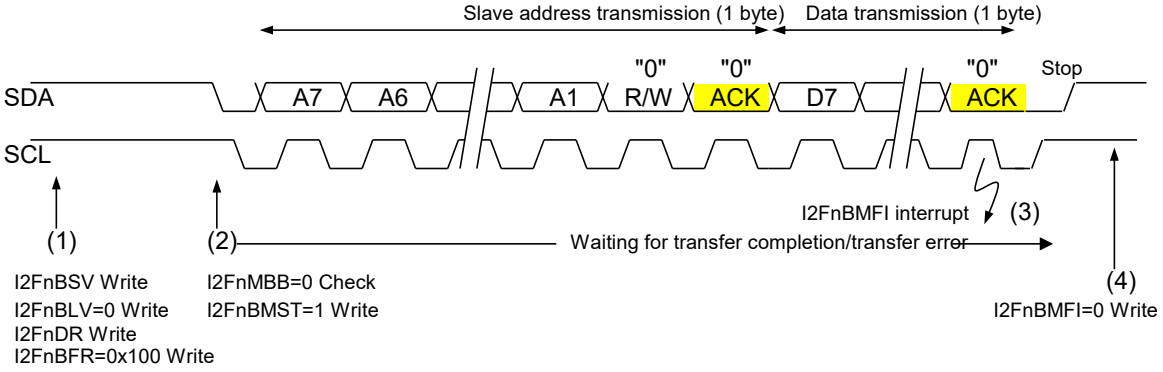


20.4.6 Waveform of Compound Format (Master Reception + Master Transmission)



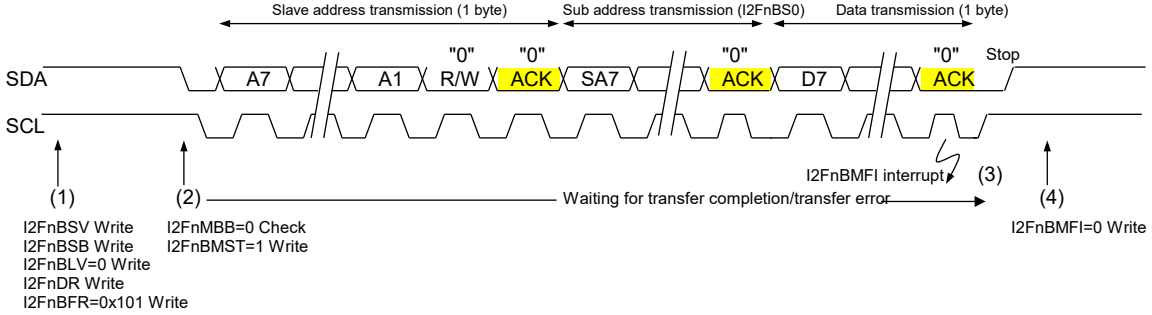
20.4.7 Waveform 1 When Using Buffer Mode

(When data length of sub address = 0, data transmission, number of transferred bytes = 1)



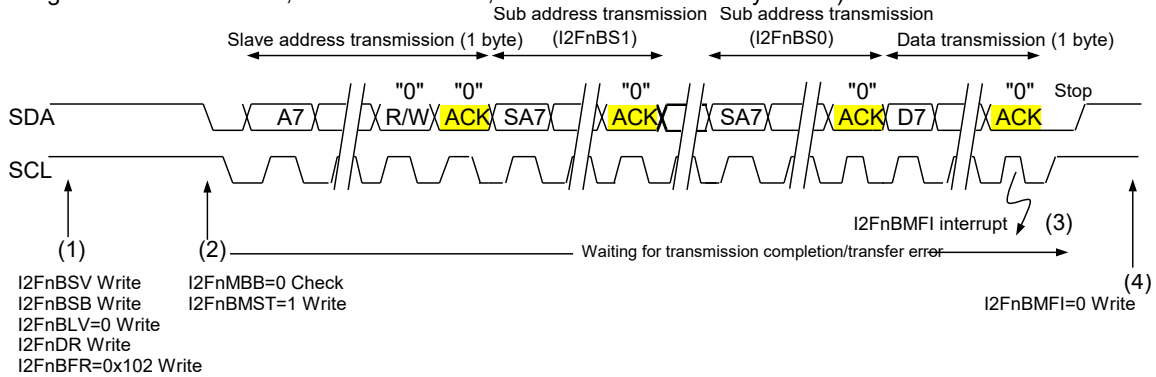
20.4.8 Waveform 2 When Using Buffer Mode

(When data length of sub address = 1, data transmission, number of transferred bytes = 1)



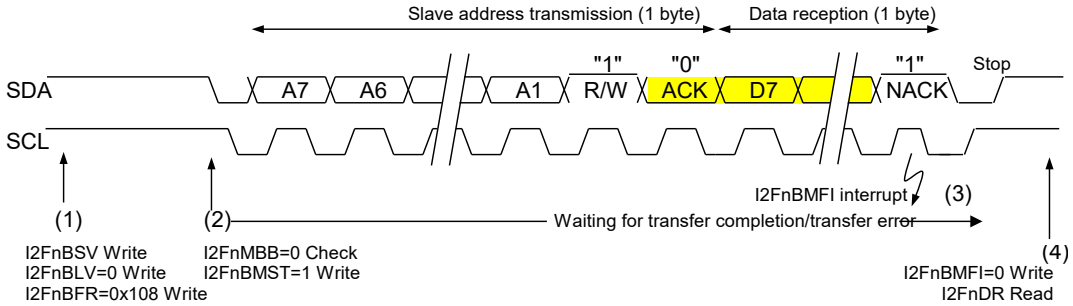
20.4.9 Waveform 3 When Using Buffer Mode

(When data length of sub address = 2, data transmission, number of transferred bytes = 1)



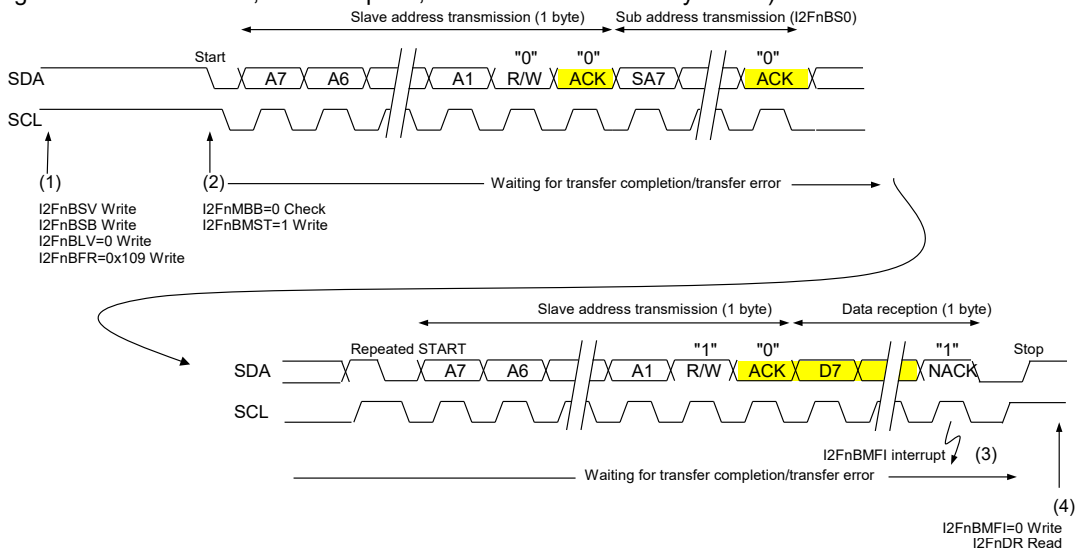
20.4.10 Waveform 4 When Using Buffer Mode

(When data length of sub address = 0, data reception, number of transferred bytes = 1)



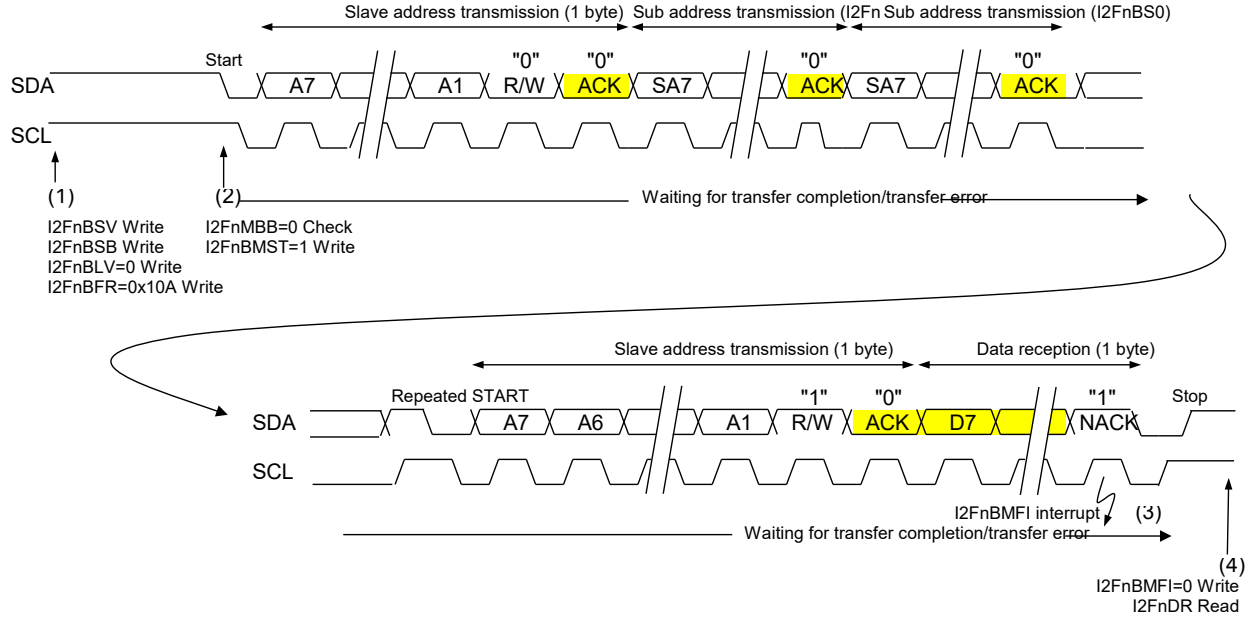
20.4.11 Waveform 5 When Using Buffer Mode

(When data length of sub address = 1, data reception, number of transferred bytes = 1)



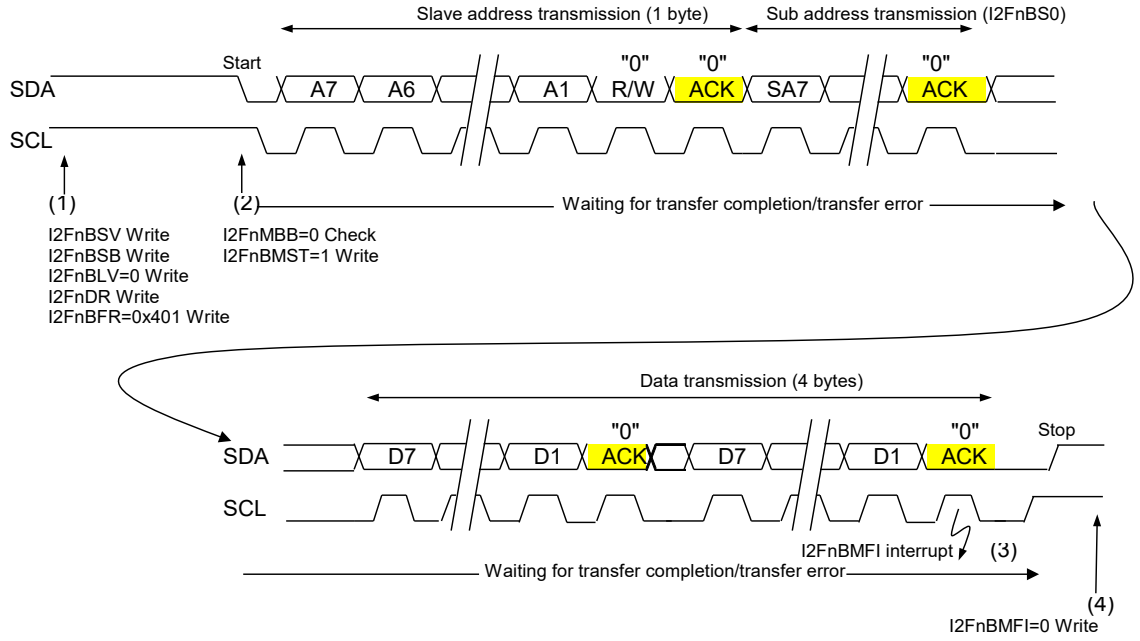
20.4.12 Waveform 6 When Using Buffer Mode

(When data length of sub address = 2, data reception, number of transferred bytes = 1)



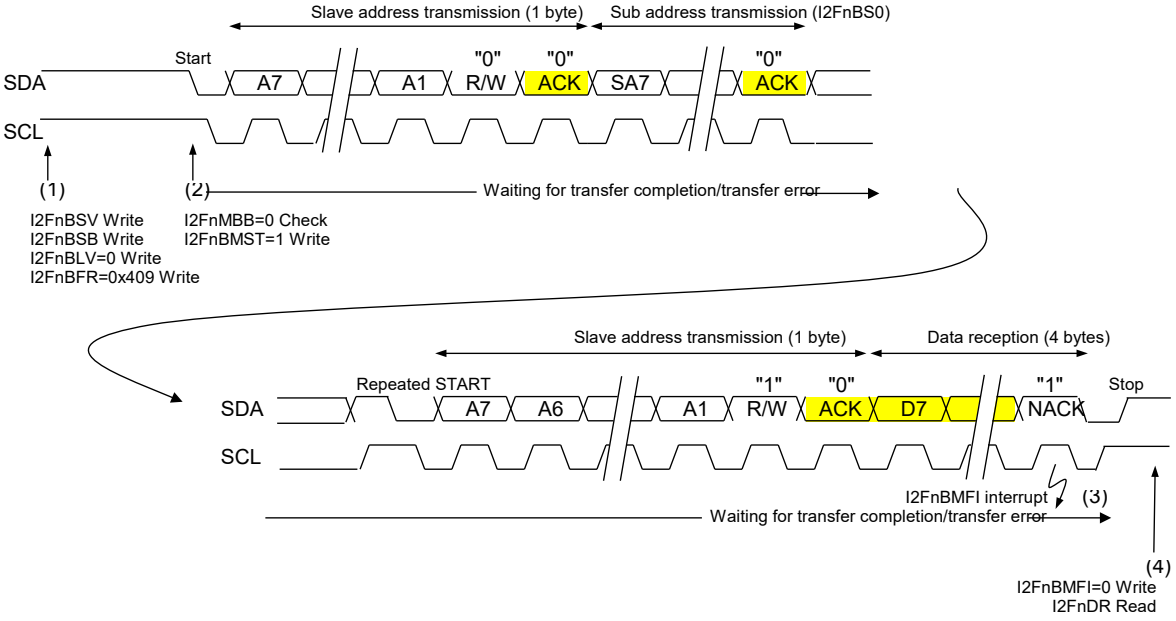
20.4.13 Waveform 7 When Using Buffer Mode

(When data length of sub address = 1, data transmission, number of transferred bytes = 4)



20.4.14 Waveform 8 When Using Buffer Mode

(When data length of sub address = 1, data reception, number of transferred bytes = 4)



20.5 Restrictions

[I²C Master transmission mode]

- If there is a NACK response after data is transferred in the I²C master transmit mode, a STOP condition is automatically sent at the same time it enters the IDLE state. To resume the transmission, it is necessary to set the I2F0MSTA of the control register to "1" and send a START condition again.

- In the I²C master transmit mode, be sure to write the first byte of the transmit data before setting I2F0MSTA = "1" (sending START condition). (The data transmission is automatically started after sending START condition) Write subsequent bytes of the transmit data after the DR_LD status is set to 1. Any transmit data written to the data register with the DR_LD status set to "0" cannot be guaranteed.

- In the I²C master transmit mode, be sure to set the STOP condition (I2F0MSTA = "0") after transferring the data (after the I2F0MCF status is set to "1"). The STOP condition sending is started after the ACK response cycle. If it is set at any other timing than an ACK cycle, the STOP condition will be sent immediately, which may cause a failure.

20.6 Pin Settings

To enable the I2CF function, the applicable bit of each related port register needs to be set. See Chapter 22, "General Purpose Port (GPIO)" for details about the port registers.
For SCLF0 and SDAF0, the ports can be selected from several possibilities.
Be sure to select one of the following combinations of ports for SCLF0/SDAF0. Choose only one combination.

Table 20-4 Pin Combination

	I²C Pins	Combination 1	Combination 2
I2CF0	SCLF0, SDAF0	P81, P80	P73, P74

Chapter 21

(Skipped)

Chapter 22

General Purpose Port (GPIO)

22. General Purpose Port (GPIO)

22.1 Overview

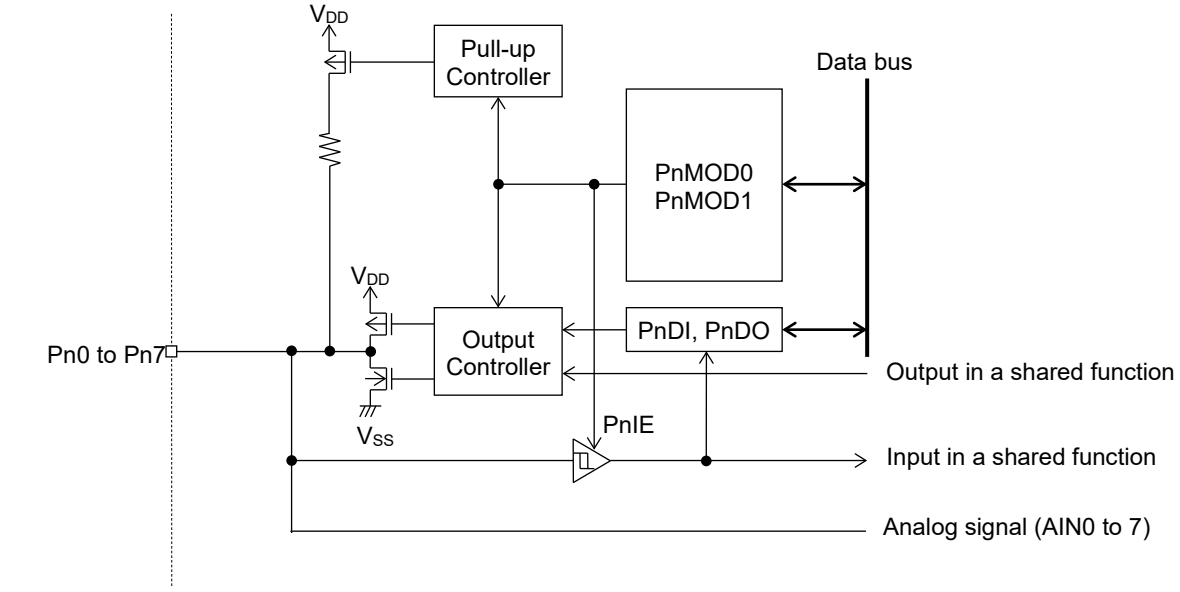
The general purpose port is used as an input port or an output port. The input and output is switchable on each pin. Max. 8 pins are available to read or to change the level of output in the same time. A general purpose port shares a number of functions. See "1.3.2 List of Pins" or "1.3.3 Description of Pins" for more detail.
The general purpose ports can be used for external interrupts and external inputs for the functional timers. Also, used as an input or an output pins in shared functions by setting the port n mode register.

22.1.1 Features

- Input or output can be chosen in each pin
- Pull-up resistor can be chosen in each pin
- CMOS output or N-channel open drain output is can be chosen in each pin
- Direct driving LEDs is supported when the N-channel open drain output is chosen

22.1.2 Configuration

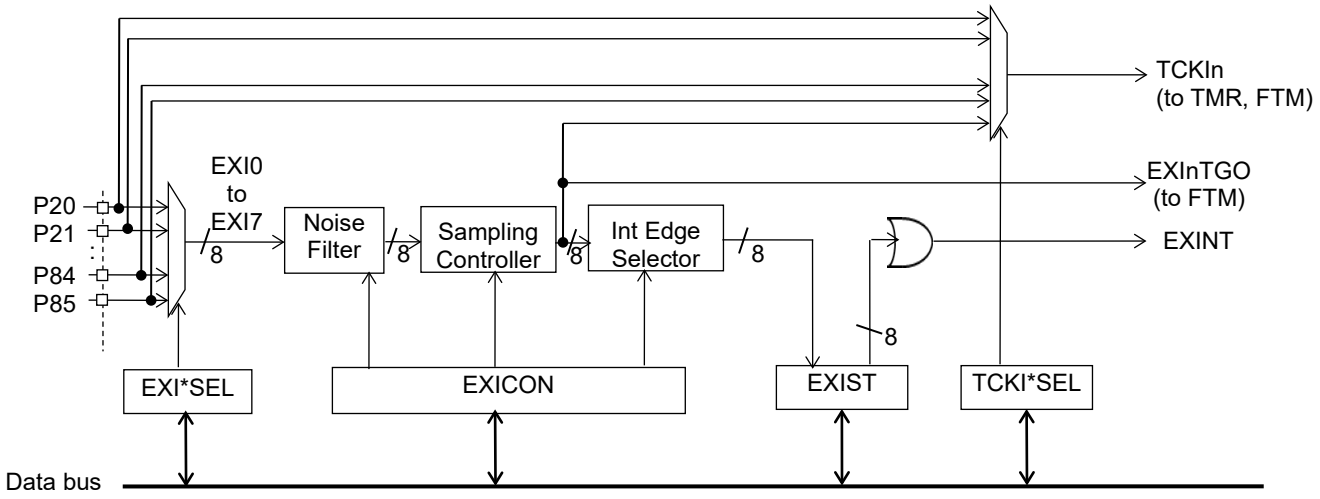
Figure 22-1 shows the configuration of the GPIO Port n.



PnDI : Port n data input register
PnDO : Port n data output register
PnMOD0 : Port n mode register 0
PnMOD1 : Port n mode register 1

Figure 22-1 Configuration of GPIO Port n (n= 2 to 8)

Figure 22-2 shows the circuit of the interrupt controller external interrupts and timer clock inputs.



EXICON : External interrupt control register
EXIST : External interrupt status register
EXI0 to 7 : External interrupt
EXI03/47SEL : External interrupt 03/47 selection register
TCKI03/47SEL : Timer clock input 03/47 selection register

Figure 22-2 Configuration of External Interrupt and Timer Clock Function

22.1.3 List of Pins

Table 22-1 List of Pins

Pin name	Description
P20 to P23	GPIO / EXI /TCKI
P30 to P37	GPIO / EXI /TCKI / AIN
P40 to P47	GPIO / EXI /TCKI
P50 to P57	GPIO / EXI /TCKI / AIN
P60 to P66	GPIO / EXI /TCKI
P70 to P77	GPIO / EXI /TCKI
P80 to P85	GPIO / EXI

22.2 Description of Registers

22.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0100	External interrupt control base address	NINT	-	-	-
0x00	Reserved	-	-	-	-
0x04	External Interrupt status register	EXIST	R/W	32	0x0000_0000
0x08	External interrupt control register	EXICON	R/W	32	0x0000_0000
0x0C	External interrupt 03 selection register	EXI03SEL	R/W	32	0x0000_0000
0x10	External interrupt 47 selection register	EXI47SEL	R/W	32	0x0000_0000
0x20	Timer clock input 03 selection register	TCKI03SEL	R/W	32	0x0000_0000
0x24	Timer clock input 47 selection register	TCKI47SEL	R/W	32	0x0000_0000
0x4001_0040	Port 2 base address	PORT2	-	-	-
0x0	Port 2 data input register	P2DI	R	32	0x0000_00FF
0x4	Port 2 data output register	P2DO	R/W	32	0x0000_0000
0x8	Port 2 mode register 0	P2MOD0	R/W	32	0x0000_0000
0x4001_0060	Port 3 base address	PORT3	-	-	-
0x0	Port 3 data input register	P3DI	R	32	0x0000_00FF
0x4	Port 3 data output register	P3DO	R/W	32	0x0000_0000
0x8	Port 3 mode register 0	P3MOD0	R/W	32	0x0000_0000
0xC	Port 3 mode register 1	P3MOD1	R/W	32	0x0000_0000
0x4001_0080	Port 4 base address	PORT4	-	-	-
0x0	Port 4 data input register	P4DI	R	32	0x0000_00FF
0x4	Port 4 data output register	P4DO	R/W	32	0x0000_0000
0x8	Port 4 mode register 0	P4MOD0	R/W	32	0x0000_0000
0xC	Port 4 mode register 1	P4MOD1	R/W	32	0x0000_0000
0x4001_00A0	Port 5 base address	PORT5	-	-	-
0x0	Port 5 data input register	P5DI	R	32	0x0000_00FF
0x4	Port 5 data output register	P5DO	R/W	32	0x0000_0000
0x8	Port 5 mode register 0	P5MOD0	R/W	32	0x0000_0000
0xC	Port 5 mode register 1	P5MOD1	R/W	32	0x0000_0000
0x4001_00C0	Port 6 base address	PORT6	-	-	-
0x0	Port 6 data input register	P6DI	R	32	0x0000_00FF
0x4	Port 6 data output register	P6DO	R/W	32	0x0000_0000
0x8	Port 6 mode register 0	P6MOD0	R/W	32	0x0000_0000
0xC	Port 6 mode register 1	P6MOD1	R/W	32	0x0000_0000
0x4001_00E0	Port 7 base address	PORT7	-	-	-
0x0	Port 7 data input register	P7DI	R	32	0x0000_00FF
0x4	Port 7 data output register	P7DO	R/W	32	0x0000_0000
0x8	Port 7 mode register 0	P7MOD0	R/W	32	0x0000_0000
0xC	Port 7 mode register 1	P7MOD1	R/W	32	0x0000_0000
0x4001_0100	Port 8 base address	PORT8	-	-	-
0x0	Port 8 data input register	P8DI	R	32	0x0000_00FF
0x4	Port 8 data output register	P8DO	R/W	32	0x0000_0000
0x8	Port 8 mode register 0	P8MOD0	R/W	32	0x0000_0000
0xC	Port 8 mode register 1	P8MOD1	R/W	32	0x0000_0000

22.2.2 Port n Data Input Register (PnDI : n =2 to 8)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	PnDIB							
R/W	–	–	–	–	–	–	–	–	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

PnDI is a SFR used to read the level of the port n pin.
Enable or disable the input by using the port n mode register 0/1. Reading value from the bits of PnDI register that have no corresponding pins is always “1”.

Bit No	Bit name	Description
7 to 0	PnDIB[m] (m= Bit No)	PnDIB bits are used to read the input level of the port n pins. 0: Input level of Pnm pin is “L” 1: Input level of Pnm pin is “H” or input disabled state (Initial value)

22.2.3 Port n Data Output Register (PnDO : n =2 to 8)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	PnDOB							
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PnDO is a SFR used to write output data of the port n pin.
Enable or disable the input or output by using the port n mode register 0/1. Write “0” to the bits of PnDO that have no corresponding pin.

Bit No	Bit name	Description
7 to 0	PnDOB[m] (m= Bit No)	PnDOB bits are used to set the output level of the port n pins. 0: Output level of Pnm pin is “L” (Initial value) 1: Output level of Pnm pins is “H”

22.2.4 Port n Mode Register 0 (PnMOD0 : n=2 to 8)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	Pn3MD		Pn3OD	Pn3PU	Pn3OE	Pn3IE	–	–	Pn2MD		Pn2OD	Pn2PU	Pn2OE	Pn2IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	Pn1MD		Pn1OD	Pn1PU	Pn1OE	Pn1IE	–	–	Pn0MD		Pn0OD	Pn0PU	Pn0OE	Pn0IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PnMOD0 is a SFR to choose the input/output mode, input/output status, and shared function of Pnm pins (m = 0 to 3). Write “0” to the bits of PnMOD0 register that have no corresponding pins.

Bit No	Bit name	Description
29 to 28, 21 to 20, 13 to 12, 5 to 4	Pn3MD Pn2MD Pn1MD Pn0MD	This is used to choose the shared function of target pin. For the details of the shared function, see "1.3.2 List of Pins". 00: Primary function (Initial value) 01: Secondary function 10: Tertiary function 11: Quaternary function
27, 19, 11, 3	Pn3OD Pn2OD Pn1OD Pn0OD	This is used to choose the output type of target pin. 0: CMOS output (Initial value) 1: N-channel open drain output. An LED is directly drive-able by enlarging the current.
26, 18, 10, 2	Pn3PU Pn2PU Pn1PU Pn0PU	This is used to enable the internal pull-up resistor of target pin. 0: Disabled (Initial value) 1: Enabled
25, 17, 9, 1	Pn3OE Pn2OE Pn1OE Pn0OE	This is used to enable the output of target pin. When PWM output (quaternary function) is selected, an output enable of corresponding pin is controlled by the peripheral NTMS. Therefore, a setting by this bit is ignored. Also, when SCKF/SOUTF/SSNF output (secondary function) is selected, an output enable of corresponding pin is controlled by both of this bit and the peripheral SSIOF. 0: Disabled (Initial value) 1: Enabled
24, 16, 8, 0	Pn3IE Pn2IE Pn1IE Pn0IE	This is used to enable the input of target pin. 0: Disabled (Initial value) 1: Enabled

22.2.5 Port n Mode Register 1 (PnMOD1 : n=3 to 8)

Offset : 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	Pn7MD		Pn7OD	Pn7PU	Pn7OE	Pn7IE	–	–	Pn6MD		Pn6OD	Pn6PU	Pn6OE	Pn6IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	Pn5MD		Pn5OD	Pn5PU	Pn5OE	Pn5IE	–	–	Pn4MD		Pn4OD	Pn4PU	Pn4OE	Pn4IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PnMOD1 is a SFR to choose the input/output mode, input/output status, and shared function of Pnm pins (m = 4 to 7). Write “0” to the bits of PnMOD1 register that have no corresponding pins.

Bit No	Bit name	Description
29 to 28, 21 to 20, 13 to 12, 5 to 4	Pn7MD Pn6MD Pn5MD Pn4MD	This is used to choose the shared function of target pin. For the details of the shared function, see "1.3.2 List of Pins". 00: Primary function (Initial value) 01: Secondary function 10: Tertiary function 11: Quaternary function
27, 19, 11, 3	Pn7OD Pn6OD Pn5OD Pn4OD	This is used to choose the output type of target pin. 0: CMOS output (Initial value) 1: N-channel open drain output. An LED is directly drive-able by enlarging the current.
26, 18, 10, 2	Pn7PU Pn6PU Pn5PU Pn4PU	This is used to enable the internal pull-up resistor of target pin. 0: Disabled (Initial value) 1: Enabled
25, 17, 9, 1	Pn7OE Pn6OE Pn5OE Pn4OE	This is used to enable the output of target pin. When PWM output (quaternary function) is selected, an output enable of corresponding pin is controlled by the peripheral NTMS. Therefore, a setting by this bit is ignored. Also, when SCKF/SOUTF/SSNF output (secondary function) is selected, an output enable of corresponding pin is controlled by both of this bit and the peripheral SSIOF. 0: Disabled (Initial value) 1: Enabled
24, 16, 8, 0	Pn7IE Pn6IE Pn5IE Pn4IE	This is used to enable the input of target pin. 0: Disabled (Initial value) 1: Enabled

22.2.6 External Interrupt Status Register (EXIST)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	QEXI7	QEXI6	QEXI5	QEXI4	QEXI3	QEXI2	QEXI1	QEXI0
R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EXIST is a special function register (SFR) used to hold each interrupt request.
The bit which interrupt request is occurred can be clear by writing "1" to this bit.

[Description of each bit]
It is flag of the corresponding interrupt request.
0: No request
1: Request

Bit No	Bit name	Description (corresponding interrupt)
7	QEXI7	External interrupt 7 (EXI7)
6	QEXI6	External interrupt 6 (EXI6)
5	QEXI5	External interrupt 5 (EXI5)
4	QEXI4	External interrupt 4 (EXI4)
3	QEXI3	External interrupt 3 (EXI3)
2	QEXI2	External interrupt 2 (EXI2)
1	QEXI1	External interrupt 1 (EXI1)
0	QEXI0	External interrupt 0 (EXI0)

22.2.7 External Interrupt Control Register (EXICON)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EXI7 FL	EXI7 SM	EXI7E		EXI6 FL	EXI6 SM	EXI6E		EXI5 FL	EXI5 SM	EXI5E		EXI4 FL	EXI4 SM	EXI4E	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXI3 FL	EXI3 SM	EXI3E		EXI2 FL	EXI2 SM	EXI2E		EXI1 FL	EXI1 SM	EXI1E		EXI0 FL	EXI0 SM	EXI0E	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EXICON is a special function register (SFR) used to set the interrupt edge, filter, sampling of external interrupt.

Bit No	Bit name	Description
31, 27, 23, 19, 15, 11, 7, 3	EXI7FL EXI6FL EXI5FL EXI4FL EXI3FL EXI2FL EXI1FL EXI0FL	This is used to enable an analog noise filter. 0: Disabled (Initial value) 1: Enabled
30, 26, 22, 18, 14, 10, 6, 2	EXI7SM EXI6SM EXI5SM EXI4SM EXI3SM EXI2SM EXI1SM EXI0SM	This is used to enable sampling function. The sampling clock is T16KHZ of the low-speed time base counter (LTBC). 0: Disabled (Initial value) 1: Enabled
29 to 28, 25 to 24, 21 to 20, 17 to 16, 13 to 12, 9 to 8, 5 to 4, 1 to 0	EXI7E EXI6E EXI5E EXI4E EXI3E EXI2E EXI1E EXI0E	This is used to select the interrupt mode. 00: Interrupt disabled (Initial value) 01: Falling-edge interrupt 10: Rising-edge interrupt 11: Both-edge interrupt

[Note]

- **In STOP mode, since the sampling clock (T16KHZ) stops, no sampling is performed regardless of the values set in EXI7-0SM.**

22.2.8 External interrupt 03 Selection Register (EXI03SEL)

Offset : 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EXI3S								EXI2S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXI1S								EXI0S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EXI03SEL is a special function register (SFR) used to select the port used as EXI0 to EXI3.

Bit No	Bit name	Description
31 to 24	EXI3S	When assigned port is Pnm, set value 'n' to bit [7:4] and value 'm' to bit [3:0].
23 to 16	EXI2S	Table 22-2 shows setting list.
15 to 8	EXI1S	For example: the P41 is assigned to EXI0 when EXI0S=0x41.
7 to 0	EXI0S	

22.2.9 External interrupt 47 Selection Register (EXI47SEL)

Offset : 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EXI7S								EXI6S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EXI5S								EXI4S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EXI47SEL is a special function register (SFR) used to select the port used as EXI4 to EXI7.

Bit No	Bit name	Description
31 to 24	EXI7S	When assigned port is Pnm, set value 'n' to bit [7:4] and value 'm' to bit [3:0].
23 to 16	EXI6S	Table 22-2 shows setting list.
15 to 8	EXI5S	For example: the P41 is assigned to EXI4 when EXI4S=0x41.
7 to 0	EXI4S	

Table 22-2 External interrupt pin setting (`-*` : setting prohibited)

EXInS[3:0] / EXInS[7:4]	0	1	2	3	4	5	6	7
0	-*	-*	-*	-*	-*	-*	-*	-*
1	-*	-*	-*	-*	-*	-*	-*	-*
2	P20	P21	P22	P23	-*	-*	-*	-*
3	P30	P31	P32	P33	P34	P35	P36	P37
4	P40	P41	P42	P43	P44	P45	P46	P47
5	P50	P51	P52	P53	P54	P55	P56	P57
6	P60	P61	P62	P63	P64	P65	P66	-*
7	P70	P71	P72	P73	P74	P75	P76	P77
8	P80	P81	P82	P83	P84	P85	-*	-*

22.2.10 Timer Clock Input 03 Selection Register (TCKI03SEL)

Offset : 0x0C

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCKI3S								TCKI2S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCKI1S								TCKI0S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCKI03SEL is a special function register (SFR) used to select the port used as TMCKI0 to TMCKI3.

Bit No	Bit name	Description
31 to 24	TCKI3S	When assigned port is Pnm, set value 'n' to bit [7:4] and value 'm' to bit [3:0].
23 to 16	TCKI2S	Table 22-3 shows setting list.
15 to 8	TCKI1S	For example: the P41 is assigned to TMCKI0 when TCKI0S=0x41.
7 to 0	TCKI0S	

22.2.11 Timer Clock Input 47 Selection Register (TCKI47SEL)

Offset : 0x10

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCKI7S								TCKI6S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCKI5S								TCKI4S							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCKI47SEL is a special function register (SFR) used to select the port used as TMCKI4 to TMCKI7.

Bit No	Bit name	Description
31 to 24	TCKI7S	When assigned port is Pnm, set value 'n' to bit [7:4] and value 'm' to bit [3:0].
23 to 16	TCKI6S	Table 22-3 shows setting list.
15 to 8	TCKI5S	For example: the P41 is assigned to TMCKI0 when TCKI0S=0x41.
7 to 0	TCKI4S	

Table 22-3 Timer clock input pin setting (`-*` : setting prohibited)

TCKInS[7:4] \ TCKInS[3:0]	0	1	2	3	4	5	6	7
0	-*	-*	-*	-*	-*	-*	-*	-*
1	-*	-*	-*	-*	-*	-*	-*	-*
2	P20	P21	P22	P23	-*	-*	-*	-*
3	P30	P31	P32	P33	P34	P35	P36	P37
4	P40	P41	P42	P43	P44	P45	P46	P47
5	P50	P51	P52	P53	P54	P55	P56	P57
6	P60	P61	P62	P63	P64	P65	P66	-*
7	P70	P71	P72	P73	P74	P75	P76	P77
8	P80	P81	P82	P83	P84	P85	-*	-*
F	EXI0TGO	EXI1TGO	EXI2TGO	EXI3TGO	EXI4TGO	EXI5TGO	EXI6TGO	EXI7TGO

22.3 Description of Operation

22.3.1 Input Function

Each pin of general purpose port n sets the PnmIE bit of the PnMOD0/1 register to enter the state where input is enabled. In the state with input enabled, the pin level can be read using the PnDI. In addition, pull-up can be enabled by setting the PnmPU bit of the PnMOD0/1 register.
At a system reset, input disabled and no pull-up are selected as the initial status of pins.

22.3.2 Output Function

Each pin of general purpose port n sets the PnmOD bit of the PnMOD0/1 register to choose either CMOS output or N-channel open drain output as an output type and sets PnmOE bit of the PnMOD0/1 register to enter the state where output is enabled.
In the state with output enabled, “L” or “H” level is output to each pin of the general purpose port according to the value set in the PnDO.
At a system reset, output disabled and CMOS output are selected as the initial status.

22.3.3 Primary Functions Other than Input/Output Function

External interrupt input, timer clock input, analog input for SA-ADC and comparator can be used as the primary function other than the input/output function.
When using as external interrupt input or timer clock input, set the PnMOD0/1 register of the applicable port to input enabled (PnmIE bit="1").
When using as analog input for SA-ADC or comparator, set the PnMOD0/1 register of the applicable port to input disabled (PnmIE bit="0" and PnmOE bit="0").

22.3.4 Shared Functions

Each pin of general purpose port n can use secondary to quaternary functions as the shared function.
Set the PnmMOD of the PnMOD0/1 register to choose each of the primary to quaternary functions.

22.3.5 External interrupt

When an interrupt edge selected with the external interrupt control register (EXICON) occurs at one of external interrupts EXI0 to 7, any of the maskable EXI0 to EXI7 interrupts occurs.
It is possible to perform the filtering with noise filtering and/or sampling (2 clocks sampling with T16KHZ that is 2 dividing of LSCLK) for an external pin input.
The EXIINT interrupt is an OR logic with eight statuses of QEXI0 to 7. The EXIINT will not deassert unless all QEXIs are cleared.
The signals for the trigger of FTM (EXI[7:0]TGO) are signals that have been controlled by noise filter and sampling according to the EXICON settings.

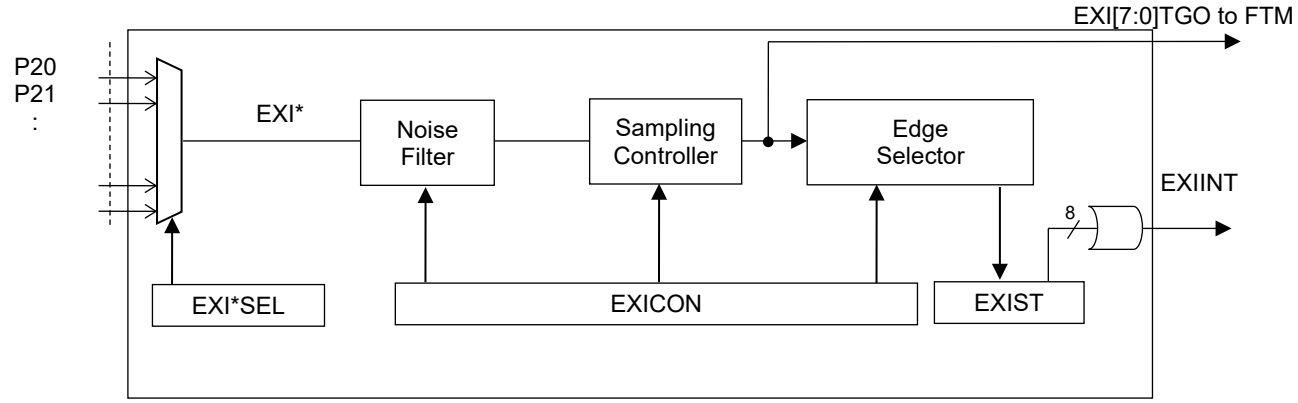
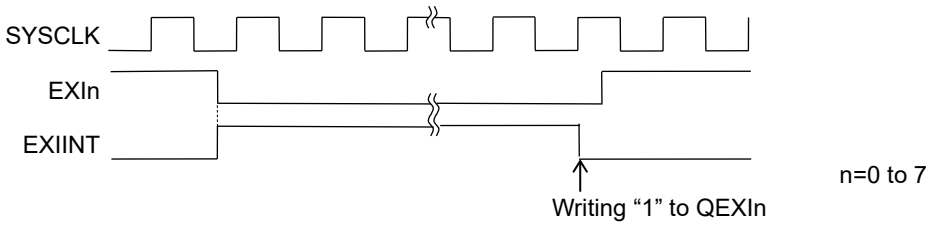
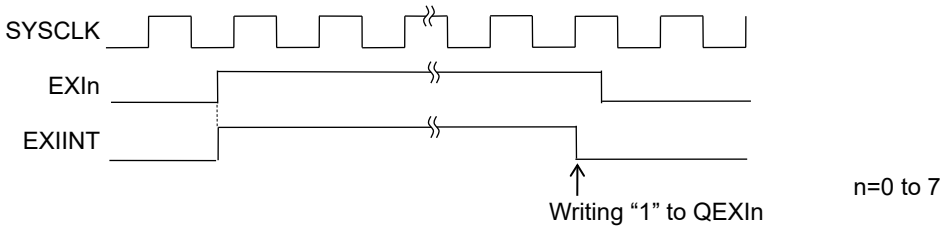


Figure 22-3 External Interrupt Generation circuit

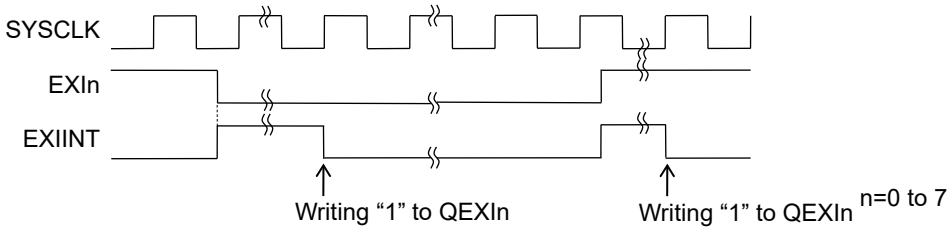
Figure 22-4 shows the interrupt timing in rising-edge interrupt mode, in falling-edge interrupt mode, and in both-edge interrupt mode without sampling, and in rising-edge interrupt mode with sampling.



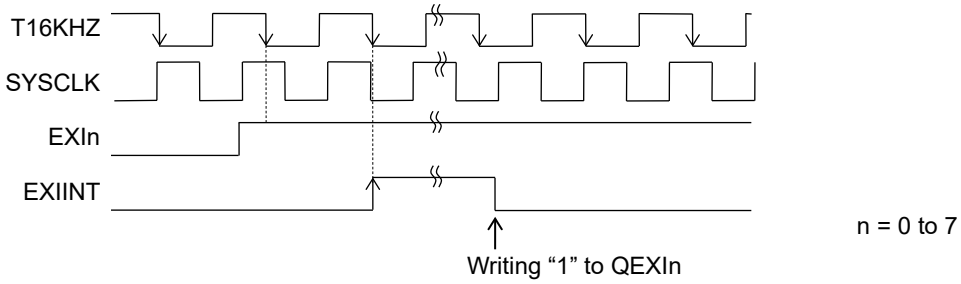
(a) When Falling-Edge Interrupt Mode without Sampling is Selected



(b) When Rising-Edge Interrupt Mode without Sampling is Selected



(c) When Both-Edge Interrupt Mode without Sampling is Selected



(d) When Rising-Edge Interrupt Mode with Sampling is Selected

Figure 22-4 External Interrupt Generation Timing

Figure 22-5 shows the interrupt generation timing with multiple EXIs in rising-edge interrupt mode.

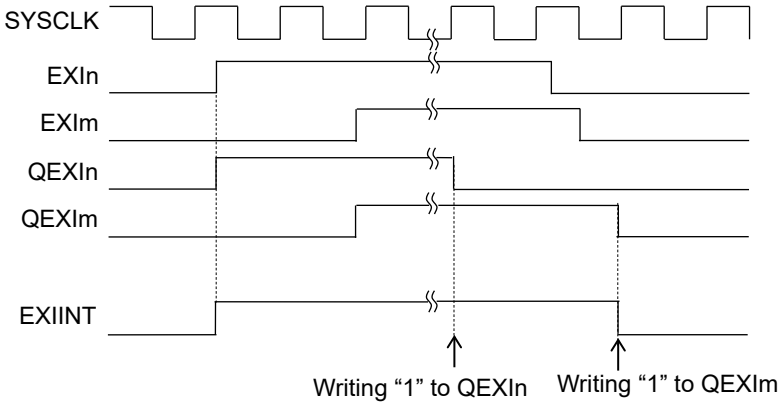


Figure 22-5 External Interrupt Generation Timing with multiple EXIs

22.3.6 Timer Clock Input

Any pins or EXIs are assigned to TMCKI0 to TMCKI7 which is used to clock of the timers or functional timers.

Chapter 23

CRC Calculator

23. CRC Calculator

23.1 Overview

This LSI has the CRC (Cyclic Redundancy Check) generator that performs CRC calculation and generates the CRC data used for error detection in serial communications.

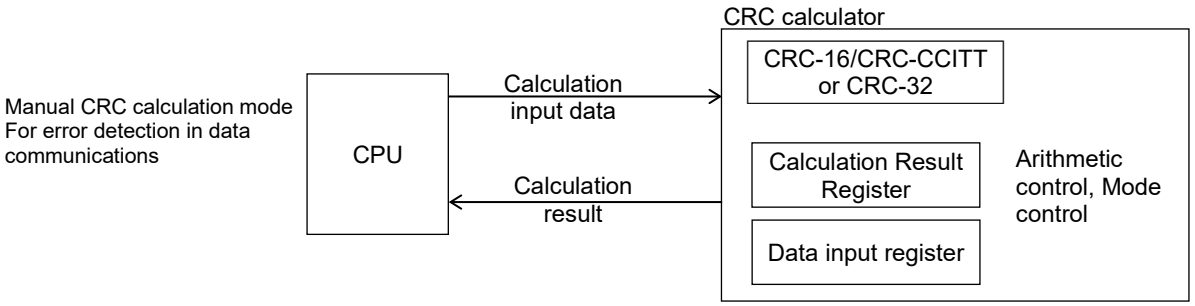


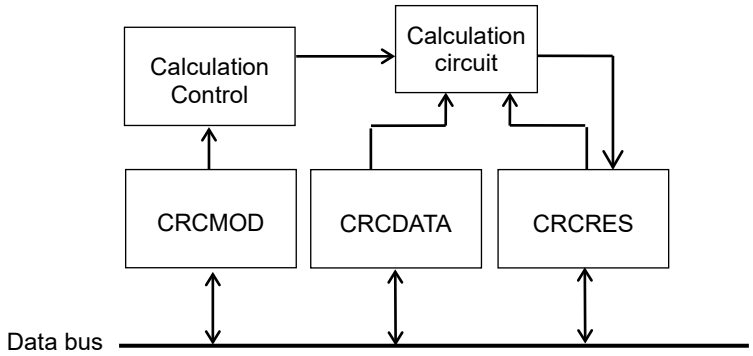
Figure 23-1 CRC Calculator Overview

23.1.1 Features

- Manual CRC calculation mode
  Generates CRC data from data set in CRC calculation register by the software
  Calculation unit is 8bit or 32bit
- Generator polynomial:
  CRC-16/CRC-CCITT ($X^{16}+X^{12}+X^5+1$)
  CRC-32 ($X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X^1+1$)
- MSB first or LSB first selectable

23.1.2 Configuration

Figure 23-2 shows the configuration of the CRC calculator.



CRCMOD : CRC Calculation Mode Register
CRCDATA : CRC Calculation Data Register
CRCRES : CRC Calculation Result Register

Figure 23-2 Configuration of CRC Generator

23.2 Description of Registers

23.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4400_0000	CRC base address	CRC	-	-	-
0x00	CRC Calculation Data Register	CRCDATA	R/W	32	0x0000_0000
0x04	CRC Calculation Result Register	CRCRES	R/W	32	0xFFFF_FFFF
0x08	CRC Calculation Mode Register	CRCMOD	R/W	32	0x0000_0000

23.2.2 CRC Calculation Data Register (CRCDATA)

Offset : 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	d[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDATA is a special function register (SFR) used to set the CRC calculation data.
Set it by 8 or 32 bits. One clock after writing data to the CRCDATA, the calculation result is stored in the CRC Calculation Result Register (CRCRES).

When CRCSIZ bit of CRCMOD register is set to “0” (8 bit mode), d7 to d0 bits are valid.
When CRCSIZ bit of CRCMOD register is set to “1” (32 bit mode), d31 to d0 bits are valid.

23.2.3 CRC Calculation Result Register (CRCRES)

Offset : 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	d[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCRES is a special function register (SFR). The CRC calculation result is stored by the hardware.
Set data to the CRCRES as an initial data for the CRC calculation.

23.2.4 CRC Calculation Mode Register (CRCMOD)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	CRCMD	CRCSIZ	CRCDIR	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCMOD is a special function register (SFR) used to control the CRC calculation mode.

Bit No	Bit name	Description
3	CRCMD	This bit is used to choose the generator polynomial of CRC calculation. 0: CRC16／CRC-CCITT (Initial value) 1: CRC32
2	CRCSIZ	This bit is used to choose the bit width of input data. 0: 8 bit (Initial value) 1: 32 bit
1	CRCDIR	This bit is used to choose the shift direction of the CRC calculation. 0: LSB first (Initial value) 1: MSB first

[Note]

- **When CRCSIZ is “1” (32bit is selected), CRC calculation is four-byte length. Generate an expected value by four bytes.**

23.3 Description of Operation

The CRC calculation result is outputted to the CRC calculation result register (CRCRES) by writing the initial value to the CRC calculation result register (CRCRES) then writing data to 8-bit or 32-bit CRC calculation data register (CRCDATA). For data error detection in serial communication, etc., presence of errors can be detected by transferring data with the calculation result attached when transmission and performing the same CRC calculation in the reception side.

23.3.1 Example of Use

The following chart shows the process flow of serial transmission with the CRC calculation result attached to data. In this example, 11-byte data with 0x21 in the beginning is used as transmit data, and calculation result is obtained.
Transmission and CRC calculation data: 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x81, 0x7F

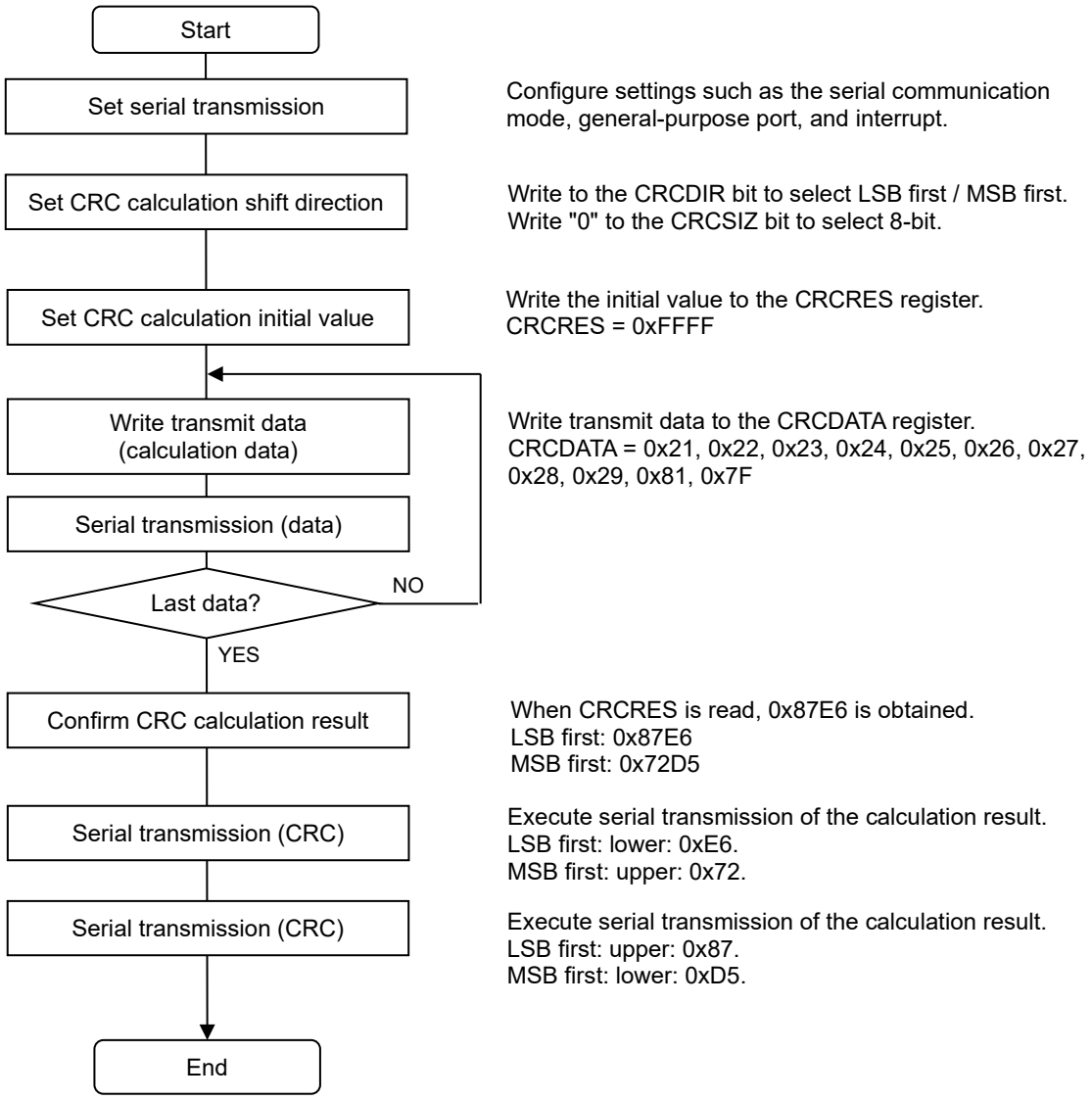


Figure 23-3 CRC Calculation Processing Flow 1 (Serial Transmission)

The following chart shows the CRC calculation process flow with the CRC calculation result attached to the serial receive data.
In this example, 13-byte data with 0x21 in the beginning is used as calculation data. From the calculation data, calculation result is obtained. The first 11 bytes of the CRC calculation result is added to the last two bytes.

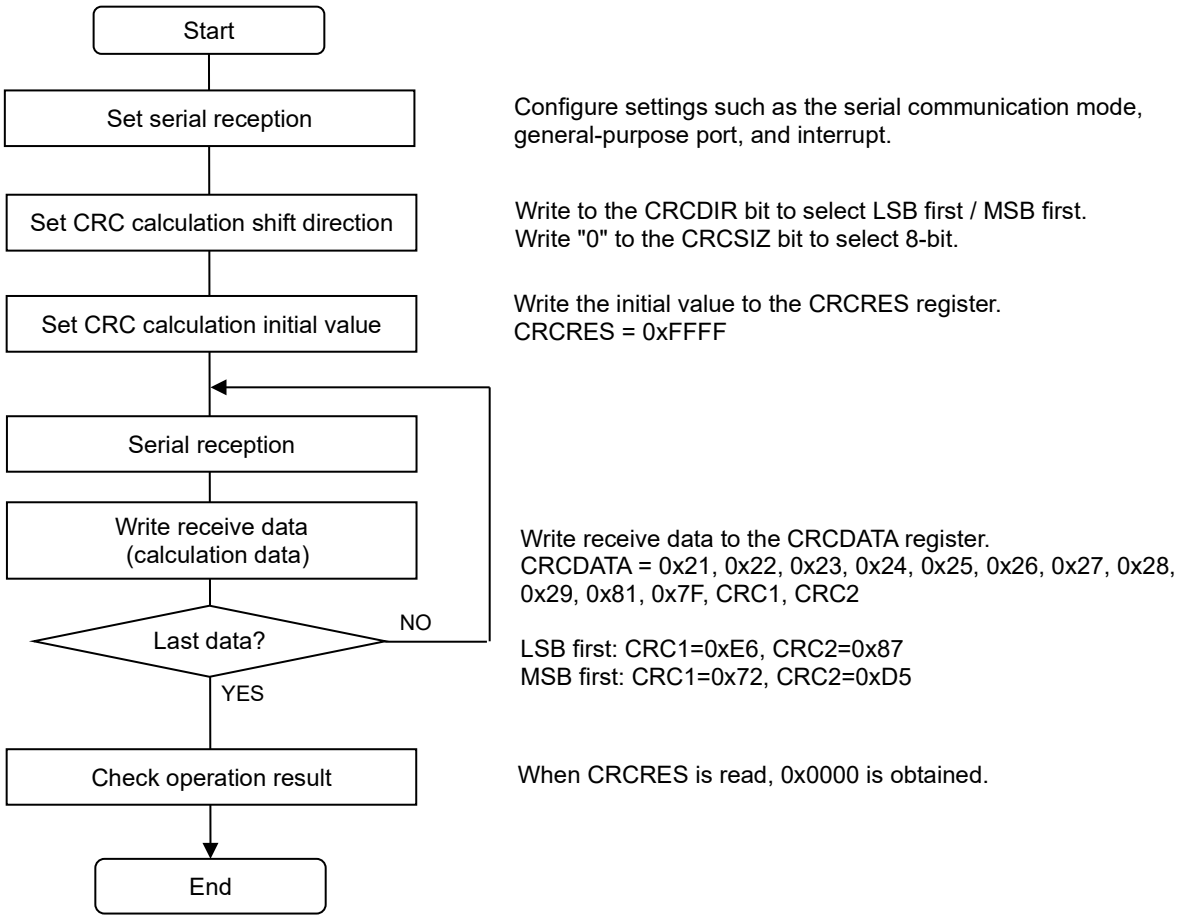


Figure 23-4 CRC Calculation Processing Flow 2 (Serial Reception)

23.3.2 Operation Timing Chart

Set the initial value of CRC calculation in the CRCRES register. When 8-bit or 32-bit data is written to the CRCDATA register, the calculation result is stored in the CRCRES register on the next clock rising-edge. The CRC calculation result can be checked anytime by reading the CRCRES register.
Figure 23-5 shows the operation timing chart of CRC calculation

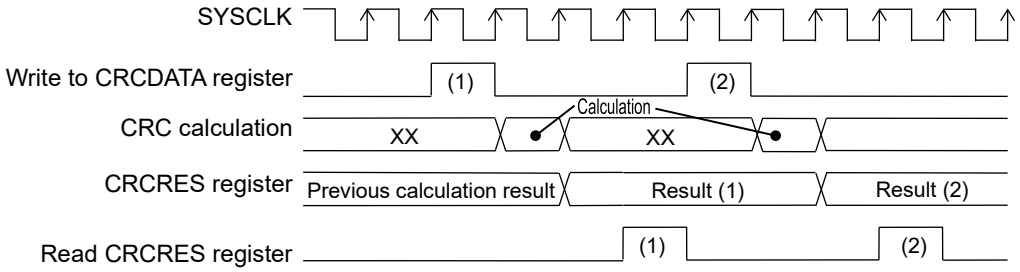


Figure 23-5 Timing Chart of CRC Calculation

23.3.3 Flow when performing another CRC calculation during CRC calculation using DMAC

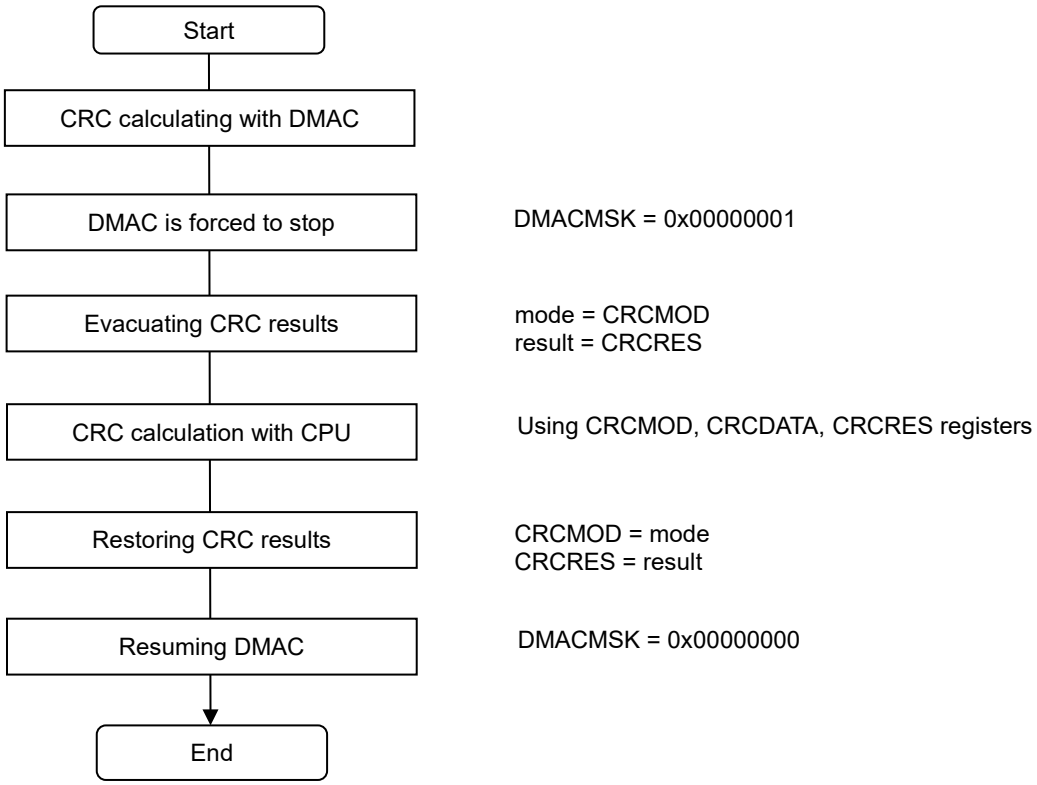


Figure 23-6 CRC Calculation Processing Flow 3 (CRC calculation in progress)

Chapter 24

(Skipped)

Chapter 25

Solist-AI™ Acceralator

25. Solist-AI™ Acceralator (ML_ACC)

25.1 Overview

A built-in Solist-AI™ acceralator can perform high-speed calculations that are heavily used by AI.

25.1.1 Features

- Not only prediction but also learning can be performed on the device. (No need for server/cloud/network connection)
  Detect anomaly conditions by learning normal conditions for each individual.
- Capable of high-speed execution of calculations used in AI processing.
  - Addition, subtraction and multiplication of scalars, vectors, and (non-square) matrices are possible.
  - Enables FFT calculation processing useful for vibration sensor data processing.
  - Calculations can be executed without CPU load.
- Data format: bFloat16 (Built-in integer to bFloat16 conversion function.)
- Low power consumption/low cost. (Compared to FPGA/GPU etc.), High-speed processing. (Compared to software processing.)
- One-stop utility using model-based technology.
  Provides software library.
- Built-in bFloat16-format uniform distribution pseudo-random number generator that can be used as a fixed table.
- Application example.
  - Motor + acceleration sensor + AI: Early detection of bearing damage.
  - Motor + current sensor + AI : Detection of poor lubrication/contamination of foreign objects.
  - Thermography camera + AI : Accurate detection of abnormal heat generation.
  - FA sensor + AI : Early detection of random failures and abnormal conditions.

25.1.2 Configuration

Figure 25-1 shows configuration of ML_ACC.

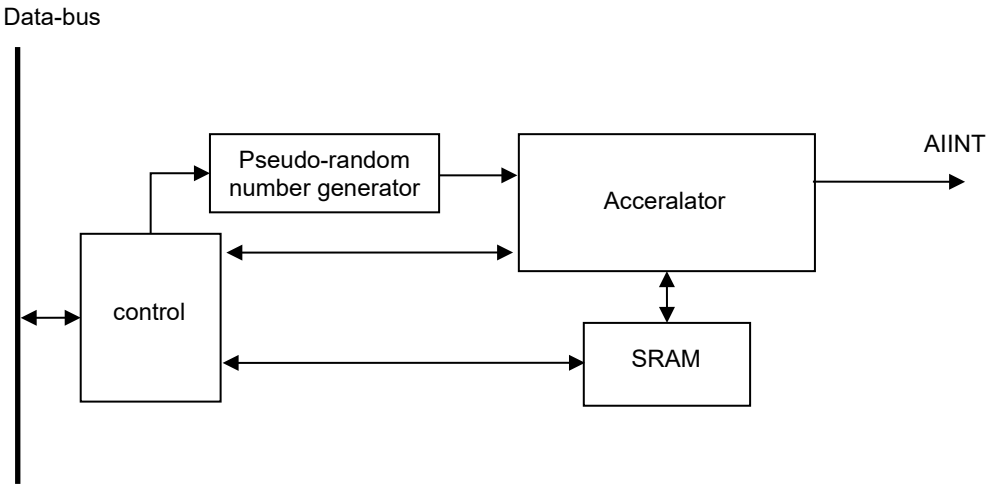


Figure 25-1 Configuration of ML_ACC

25.2 Description of Operation

See to the manual of software (library, driver etc.) for details on how to use the ML_ACC.

[Note]
- **Do not input or retrieve data while the ML_ACC is running (i.e. in the busy state).**

25.2.1 RAM Parity Error

A data written to RAM from somewhere is added parity 1 bit per byte.
If the AI_PAR_EN of the MCUINTEN register of the system control function is set to "1", the AI_PAR of the MCUISTAT register is set to "1" and the MCU interrupt request is occurred when a parity error is detected when RAM is read.

[Note]
- **Since the initial value of RAM is indefinite, a parity error may occur if it is read without writing it once.**

Chapter 26

Successive Approximation Type A/D Converter (SA-ADC)

26. Successive Approximation Type A/D Converter (SA-ADC)

26.1 Overview

This LSI has 2 units of the Successive Approximation type A/D Converter (SA-ADC) that converts an analog input level to a digital value.

26.1.1 Features

- Resolution : 12 bit
- Conversion time : Minimum 0.917 μs / input channel (when conversion clock is 24MHz)
- Number of unit : 2
- Number of input channel : 12
- Reference voltage: Voltage input from External reference voltage (VREF pin)
- Sampling time can be chosen
- Consecutive scan conversion function for target input channels
- Consecutive scan conversion with a specific interval time
- One conversion result register for each input channel
- Upper /Lower limit is configurable for the conversion result, generates an interrupt
- A/D converter self test function (full scale, zero scale, internal reference voltage)
- Simultaneous conversion of 2inputs with 2 units is possible.
- Following triggers is available to start the A/D conversion
  - Timer interrupt (TMnINT)
  - Functional timer trigger (FTMnTGO)
  - Three-phase motor control PWM trigger (NTMSnTGO)

26.1.2 Configuration

Figure 26-1 and Figure 26-2 shows the configuration of SA-ADC.

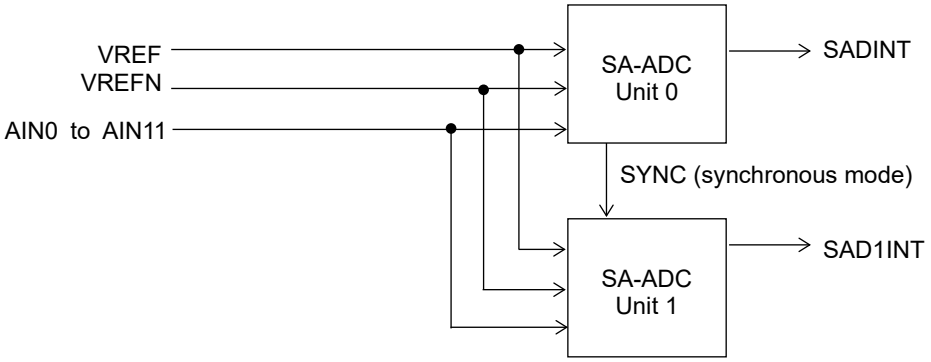
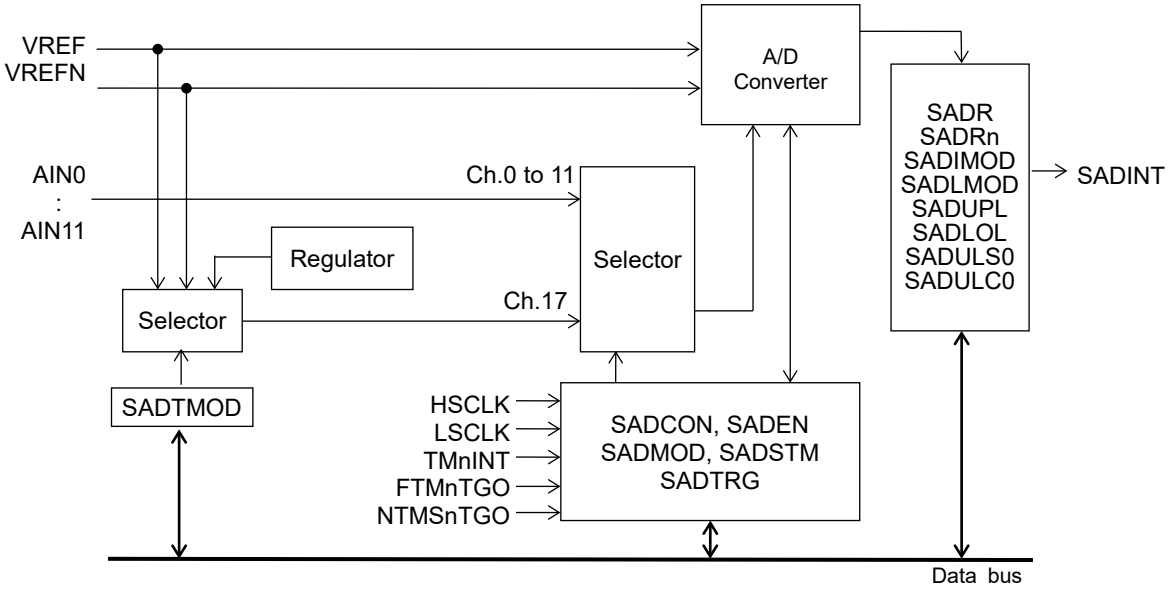


Figure 26-1 Units Configuration of SA-ADC



SADCON	:	SA-ADC control register	SADINT	:	SA-ADC interrupt request
SADEN	:	SA-ADC enable register	TMnINT	:	Timer n interrupt
SADMOD	:	SA-ADC mode register	FTMnTGO	:	Functional timer n trigger
SADSTM	:	SA-ADC conversion interval register	NTMSnTGO	:	Three-phase motor control PWM trigger
SADR	:	SA-ADC result register			
SADRn	:	SA-ADC result register n (n=0 to 11)			
SADIMOD	:	SA-ADC interrupt mode register			
SADLMOD	:	SA-ADC upper/lower limit mode register			
SADUPL	:	SA-ADC upper limit setting register			
SADLOL	:	SA-ADC lower limit setting register			
SADULS0	:	SA-ADC upper/lower limit status register			
SADULC0	:	SA-ADC upper/lower limit status clear register			
SADTRG	:	SA-ADC trigger register			
SADTMOD	:	SA-ADC test mode register			

Figure 26-2 Configuration of SA-ADC

26.1.3 List of Pins

Table 26-1 Pins

Pin name	I/O	Description
P32/AIN0	I	SA-ADC analog input channel 0
P33/AIN1	I	SA-ADC analog input channel 1
P34/AIN2	I	SA-ADC analog input channel 2
P35/AIN3	I	SA-ADC analog input channel 3
P36/AIN4	I	SA-ADC analog input channel 4
P37/AIN5	I	SA-ADC analog input channel 5
P50/AIN6	I	SA-ADC analog input channel 6
P51/AIN7	I	SA-ADC analog input channel 7
P52/AIN8	I	SA-ADC analog input channel 8
P55/AIN9	I	SA-ADC analog input channel 9
P56/AIN10	I	SA-ADC analog input channel 10
P57/AIN11	I	SA-ADC analog input channel 11
VREF	I	Reference voltage positive input for SA-ADC
VREFN	I	Reference voltage negative input for SA-ADC

26.2 Description of Registers

26.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4200_1000	SA-ADC0 base address	SADC0	-	-	-
0x4200_1100	SA-ADC1 base address	SADC1	-	-	-
0x00	SA-ADC mode register	SADMOD	R/W	32	0x0000_0100
0x04	SA-ADC control register	SADCON	R/W	32	0x0000_0000
0x08	SA-ADC conversion interval register	SADSTM	R/W	32	0x0000_0000
0x0C	Reserved	-	R/W	32	0x0000_0010
0x10	SA-ADC interrupt mode register	SADIMOD	R/W	32	0x0000_0000
0x14	SA-ADC trigger register	SADTRG	R/W	32	0x0000_0000
0x18	SA-ADC enable register	SADEN	R/W	32	0x0000_0000
0x1C	SA-ADC upper/lower limit mode register	SADLMOD	R/W	32	0x0000_0000
0x20	SA-ADC upper limit setting register	SADUPL	R/W	32	0x0000_FFF0
0x24	SA-ADC lower limit setting register	SADLOL	R/W	32	0x0000_0000
0x28	SA-ADC upper/lower limit status register	SADULS	R	32	0x0000_0000
0x2C	SA-ADC upper/lower limit status clear register	SADULC	W	32	0x0000_0000
0x30	SA-ADC test mode register	SADTMOD	R/W	32	0x0000_0000
0x34	SA-ADC result register	SADR	R	32	0x0000_0000
0x38	SA-ADC result register 0	SADR0	R	32	0x0000_0000
0x3C	SA-ADC result register 1	SADR1	R	32	0x0000_0000
0x40	SA-ADC result register 2	SADR2	R	32	0x0000_0000
0x44	SA-ADC result register 3	SADR3	R	32	0x0000_0000
0x48	SA-ADC result register 4	SADR4	R	32	0x0000_0000
0x4C	SA-ADC result register 5	SADR5	R	32	0x0000_0000
0x50	SA-ADC result register 6	SADR6	R	32	0x0000_0000
0x54	SA-ADC result register 7	SADR7	R	32	0x0000_0000
0x58	SA-ADC result register 8	SADR8	R	32	0x0000_0000
0x5C	SA-ADC result register 9	SADR9	R	32	0x0000_0000
0x60	SA-ADC result register 10	SADR10	R	32	0x0000_0000
0x64	SA-ADC result register 11	SADR11	R	32	0x0000_0000

26.2.2 SA-ADC Mode Register (SADMOD)

Offset : 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	SAINITT				–	–	–	SASHT				
R/W	–	–	–	–	R/W	R/W	R/W	R/W	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	SABC			-	SACK			SYNC	-	-	-	-	-	SAINIT	SALP
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

This is a SFR to set an operation mode and operating clock frequency for SA-ADC.
When SYNC bit is "1", a reading this register of unit 1 except the SYNC bit is disabled, and reading value is 0x00000080.

Bit No	Bit name	Description
27 to 24	SAINITT	This is used to be configured amplifier stability time at conversion starting. Wait time for stability [s] = setting value / SAD_CLK frequency This time should be equal or more than 0.5µs. When SAINIT="1", it is included discharge time for sample hold capacitor. In this case, this time should be equal or more than 0.625µs. Table 26-2 shows example for typical setting.
20 to 16	SASHT	This is used to set the sampling time. Sampling time [s] = (setting value + 1) / SAD_CLK frequency This setting value should be equal or more than 3. And in the case that an impedance of analog input is equal or less than 1 kΩ, this time should be equal or more than 0.33µs at V_{REF} ≥ 4.5V, 0.5µs at V_{REF} ≥ 2.7V, 0.6µs at V_{REF} ≥ 2.4V, 1.6µs at V_{REF} ≥ 2.1V. Table 26-3 shows example for typical setting. Make decision on the value with external impedance of input pin. See "26.4.1 Sampling Time Setting" for detail.
14 to 12	SABC	This is used to setting the adjustment for characteristics of SA-ADC. Set this bit to 0x4 before starting conversion.
10 to 8	SACK	This is used to choose the frequency of the A/D conversion operating clock (SAD_CLK). The SAD_CLK frequency should be equal or lower 24MHz as nominal value. 000: OSCLK 001: OSCLK / 2 010: OSCLK / 4 011: OSCLK / 8 100: OSCLK / 16 101: OSCLK / 3 110: OSCLK / 6 111: LSCLK The following formula is calculated A/D conversion time without discharge/amp. stability time. A/D conversion time [s] = (SASHT4 to SASHT0 value + 15) / SAD_CLK frequency
7	SYNC	This is used to set the synchronous mode of simultaneous conversion of unit 0 and unit 1. This bit in the unit 0 has no function. 0: Normal mode (Initial value) 1: Synchronous mode
1	SAINIT	This is used to control whether or not to discharge the electrical charge remained in the sample hold capacitor on the previous A/D conversion, before starting the next SA-ADC conversion. 0: Without discharge (Initial value) 1: With discharge
0	SALP	This bit is used to choose whether the A/D conversion is performed once only for each input channel or consecutively. The conversion interval time in the consecutive scan A/D conversion mode is specified in the SADSTM register. 0: One time conversion (Initial value) 1: Consecutive scan conversion

[Note]

- **If using both unit 0 and 1 with SYNC="0", the conversion accuracy may be affected by the timing.**

Table 26-2 Example for SAINITT setting

SAD_CLK	SAINITT SAINIT=1 (Discharge time / Amp. Stability time > 0.625us)	SAINITT SAINIT=0 (Amp. Stability time > 0.5us)
24 MHz	1111	1100
20 MHz	1100	1001
16 MHz	1010	1000
12 MHz	1000	0110
6 MHz	0100	0011
≤ 3 MHz	0010	0010

Table 26-3 Example for SASHT setting

SAD_CLK	SASHT (Sampling time)			
	V_{REF} ≥ 2.1V	V_{REF} ≥ 2.4V	V_{REF} ≥ 2.7V	V_{REF} ≥ 4.5V
24 MHz	-	-	-	00111
20 MHz	-	-	-	00110
Up to16 MHz	-	-	00111	00101
Up to 8 MHz	-	00111	00011	00011
2.5 MHz	01001	00011	00011	00011
32.768 kHz	00011	00011	00011	00011

26.2.3 SA-ADC Control Register (SADCON)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SATGEN	SARUN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SADCON is a SFR used to control the operation of the A/D converter.
When SYNC bit of SADMOD register is “1”, a reading /writing this register of unit 1 is disabled, and reading value is 0x00000000.

Bit No	Bit name	Description
1	SATGEN	This bit is used to enable starting the A/D conversion by the trigger events. 0: Disabled (Initial value) 1: Enabled
0	SARUN	This bit is used to start or stop the A/D conversion. When "0" is written to SALP bit and the A/D conversion on the largest number of channel is ended, this SARUN bit is automatically reset to "0". When "1" is written to SALP, the A/D conversion repeats until the SARUN bit is reset to "0" by the software. 0: Stops the conversion (Initial value) 1: Starts the conversion

[Note]

- **Start the A/D conversion with one or more channels chosen by the SA-ADC enable registers (SADEN). If no channel is chosen, the operation does not start.**
- **Enter STOP mode after checking SARUN bit is "0".**
- **When SACK is set to 0x7, it takes max. 3 clocks of the low-speed clock (LSCLK) to start or stop the A/D conversion after setting or resetting the SARUN bit.**

26.2.4 SA-ADC Conversion Interval Register (SADSTM)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SASTM															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to set the interval time in the consecutive scan A/D conversion mode.
When SYNC bit of SADMOD register is “1”, a reading /writing this register of unit 1 is disabled, and reading value is 0x00000000.

Bit No	Bit name	Description
15 to 0	SASTM	SASTM is used to set A/D conversion interval time in the consecutive scan conversion mode. A/D conversion interval time = SAD_CLK cycle x SADSTM setting value If SAINIT=0 : Do not set anything other than 0x0000 If SAINIT=1 : 0x0000 to 0xFFFF can be set. When multiple input channels are selected, the conversion interval is the time from the completion of the conversion of the last channel to the start of conversion of the first channel. For an example, supposing to A/D convert channel 2 and channel 5, the A/D conversion interval time means the time after the channel 2 and channel 5 are A/D converted consecutively and before the A/D conversion of channel 2 is started.

26.2.5 Reserved

Offset : 0x0C

This register is reserved. Do not change from initial value.

26.2.6 SA-ADC Interrupt Mode Register (SADIMOD)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SADIMD	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to choose the interrupt mode of the SA-ADC.
When SYNC bit of SADMOD register is “1”, a reading /writing this register of unit 1 is disabled, and reading value is 0x00000000.

Bit No	Bit name	Description
1	SADIMD[1]	This bit is used to choose the occurrence timing of SA-ADC interrupt request with upper/lower limit detection function. 0: Makes the interrupt request at a timing corresponding to SADIMD[0] setting, only when the detection function result coincides. (Initial value) 1: Makes the interrupt request at a timing corresponding to SADIMD[0] setting without the detection function result.
0	SADIMD[0]	This bit is used to choose the occurrence timing of SA-ADC interrupt request. 0: Makes the interrupt request after the A/D conversion is completed on all channels (Initial value) 1: Makes the interrupt request whenever the A/D conversion is completed on each channel

26.2.7 SA-ADC Trigger Register (SADTRG)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	SASTS				
R/W	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to control the trigger event for the SA-ADC.
When SYNC bit of SADMOD register is “1”, a reading /writing this register of unit 1 is disabled, and reading value is 0x00000000.

Bit No	Bit name	Description
4 to 0	SASTS	This is used to choose the source of the trigger event to start SA-ADC. 00000: TM0INT 00001: TM1INT 00010: TM2INT 00011: TM3INT 001xx: Setting prohibited 01000: FTM0TGO 01001: FTM1TGO 01010: Setting prohibited 011xx: Setting prohibited 10000: NTMS0TGO0 10001: NTMS0TGO1 10010: NTMS1TGO0 10011: NTMS1TGO1 10100: NTMS2TGO0 10101: NTMS2TGO1 Other: Setting prohibited

26.2.8 SA-ADC Enable Register (SADEN)

Offset : 0x18

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SACH17	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SACH											
R/W	-	-	-	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to choose channels of the A/D converter and enable/disable the conversion.
When multiple bits of SACH and SACH17 are set to "1", the A/D conversion starts in the order of smaller channel number.

[Description of each bit]
This is used to enable conversion on the corresponding input channel.
0: Disabled (Initial value)
1: Enabled

Bit No	Bit name	Description (corresponding input channel)
17	SACH17	Input channel 17 (FS/ZS/VBGBUF)
11	SACH[11]	Input channel 11
10	SACH[10]	Input channel 10
9	SACH[9]	Input channel 9
8	SACH[8]	Input channel 8
7	SACH[7]	Input channel 7
6	SACH[6]	Input channel 6
5	SACH[5]	Input channel 5
4	SACH[4]	Input channel 4
3	SACH[3]	Input channel 3
2	SACH[2]	Input channel 2
1	SACH[1]	Input channel 1
0	SACH[0]	Input channel 0

[Note]
- **Do not start the A/D conversion when the all bits of SACH and SACH17 are "0". In that case SARUN bit of SADCON register does not get to "1".**
- **SARUN will not be “1” if SADTM in the SADTMOD register is 0x0 even if SACH17 is “1”.**
- **In simultaneous conversion mode, set the number of input channels to be A/D conversion to the same number for units 0 and 1.**
- **Simultaneous A/D conversions to the same input channel decreases the conversion accuracy. Also Set sampling time considering that the sample hold capacitor is double.**

26.2.9 SA-ADC Upper/Lower Limit Mode Register (SADLMOD)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	SALMD		–	–	–	–	–	–	–	SALEN
R/W	–	–	–	–	–	–	R/W	R/W	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to set modes in the A/D conversion result upper/lower limit detection function.

Bit No	Bit name	Description
9 to 8	SALMD	This is used to set a condition of the A/D conversion result upper/lower limit detection. If the condition is satisfied, corresponding bits of the SA-ADC upper/lower status registers (SADULS) get to "1" and generates the SA-ADC interrupt request. 00: SADLOL value ≤ A/D conversion result ≤ SADUPL value (Initial value) 01: A/D conversion result > SADUPL value 10: A/D conversion result < SADLOL value 11: A/D conversion result > SADUPL value or A/D conversion result < SADLOL value
0	SALEN	This bit is used to enable the A/D conversion result upper/lower limit detection function. If the interrupt occurred by satisfying the upper/lower limit detection condition, check the SA-ADC upper/lower status registers (SADULS) to see which channel of A/D conversion result matched to the condition. SADULS is not updated when this bit is "0". 0: Disabled (Initial value) 1: Enabled

26.2.10 SA-ADC Upper Limit Setting Register (SADUPL)

Offset : 0x20

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–	–	–
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

This is a SFR used to set the upper limit of A/D conversion result.

26.2.11 SA-ADC Lower Limit Setting Register (SADLOL)

Offset : 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to set the lower limit of A/D conversion result.

26.2.12 SA-ADC Upper/Lower Limit Status Register (SADULS)

Offset : 0x28

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SAULS											
R/W	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a read-only SFR used to indicate whether the A/D conversion result matches to the condition of upper/lower limit.

[Description of each bit]
This indicates that a result of the corresponding input channel matches to a condition that is set SALMD.
0: Unmatched (Initial value)
1: Matched

Each bit is forcibly cleared to "0" by writing “1” corresponding bit in the SADULC register.
When using the A/D conversion result upper/lower limit detection function (SALEN=1), the interrupt request is generated at any bit of SADULS is “1” and a timing configured on the SADMOD register.
Refer to Figure 26-10 to Figure 26-12 for the timing of the interrupt and updates of detection result.

Bit No	Bit name	Description (corresponding input channel)
11	SAULS[11]	Input channel 11
10	SAULS[10]	Input channel 10
9	SAULS[9]	Input channel 9
8	SAULS[8]	Input channel 8
7	SAULS[7]	Input channel 7
6	SAULS[6]	Input channel 6
5	SAULS[5]	Input channel 5
4	SAULS[4]	Input channel 4
3	SAULS[3]	Input channel 3
2	SAULS[2]	Input channel 2
1	SAULS[1]	Input channel 1
0	SAULS[0]	Input channel 0

[Note]
- **In the one time conversion mode (SALP bit = “0”), confirm the corresponding bit of SAULS is “0” before setting SARUN bit to “1”.**
- **In the consecutive scan conversion mode (SALP bit = “1”), confirm the corresponding bit of SAULS is “0”, before the next A/D conversion ends.**

26.2.13 SA-ADC Upper/Lower Limit Status Clear Register (SADULC)

Offset : 0x2C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	SAULC											
R/W	-	-	-	-	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a write-only SFR used to clear the A/D conversion result matches to the condition of upper/lower limit.

[Description of each bit]
It is used to clear the upper/lower limit status of the corresponding input channel.
Writing “0”: Invalid (initial value)
Writing “1”: Clear the status

Bit No	Bit name	Description (corresponding input channel)
11	SAULC[11]	Input channel 11
10	SAULC[10]	Input channel 10
9	SAULC[9]	Input channel 9
8	SAULC[8]	Input channel 8
7	SAULC[7]	Input channel 7
6	SAULC[6]	Input channel 6
5	SAULC[5]	Input channel 5
4	SAULC[4]	Input channel 4
3	SAULC[3]	Input channel 3
2	SAULC[2]	Input channel 2
1	SAULC[1]	Input channel 1
0	SAULC[0]	Input channel 0

26.2.14 SA-ADC Result Register (SADR)

Offset : 0x34

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	–	–	–	–
R/W	R	R	R	R	R	R	R	R	R	R	R	R	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a read-only SFR used to store the A/D conversion results (A/D converter test function).
The A/D conversion results of all channels are stored to this register. The result of each channel is overwritten.
The A/D conversion test result on channel 17 is stored to this register only.

26.2.15 SA-ADC Result Register n (SADRn : n = 0 to 11)

Offset : 0x38 (SADR0), 0x3C (SADR1), 0x40 (SADR2), 0x44 (SADR3),
0x48 (SADR4), 0x4C (SADR5), 0x50 (SADR6), 0x54 (SADR7),
0x58 (SADR8), 0x5C (SADR9), 0x60 (SADR10), 0x64 (SADR11)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	–	–	–	–
R/W	R	R	R	R	R	R	R	R	R	R	R	R	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to store the SA-ADC conversion results on channels 0 to 11.
The A/D conversion result of each channel can be read from SADRn.

Register name	Input channel
SADR0	Input channel 0 (AIN0)
SADR1	Input channel 1 (AIN1)
SADR2	Input channel 2 (AIN2)
SADR3	Input channel 3 (AIN3)
SADR4	Input channel 4 (AIN4)
SADR5	Input channel 5 (AIN5)
SADR6	Input channel 6 (AIN6)
SADR7	Input channel 7 (AIN7)
SADR8	Input channel 8 (AIN8)
SADR9	Input channel 9 (AIN9)
SADR10	Input channel 10 (AIN10)
SADR11	Input channel 11 (AIN11)

26.2.16 SA-ADC Test Mode Register (SADTMOD)

Offset : 0x30

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SADTM	
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to control the SA-ADC test function.

This function is used to check if the SA-ADC and the analog switch work correctly,
by performing the A/D conversion for the full scale, zero scale and the internal reference voltage (approx. 1.0 V).
The A/D conversion result is stored in the SA-ADC result register (SADR) only.

Also, the AIN0-11 input level is measured by using a measurement value of internal reference voltage.
For example:
1: Convert at SACH17=1, SADTM1-0=3, where the result is “a”.
2: Convert at SACHn=1, where the result is “b”.
An input level from AINn is b/a [V].

The continuous measurement can be by setting “1” to the SACHn and SACH17. Read the results from SADRn and SADR.

Bit No	Bit name	Description
1 to 0	SADTM	This is used to select input to channel 17. 00: No used test function (Initial value) 01: Full scale level 10: Zero scale level 11: Internal reference voltage (approx. 1.0V)

[Note]

- **The conversion accuracy of the channel 17 input does not guarantee the characteristics of the data sheet.**

26.3 Description of Operation

26.3.1 Operation of Successive Approximation Type A/D Converter

Figure 26-3 shows a setting example for one-time A/D conversion used AIN1 and AIN0.

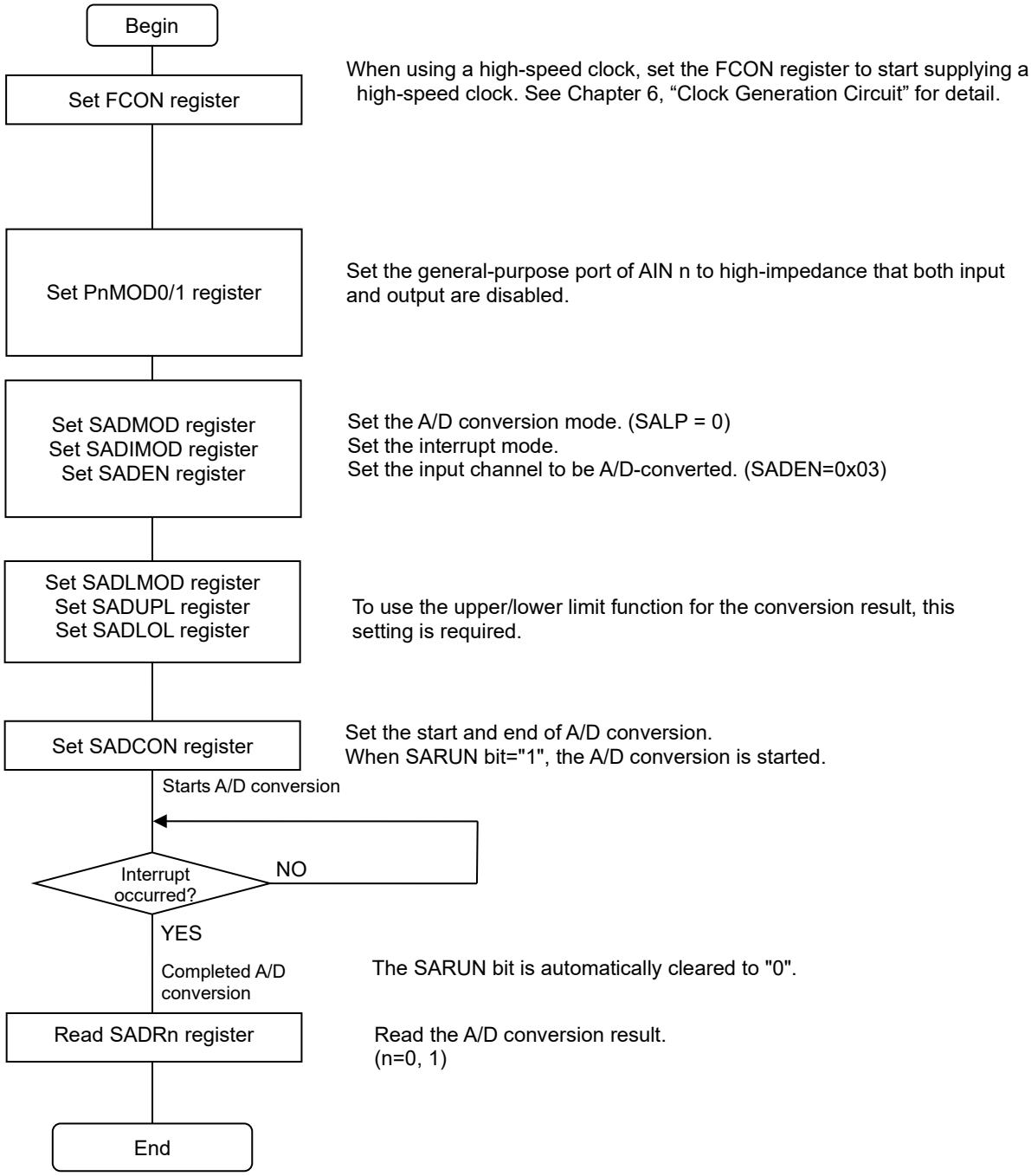


Figure 26-3 Example for A/D Convertor Setting

Figure 26-4 shows a setting example when one-time A/D conversion is performed in HALT mode using AIN1 and AIN0.

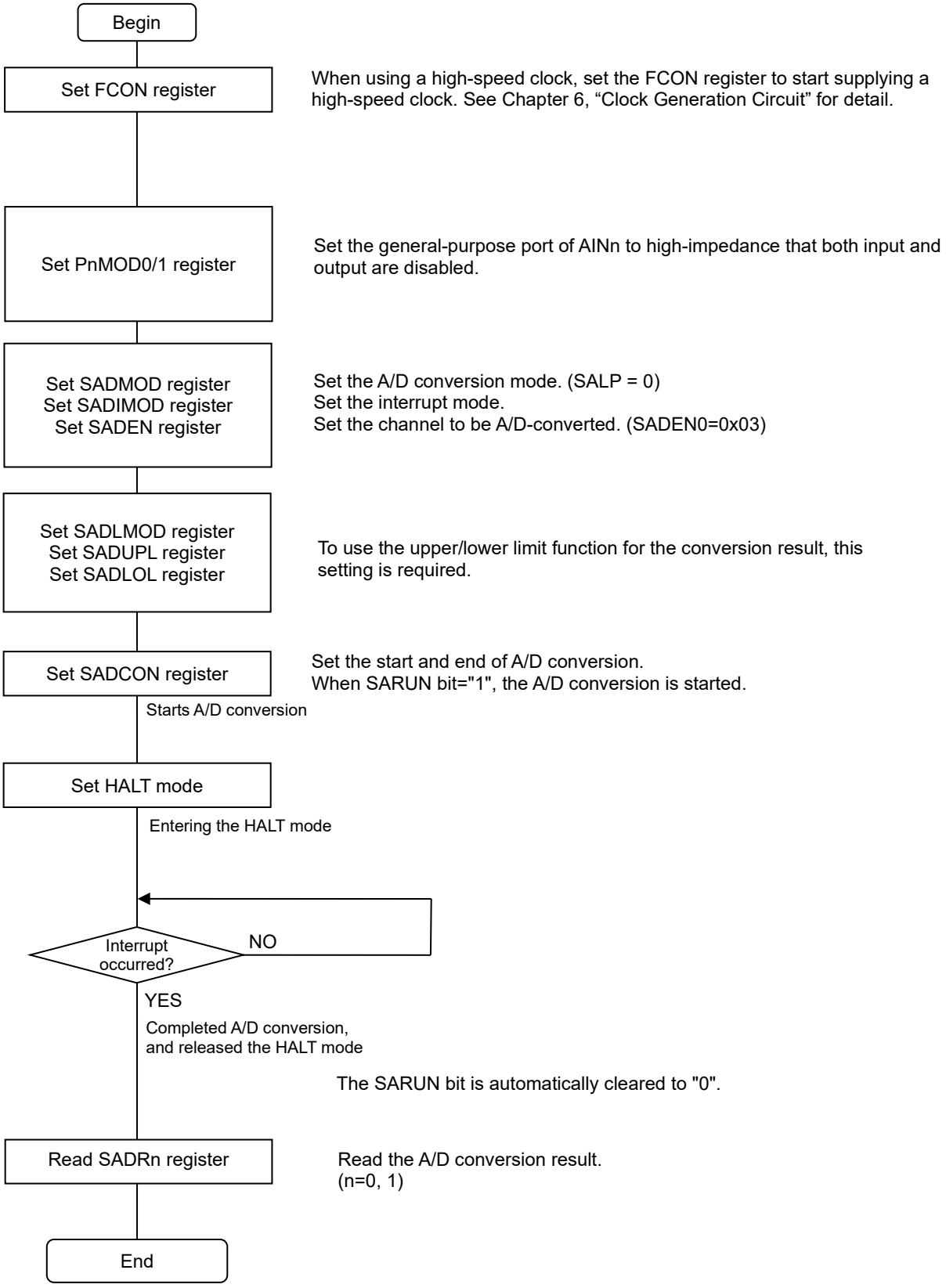


Figure 26-4 Example for A/D Conversion Setting (Converting in HALT mode)

Figure 26-5 shows a setting example when one-time A/D conversion is performed using AIN1 and AIN0 starting by a trigger event.

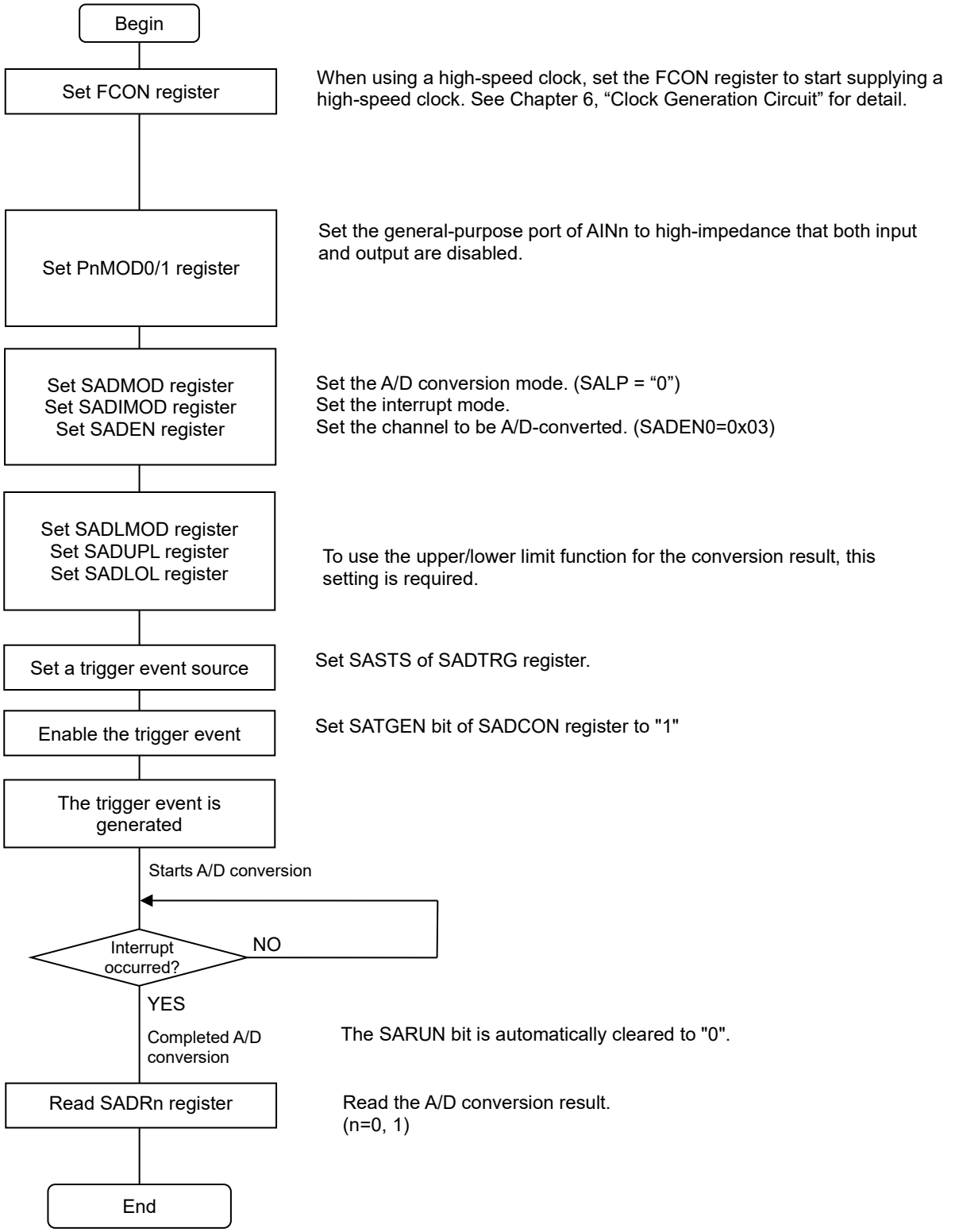
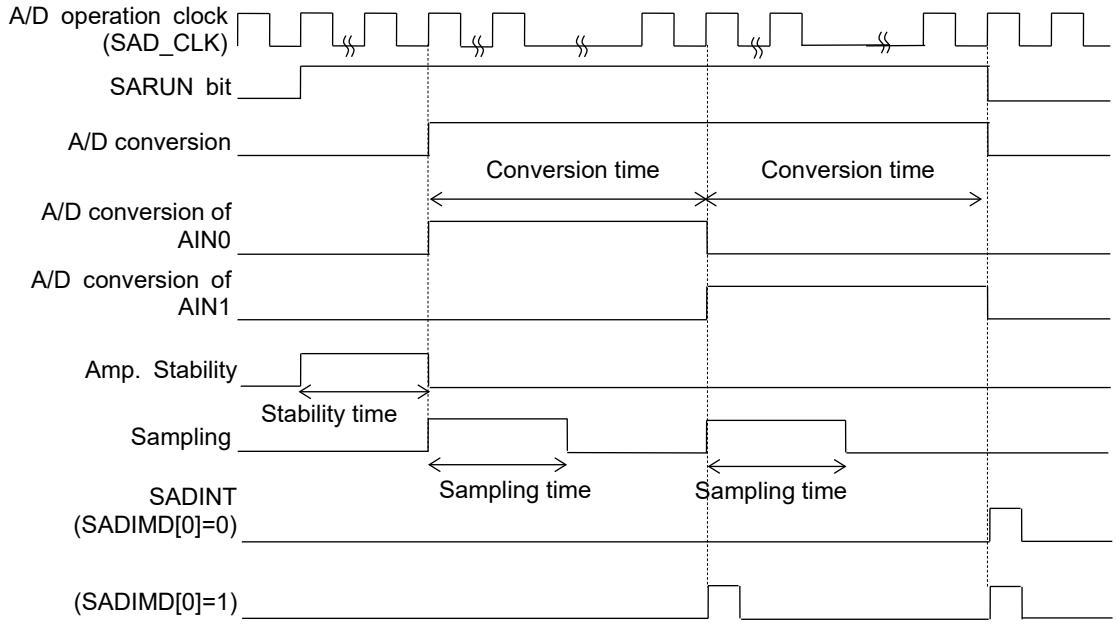


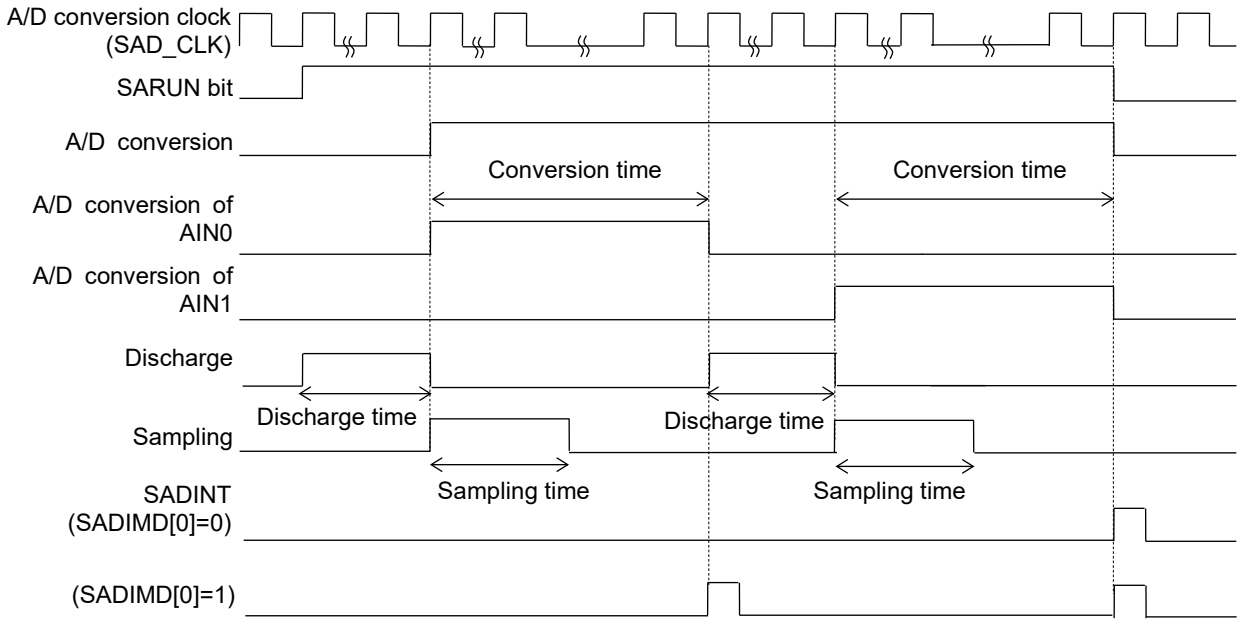
Figure 26-5 Example for A/D Conversion Setting (Start converting by a trigger event)

Figure 26-6 and Figure 26-7 show operation waveforms when one-time A/D conversion is performed using AIN1 and AIN0.



Operating clock (SAD_CLK) is configured by SACK of SADMOD register.
Amp stability time is configured by SAINITT of SADMOD register.
Sampling time is configured by SASHT of SADMOD register.
Interrupt mode is configured by SADIMOD register.

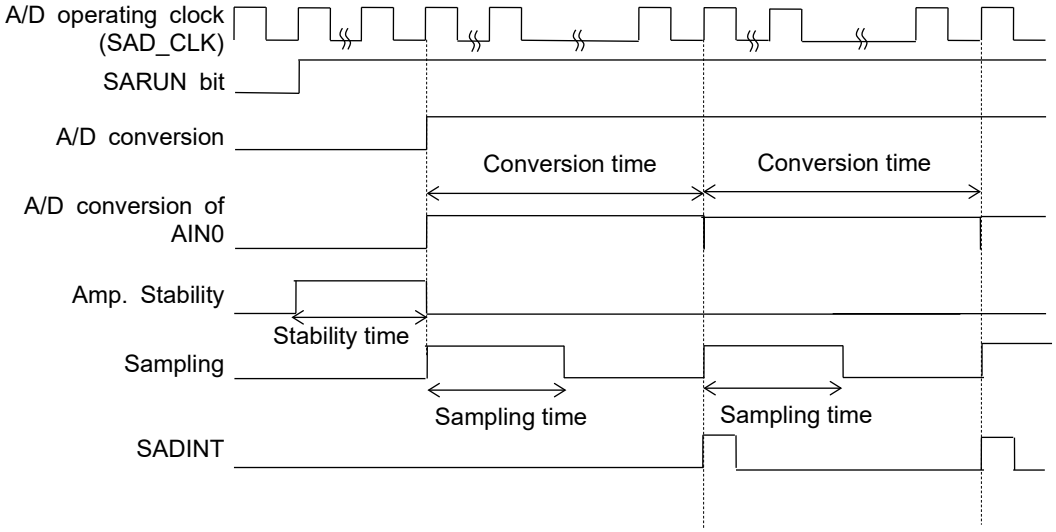
Figure 26-6 Operation Waveforms of A/D Conversion (One-time Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK of SADMOD register.
Discharge enabling and time are configured by SAINIT and SAINITT of SADMOD register.
Sampling time is configured by SASHT of SADMOD register.
Interrupt mode is configured by SADIMOD register.

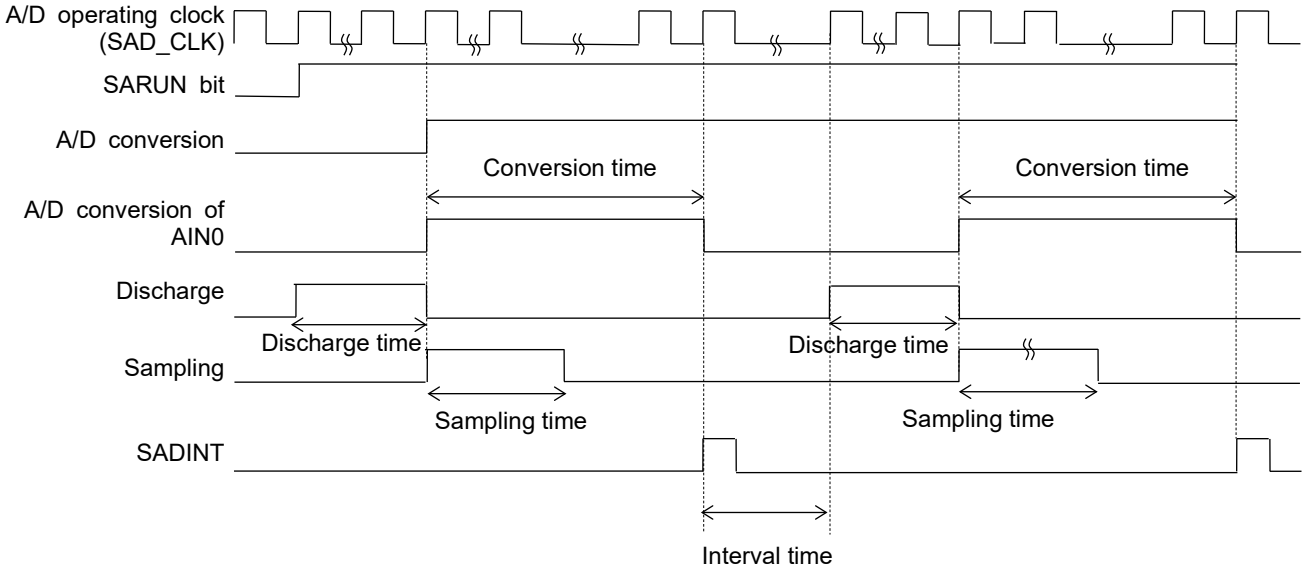
Figure 26-7 Operation Waveforms of A/D Conversion (One-time Conversion, With Discharge)

Figure 26-8 and Figure 26-9 show the operation waveforms when the continuous A/D conversion is performed using AIN0.



Operating clock (SAD_CLK) is configured by SACK of SADMOD register.
Amp stability time is configured by SAINITT of SADMOD register.
Sampling time is configured by SASHT of SADMOD register.
Interval time setting by SADSTM register is prohibit if SADINIT=1.

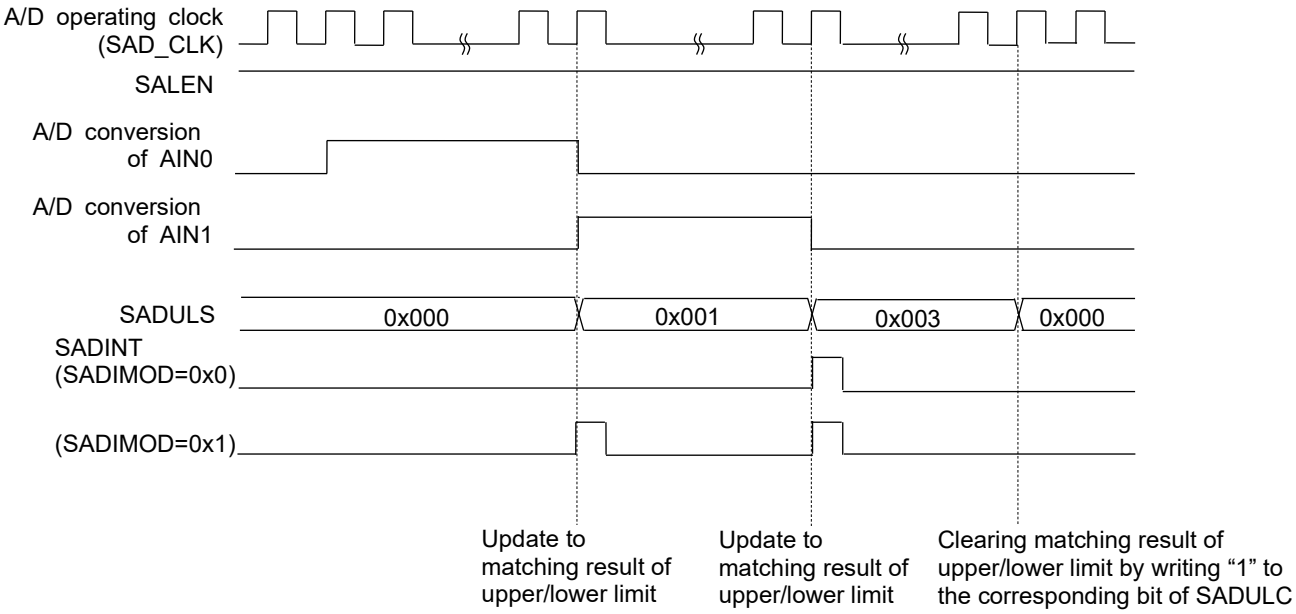
Figure 26-8 A/D Operation Waveforms of A/D Conversion (Continuous Conversion, Without Discharge)



Operating clock (SAD_CLK) is configured by SACK of SADMOD register.
Discharge enabling and time are configured by SAINIT and SAINITT of SADMOD register.
Sampling time is configured by SASHT of SADMOD register.
Interval time is configured by SADSTM register.

Figure 26-9 Operation Waveforms of A/D Conversion (Continuous Conversion, With Discharge)

Figure 26-10 to Figure 26-12 show the operation waveforms when an A/D conversion is performed with upper/lower limit function using AIN1 and AIN0.



A reflection of writing “1” to the corresponding bit of SADULC register to clear matching result of upper/lower limit, is delayed maximum 1clock of the SAD_CLK.

Figure 26-10 Operation Waveforms of A/D Conversion with Upper/Lower Limit (SADIMD[1] = 0 , matched limit ranges)

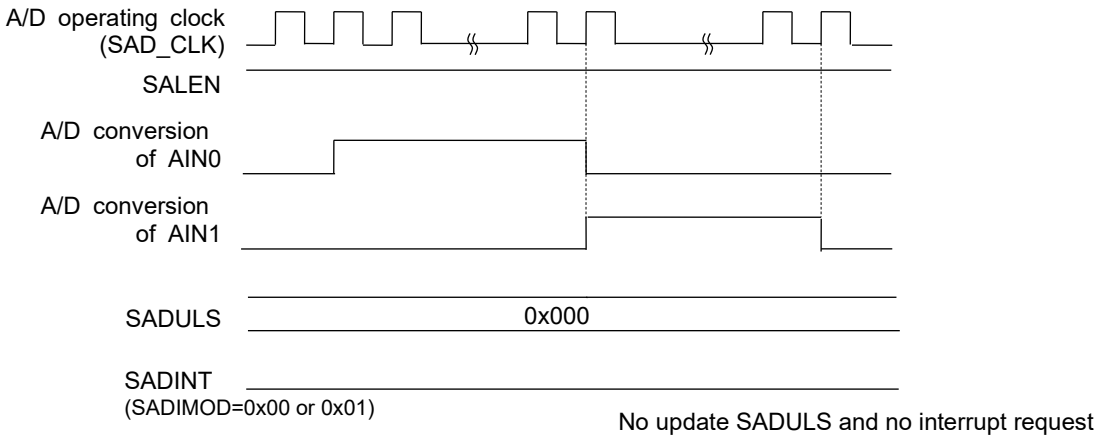
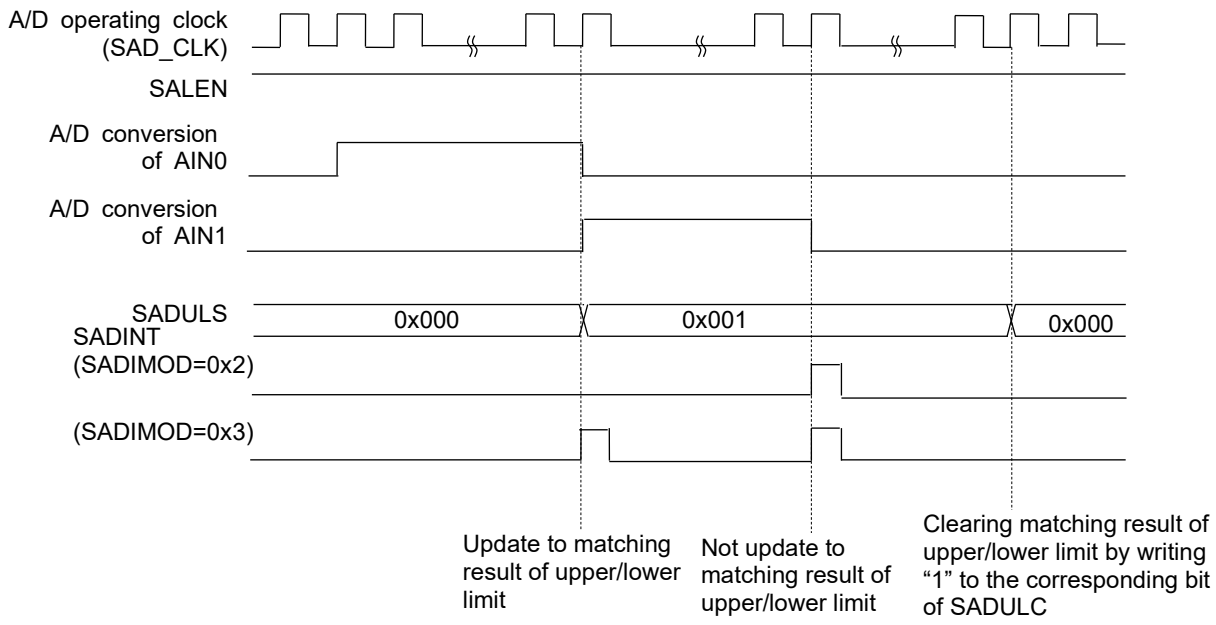


Figure 26-11 Operation Waveforms of A/D Conversion with Upper/Lower Limit (SADIMD[1] = 0 , not matched limit ranges)



The interrupt occurs regardless of matching result of upper/lower limit.

Figure 26-12 Operation Waveforms of A/D Conversion with Upper/Lower Limit (SADIMD1 = 1)

26.3.2 How to test the Successive Approximation Type A/D Converter

The self test can be performed by A/D-converting the full scale, zero scale and internal reference voltage.
Follow this procedure to check if the successive approximation type A/D converter works correctly. (n=0 to 11)

(1) A/D convert AINn pin. (conversion result 1)
(2) A/D convert AIN17=full scale by setting the SADTMOD register (SADTM=0x1).
(3) A/D convert the AINn pin. (conversion result 2)
(4) A/D convert AIN17=zero scale by setting the SADTMOD register (SADTM=0x2).
(5) A/D convert the AINn pin. (conversion result 3)
(6) A/D convert AIN17=internal reference voltage(approx.1.0V) by setting the SADTMOD register (SADTM=0x3).
(7) A/D convert the AINn pin. (conversion result 4)
(8) Confirm conversion result 1 ≈ conversion result 2 ≈ conversion result 3 ≈ conversion result 4.
Use the same AINn pin for the A/D conversion in (1), (3), (4) and (7).
(9) Confirm the conversion result in (2), (4) and (6) is different each other and also different from the result in (1), (3), (5) and (7).

26.3.3 Simultaneous conversion of 2 channels

If SYNC bit in the SADMOD register of unit 1 is set to "1", simultaneous conversion between units 0 and 1 can be performed.
Table 26-4 shows registers to use.
When using simultaneous conversion, set the number of input channels to be A/D conversion to the same number for unit 0 and unit 1.

Table 26-4 Available registers in simultaneous conversion
(A: Setting is available. N: Setting is unavailable.)

Name	Symbol	Unit 0	Unit 1
SA-ADC mode register (except SYNC bit)	SADMOD	A	N
(SYNC bit)		N	A
SA-ADC control register	SADCON	A	N
SA-ADC conversion interval register	SADSTM	A	N
SA-ADC interrupt mode register	SADIMOD	A	N
SA-ADC trigger register	SADTRG	A	N
SA-ADC enable register	SADEN	A	A
SA-ADC upper/lower limit mode register	SADLMOD	A	A
SA-ADC upper limit setting register	SADUPL	A	A
SA-ADC lower limit setting register	SADLOL	A	A
SA-ADC upper/lower status register	SADULS	A	A
SA-ADC upper/lower status clear register	SADULC	A	A
SA-ADC test mode register	SADTMOD	A	A
SA-ADC result register	SADR	A	A
SA-ADC result register 0	SADR0	A	A
SA-ADC result register 1	SADR1	A	A
SA-ADC result register 2	SADR2	A	A
SA-ADC result register 3	SADR3	A	A
SA-ADC result register 4	SADR4	A	A
SA-ADC result register 5	SADR5	A	A
SA-ADC result register 6	SADR6	A	A
SA-ADC result register 7	SADR7	A	A
SA-ADC result register 8	SADR8	A	A
SA-ADC result register 9	SADR9	A	A
SA-ADC result register 10	SADR10	A	A
SA-ADC result register 11	SADR11	A	A

26.4 Notes on SA-ADC

26.4.1 Sampling Time Setting

Sampling time of the SA-ADC should satisfy the following formula:

$$\text{Sampling time} > 9(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

To calculate sampling time more precisely, use the following formula:

$$\text{Sampling time} = \left\{log_e(2^n) + log_e\left(\frac{C_{SAMPLE}}{C_{SAMPLE} + C_{PARA}}\right)\right\}(C_{SAMPLE} + C_{PARA})(R_1 + R_2)$$

C_{PARA} varies depending on board-layout and connected parts. Please check the accuracy of SA-ADC with the actual board.

R_1	: Input impedance of external resistor
R_2	: Internal resistor value which is the sum of the internal resistor and the ON register of the switch
C_{SAMPLE}	: Sample hold capacitor
C_{PARA}	: Parasitic capacitance of the A/D input line. (Measure the capacitance between the A/D input line and V_{SS}.)
n	: Resolution of SA-ADC

The following diagram shows the equivalent circuit in this case:

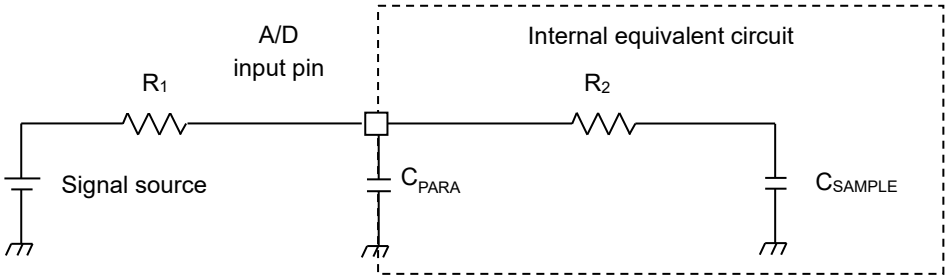


Figure 26-13 Internal equivalent circuit

Table 26-5 Const (reference value)

V_{REF}	R_2[kΩ]	C_{SAMPLE}[pF]
2.1V ≤ V_{REF} < 2.4V	8	5
2.4V ≤ V_{REF} < 2.7V	6	5
2.7V ≤ V_{REF} < 4.5V	5	5
4.5V ≤ V_{REF} ≤ 5.5V	3	5

If a range of $V_{REF,}$ may cross voltage conditions in Table 26-5, set the sampling time to satisfy the lower voltage side.
If the sampling time above is unsatisfied, connect the external capacitor C_1 near by A/D input pin to satisfy the following formula.

$$(C_1 + C_{PARA}) > 2^n C_{SAMPLE}$$

$$\text{Sampling time} > 9C_{SAMPLE}R_2$$

C_1 : External capacitor

The equivalent circuit when the external capacitor C1 is connected is as follows:

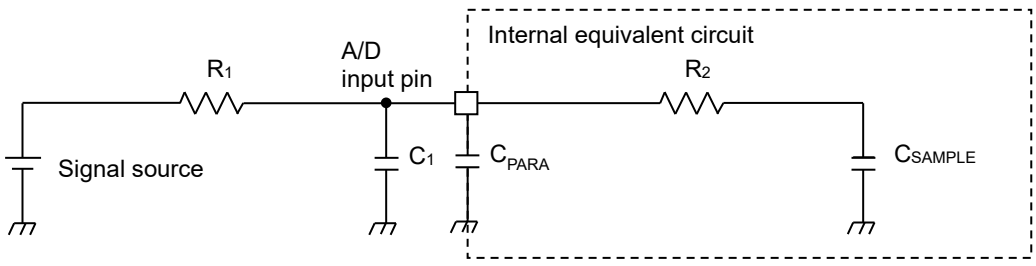


Figure 26-14 Internal equivalent circuit (connected C_1)

Note that the voltage at the A/D input pin transitionally changes due to the external capacitor C_1 and the external resistor R1. Therefore, when sampling data, wait until the voltage is stabilized. If the stabilization timing is unknown, perform A/D conversion once, then wait for time constant τ ($=R_1C_1$) to 4τ or so and perform A/D conversion again. Confirm that the difference between values is small, and then sample data.

26.4.2 Noise Suppression

In order to prevent deterioration in accuracy of A/D conversion, operate the A/D converter in the environment with little noise.
The following processes are recommended for noise reduction:

- Perform A/D conversion in the HALT mode.
- Do not have clock input/output to and from a pin located in the vicinity of the pin in which A/D conversion is in progress.
- Do not have clock input/output to and from the pin in which A/D conversion is in progress and other A/D conversion pins.

In addition, the capacitor for noise suppression should be connected between VREF and VSS, as well as between VDD and VSS. When connecting, place the capacitor in the immediate vicinity of LSI using short wiring.

Chapter 27

(Skipped)

Chapter 28

Analog Comparator

28. Analog Comparator

28.1 Overview

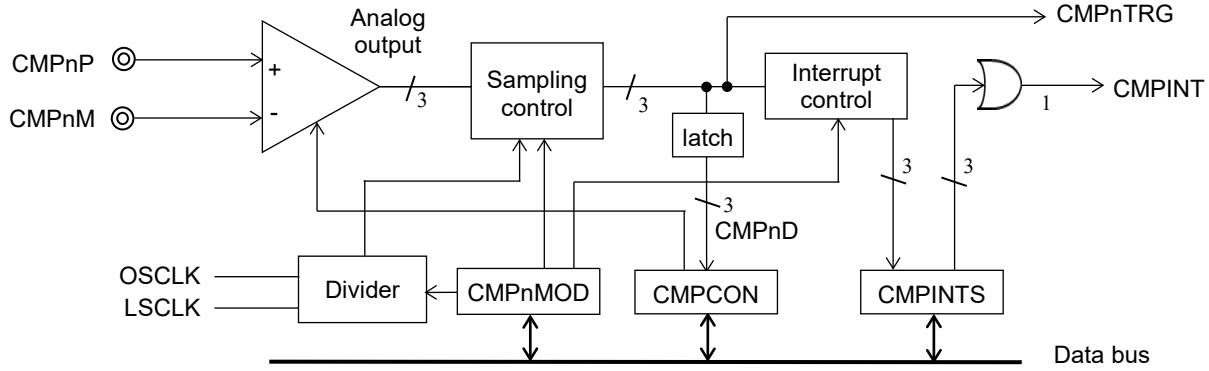
The analog comparator compares voltages input to the two pins.

28.1.1 Features

- Built-in 3 units
- It compares voltages input to the two pins
- The compare interrupt request timing is selectable from the following three kinds:
  – rising edge, falling edge and both edges of the compared result output
- Selectable whether sampling the compared result output.
- The HSCLK or OSCLK divided by 1 to 128 can be selected as sampling clock.
- Analog comparator output can be used as an emergency stop trigger source for functional timers and NTMS timers.

28.1.2 Configuration

Figure 28-1 shows the configuration of the analog comparator.



CMPCON : Comparator control register
CMPnMOD : Comparator n mode register
CMPnD : Comparator n result
CMPINT : Comparator interrupt
CMPnTRG : Comparator n trigger output
(n = 0 to 2)

Figure 28-1 Configuration of Analog Comparator

28.1.3 List of Pins

The input/output pins of the analog comparator are assigned to the primary function of the general-purpose port.

Table 28-1 Pins

Functional pin name	I/O	Description	LSI pin name
CMP0P	I	Analog comparator 0 non-inverting input	P46
CMP0M	I	Analog comparator 0 inverting input	P47
CMP1P	I	Analog comparator 1 non-inverting input	P80
CMP1M	I	Analog comparator 1 inverting input	P81
CMP2P	I	Analog comparator 2 non-inverting input	P82
CMP2M	I	Analog comparator 2 inverting input	P83

28.2 Description of Registers

28.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4200_2000	Analog comparator base address	CMP	-	-	-
0x00	Comparator control register	CMPCON	R/W	32	0x0000_0000
0x04	Comparator 0 mode register	CMP0MOD	R/W	32	0x0000_0000
0x08	Comparator 1 mode register	CMP1MOD	R/W	32	0x0000_0000
0x0C	Comparator 2 mode register	CMP2MOD	R/W	32	0x0000_0000
0x24	Comparator interrupt status register	CMPINTS	R/W	32	0x0000_0000

28.2.2 Comparator Control Register (CMPCON)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	CMP2D	CMP1D	CMP0D	–	–	–	–	–	CMP2EN	CMP1EN	CMP0EN
R/W	–	–	–	–	–	R/W	R/W	R/W	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMPCON is a special function register (SFR) used to control the analog comparators.

Bit No	Bit name	Description
10	CMP2D	This bit indicates comparison result of the comparator 2. The last comparison result is retained when the comparator is stopped. 0: CMP2P < CMP2M 1: CMP2P > CMP2M
9	CMP1D	This bit indicates comparison result of the comparator 1. The last comparison result is retained when the comparator is stopped. 0: CMP1P < CMP1M 1: CMP1P > CMP1M
8	CMP0D	This bit indicates comparison result of the comparator 0. The last comparison result is retained when the comparator is stopped. 0: CMP0P < CMP0M 1: CMP0P > CMP0M
2	CMP2EN	This bit is used to enable the operation of the comparator 2. 0: Disabled (Initial value) 1: Enabled
1	CMP1EN	This bit is used to enable the operation of the comparator 1. 0: Disabled (Initial value) 1: Enabled
0	CMP0EN	This bit is used to enable the operation of the comparator 0. 0: Disabled (Initial value) 1: Enabled

[Note]

- **When using the analog comparator, set the target pin to high-impedance (input and output disabled) by the port n mode register 0/1 (n=4,8), otherwise a shoot-through current may flow.**
- **An influence of the noise is reducible by preventing the switching of neighboring pins when the comparator enables.**
- **After CMPnEN bit is set to "1", it takes up to 100µs for the operation of the analog comparator to stabilize. Wait until this stabilization time has elapsed to use the output of an analog comparator.**

28.2.3 Comparator n Mode Register (CMPnMOD : n=0 to 2)

Offset : 0x04 (CMP0MOD), 0x08 (CMP1MOD), 0x0C (CMP2MOD)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMPMD	–	–	–	–	–	–	–	–	CMPDIV			CMPCS		CMPE	
R/W	R/W	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMPnMOD is a special function register (SFR) used to set the operation mode of the analog comparator n.

Bit No	Bit name	Description
15	CMPMD	Set this bit to "1" before the corresponding comparator operation is enabled.
6 to 4	CMPDIV	This is used to choose dividing ratio for the sampling clock source selected CMPCS. 000: No dividing (Initial value) 001: Divided by 2 010: Divided by 4 011: Divided by 8 100: Divided by 16 101: Divided by 32 110: Divided by 64 111: Divided by 128
3 to 2	CMPCS	This is used to choose the sampling clock source or no sampling. 00: No sampling (Initial value) 01: Sampling with OSCLK 10: Sampling with LSCLK 11: Setting prohibited (as same as setting value 2'b10)
1 to 0	CMPE	This is used to choose the timing of interrupt request. 00: Disabled interrupt. (Initial value) 01: Falling-edge 10: Rising-edge 11: Both-edge

[Note]

- **Write CMPnMOD register when the comparator stops (CMPnEN bit of CMPCON register is "0"), otherwise the comparison result is unguaranteed.**
- **In the STOP mode, the comparator works without sampling regardless the setting in CMPCS.**
- **When the OSCLK is chosen for the sampling clock and the high-speed clock is not oscillating, the sampling circuit does not work. When using analog comparator in this case, choose "No sampling" or "Sampling with LSCLK" for sampling condition. For how to enable the high-speed clock oscillation, see Chapter 6 "Clock Generation Circuit".**

28.2.4 Comparator Interrupt Status Register (CMPINTS)

Offset : 0x24

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	CMP2IS	CMP1IS	CMP0IS
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMPINTS is a special function register (SFR) used to indicate interrupt status of the analog comparators.

Bit No	Bit name	Description
2	CMP2IS	This indicates interrupt request of the comparator 2. The request is cleared to “0” when this bit is written to “1”. 0: No interrupt request (Initial value) 1: Has interrupt request
1	CMP1IS	This indicates interrupt request of the comparator 1. The request is cleared to “0” when this bit is written to “1”. 0: No interrupt request (Initial value) 1: Has interrupt request
0	CMP0IS	This indicates interrupt request of the comparator 0. The request is cleared to “0” when this bit is written to “1”. 0: No interrupt request (Initial value) 1: Has interrupt request

28.3 Description of Operation

28.3.1 Analog Comparator Operation

Figure 28-2 shows an analog comparator operation overview.

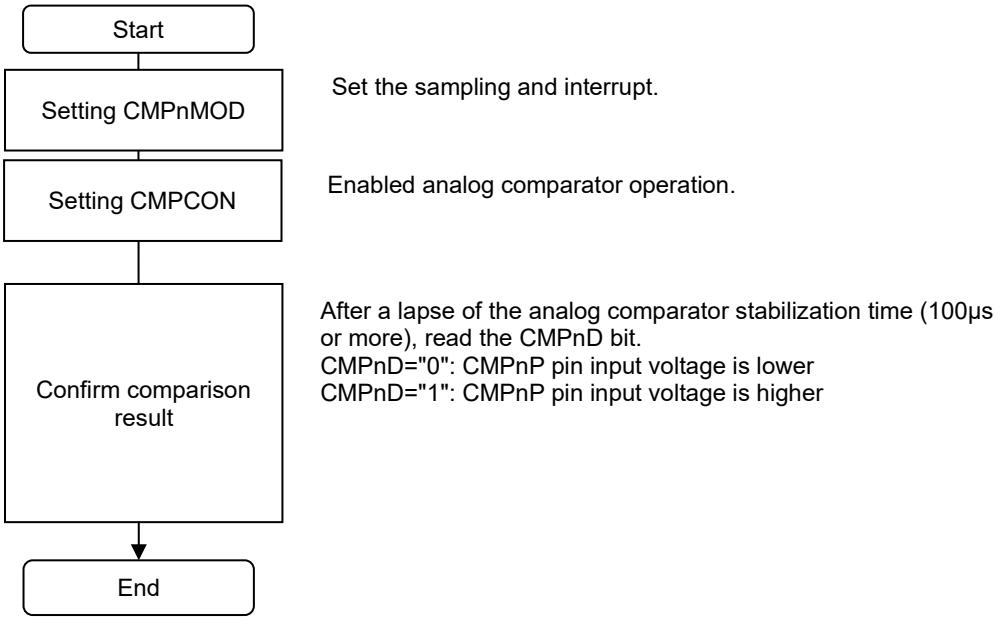


Figure 28-2 Analog comparator operation overview

Figure 28-3 shows an example of the analog comparator operation timing.

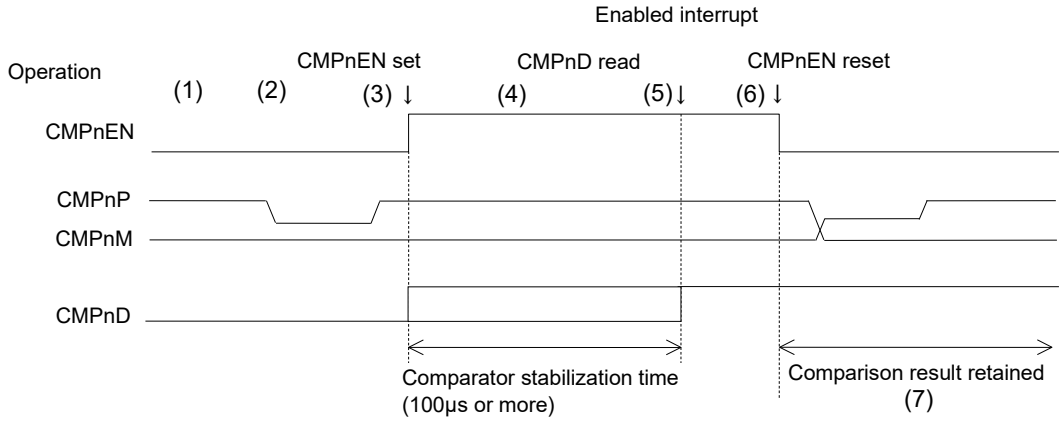


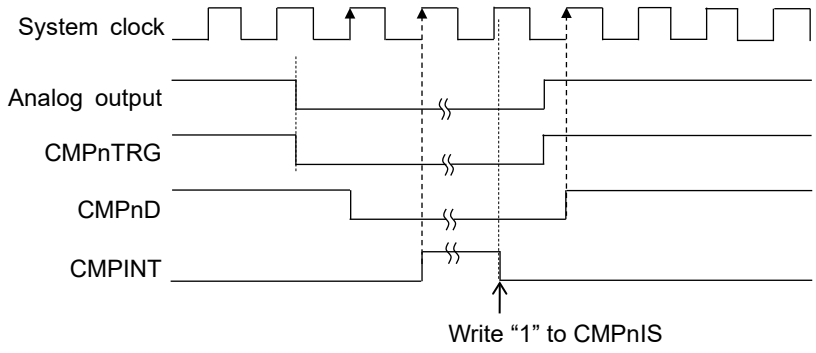
Figure 28-3 Example of Analog Comparator Operation Timing

Operation shown in Figure 28-3 above is described below.

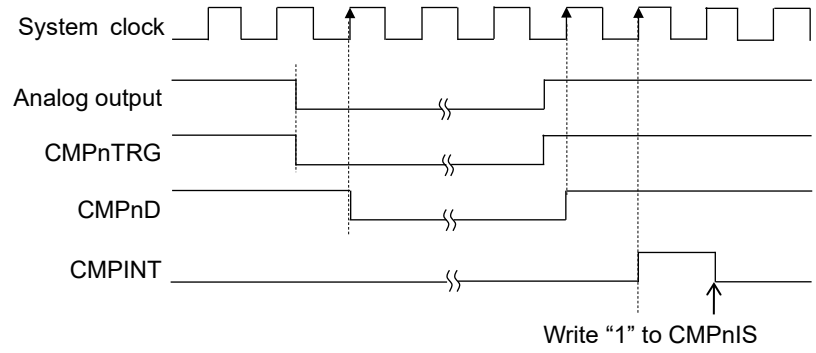
1) To operate the analog comparator, first set the following configuration:
   - Set the general-purpose port used for the analog comparator to high-impedance that both input and output is disabled.
   - If using the OSCLK for the sampling clock, enable the high-speed clock. See Chapter 6, “Clock generation circuit” for detail.
2) Choose the interrupt mode and sampling conditions using the CMPnMOD register.
3) Write "1" to the CMPnEN bit to enable the analog comparator operation.
4) Wait for the stabilization time (100 µs or more) of the analog comparator.
5) Read the comparison result from the CMPnD bit. Enable the interrupt, if using interrupt.
6) Write "0" to the CMPnEN bit to disable the analog comparator operation. The CMPnEN is not needed to disable, if the operation will be continued.
7) The CMPnD bit may be read after "0" is written to the CMPnEN bit because the CMPnD bit holds the comparison result at the time when "0" is written to the CMPnEN bit.

28.3.2 Interrupt Request

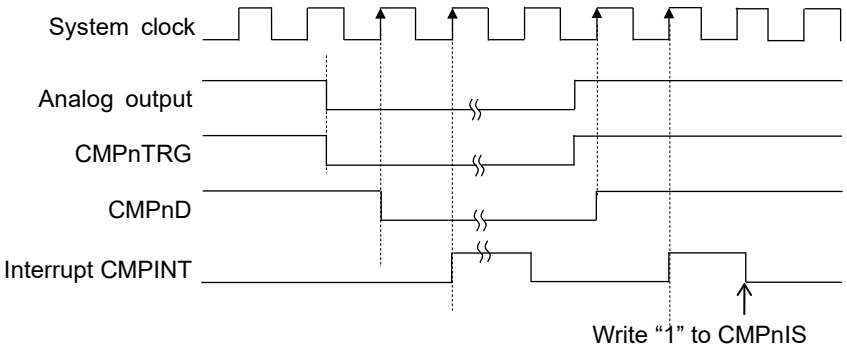
If the interrupt edge chosen in the CMPnE of the CMPnMOD register is detected, the analog comparator interrupt (CMPINT) is generated.
The CMPINT interrupt is the OR logic of the three sources of CMP0 to 2. The CMPINT will contenue to request until all CMPnISs have been cleared.
Figure 28-4 shows the interrupt generation timings without sampling (when the falling-edge/rising-edge/both-edge interrupt mode is chosen).
Figure 28-5 shows the interrupt generation timing with sampling (when the rising-edge interrupt mode is chosen).
Figure 28-6 shows the interrupt generation timing in the STOP mode.
Figure 28-7shows the interrupt generation timing by multiple comparators (when the rising-edge interrupt mode is chosen).



(a) In falling-edge interrupt mode



(b) In rising-edge interrupt mode



(c) In both-edge interrupt mode

Figure 28-4 Analog Comparator Interrupt Generation Timing (without sampling)

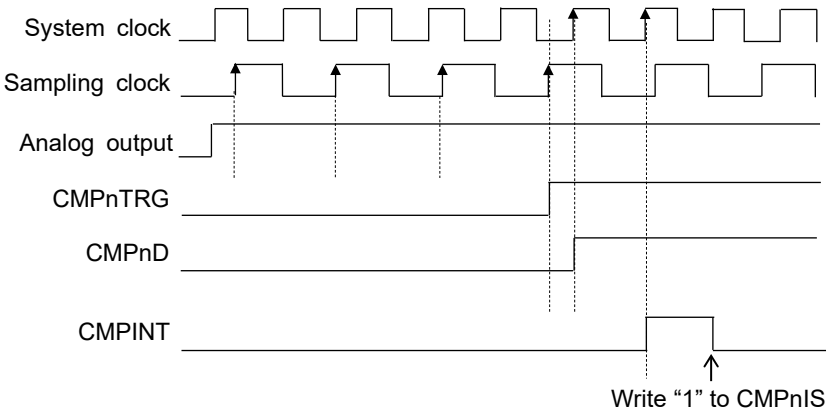


Figure 28-5 Analog Comparator Interrupt Generation Timing (with sampling in the rising-edge interrupt mode)

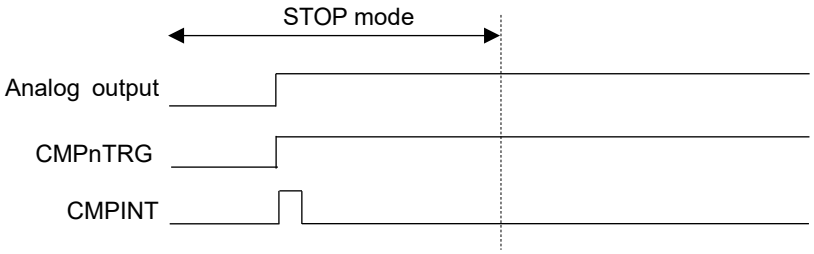


Figure 28-6 Analog Comparator Interrupt Generation Timing in the STOP mode (in the rising-edge interrupt mode)

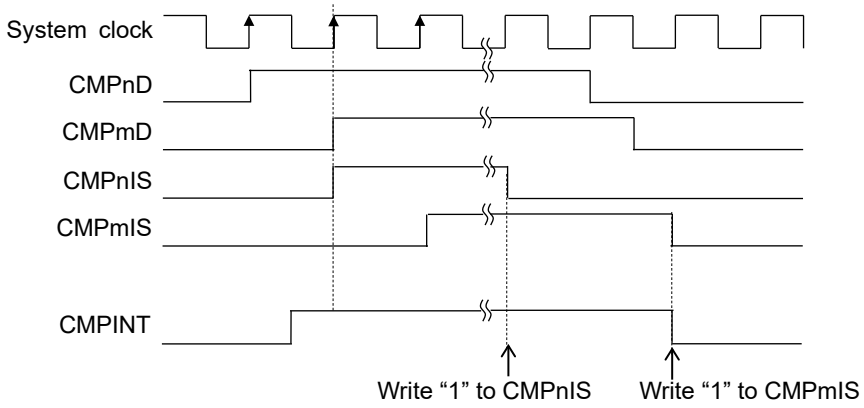


Figure 28-7 Analog Comparator Interrupt Generation Timing by multiple comparators (in the rising-edge interrupt mode)

Chapter 29

Flash Memory

29. Flash Memory

29.1 Overview

This LSI has the flash memory in the program memory space and data flash area. For details of the program memory space and data flash area, see Chapter 3 " Memory Space".

Table 29-1 The ways of programming the flash memory

Programming method	Tool/Register/Communication	Reference Chapter
Programming by the on-chip debug function	On-chip debug emulator	Chapter 32 "On chip Debug function"
Self-Programming by using the special function register(SFR)	Special Function Registers (SFRs) for programming the flash memory	Section 29.3 "Self-programming"

Table 29-2 and Table 29-3 show overview of the program memory space and data flash area.

Table 29-2 Flash memory overview (Size and Address)

Part name	Program memory space		Data flash area	
	Size	Address	Size	Address
ML63Q2537 ML63Q2557	256K Byte	0x1000_0000 to 0x1003_FFFF	8K Byte (256 Byte x 32 sector)	0x1800_0000 to 0x1800_1FFF
ML63Q2534 ML63Q2554	128K Byte	0x1000_0000 to 0x1001_FFFF	8K Byte (256 Byte x 32 sector)	0x1800_0000 to 0x1800_1FFF

Table 29-3 Flash memory overview (Functions and Characteristics)

Item		Program memory space	Data flash area
Reading unit		32-bit / 16-bit / 8-bit	16-bit / 8-bit
Erasing and programming unit	Block erasing	32K Byte	All area
	Sector erasing	2K Byte	256 Byte
	Programming	4 Byte (32 bit)	1 Byte (8 bit)
Erasing and programming time	Block erasing	Max. 50ms	Max. 50ms
	Sector erasing		
	Programming	Max. 80μs	Max. 40μs
Programming cycle		100 times	10,000 times
Erasing and programming temperature		0°C to +40°C	-40°C to +85°C
Background operation (BGO) function		-	Yes

29.2 Description of Registers

29.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0400	Flash ROM control base address	FLASHCNT	-	-	-
0x00	Flash address register	FLASHA	R/W	32	0x1000_0000
0x04	Flash data register	FLASHD	R/W	32	0xFFFF_FFFF
0x08	Flash control register	FLASHCON	W	32	0x0000_0000
0x0C	Flash acceptor	FLASHACP	W	32	0x0000_0000
0x10	Reserved	-	R	32	0x0000_0000
0x14	Flash self register	FLASHSLF	R/W	32	0x0000_0000
0x18	Flash status register	FLASHSTA	R	32	0x0000_0000

29.2.2 Flash Address Register (FLASHA)

Offset : 0x00

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FA[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FA[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLASHA is special function register (SFR) to set the erasing or programming address.
Table 29-4 shows address setting for block erasing, and Table 29-5 shows address setting for sector erasing.

29.2.3 Flash Data Register (FLASHD)

Offset : 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ED[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ED[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLASHD is special function register (SFR) to set the write data.
When data is written to this register, it starts programming to flash memory.

There are some differences for programming the program memory space and the data flash area.

Programming target	Register	Description
Program memory space	Four bytes specified in FLASHD	-
Data flash area	One byte in FLASHD register. (FD[7:0])	FD[31:8] bits data are invalid.

[Note]
- **Note that programming for the program memory space is performed by the unit of 4 bytes. Because of this, the setting values in the FA[1:0] bits are ignored.**
- **During programming data-flash, a CPU can execute instruction by the background operation function; BGO. Confirm FDPRSTA bit of FLASHSTA register for completion of programming.**
- **Erase data in the addresses in advance. Programmed data without erase is unguaranteed.**
- **Do not read or program unused areas to prevent the CPU works incorrectly.**

Table 29-4 Address setting for sector erase

Area	Block	Address	Size	FLASHA register
Program memory	block 0	0x1000_0000 to 0x1000_7FFF	32KByte	0x1000_0000
	block 1	0x1000_8000 to 0x1000_FFFF	32KByte	0x1000_8000
	block 2	0x1001_0000 to 0x1001_7FFF	32KByte	0x1001_0000
	block 3	0x1001_8000 to 0x1001_FFFF	32KByte	0x1001_8000
	block 4 *	0x1002_0000 to 0x1002_7FFF	32KByte	0x1002_0000
	block 5 *	0x1002_8000 to 0x1002_FFFF	32KByte	0x1002_8000
	block 6 *	0x1003_0000 to 0x1003_7FFF	32KByte	0x1003_0000
	block 7 *	0x1003_8000 to 0x1003_FFFF	32KByte	0x1003_8000
Data Flash	block 0	0x1800_0000 to 0x1800_1FFF	8KByte	0x1800_0000

*: ML63Q2537/ML63Q2557 only

Table 29-5 Address setting for sector erase

Area	Sector	Address	Size	FLASHA register
Program memory	sector 0	0x1000_0000 to 0x1000_07FF	2KByte	0x1000_0000
	sector 1	0x1000_0800 to 0x1000_0FFF	2KByte	0x1000_0800
	:	:	:	:
	sector 30	0x1000_F000 to 0x1000_F7FF	2KByte	0x1000_F000
	sector 31	0x1000_F800 to 0x1000_FFFF	2KByte	0x1000_F800
	sector 32	0x1001_0000 to 0x1001_07FF	2KByte	0x1001_0000
	sector 33	0x1001_0800 to 0x1001_0FFF	2KByte	0x1001_0800
	:	:	:	:
	sector 62	0x1001_F000 to 0x1001_F7FF	2KByte	0x1001_F000
	sector 63	0x1001_F800 to 0x1001_FFFF	2KByte	0x1001_F800
	sector 64 *	0x1002_0000 to 0x1002_07FF	2KByte	0x1002_0000
	sector 65 *	0x1002_0800 to 0x1002_0FFF	2KByte	0x1002_0800
	:	:	:	:
	sector 94 *	0x1002_F000 to 0x1002_F7FF	2KByte	0x1002_F000
	sector 95 *	0x1002_F800 to 0x1002_FFFF	2KByte	0x1002_F800
	sector 96 *	0x1003_0000 to 0x1003_07FF	2KByte	0x1003_0000
	sector 97 *	0x1003_0800 to 0x1003_0FFF	2KByte	0x1003_0800
	:	:	:	:
	sector 126 *	0x1003_F000 to 0x1003_F7FF	2KByte	0x1003_F000
	sector 127 *	0x1003_F800 to 0x1003_FFFF	2KByte	0x1003_F800
Data Flash	sector 0	0x1800_0000 to 0x1800_00FF	256Byte	0x1800_0000
	sector 1	0x1800_0100 to 0x1800_01FF	256Byte	0x1800_0100
	:	:	:	:
	sector 30	0x1800_1E00 to 0x1800_1EFF	256Byte	0x1800_1E00
	sector 31	0x1800_1F00 to 0x1800_1FFF	256Byte	0x1800_1F00

*: ML63Q2537/ML63Q2557 only

29.2.4 Flash Control Register (FLASHCON)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	FSERS	FERS
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a write-only SFR used to control the block erasing and sector erasing for the flash memory. This register always returns 0x00 for reading.

Bit No	Bit name	Description
1, 0	FSERS, FSRS	There are used to start the sector erasing or block erasing. 00: Setting prohibited (No function) (Initial value) 01: Start block erasing 10: Start sector erasing 11: Setting prohibited (No function)

[Note]

- **In processing to program or erase of Data flash memory, reading to Data flash memory is prohibited.**
- **In processing to program or erase of the flash memory, access to flash memory by DMAC is prohibited.**

29.2.5 Flash Acceptor (FLASHACP)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	FAC							
R/W	–	–	–	–	–	–	–	–	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a write-only SFR used to accept for erasing/programming the flash memory in order to prevent an unintended. erasing/programming operation.

Bit No	Bit name	Description
7 to 0	FAC	When 0xFA and 0xF5 are written to the FLASHACP in this order, the erasing or programming function is enabled only once. For subsequent erasing or programming, 0xFA and 0xF5 must be written to FLASHACP each time. Even if other instructions are executed between the instruction that writes 0xFA and 0xF5 to the FLASHACP, the erasing or programming function is still valid. If data other than 0xF5 is written to the FLASHACP after 0xFA is written, 0xFA becomes invalid. In this case, it needs to write 0xFA again.

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

29.2.6 Flash Self Register (FLASHSLF)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	FSELF
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a SFR used to enable erasing and programming the flash memory. When system clock is the low-speed clock, it is not writable.

Bit No	Bit name	Description
0	FSELF	This bit is used to enable erasing and programming the flash memory. This bit is kept after completed erasing/programming. 0: Disabled (Initial value) 1: Enabled

[Note]

- **A flash memory data in processing to program is not guaranteed, if this register is written any data when FLASHSTA is not 0x0.**

29.2.7 Flash Status Register (FLASHSTA)

Offset : 0x18

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	FCPRSTA	FCERSTA	FDPRSTA	FDERSTA
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This is a read-only SFR used to check status of the flash memory.
FLASHSTA is used to erase/program data flash by code on flash memory, or erase/program flash memory by code on work RAM.

Table 29-6 Usage condition of FLASHSTA register

Executed code Placement	Target area	Availability to read this register while erasing/programming	Description
Flash memory	Program memory space	Unavailable	Do not use the FLASHSTA. This is because the CPU stops while the program memory space is being erased/programmed, so the FLASHSTA register cannot be read.
	Data flash area	Available	As the Background Operation (BGO) function allows the CPU to continue running the program codes, make a process for the next erasing and programming by checking the FDERSTA bit or FDPRSTA bit to see if the erasing or programming is completed.
Work RAM	Program memory space		Start erasing/programming the flash checking the bit is "0".
	Data flash area		

Bit No	Bit name	Description
3	FCPRSTA	This indicates whether the program flash area is in the state of programming. 0: Not in the state of programming (Initial value) 1: In the state of programming
2	FCERSTA	This indicates whether the program flash area is in the state of erasing. 0: Not in the state of erasing. (Initial value) 1: In the state of erasing
1	FDPRSTA	This indicates whether the data flash area is in the state of programming. 0: Not in the state of programming (Initial value) 1: In the state of programming
0	FDERSTA	This indicates whether the data flash area is in the state of erasing. 0: Not in the state of erasing. (Initial value) 1: In the state of erasing

29.3 Self-programming

The self-programming is the function to program (erase and program) the program memory space and data flash area using SFRs.
Table 29-7 shows the self-programming specifications for each of the program memory space and data flash area.

Table 29-7 Self-programming of Program Memory Space and Data Flash Area

		Program memory space	Data flash area
Programming unit	Block erasing	32K Byte	All area
	Sector erasing	2K Byte	256 Byte
	Programming	4 Byte	1 Byte
CPU operation during block/sector erase or program		Stop program processing (after completion of erasing/programming, resume program processing)	Continue program processing through the background operation (BGO) function (*3)
Confirmation of end of block/sector erasing or programming		Confirmation not required (as program run is stopped during erasing/programming)	Confirmation can be made through FLASHSTA register
Target area where block/sector erasing has been applied		Every bit becomes "1" (the bit written with "0" by writing becomes "0" from "1")	
Note on data programming		Erase the area to be reprogrammed (data programmed without erasing is unguaranteed)	
Function to prevent unintended erasing/programming		Flash self-register (FLASHSLF) and flash acceptor (FLASHACP) incorporated (*1)	
Flash memory erasing/programming		Supported only when system clock is the high-speed clock (*2)	
Note on user program programming		Before programming the user program, prepare a program for self-programming in the program code area which is not erased/reprogrammed	-
Remapping function		User program update, etc. can be performed by simultaneously using remapping function	-

*1: After the programming is enabled by the FLASHSLF register, if 0xFA and 0xF5 are written to the flash acceptor (FLASHACP), block/sector erase or reprogram is enabled only once.
*2: See Chapter 6 "Clock Generation Circuit" for enabling oscillation of the high-speed oscillation circuit and switching the system clock.
*3: At the start of BGO, the read from flash memory wait for 1μs.

29.3.1 Notes on Debugging Self-programming Code

When debugging the area within the scope of program for self-programming (from setting the flash acceptor to writing the flash data register), use the debugger according to the precautions described in Table 29-8.

Table 29-8 Notes on Debugging Self-programming

Limited function	Notes
Breakpoint setting	Do not perform the real time execution with break points set in the scope of program for self-programming (from setting the flash acceptor to setting the flash data register). Otherwise, the flash memory may not be reprogrammed if break points occur within the scope of program for self-programming.
Step execution	Do not perform the step execution within the scope of program for self-programming. Otherwise, the flash memory may not be reprogrammed if the step execution is performed within the scope of program for self-programming.

29.3.2 Programming Program Memory Space

In the program memory space (flash memory), block erase in units of 32 Kbytes, sector erase in units of 2 Kbyte, and reprogram in units of 4 bytes can be executed.
Figure 29-1 shows the flow diagram for erasing the program memory space.

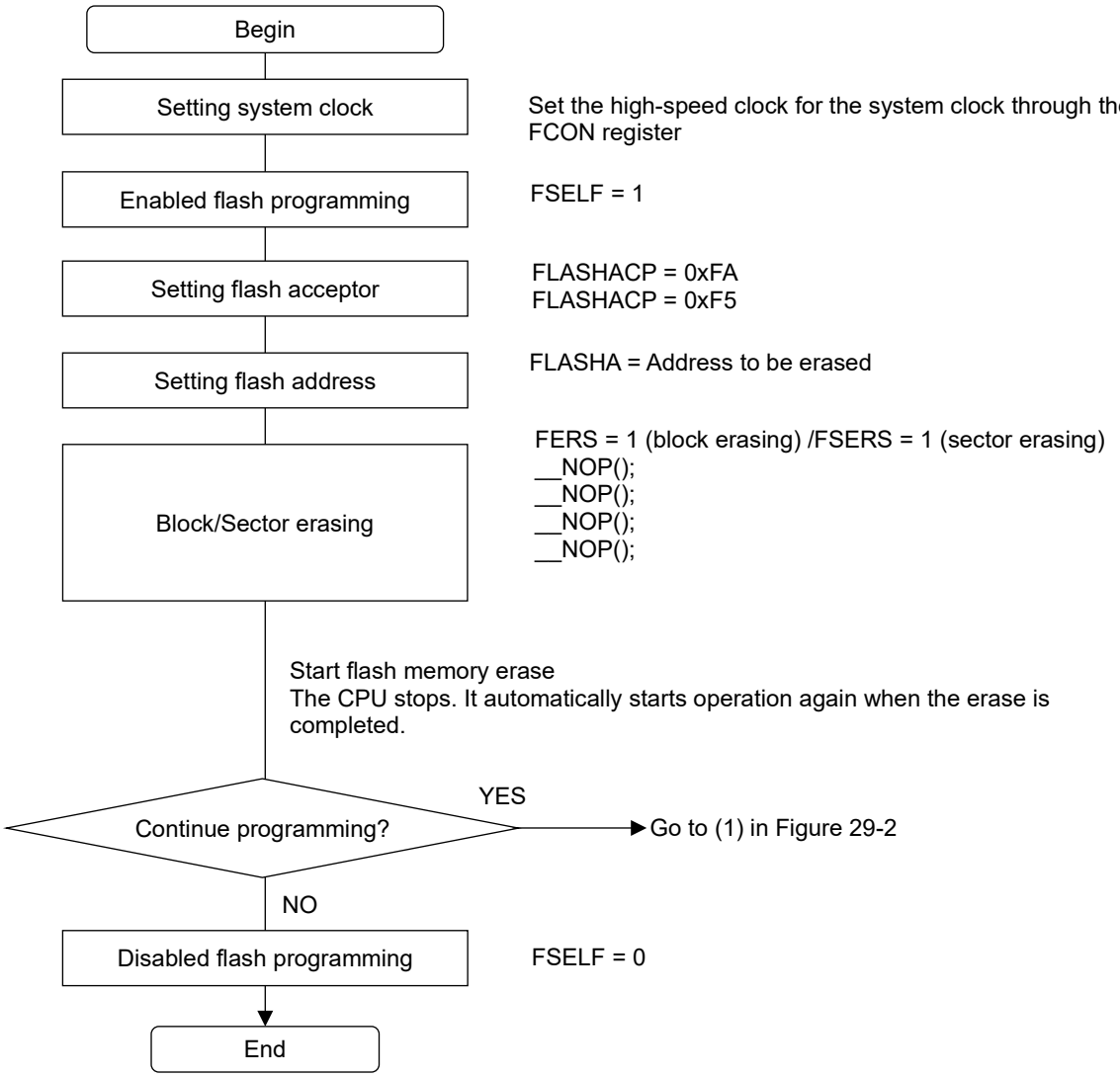


Figure 29-1 Flow Diagram for Erasing Program Memory Space

[Note]

- **Only erase areas irrelevant to program processing. If erasing the area where program processing is in progress, the LSI works incorrectly.**
- **During block/sector erasing, the CPU stops the operation for maximum 50 ms whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For block/sector erasing, place four NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 29-2 shows the flow diagram for programming the program memory space.

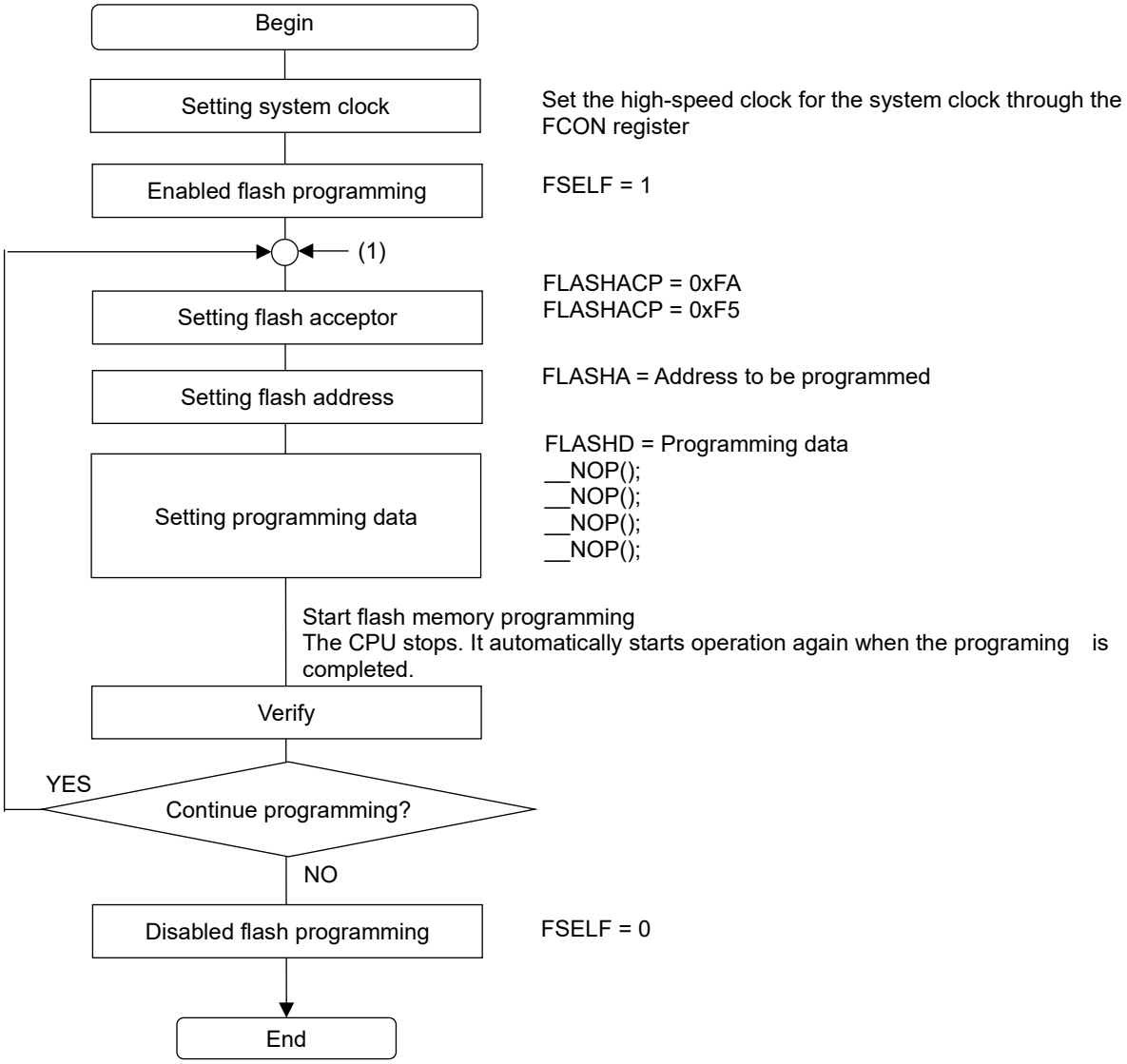


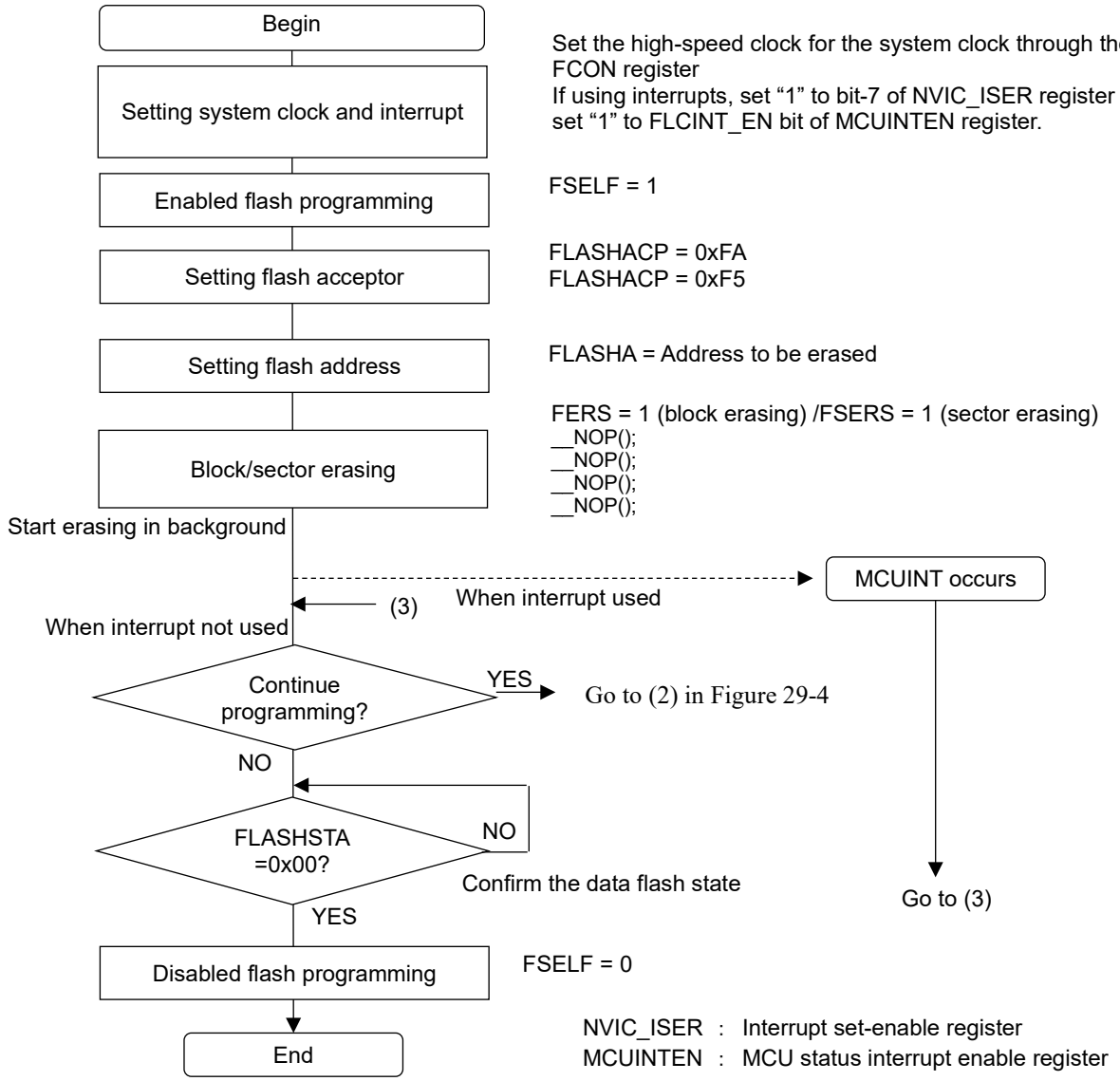
Figure 29-2 Flow Diagram for Programming Program Memory Space

[Note]

- **Only programming areas irrelevant to program processing. If programming the area where program processing is in progress, the LSI works incorrectly.**
- **During the programming, the CPU stops the operation for maximum 80 µs whereas peripheral circuits continue operation. Therefore, clear the WDT counter accordingly.**
- **For data programming setting, place four NOP instructions following the instruction used to set the programming data in the FLASHD register.**

29.3.3 Programming Data Flash Area

In the data flash area (flash memory), block erase in all area, sector erase in units of 256 bytes, and programming in units of 1 byte can be executed. During block/sector erase or program in the data flash area, the CPU continues program processing using the background operation (BGO) function.
Figure 29-3 shows the flow diagram for erasing the data flash area.



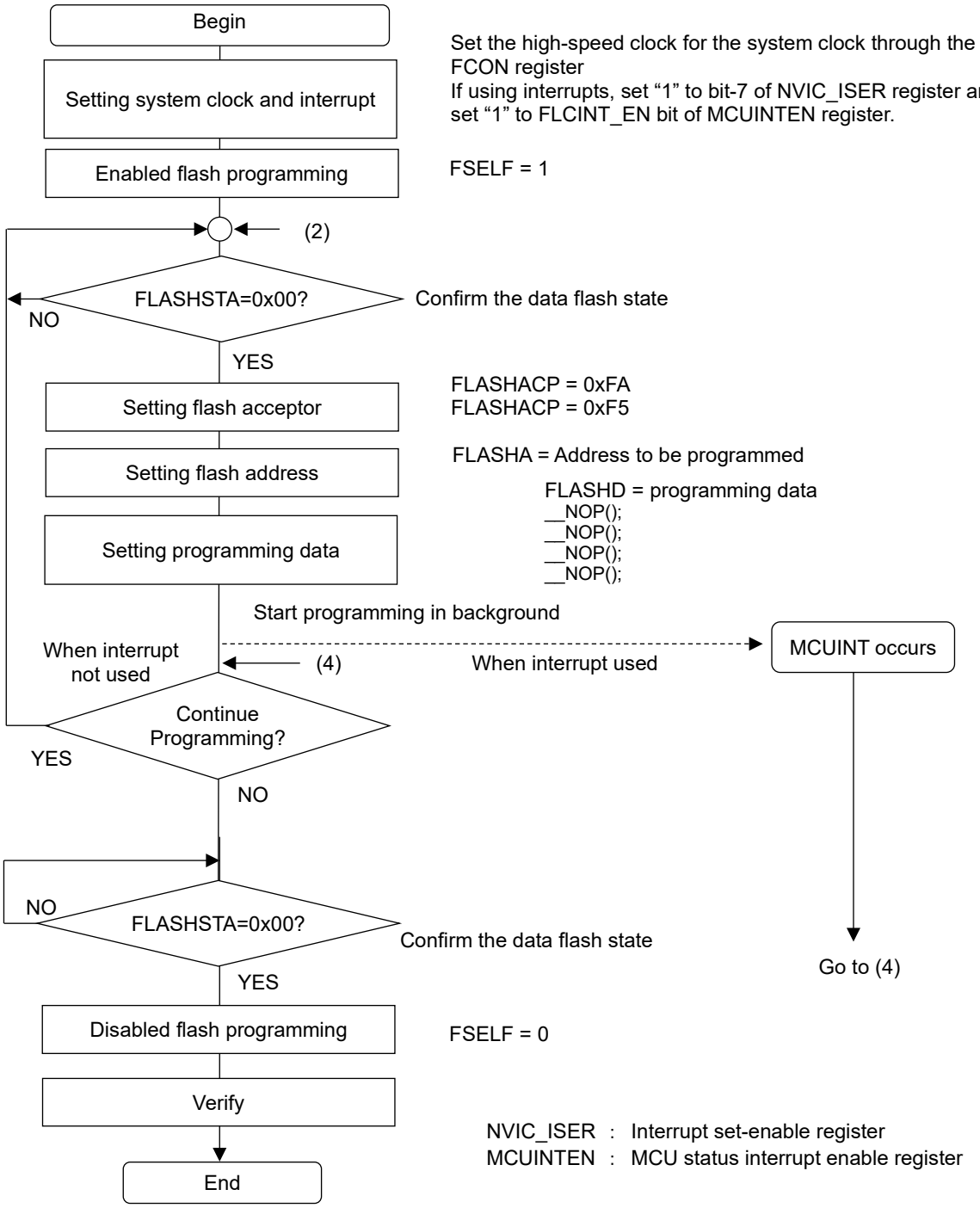
See Chapter 5, "System Control Function" and Chapter 7, “Interrupts” for using interrupts.

Figure 29-3 Flow Diagram for Erasing Data Flash Area

[Note]

- **The CPU continues program processing even while data flash erasing is in progress. Do not enter the STOP mode during the erasing. In addition, set the FSELF bit of the FLASHSLF register to "0" after the erasing is completed.**
- **The data flash area is unreadable during erasing.**
- **For block/sector erasing, place four NOP instructions following the instruction used to set FERS/FSERS bits of the FLASHCON register to "1".**

Figure 29-4 shows the flow diagram for programming the data flash area.



See Chapter 5, "System Control Function" and Chapter 7, “Interrupts” for using interrupts.

Figure 29-4 Flow Diagram for Programming Data Flash Area

[Note]

- **The CPU continues program processing even while data flash programming is in progress. Do not enter the STOP mode during the programming. In addition, set the FSELF bit of the FLASHSLF register to "0" (erase/program disabled) after the programming ended.**
- **The data flash area is unreadable during programming.**
- **For data programming setting, place four NOP instructions following the instruction used to set the programming data in the FLASHD register.**

29.3.4 Notes on use of self-programming

Table 29-9 shows the notes on the use of self-programming (block erasing/sector erasing/programming).

Table 29-9 Notes on Use of Self-programming

Item	Notes
System clock during use of self-programming	Set to high-speed clock. See "Chapter 6 Clock Generation Circuit" for enabling the high-speed clock oscillation and switching the system clock.
If power outage or forced termination due to a reset occurs	Data in flash memory is not guaranteed. Perform block/sector erase again then program data.
If LSI does not start up due to occurrence of power outage or forced termination during programming (*1)	Program the program again using on-chip debug emulator.
Access to SFRs related flash control.	Do not perform to write to the FLASHACP/FLASHSLF register during self-programming; when FLASHSTA is not 0x0. If this happens, the data in the flash memory that is being processed is not guaranteed.

*1 : While programming the block or sector including address 0x0000_0000 of the program area is in progress.

Chapter 30

Voltage Level Supervisor (VLS)

30. Voltage Level Supervisor (VLS)

30.1 Overview

This LSI has one unit of built-in Voltage Level Supervisor (VLS) that detects whether the voltage level of VDD is lower or higher than the selected threshold voltage.

30.1.1 Features

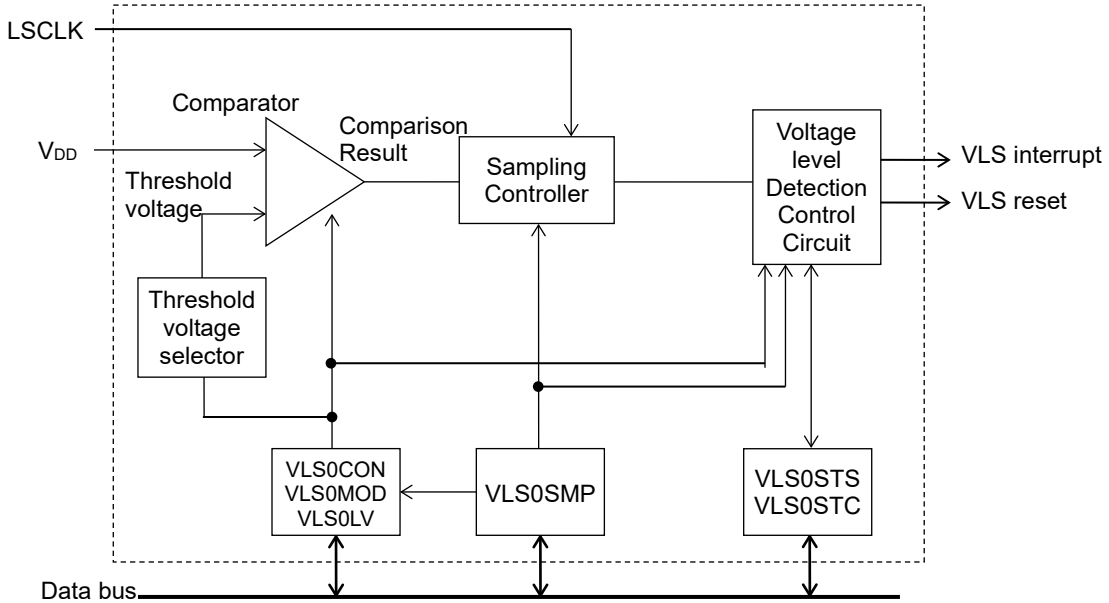
- Threshold voltage: Selectable from 10 values
- Operation mode: Supervisor mode (continuous detection) or single mode (one detection)

Mode	Description
Single mode 1	Detect the voltage level of V_{DD} only once. The interrupt occurs after detecting the voltage of V_{DD}, indicates the detection has been completed.
Single mode 2	Detect the voltage level of V_{DD} only once. The interrupt occurs after detecting the voltage of V_{DD} is lower than the threshold voltage, indicates the MCU is in the low voltage condition.
Supervisor mode	Detect continuously the voltage level of V_{DD}, suitable for always detecting the low voltage level of V_{DD} and generating the interrupt or reset. The interrupt or reset occurs according to the setting in the VLS0MOD register. The VLS reset function is available by choosing the supervisor mode.

- Voltage level supervisor reset (VLS reset). See Chapter 4, "Reset Function".
- Voltage level supervisor interrupt (VLS interrupt). See Chapter 7, "Interrupts".

30.1.2 Configuration

Figure 30-1 shows the configuration of the VLS.



VLS0CON : Voltage level supervisor control register
VLS0MOD : Voltage level supervisor mode register
VLS0LV : Voltage level supervisor level register
VLS0SMP : Voltage level supervisor sampling register
VLS0STS : Voltage level supervisor status register
VLS0STC : Voltage level supervisor status clear register

Figure 30-1 Configuration of Voltage Level Supervisor

30.2 Description of Registers

30.2.1 List of Registers

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4200_0000	VLS base address	VLS	-	-	-
0x00	Voltage level supervisor control register	VLS0CON	R/W	32	0x0000_0000
0x04	Voltage level supervisor mode register	VLS0MOD	R/W	32	0x0000_0000
0x08	Voltage level supervisor level register	VLS0LV	R/W	32	0x0000_0009
0x0C	Voltage level supervisor sampling register	VLS0SMP	R/W	32	0x0000_0000
0x10	Voltage level supervisor status register	VLS0STS	R	32	0x0000_0000
0x14	Voltage level supervisor status clear register	VLS0STC	W	32	0x0000_0000

30.2.2 Voltage Level Supervisor Control Register (VLS0CON)

Offset : 0x00

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VLS0EN
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLS0CON is a special function register (SFR) used to control the voltage level detector.
This register is not initialized by VLS reset.

Bit No	Bit name	Description
0	VLS0EN	This bit is used to control the VLS operation. 0: OFF (Initial value) 1: ON

[Note]

- **There is a limitation in each mode for entering the STOP mode while the VLS is running.**

Mode	Description
Running in the supervisor mode	The MCU can enter the STOP mode only when the VLS0RF bit is "1".
Running in the single mode	It is prohibit to enter the STOP mode. Wait for the single-mode operation to stop (VLS0EN = “0”).

30.2.3 Voltage Level Supervisor Mode Register (VLS0MOD)

Offset : 0x04

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	VLS0AMD		–	–	VLS0SEL	
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLS0MOD is a special function register (SFR) used to control the operation mode of the VLS (Voltage Level Supervisor).
Set this register only when the VLS is stopped (VLS0EN bit of VLS0CON register is "0").
This register is not initialized when VLS reset is occurred.

Bit No	Bit name	Description
5 to 4	VLS0AMD	VLS0AMD is used to choose the VLS0 operating mode. 00: Single mode 1 (Initial value) It detects the voltage level of V_{DD} only once. When VLS0SEL is "0x2", the interrupt occurs when detecting the voltage level of VDD. The result can be checked by reading VLS0FS bit of VLS0STS register. 01: Single mode 2 It detects the voltage level of V_{DD} only once. When VLS0SEL is "0x2", the interrupt occurs when detecting the voltage level of V_{DD} is lower than the threshold voltage (when the VLS0FS of VLS0STS register is "1"). 1x: Supervisor mode It always detects the voltage level of V_{DD}. The interrupt or reset occurs depending on the setting of VLS0SEL.
1 to 0	VLS0SEL	VLS0SEL is used to set a function. 00: No function (Initial value) 01: Enabled reset 10: Enabled interrupt request 11: Enabled reset

[Note]

- **There is a limitation in each mode for entering the STOP mode while the VLS is running.**

Mode	Description
Running in the supervisor mode	The MCU can enter the STOP mode only when the VLS0RF bit is "1".
Running in the single mode	The MCU is unable to enter the STOP mode.

30.2.4 Voltage Level Supervisor Level Register (VLS0LV)

Offset : 0x08

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	VLSLV			
R/W	–	–	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

VLS0LV is a special function register (SFR) used to set the detection voltage.
Set this register only when the VLS is stopped (VLS0EN bit of VLS0CON register is "0").
This register is not initialized when VLS reset is occurred.

Bit No	Bit name	Description
3 to 0	VLSLV	VLSLV is used to choose the threshold voltage (V_{VLSF} / V_{VLSR}) of VLS detected while the V_{DD} is falling or rising. The VLS has hysteresis characteristics. For the characteristics of threshold voltage detected while the V_{DD} is falling or rising, see the data sheet of each product. 0000: 3.99V 0001: 3.68V 0010: 3.05V 0011: 2.96V 0100: 2.84V 0101: 2.76V 0110: 2.66V 0111: 2.54V 1000: 2.45V 1001: 2.35V (Initial value) Other: Setting prohibited (2.35V)

30.2.5 Voltage Level Supervisor Sampling Register (VLS0SMP)

Offset : 0x0C

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	VLS0DIV		VLS0 SM1	–	–	–
R/W	–	–	–	–	–	–	–	–	–	–	R/W	R/W	R/W	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLS0SMP is a special function register (SFR) used to control sampling the voltage level detection.
Set this register only when the VLS0 is stopped (VLS0EN bit of VLS0CON register is "0").
This register is not initialized when VLS reset is occurred.

Bit No	Bit name	Description
5 to 4	VLS0DIV	This bit is used to set dividing rate of the sampling clock. 00: No dividing (Initial value) 01: Divided by 2 10: Divided by 4 11: Divided by 8
3	VLS0SM1	This bit is used to choose the sampling clock source for detecting the voltage level. 0: No sampling (Initial value) 1: LSCLK

[Note]

- **In the STOP mode, the sampling clock stops and the VLS works without sampling regardless the setting in VLS0SM1 bit.**

30.2.6 Voltage Level Supervisor Status Register (VLS0STS)

Offset : 0x10

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	VLS0RFS	VLS0FS	VLS0ENS
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLS0STS is a special function register (SFR) used to display the status of the voltage level detector.
This register is not initialized when VLS reset is occurred.

Bit No	Bit name	Description
2	VLS0RFS	This bit is used to indicate whether the voltage level detection result is valid or not. This bit is valid only in the supervisor mode and fixed to "0" in the single mode. 0: The VLS circuit is stopped or VLS is being stabilized (Initial value) 1: The VLS detection result is valid (readable)
1	VLS0FS	This bit for monitoring the voltage level retains the last detection result. Also, this bit is cleared to “0” at the start of the VLS operation. 0: V_{DD} is higher than the threshold voltage (Initial value) 1: V_{DD} is lower than the threshold voltage
0	VLS0ENS	This bit indicates the VLS operation state. 0: VLS OFF (Initial value) 1: VLS ON

30.2.7 Voltage Level Supervisor Status Clear Register (VLS0STC)

Offset : 0x14

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VLS0FC	–
R/W	–	–	–	–	–	–	–	–	–	–	–	–	–	–	W	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VLS0STC is a special function register (SFR) used to clear the status of the voltage level detector.

Bit No	Bit name	Description
1	VLS0FC	This bit is cleared VLS0FS bit of VLS0STS register to "0" by writing "1" to this bit, but not cleared by writing "0".

30.3 Description of Operation

VLS can be used to verify if VDD is lower or higher than the specified threshold voltage. In addition, it generates VLS interrupt or VLS reset. VLS has hysteresis characteristics. See the data sheet of each product for characteristics of the threshold voltage at power voltage fall / rise.
The following two operation modes are available for VLS:

- Supervisor mode :

Operation	"1" is written to VLS0EN to enable operation of VLS, and then detecting the voltage is executed. The result is notified of through the VLS0RFS flag as at the time the detection result becomes valid. The detection continues.	
Function	Voltage variations detection interrupt	The interrupt is generated when the power voltage becomes lower or higher than the threshold voltage.
	Low voltage detection reset	The reset can be generated when the power voltage becomes lower than the threshold voltage.

- Single mode :

Operation	"1" is written to VLS0EN to enable operation of VLS, and then detecting the voltage is executed. "0" is automatically written to VLS0EN to end the detection when the detection result becomes valid.	
Function	Single mode 1: Interrupt that indicates the detecting voltage has been completed	The interrupt is generated at the time of completion of the voltage detection.
	Single mode 2: Low voltage detection interrupt	The interrupt is generated when the power voltage becomes lower than the threshold voltage.

30.3.1 Supervisor Mode

In the supervisor mode, the voltage level of VDD can be constantly detected. This mode is suitable for using the reset when the low voltage is detected, or the interrupt when the voltage variations is detected.
Figure 30-2 shows the flow chart for starting the VLS in the supervisor mode.

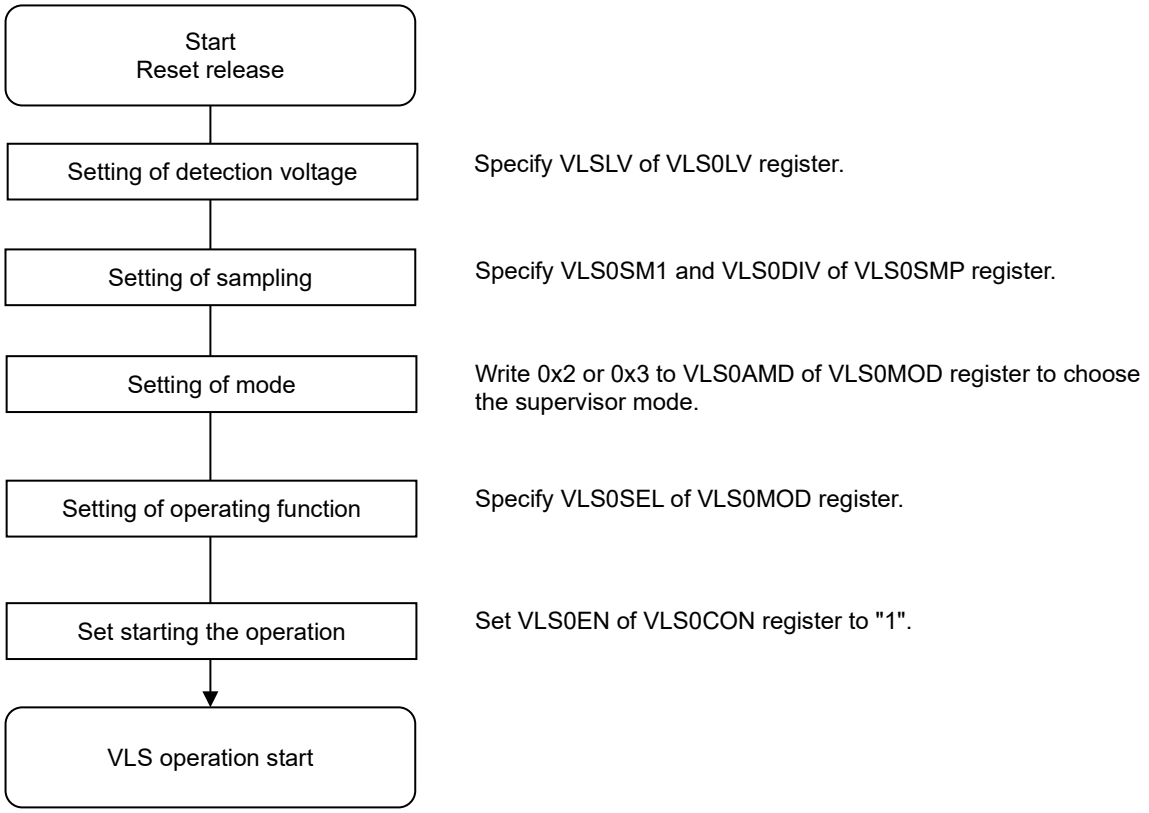


Figure 30-2 Flow chart for starting the VLS in the supervisor mode

30.3.1.1 Reset Output

Figure 30-3 shows the operation timing chart when the VLS reset output without sampling is specified.

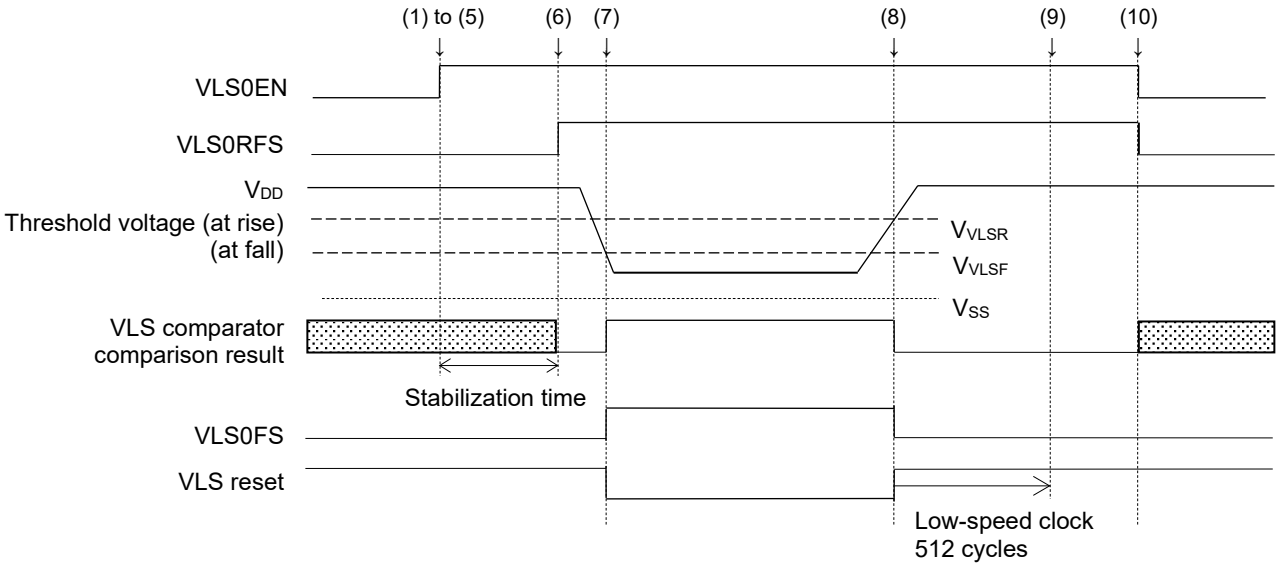


Figure 30-3 Operation Timing Chart When the VLS Reset Output without Sampling is specified

The operation shown in Figure 30-3 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD of VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x1” to VLS0SEL of the VLS0MOD register in order to enable the VLS reset.
(5) Set the VLS0EN bit of the VLS0CON register to "1" (VLS starts operation in the supervisor mode).
(6) After approximately 300 μs passed, the detection result of VLS becomes stabilized and the VLS0RFS bit of the VLS0STS register is set to "1" (value of the voltage level supervisor bit (VLS0FS) is read in software) (*1).
(7) When the power voltage (V_{DD}) becomes below the threshold voltage V_{VLSF}, the VLS0FS bit is set to "1" to generate the VLS reset.
(8) If V_{DD} becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0FS bit is cleared to "0" to release the VLS reset.
(9) The CPU starts after 512 cycles of low-speed clock.
(10) Write "0" to the VLS0EN bit to disable VLS operation.

*1 : VLS0FS bit/interrupt/reset is masked until the VLS0RFS bit becomes "1".

Figure 30-4 shows the operation timing chart when the VLS reset output with sampling is specified.

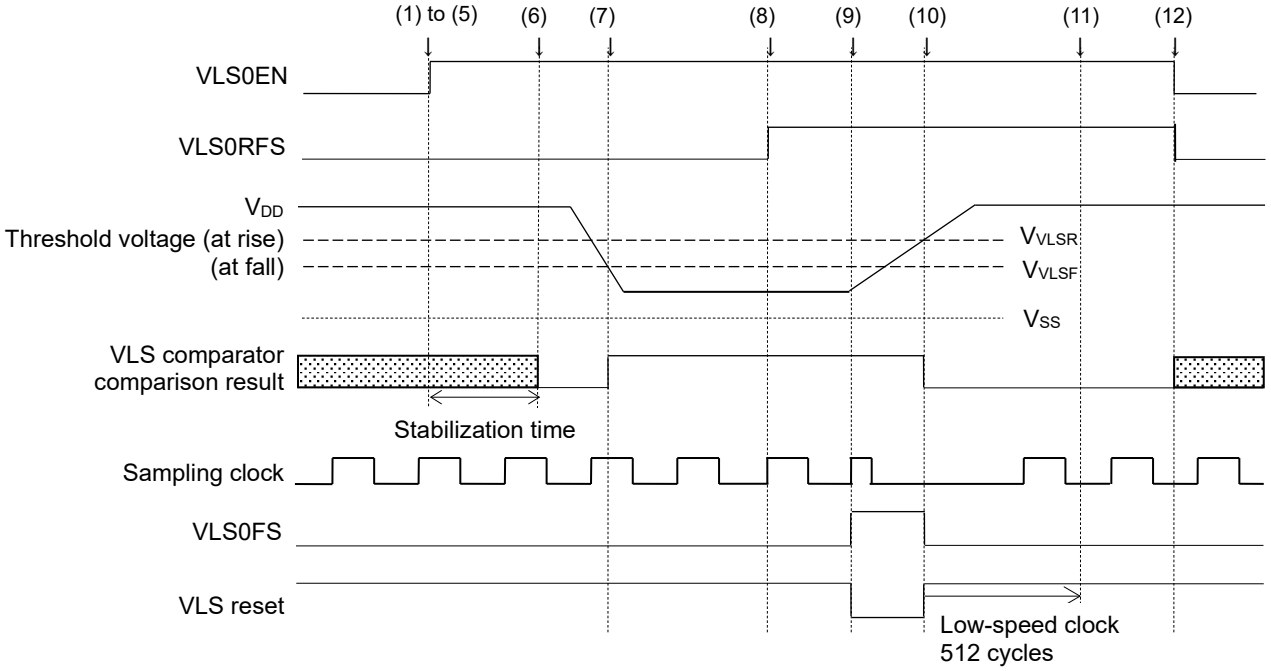


Figure 30-4 Operation Timing Chart When the VLS Reset Output with Sampling is specified

The operation shown in Figure 30-4 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "Sampling with LSCLK" by the VLS0SM1 bit of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD of the VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x1” to VLS0SEL of the VLS0MOD register in order to enable the VLS reset.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) Wait until the comparison result of the VLS comparator is stabilized (approx. 300 μs).
(7) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(8) Once the comparison result of the VLS comparator is stabilized, the VLS0RFS bit is set to "1" after three cycles of the sampling clock.
(9) If the comparison result of the VLS comparator is below the threshold voltage (V_{VLSF}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0FS bit is set to "1" and the VLS reset is generated.
(10) If the comparison result of the VLS comparator becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0FS bit is cleared to "0" to release the VLS reset.
(11) The CPU starts after 512 cycles of low-speed clock. The VLS does not operate while the sampling clock is stops.
(12) Write "0" to the VLS0EN bit to disable VLS operation.

[Note]

- **Entering the STOP mode is not allowed during the VLS stabilization time. If entering the STOP mode after the supervisor mode is enabled, make sure that the VLS0RFS bit is set to "1", and then enter the STOP mode.**
- **If you want to use the VLS reset function like a reset IC, start the VLS when the CPU initially runs after the power up.**

30.3.1.2 Interrupt Output

Figure 30-5 shows an example of the operation timing chart when the VLS interrupt output without sampling is specified.

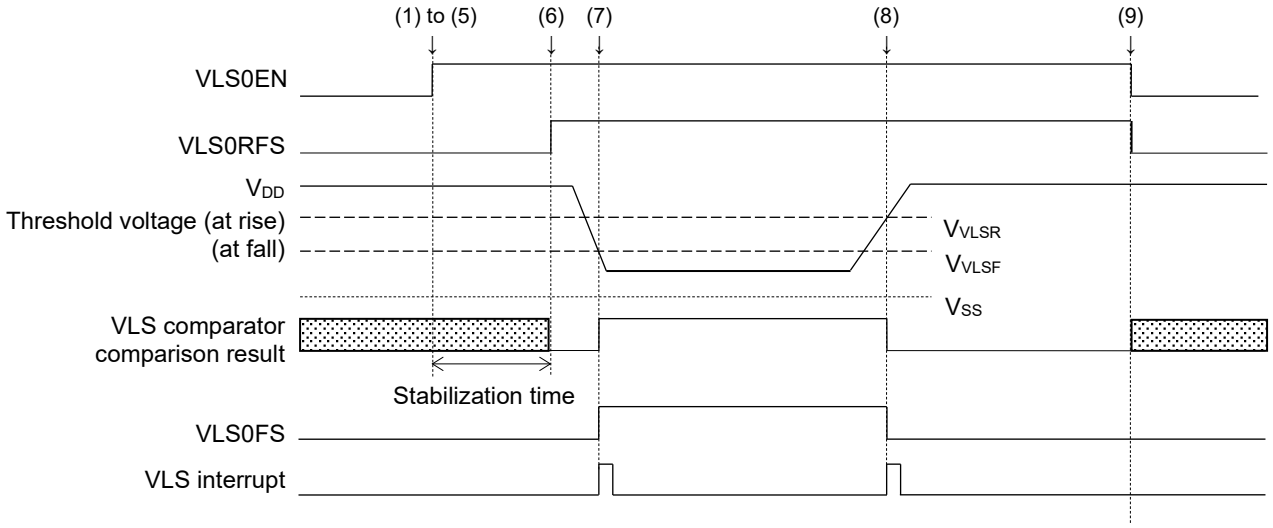


Figure 30-5 Operation Timing Chart When the VLS Interrupt Output without Sampling is specified

The operation shown in Figure 30-5 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 bit of the VLS0SMP register.
(3) Write 0x2 or 0x3 to VLS0AMD of VLS0MOD register in order to choose the supervisor mode.
(4) Write 0x2 to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) When the comparison result of the VLS comparator is stabilized, the VLS0RFS bit is set to "1".
(7) When V_{DD} becomes below the threshold voltage (V_{VLSF}), the VLS0FS bit is set to "1" to generate the VLS interrupt.
(8) If V_{DD} becomes equal to or above the threshold voltage (V_{VLSR}), the VLS0FS bit is cleared to "0" to generate the VLS interrupt.
(9) Write "0" to the VLS0EN bit to disable VLS operation.

Figure 30-6 shows an example of the operation timing chart when the VLS interrupt output with sampling is specified.

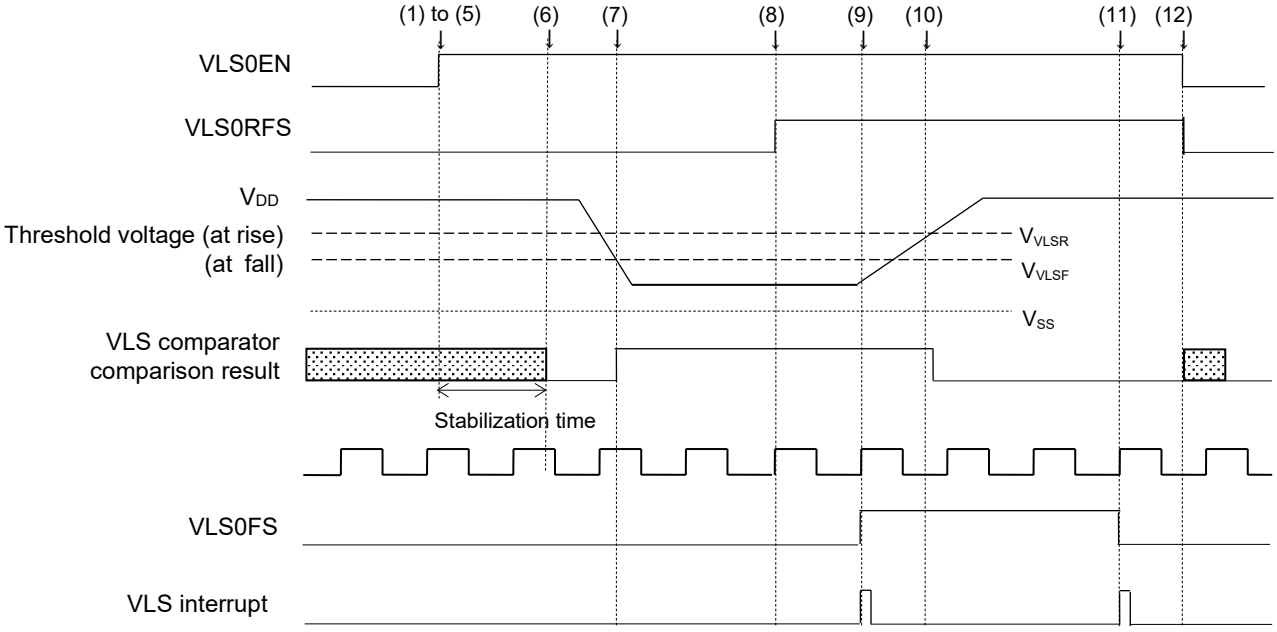


Figure 30-6 Operation Timing Chart When the VLS Interrupt Output with Sampling is specified

The operation shown in Figure 30-6 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "Sampling with LSCLK" by the VLS0SM1 of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV of the VLS0SMP register.
(3) Write "0x2" or "0x3" to VLS0AMD of VLS0MOD register in order to choose the supervisor mode.
(4) Write “0x2” to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) Wait until the comparison result of the VLS comparator is stabilized (approx. 300 μs).
(7) V_{DD} becomes below the threshold voltage (V_{VLSF}).
(8) Once the comparison result of the VLS comparator is stabilized, the VLS0RFS bit is set to "1" after three cycles of the sampling clock.
(9) If the comparison result of the VLS comparator is below the threshold voltage (V_{VLSF}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0FS bit is set to "1" and the VLS interrupt is generated.
(10) The comparison result of the VLS comparator becomes equal to or above the threshold voltage (V_{VLSR}).
(11) If the comparison result of the VLS comparator is equal to or above the threshold voltage (V_{VLSR}) and this condition continues for the duration of three cycles or more of the sampling clock, the VLS0FS bit is cleared to "0" and the VLS interrupt is generated.
(12) Write "0" to the VLS0EN bit to disable VLS operation.

[Note]

- **Entering the STOP mode is not allowed during the VLS stabilization time. If entering the STOP mode after the supervisor mode is enabled, make sure that the VLS0RFS bit is set to "1", and then enter the STOP mode.**
- **When VLS is stopped (VLS0EN bit="0") while the V_{DD} is lower than the specified threshold voltage (VLS0F bit="1"), the VLS interrupt is generated.**

30.3.2 Single Mode

In the single mode, the software waits for the VLS interrupt to detect the voltage. It is useful for intermittently checking V_{DD}.
Figure 30-7 shows the flow chart for starting the VLS in the single mode.

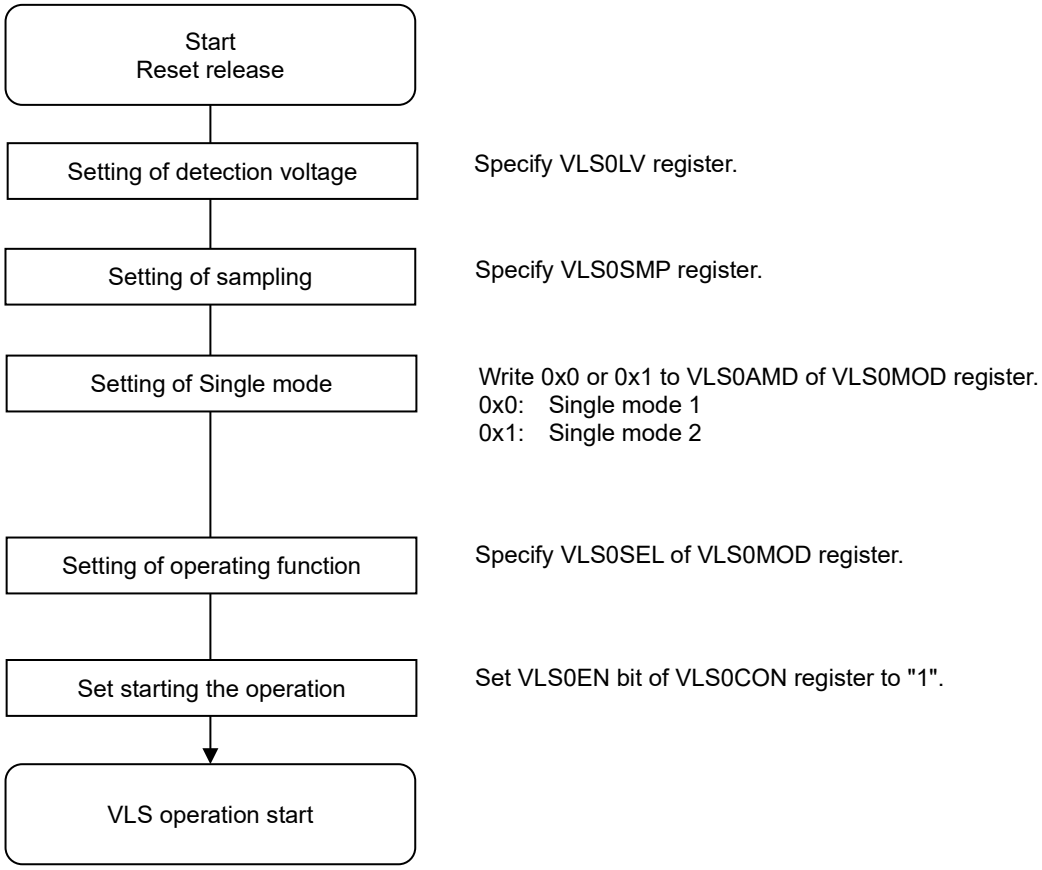


Figure 30-7 Flow chart for starting the VLS in the single mode

30.3.2.1 Single Mode 1

The single mode 1 always generates the interrupt at completing the detection.
Figure 30-8 shows an example of the operation timing chart without sampling in single mode 1.

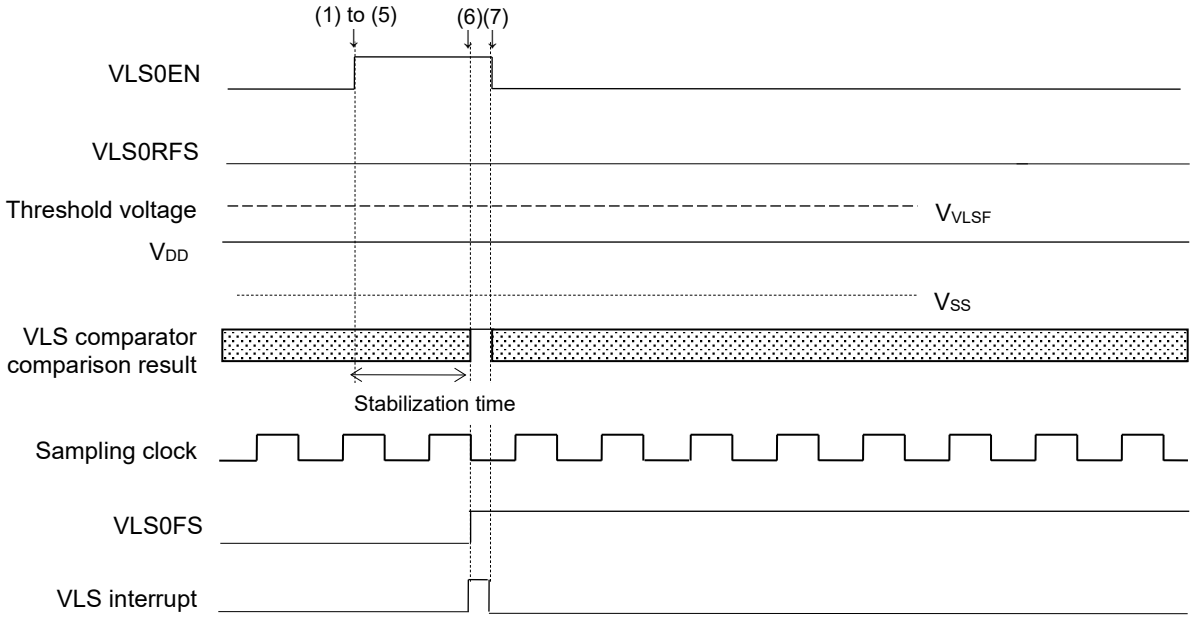


Figure 30-8 Operation Timing Chart without Sampling (Single Mode 1)

The operation shown in Figure 30-8 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 of the VLS0SMP register.
(3) Write 0x0 to VLS0AMD of VLS0MOD register in order to choose the single mode 1.
(4) Write 0x2 to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) If V_{DD} is below the threshold voltage (V_{VLSF}) when the comparison result of the VLS comparator is stabilized (*1), the VLS0FS bit is set to "1" and the VLS0 interrupt (detection complete) is generated. The VLS interrupt (detection complete) is generated regardless of the detection result of V_{DD}.
(7) After the interrupt is generated, the VLS0EN bit is cleared to "0" and VLS operation is disabled.
(8) Read the VLS0FS bit to confirm the detection result.

*1 : Stabilization time: Approx. 300 µs (approx. 300 µs + sampling clock cycle x 3 when sampling is enabled)

Figure 30-9 shows an example of the operation timing chart with sampling in single mode 1.

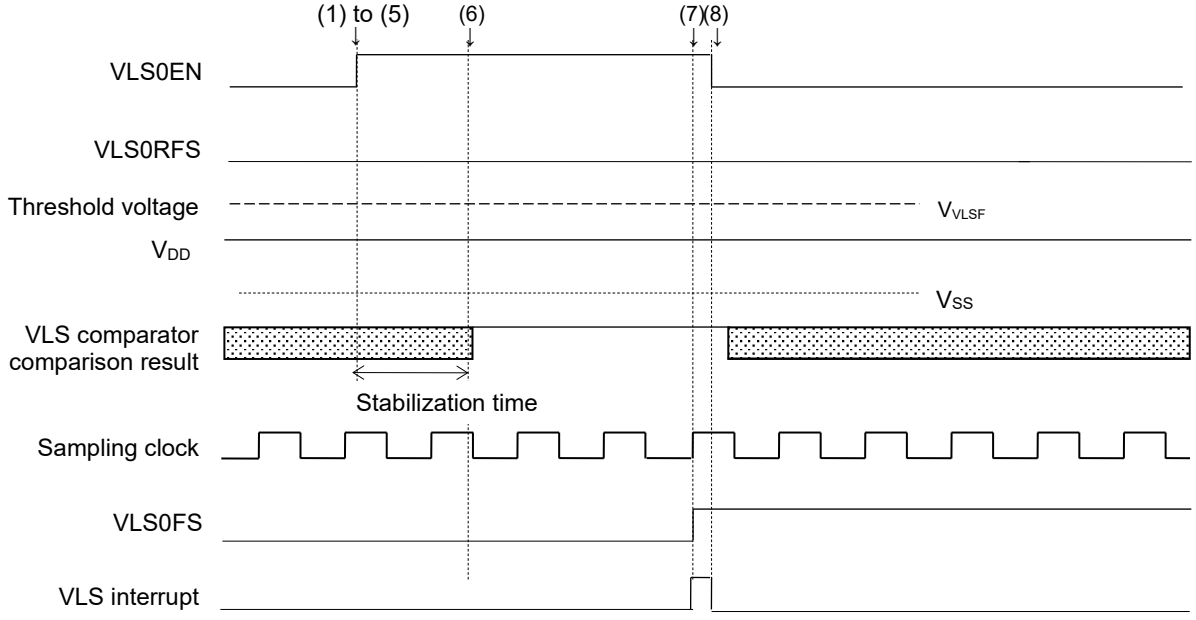


Figure 30-9 Operation Timing Chart with Sampling (Single Mode 1)

The operation shown in Figure 30-9 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "Sampling with LSCLK" by the VLS0SM1 of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV of the VLS0SMP register.
(3) Write 0x0 to VLS0AMD of VLS0MOD register in order to choose the single mode 1.
(4) Write 0x2 to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) Wait until the comparison result of the VLS comparator is stabilized (approx. 300 μs).
(7) If V_{DD} is below the threshold voltage (V_{VLSF}) after three cycles of the sampling clock, the VLS0FS bit is set to "1" and the VLS interrupt (detection complete) is generated. The VLS interrupt (detection complete) is generated regardless of the detection result of V_{DD}.
(8) After the interrupt is generated, the VLS0EN bit is cleared to "0" and VLS operation is disabled.
(9) Read the VLS0FS bit to confirm the detection result.

[Note]

- **Entering the STOP mode is not allowed while the single mode operation is in progress. Enter the STOP mode after the single mode operation is completed (VLS0EN bit="0").**

30.3.2.2 Single Mode 2

The single mode 2 generates the interrupt when the VDD is lower than the threshold voltage.
Figure 30-10 shows an example of the operation timing chart without sampling in single mode 2.

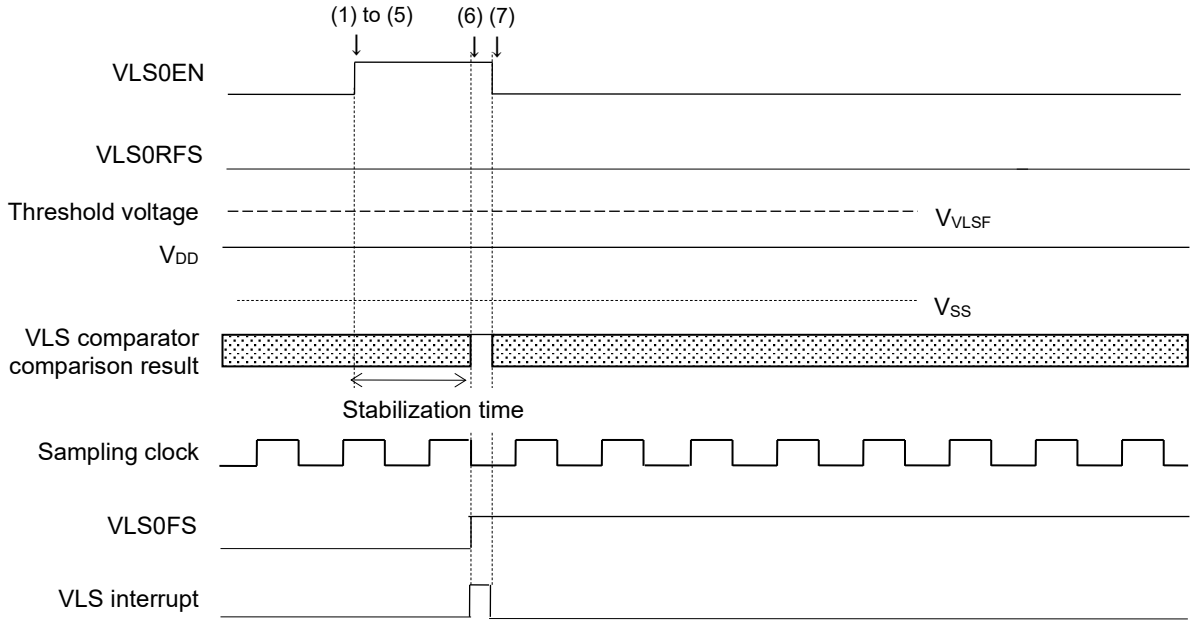


Figure 30-10 Operation Timing Chart without Sampling (Single Mode 2)

The operation shown in Figure 30-10 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "No sampling" by the VLS0SM1 of the VLS0SMP register.
(3) Write 0x1 to VLS0AMD of VLS0MOD register in order to choose the single mode 2.
(4) Write 0x2 to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS.
(6) If V_{DD} is below the specified threshold voltage (V_{VLSF}) when the comparison result of the VLS comparator is stabilized, the VLS0FS bit is set to "1" and the VLS interrupt (low voltage) is generated. If V_{DD} is higher than the specified threshold voltage (V_{VLSF}), the VLS0FS bit is cleared to "0" and the VLS interrupt (low voltage) is not generated.
(7) The VLS0EN bit is set to "0" and VLS is disabled regardless of whether the VLS interrupt occurs or not.

Figure 30-11 shows an example of the operation timing chart with sampling in single mode 2.

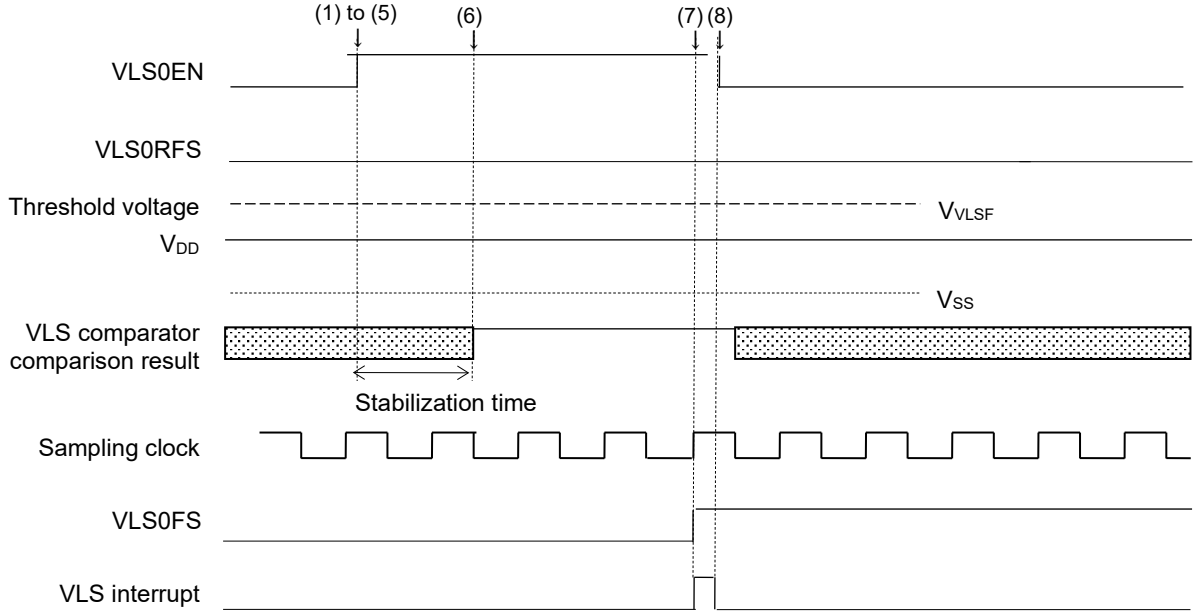


Figure 30-11 Operation Timing Chart with Sampling (Single Mode 2)

The operation shown in Figure 30-11 is described below:

(1) Choose a detection voltage by the VLSLV of the VLS0LV register.
(2) Choose "Sampling with LSCLK" by the VLS0SM1 of the VLS0SMP register. And specify sampling clock dividing ratio by the VLS0DIV of the VLS0SMP register.
(3) Write 0x1 to VLS0AMD of VLS0MOD register in order to choose the single mode 2.
(4) Write 0x2 to VLS0SEL of the VLS0MOD register in order to enable the VLS interrupt.
(5) Write "1" to the VLS0EN bit to enable VLS operation.
(6) Wait until the comparison result of the VLS comparator is stabilized (approx. 300µs).
(7) If V_{DD} is below the threshold voltage (V_{VLSF}) after three cycles of the sampling clock, the VLS0FS bit is set to "1" and the VLS interrupt (low voltage) is generated. If V_{DD} is equal to or above the threshold voltage (V_{VLSF}), the VLS0FS bit is cleared to "0" and the VLS interrupt (low voltage) is not generated.
(8) The VLS0EN bit is set to "0" and VLS is disabled regardless of whether the VLS interrupt occurs or not.

[Note]
- **Entering the STOP mode is not allowed while the single mode operation is in progress. Enter the STOP mode after the single mode operation is completed (VLS0EN bit ="0").**
- **If V_{DD} is higher than the specified threshold voltage, the VLS interrupt is not generated.**

Chapter 31

Code Option

31. Code Option

31.1 Overview

The code option is used to choose a watchdog timer operation etc. depending on values written in the code option area of the program memory area.

When the microcontroller starts up due to a system reset, the hardware automatically references the data in the code options area and configures each selected function.
The code option area can be erased or programmed through the on-chip debug function, or self-rewrite function of flash memory.

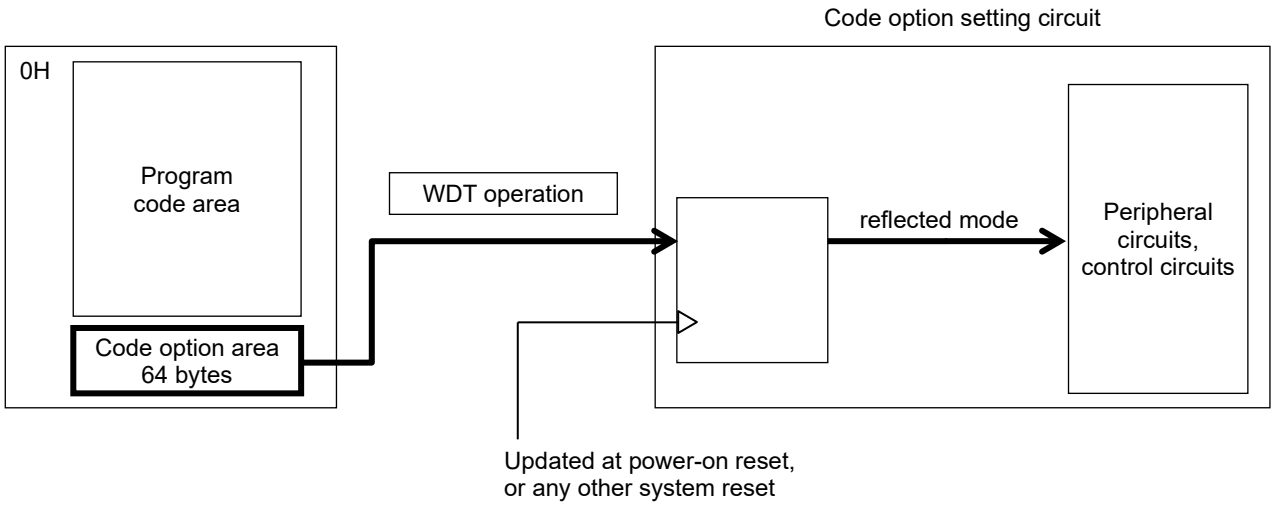


Figure 31-1 Code Option Overview

31.1.1 Function List

- Readable configured code options from SFRs
- Enabling or disabling the watchdog timer operation
- Enabling or disabling the watchdog timer operation at HALT

31.2 Description of Registers

31.2.1 Reading from SFRs

The address of code options area in the ROM space is a different for each product.
The address of SFR to read the code option setting is same for all products.

Address/Offset	Name	Symbol	R/W	Size	Initial value
0x4000_0480	Code option base address	CODEOP	-	-	-
0x04	Code Option 0	CODEOP0	R	32	0xFFFF_FFFF
0x08	Code Option 1	CODEOP1	R	32	0xFFFF_FFFF

[Note]
- **There are available to read the code option values from SFRs, if INITE flag bit of Reset Status Register (RSTAT) is "0".**
- **The initial value above indicates the value of the blank product. After the code is programmed, the value is reflected.**

31.2.2 Code Option 0 (CODEOP0)

Offset : 0x04

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	WDTPWMD0	-	WDTMD
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This is a special function register (SFR) used to read programmed code option.

Bit No	Bit name	Description
2	WDTPWMD0	This bit shows enabled/disabled the watchdog timer (WDT) operation in HALT mode. 0: Disabled 1: Enabled
0	WDTMD	This bit shows enabled/disabled the watchdog timer (WDT) operation. 0: Disabled 1: Enabled

31.2.3 Code Option 1 (CODEOP1)

Offset : 0x08

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This is a special function register (SFR) used to read programmed code option.

31.3 Code Option Data Setting

The address mappings for the code option area is shown in Table 31-1.
Fill the reserved area and reserved bits in the code option 0/1 with "1".

Table 31-1 Address Mappings for the Code Option Area

Flash address		Description
ML63Q2537/ML63Q2557	ML63Q2534/ML63Q2554	
0x1003_FFC0 to 0x1003_FFCF	0x1001_FFC0 to 0x1001_FFCF	Reserved area.
0x1003_FFD0	0x1001_FFD0	Code Option 0
0x1003_FFD4	0x1001_FFD4	Code Option 1
0x1003_FFD8 to 0x1003_FFDF	0x1001_FFD8 to 0x1001_FFDF	Reserved area
0x1003_FFE0 to 0x1003_FFFF	0x1001_FFE0 to 0x1001_FFFF	Reserved area.

Chapter 32

On-Chip Debug Function

32. On-Chip Debug Function

32.1 Overview

This LSI implements a SW-DP (serial wire debug port) as the debug interface.
The connection example is shown in Figure 32-1. For details, see the debugger manual.

[Note]
- **The timer and WDT clock are stopped at break in the debugger.**

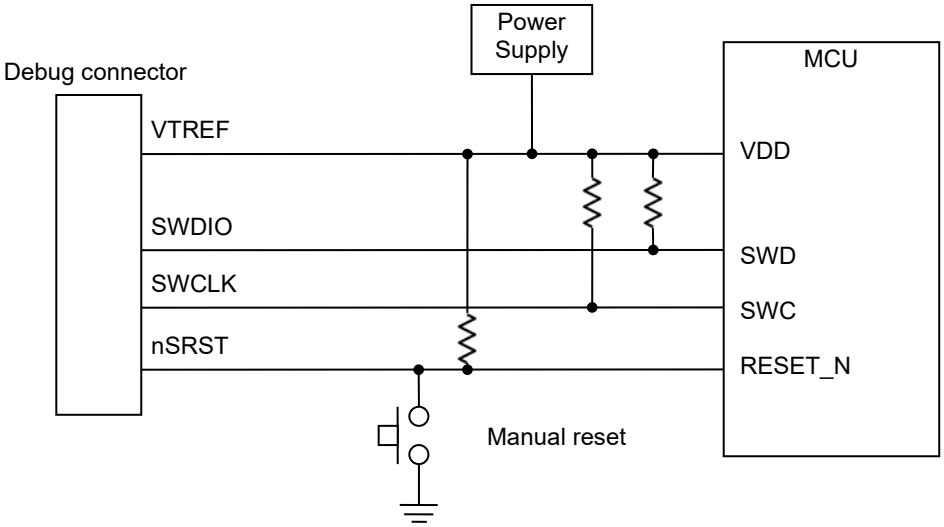


Figure 32-1 Connection with Debug Connector

32.2 Restriction

32.2.1 Standby Mode

When connecting the debugger, HALT mode is not worked. Therefore, this mode is worked as sleep mode that only CPU is stopped.
When connecting the debugger, STOP mode is not worked. Therefore, this mode is worked as sleep-deep mode.
The peripheral circuits are supplied clocks and continue to operate.
Verify the behavior of these mode without the debugger.

[Note]
- **Since the clock does not stop when the debugger is connected, an interrupt will occur even if the STOP mode entered.**

32.2.2 Operation of Peripheral Circuits during Breaks in the On-Chip Debug Mode

The following peripheral circuits stop working during breaks by the debugger.
- Counting operation of Timer, WDT, LTBC, RTC, and 1kHz timer
- Sampling operation of external interrupt circuit, because T16KHZ clock of LTBC is stopped.
- Clock of Three phase motor control PWM. It can be configured stop/operation by timer unit 0 break setting register.
- Conversion operation of SA-ADC. If the conversion is in progress at the time of the break, it will be stopped after the conversion is completed.
- VLS interrupt / reset
- Analog comparator interrupt request

Depending on the display of the memory window, etc., the SFR whose status changes when it is read, may be affected.
However, the following actions are disabled during the break and the state is maintained.
- Clearing capture status in the FTM
- Status update due to read reception buffer in the UART/SSIO/I^2C with FIFO.

[Note]
- **The clock of the three-phase motor control PWM should be set to "Run during break" for fail-safe.**

Revision History

REVISION HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL63Q2500-01	2025.01.20	—	—	Final edition 1st Revision