dsPIC33CDVL64MC106 FAMILY

16-Bit Digital Signal Controllers with High-Resolution PWM, Op Amps, Advanced Analog, MOSFET Driver and LIN Transceiver

Operating Conditions

- 3.0V to 3.6V: -40°C to +125°C, DC to 100 MIPS
- 3.0V to 3.6V: -40°C to +150°C, DC to 70 MIPS

High-Performance 16-Bit DSP RISC CPU

- 16-Bit Wide Data Path
- Code Efficient (C and Assembly) Architecture
- 40-Bit Wide Accumulators
- Single-Cycle (MAC/MPY) with Dual Data Fetch
- Single-Cycle, Mixed-Sign Multiply:
  - 32-bit multiply support
- Fast Six-Cycle Divide
- Zero Overhead Looping

High-Resolution PWM

- Four PWM Pairs
- Up to 2 ns PWM Resolution
- Dead Time for Rising and Falling Edges
- Dead-Time Compensation
- Clock Chopping for High-Frequency Operation
- PWM Support for:
  - DC/DC, AC/DC, inverters, PFC, lighting
  - BLDC, PMSM, ACIM, SRM motors
- Fault and Current Limit Inputs
- Flexible Trigger Configuration for ADC Triggering

High-Speed Analog-to-Digital Converter

- Up to 15 A/D inputs
- 12-Bit Resolution
- One Shared SAR ADC Core
- Up to 3.5 Msps Conversion Rate per Core
- Dedicated Result Buffer for Each Analog Channel
- Flexible and Independent ADC Trigger Sources
- Four Digital Comparators
- Four Oversampling Filters

Microcontroller Features

- High-Current I/O Sink/Source
- Edge or Level Change Notification Interrupt on I/O Pins
- Peripheral Pin Select (PPS) Remappable Pins
- Up to 64 Kbytes Flash Memory:
  - 10,000 erase/write cycle endurance
  - 20 years minimum data retention
  - Self-programmable under software control
  - Programmable code protection
  - Error Code Correction (ECC)
  - Flash OTP by ICSP™ Write Inhibit
- Eight Kbytes SRAM Memory:
  - SRAM Memory Built-In Self-Test (MBIST)
- Multiple Interrupt Vectors with Individually Programmable Priority
- Four Sets of Interrupt Context Saving Registers which Include Accumulator and STATUS for Fast Reserved Interrupt Handling
- Four External Interrupt Pins
- Watchdog Timer (WDT)
- Windowed Deadman Timer (DMT)
- Fail-Safe Clock Monitor (FSCM) with Dedicated Oscillator
- Selectable Oscillator Options Including:
  - High-precision, 8 MHz internal Fast RC (FRC) Oscillator
  - Primary high-speed, crystal/resonator oscillator or external clock
  - Primary PLL, which can be clocked from FRC or crystal oscillator
- Low-Power Management modes (Sleep and Idle)
- Power-on Reset and Brown-out Reset
- On-Board Capacitorless Regulator
- 384 Bytes of One-Time-Programmable (OTP) Memory

Peripheral Features

- Two Four-Wire SPI modules (up to 50 Mbps):
  - 16-byte FIFO
  - Variable width
  - I2S mode
- One I2C Host and Client w/Address Masking and IPMI Support
- Three Protocol UARTs with Automated Handling Support for:
  - LIN 2.2
  - DMX
  - Smart card (ISO 7816)
- One SENT module
- Timers/Counters:
  - One dedicated 16-bit timer/counter
- Four Single Output Capture/Compare/PWM/ Timer (SCCP) Modules:
  - Flexible configuration as PWM, input capture, output compare or timers
  - Two 16-bit timers or one 32-bit timer in each module
  - PWM resolution down to 2.5 ns
  - Single PWM output
- One Quadrature Encoder Interface (QEI):
  - Four inputs: Phase A, Phase B, Home, Index
  - One 32-bit timer/counter (in QEI module, available if encoder is not used)
- Reference Clock Output (REFCLKO)
- Four Configurable Logic Cells (CLC) with Internal Connections to Select Peripherals and PPS
- Four-Channel Hardware DMA
- 32-Bit CRC Calculation Module
- Peripheral Trigger Generator (PTG):
  - 16 possible trigger sources to other peripheral modules
  - CPU-independent state machine-based instruction sequencer
  - Two 16-bit general purpose timers

Analog Features

- One Fast Analog Comparator with Input Multiplexing
- Three Operational Amplifiers
- One 12-Bit PDM DAC with Slope Compensation
- One Output DAC Buffer

Debug Features

- Three Programming and Debugging Interfaces:
  - Two-wire ICSP™ interface with non-intrusive access and real-time data exchange with application
- Three Complex, Five Simple Breakpoints
- IEEE Standard 1149.2 Compatible (JTAG) Boundary Scan

Safety Features

- Backup Fast RC Oscillator (BFRC)
- Brown-out Reset (BOR)
- Capless Internal Voltage Regulator
- Clock Monitor System with Backup Oscillator
- CodeGuard™ Security
- Cyclic Redundancy Check (CRC)
- Dual Watchdog Timer (WDT)
- Fail-Safe Clock Monitoring (FSCM)
- Flash Error Correcting Code (ECC)
- Flash OTP by ICSP™ Write Inhibit
- RAM Memory Built-In Self-Test (MBIST)
- Two-Speed Start-up
- Virtual Pins for Redundancy and Monitoring
- Windowed Deadman Timer (DMT)

MOSFET Gate Driver Module (Based on MCP8021 Device)

- Three Half-Bridge Drivers Configured to Drive External High-Side NMOS and Low-Side NMOS MOSFETs:
  - Peak output current: 0.5A @ 12V
  - Shoot-through protection
  - Overcurrent and short-circuit protection
- Fixed Output Linear Regulator:
  - 3.3V @ 70 mA
  - True Current Foldback
- Protection Features:
  - Gate Drive Undervoltage Lockout: 4.5V
  - Supply Voltage Undervoltage Shutdown: 4.5V
  - Supply Voltage Undervoltage Lockout (UVLO): 6.25V
  - Overvoltage Lockout (OVLO): 32V
  - Transient (100 ms) Voltage Tolerance: 40V
  - Power Module Thermal Shutdown

LIN Transceiver Module (Based on ATA663211 Device) (dsPIC33CDVL64MC106 Only)

- LIN_BUS Voltage Up to 40V
- LIN_V_{DD} Voltage = 5V to 28V
- Very Low Supply Current:
  - Sleep mode: typically 9 µA
  - Fail-Safe mode: typically 80 µA
  - Normal mode: typically 250 µA
- Fully Compatible with 3.3V and 5V Devices
- LIN Physical Layer According to LIN 2.0, 2.1, 2.2, 2.2A and SAEJ2602-2
- Wake-up Capability via LIN_BUS Activity (100 µs dominant)
- External Wake-up via LIN_WKIN Pin (100 µs low level)

Functional Safety Collaterals

- Class B Safety Library – IEC 60730
- For ASIL B and Beyond Applications – ISO 26262
- FMEDA Computation Spreadsheet (evaluation of Random Hardware Failures Metric)
- Functional Safety Manual
- Functional Safety Diagnostics Suite

Qualification

- AEC-Q100 REV G (Grade 1: -40°C to +125°C)
- AEC-Q100 REV G (Grade 0: -40°C to +150°C)

dsPIC33CDVL64MC106

The dsPIC33CDVL64MC106 family device features are listed in Table 1.

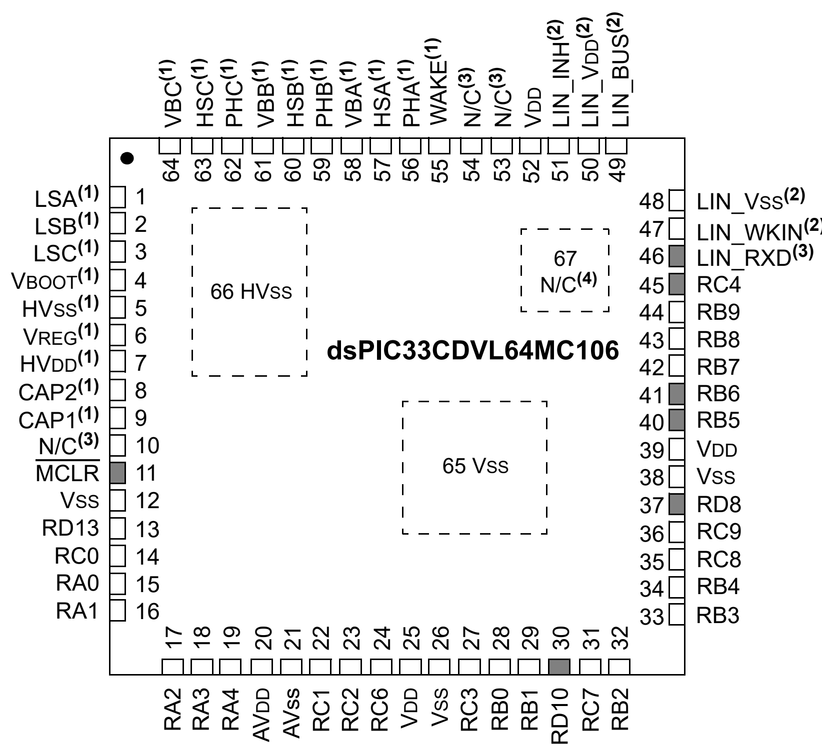
TABLE 1: dsPIC33CDVL64MC106 DEVICE FEATURES

Device	Pins	Flash (Kbytes)	SRAM (Kbytes)	DMA # of Ch	GPIO/PPS	LIN Transceiver	MOSFET Gate Driver	16-Bit Timer	SCCP	UART	SPI	I^2C	SENT	MC PWM	PWM Res (ns)	QEI	12-Bit ADC Module	ADC Channels	Op Amp	Analog Comp w/12-Bit DAC	PTG	CLC	REFO	CRC	WDT/DMT
dsPIC33CDVL64MC106	64	64	8	4	27/22	1	1	1	4	3	2	1	1	4x2	2	1	1	15	3	1	1	4	1	1	1/1
dsPIC33CDV64MC106	64	64	8	4	30/25	0	1	1	4	3	2	1	1	4x2	2	1	1	15	3	1	1	4	1	1	1/1

Pin Diagrams

64-Pin VGQFN

= 5.5 VDC Tolerant



Note 1: These pins are specific to the MOSFET gate driver module.

2: These pins are specific to the LIN transceiver module.

3: Pin is connected to a device interconnect; see Table 1-1 for more information.

4: Pin has no electrical connection to the device, but is recommended to be connected to VSS/LIN_VSS in the PCB design to improve thermal dissipation.

TABLE 2: dsPIC33CDVL64MC106 COMPLETE PIN FUNCTIONS

Pin	Function	Pin	Function
1	LSA	35	**RP56**/ASDA1/SCK2/RC8
2	LSB	36	**RP57**/ASCL1/SDI2/RC9
3	LSC	37	**RP72**/PCI19/SDO2/RD8
4	VBOOT	38	VSS
5	HVSS	39	VDD
6	VREG	40	PGD3/**RP37**/RB5
7	HVDD	41	PGC3/**RP38**/RB6
8	CAP2	42	AN2/**RP39**/RB7
9	CAP1	43	PGD1/AN10/**RP40**/SCL1/RB8
10	N/C(1)	44	PGC1/AN11/**RP41**/SDA1/RB9
11	$\overline{\text{MCLR}}$	45	**RP52**/RC4
12	VSS	46	LIN_RXD(1)
13	ANN0/**RP77**/RD13	47	LIN_WKIN
14	AN12/**RP48**/RC0	48	LIN_VSS
15	OA1OUT/AN0/CMP1A/IBIAS0/RA0	49	LIN_BUS
16	OA1IN-/RA1	50	LIN_VDD
17	OA1IN+/AN9/RA2	51	LIN_INH
18	DACOUT/AN3/CMP1C/RA3	52	VDD
19	OA3OUT/AN4/IBIAS3/RA4	53	N/C(1)
20	AVDD	54	N/C(1)
21	AVSS	55	WAKE
22	OA3IN-/AN13/CMP1B/ISRC0/**RP49**/RC1	56	PHA
23	OA3IN+/AN14/ISRC1/**RP50**/RC2	57	HSA
24	IBIAS1/**RP54**/RC6	58	VBA
25	VDD	59	PHB
26	VSS	60	HSB
27	AN15/IBIAS2/**RP51**/RC3	61	VBB
28	OSCI/CLKI/AN5/**RP32**/RB0	62	PHC
29	OSCO/CLKO/AN6/**RP33**/RB1	63	HSC
30	ISRC3/**RP74**/RD10	64	VBC
31	ISRC2/**RP55**/RC7	65	VSS(2)
32	OA2OUT/AN1/AN7/CMP1D/**RP34**/INT0/RB2	66	HVSS(2)
33	PGD2/OA2IN-/AN8/**RP35**/RB3	67	N/C(3)
34	PGC2/OA2IN+/**RP36**/RB4		

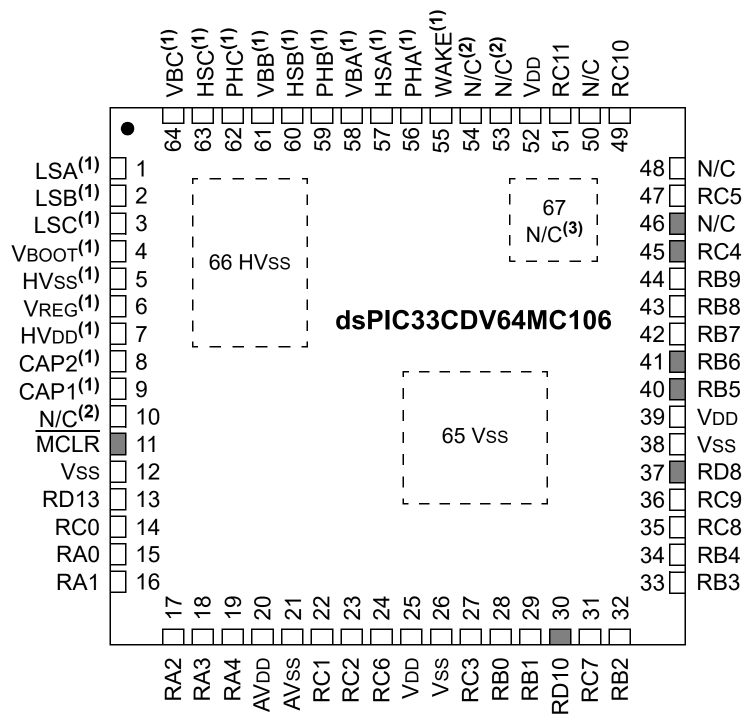
Legend: **RPn** represents remappable pins for the Peripheral Pin Select (PPS) function.

Note 1: Pin is connected to a device interconnect; see Table 1-1 for more information.

2: Pin is connected to a device exposed pad.

3: Pin has no electrical connection to the device, but is recommended to be connected to VSS in the PCB design to improve thermal dissipation.

Pin Diagrams (Continued)



Note 1: These pins are specific to the MOSFET gate driver module.

2: Pin is connected to a device interconnect; see Table 1-1 for more information.

3: Pin has no electrical connection to the device, but is recommended to be connected to Vss in the PCB design to improve thermal dissipation.

TABLE 3: dsPIC33CDV64MC106 COMPLETE PIN FUNCTIONS

Pin	Function	Pin	Function
1	LSA	35	**RP56**/ASDA1/SCK2/RC8
2	LSB	36	**RP57**/ASCL1/SDI2/RC9
3	LSC	37	**RP72**/PCI19/SDO2/RD8
4	VBOOT	38	VSS
5	HVSS	39	VDD
6	VREG	40	PGD3/**RP37**/RB5
7	HVDD	41	PGC3/**RP38**/RB6
8	CAP2	42	AN2/**RP39**/RB7
9	CAP1	43	PGD1/AN10/**RP40**/SCL1/RB8
10	N/C(1)	44	PGC1/AN11/**RP41**/SDA1/RB9
11	$\overline{\text{MCLR}}$	45	**RP52**/RC4
12	VSS	46	N/C
13	ANN0/**RP77**/RD13	47	**RP53**/RC5
14	AN12/**RP48**/RC0	48	N/C
15	OA1OUT/AN0/CMP1A/IBIAS0/RA0	49	**RP58**/RC10
16	OA1IN-/RA1	50	N/C
17	OA1IN+/AN9/RA2	51	**RP59**/RC11
18	DACOUT/AN3/CMP1C/RA3	52	VDD
19	OA3OUT/AN4/IBIAS3/RA4	53	N/C(1)
20	AVDD	54	N/C(1)
21	AVSS	55	WAKE
22	OA3IN-/AN13/CMP1B/ISRC0/**RP49**/RC1	56	PHA
23	OA3IN+/AN14/ISRC1/**RP50**/RC2	57	HSA
24	IBIAS1/**RP54**/RC6	58	VBA
25	VDD	59	PHB
26	VSS	60	HSB
27	AN15/IBIAS2/**RP51**/RC3	61	VBB
28	OSCI/CLKI/AN5/**RP32**/RB0	62	PHC
29	OSCO/CLKO/AN6/**RP33**/RB1	63	HSC
30	ISRC3/**RP74**/RD10	64	VBC
31	ISRC2/**RP55**/RC7	65	VSS(2)
32	OA2OUT/AN1/AN7/CMP1D/**RP34**/INT0/RB2	66	HVSS(2)
33	PGD2/OA2IN-/AN8/**RP35**/RB3	67	N/C(3)
34	PGC2/OA2IN+/**RP36**/RB4		

Legend: **RPn** represents remappable pins for the Peripheral Pin Select (PPS) function.

Note 1: Pin is connected to a device interconnect; see Table 1-1 for more information.

2: Pin is connected to a device exposed pad.

3: Pin has no electrical connection to the device, but is recommended to be connected to VSS in the PCB design to improve thermal dissipation.

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **doc.errors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

https://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website: **https://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our website at **https://www.microchip.com** to receive the most current information on all of our products.

Referenced Sources

This device data sheet is based on the following individual chapters of the *"dsPIC33/PIC24 Family Reference Manual"*. These documents should be considered as the general reference for the operation of a particular module or device feature.

Note:	To access the documents listed below, browse to the documentation section of the dsPIC33CDVL64MC106 product page of the Microchip website (www.microchip.com) or select a family reference manual section from the following list. In addition to parameters, features and other documentation, the resulting page provides links to the related family reference manual sections.

- **"Introduction"** (www.microchip.com/DS70573)
- **"Enhanced CPU"** (www.microchip.com/DS70005158)
- **"Data Memory"** (www.microchip.com/DS70595)
- **"dsPIC33/PIC24 Program Memory"** (www.microchip.com/DS70000613)
- **"Flash Programming"** (www.microchip.com/70000609)
- **"Reset"** (www.microchip.com/DS70602)
- **"Interrupts"** (www.microchip.com/DS70000600)
- **"I/O Ports with Edge Detect"** (www.microchip.com/DS70005322)
- **"Oscillator Module with High-Speed PLL"** (www.microchip.com/DS70005255)
- **"Direct Memory Access Controller (DMA)"** (www.microchip.com/DS30009742)
- **"High-Resolution PWM with Fine Edge Placement"** (www.microchip.com/DS70005320)
- **"12-Bit High-Speed, Multiple SARs A/D Converter (ADC)"** (www.microchip.com/DS70005213)
- **"High-Speed Analog Comparator Module"** (www.microchip.com/DS70005280)
- **"Quadrature Encoder Interface (QEI)"** (www.microchip.com/DS70000601)
- **"Multiprotocol Universal Asynchronous Receiver Transmitter (UART) Module"** (www.microchip.com/DS70005288)
- **"Serial Peripheral Interface (SPI) with Audio Codec Support"** (www.microchip.com/DS70005136)
- **"Inter-Integrated Circuit (I^2C)"** (www.microchip.com/DS70000195)
- **"Single-Edge Nibble Transmission (SENT) Module"** (www.microchip.com/DS70005145)
- **"Timer1 Module"** (www.microchip.com/DS70005279)
- **"Capture/Compare/PWM/Timer (MCCP and SCCP)"** (www.microchip.com/DS30003035)
- **"Configurable Logic Cell (CLC)"** (www.microchip.com/DS70005298)
- **"Peripheral Trigger Generator (PTG)"** (www.microchip.com/DS70000669)
- **"Current Bias Generator (CBG)"** (www.microchip.com/DS70005253)
- **"Deadman Timer (DMT)"** (www.microchip.com/DS70005155)
- **"32-Bit Programmable Cyclic Redundancy Check (CRC)"** (www.microchip.com/DS30009729)
- **"Dual Watchdog Timer"** (www.microchip.com/DS70005250)
- **"Watchdog Timer and Power-Saving Modes"** (www.microchip.com/DS70615)
- **"Programming and Diagnostics"** (www.microchip.com/DS70608)
- **"CodeGuard™ Intermediate Security"** (www.microchip.com/DS70005182)
- **"Flash Programming"** (www.microchip.com/DS70000609)

Terminology Cross Reference

Table 4 provides updated terminology for depreciated naming conventions. Register and bit names remain unchanged, however, descriptions and usage guidance may have been updated

TABLE 4: TERMINOLOGY CROSS REFERENCES

Use Case	Depreciated Term	New Term
CPU	Master	Initiator
DMA	Master	Initiator
I^2C	Master	Host
	Slave	Client
SPI	Master	Host
	Slave	Client
UART, LIN Mode	Master	Commander
	Slave	Responder
PWM	Master	Host
	Slave	Client

1.0 DEVICE OVERVIEW

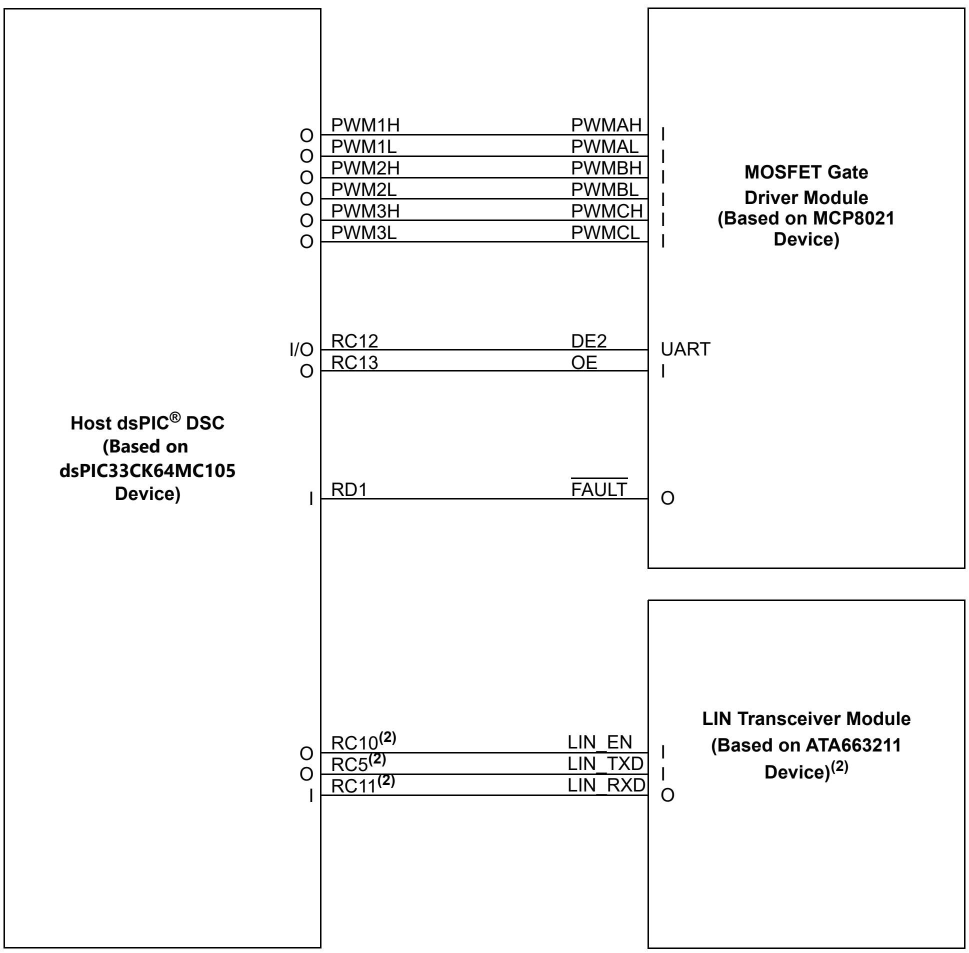
> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive resource. To complement the information in this data sheet, refer to the related section of the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip website (www.microchip.com).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

This document contains device-specific information for the dsPIC33CDVL64MC106 Digital Signal Controller (DSC) and Microcontroller (MCU) devices.

The dsPIC33CDVL64MC106 devices contain extensive Digital Signal Processor (DSP) functionality with a high-performance, 16-bit MCU architecture.

Figure 1-2 shows a general block diagram of the core and peripheral modules of the dsPIC33CDVL64MC106 family. Table 1-2 lists the functions of the various pins shown in the pinout diagrams.

FIGURE 1-1: dsPIC33CDVL64MC106 FAMILY INTERNAL CONNECTIONS BLOCK DIAGRAM



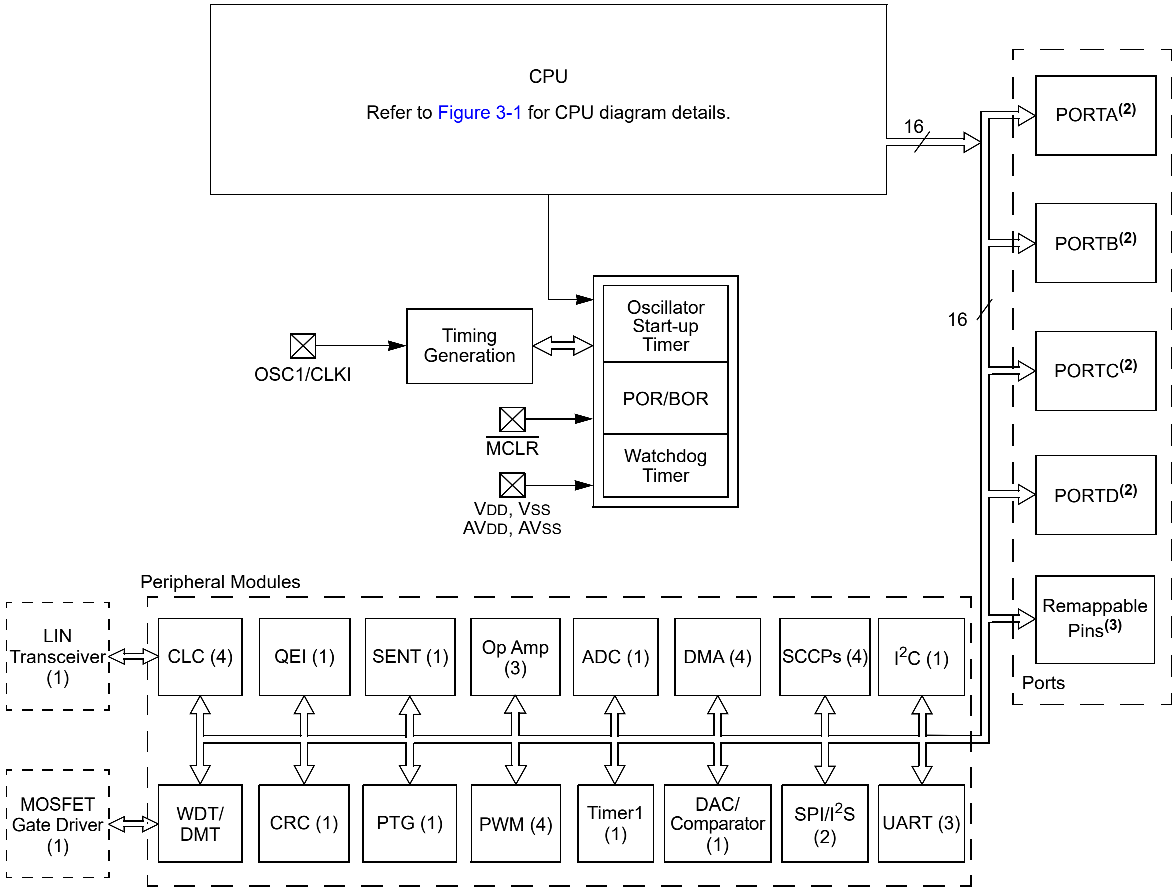
Note 1: Device Interconnections are shown in Table 1-1.
2: Only applies to devices with the LIN transceiver module. Please see Table 1 for more information.

TABLE 1-1: dsPIC33CDVL64MC106 FAMILY INTERCONNECTIONS[1]

Host dsPIC® DSC Connection	MOSFET Gate Driver Connection	External Pin
RB14/PWM1H	PWMAH	10
RB15/PWM1L	PWMAL	No
RB12/PWM2H	PWMBH	54
RB13/PWM2L	PWMBL	No
RB10/PWM3H	PWMCH	53
RB11/PWM3L	PWMCL	No
RD1	FAULT	No
RC12/RP60	DE2	No
RC13	OE	No
Host dsPIC DSC Connection	**LIN Transceiver Connection[2]**	**External Pin**
RC10	LIN_EN	No
RC5/RP53	LIN_TXD	No
RC11/RP59	LIN_RXD	46

Note 1: Interconnect is also bonded to an external device pin.
2: Only applies to devices with LIN Transceiver module.

FIGURE 1-2: dsPIC33CDVL64MC106 FAMILY BLOCK DIAGRAM[(1)]



Note 1: The numbers in the parentheses are the number of instantiations of the module indicated.

2: Not all I/O pins or features are implemented on all device pinout configurations.

3: Some peripheral I/Os are only accessible through Peripheral Pin Select (PPS).

TABLE 1-2: PINOUT I/O DESCRIPTIONS

Pin Name(1)	Pin Type	Buffer Type	PPS	Description
dsPIC® DSC Functions				
AN0-AN15	I	Analog	No	Analog input channels
ANN0	I	Analog	No	Analog negative input
CLKI	I	ST	No	External Clock (EC) source input. Always associated with OSCI pin function.
CLKO	O	—	No	In Configuration bits, it can be set to output the CPU clock. Always associated with OSCO pin function.
OSCI	I	CMOS	No	Oscillator crystal input. Connects to crystal or resonator in Crystal Oscillator mode.
OSCO	I/O	—	No	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
REFCLKI	I	ST	Yes	Reference clock input
REFCLKO	O	—	Yes	Reference clock output
INT0	I	ST	No	External Interrupt 0
INT1	I	ST	Yes	External Interrupt 1
INT2	I	ST	Yes	External Interrupt 2
INT3	I	ST	Yes	External Interrupt 3
IOCA[4:0]	I	ST	No	Interrupt-on-Change input for PORTA
IOCB[15:0]	I	ST	No	Interrupt-on-Change input for PORTB
IOCC[13:0]	I	ST	No	Interrupt-on-Change input for PORTC
IOCD1, IOCD8, IOCD10	I	ST	No	Interrupt-on-Change input for PORTD
IOCD13				
QEIAx	I	ST	Yes	QEIx Input A
QEIBx	I	ST	Yes	QEIx Input B
QEINDXx	I	ST	Yes	QEIx Index input
QEIHOMx	I	ST	Yes	QEIx Home input
QEICMPx	O	—	Yes	QEIx comparator output
RP32-RP61, RP65, RP72, RP77	I/O	ST	Yes	Remappable I/O ports
RA0-RA4	I/O	ST	No	PORTA is a bidirectional I/O port.
RB0-RB15	I/O	ST	No	PORTB is a bidirectional I/O port.
RC0-RC13	I/O	ST	No	PORTC is a bidirectional I/O port.
RD1, RD8, RD10, RD13	I/O	ST	No	PORTD is a bidirectional I/O port.
T1CK	I	ST	Yes	Timer1 external clock input
$\overline{\text{U1CTS}}$	I	ST	Yes	UART1 Clear-to-Send
$\overline{\text{U1RTS}}$	O	—	Yes	UART1 Request-to-Send
U1RX	I	ST	Yes	UART1 Receive
U1TX	O	—	Yes	UART1 Transmit
$\overline{\text{U1DSR}}$	I	ST	Yes	UART1 Data-Set-Ready
$\overline{\text{U1DTR}}$	O	—	Yes	UART1 Data-Terminal-Ready

Legend: CMOS = CMOS compatible input or output; Analog = Analog input; P = Power
ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input
PPS = Peripheral Pin Select

Note 1: Not all pins are available in all package variants. See the **"Pin Diagrams"** section for pin availability.
2: PWM4L and PWM4H pins are available on PPS.
3: Pin is connected to a device interconnect; see Table 1-1 for more information.
4: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name[1]	Pin Type	Buffer Type	PPS	Description
U2CTS	I	ST	Yes	UART2 Clear-to-Send
U2RTS	O	—	Yes	UART2 Request-to-Send
U2RX	I	ST	Yes	UART2 Receive
U2TX	O	—	Yes	UART2 Transmit
U2DSR	I	ST	Yes	UART2 Data-Set-Ready
U2DTR	O	—	Yes	UART2 Data-Terminal-Ready
U3CTS	I	ST	Yes	UART3 Clear-to-Send
U3RTS	O	—	Yes	UART3 Request-to-Send
U3RX	I	ST	Yes	UART3 Receive
U3TX	O	—	Yes	UART3 Transmit
U3DSR	I	ST	Yes	UART3 Data-Set-Ready
U3DTR	O	—	Yes	UART3 Data-Terminal-Ready
SENT1	I	ST	Yes	SENT1 input
SENT1OUT	O	—	Yes	SENT1 output
PTGTRG24	O	—	Yes	PTG Trigger Output 24
PTGTRG25	O	—	Yes	PTG Trigger Output 25
TCKI1-TCKI4	I	ST	Yes	SCCP timer inputs
ICM1-ICM4	I	ST	Yes	SCCP capture inputs
OCFA-OCFB	I	ST	Yes	SCCP Fault inputs
OCM1x-OCM4x	O	—	Yes	SCCP compare outputs
SCK1	I/O	ST	Yes	Synchronous serial clock input/output for SPI1
SDI1	I	ST	Yes	SPI1 data in
SDO1	O	—	Yes	SPI1 data out
SS1	I/O	ST	Yes	SPI1 Slave synchronization or frame pulse I/O
SCK2	I/O	ST	Yes	Synchronous serial clock input/output for SPI2
SDI2	I	ST	Yes	SPI2 data in
SDO2	O	—	Yes	SPI2 data out
SS2	I/O	ST	Yes	SPI2 Slave synchronization or frame pulse I/O
SCL1	I/O	ST	No	Synchronous serial clock input/output for I2C1
SDA1	I/O	ST	No	Synchronous serial data input/output for I2C1
ASCL1	I/O	ST	No	Alternate synchronous serial clock input/output for I2C1
ASDA1	I/O	ST	No	Alternate synchronous serial data input/output for I2C1
PCI8-PCI18	I	ST	Yes	PWM Inputs 8 through 18
PCI19	I	ST	No	PWM Input 19
PWMEA-PWMED	O	—	Yes	PWM Event Outputs A through D
PWM1L-PWM4L[2]	O	—	No	PWM Low Outputs 1 through 4
PWM1H-PWM4H[2]	O	—	No	PWM High Outputs 1 through 4
CLCINA-CLCIND	I	ST	Yes	CLC Inputs A through D
CLCxOUT	O	—	Yes	CLCx Output
CMP1A	I	Analog	No	Comparator Channel 1A
CMP1B	I	Analog	No	Comparator Channel 1B
CMP1C	I	Analog	No	Comparator Channel 1C
CMP1D	I	Analog	No	Comparator Channel 1D

Legend: CMOS = CMOS compatible input or output; Analog = Analog input; P = Power; ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input; PPS = Peripheral Pin Select

Note 1: Not all pins are available in all package variants. See the **"Pin Diagrams"** section for pin availability.

2: PWM4L and PWM4H pins are available on PPS.

3: Pin is connected to a device interconnect; see Table 1-1 for more information.

4: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name(1)	Pin Type	Buffer Type	PPS	Description
DACOUT	O	—	No	DAC output voltage
IBIAS0-IBIAS3 ISRC0-ISRC3	O O	Analog Analog	No No	50 µA Constant-Current Outputs 0 through 3 10 µA Constant-Current Outputs 0 through 3
OA1IN+ OA1IN- OA1OUT OA2IN+ OA2IN- OA2OUT OA3IN+ OA3IN- OA3OUT	I I O I I O I I O	— — — — — — — — —	No No No No No No No No No	Op Amp 1+ input Op Amp 1- input Op Amp 1 output Op Amp 2+ input Op Amp 2- input Op Amp 2 output Op Amp 3+ input Op Amp 3- input Op Amp 3 output
ADTRG31	I	ST	No	External ADC trigger source
PGD1 PGC1 PGD2 PGC2 PGD3 PGC3	I/O I I/O I I/O I	ST ST ST ST ST ST	No No No No No No	Data I/O pin for Programming/Debugging Communication Channel 1 Clock input pin for Programming/Debugging Communication Channel 1 Data I/O pin for Programming/Debugging Communication Channel 2 Clock input pin for Programming/Debugging Communication Channel 2 Data I/O pin for Programming/Debugging Communication Channel 3 Clock input pin for Programming/Debugging Communication Channel 3
$\overline{\text{MCLR}}$	I/P	ST	No	Master Clear (Reset) input. This pin is an active-low Reset to the device.
AVDD	P	P	No	Positive supply for analog modules. This pin must be connected at all times.
AVSS	P	P	No	Ground reference for analog modules. This pin must be connected at all times.
VDD	P	P	No	Positive supply for peripheral logic and I/O pins
VSS	P	P	No	Ground reference for logic and I/O pins

Legend: CMOS = CMOS compatible input or output; Analog = Analog input; P = Power; ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input; PPS = Peripheral Pin Select

Note 1: Not all pins are available in all package variants. See the **"Pin Diagrams"** section for pin availability.
2: PWM4L and PWM4H pins are available on PPS.
3: Pin is connected to a device interconnect; see Table 1-1 for more information.
4: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name[1]	Pin Type	Buffer Type	PPS	Description
MOSFET Gate Driver Module Functions				
WAKE	I			HV digital edge input, device wake-up from Sleep with internal pull-down resistor
PHA	I/O			Phase A high-side MOSFET Driver reference, back-EMF sense input
HSA	O			Phase A high-side N-channel MOSFET Driver, active-high
VBA	P			Phase A high-side MOSFET Driver bias
PHB	I/O			Phase B high-side MOSFET Driver reference, back-EMF sense input
HSB	O			Phase B high-side N-channel MOSFET Driver, active-high
VBB	P			Phase B high-side MOSFET Driver bias
PHC	I/O			Phase C high-side MOSFET Driver reference, back-EMF sense input
HSC	O			Phase C high-side N-channel MOSFET Driver, active-high
VBC	P			Phase C high-side MOSFET driver bias
LSA	O			Phase A low-side N-channel MOSFET Driver, active-high
LSB	O			Phase B low-side N-channel MOSFET Driver, active-high
LSC	O			Phase C low-side N-channel MOSFET Driver, active-high
VBOOT	P			External bootstrap circuit supply voltage output
CAP1[4]	P			Charge Pump Flying Capacitor Input 1
CAP2	P			Charge Pump Flying Capacitor Input 2
HVDD	P			Input supply
VREG	P			Linear Regulator Output: 3.3V
HVSS	P			MOSFET Driver Ground Reference
PWMAH [3]	I			Phase A high-side control, internal 47 kΩ pull-down
PWMAL[3]	I			Phase A low-side control, internal 47 kΩ pull-down
PWMBH[3]	I			Phase B high-side control, internal 47 kΩ pull-down
PWMBL[3]	I			Phase B low-side control, internal 47 kΩ pull-down
PWMCH[3]	I			Phase C high-side control, internal 47 kΩ pull-down
PWMCL[3]	I			Phase C low-side control, internal 47 kΩ pull-down
FAULT[3]	O			Digital output, active-low Fault, open-drain
DE2[3]	UART			Digital communications port, open-drain
OE[3]	I			Digital input, output enable, Fault clearing, internal 47 kΩ pull-down
LIN Transceiver Module Functions (dsPIC33CDVL64MC106 Only)				
LIN_INH	I			LIN Transceiver Inhibit, active high
LIN_BUS	I/O			LIN Communications Bus
LIN_WKin	I			LIN Wake Input
LIN_VDD[3]	P			LIN Transceiver Input Supply
LIN_VSS[3]	P			LIN Transceiver Ground Reference
LIN_EN[3]	I			Digital input, enable signal
LIN_TXD[3]	I			Transmit data input from microcontroller
LIN_RXD[3]	O			Receive data output to microcontroller, use 4.7 kΩ external pull-up and 20 pF load capacitor

Legend: CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; PPS = Peripheral Pin Select; Analog = Analog input; O = Output; P = Power; I = Input

Note 1: Not all pins are available in all package variants. See the **"Pin Diagrams"** section for pin availability.

2: PWM4L and PWM4H pins are available on PPS.

3: Pin is connected to a device interconnect; see Table 1-1 for more information.

4: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

NOTES:

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT DIGITAL SIGNAL CONTROLLERS

2.1 Basic Connection Requirements

Getting started with the dsPIC33CDVL64MC106 family devices requires attention to a minimal set of device pin connections before proceeding with development. The following is a list of pin names which must always be connected:

- All VDD and VSS pins (see **Section 2.2 "Decoupling Capacitors"**)
- All AVDD and AVSS pins regardless if the ADC module is not used (see **Section 2.2 "Decoupling Capacitors"**)
- $\overline{\text{MCLR}}$ pin (see **Section 2.3 "Master Clear (MCLR) Pin"**)
- PGCx/PGDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.4 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used (see **Section 2.5 "External Oscillator Pins"**)
- LIN transceiver pins INH and WAKE and MOSFET Gate Driver VREG output pin to ensure a WAKE event is detected when application has been put into Sleep mode (see **Section 2.6 "MOSFET Gate Driver Sleep Mode Requirements"**)

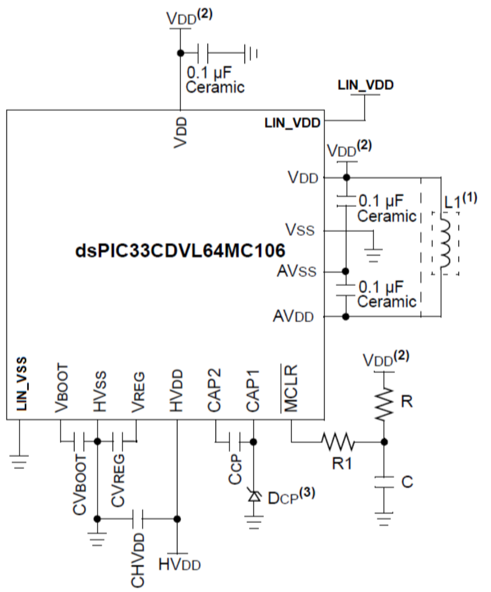
2.2 Decoupling Capacitors

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** Recommendation of 0.1 µF (100 nF), 10-20V. This capacitor should be a low-ESR and have resonance frequency in the range of 20 MHz and higher. It is recommended to use ceramic capacitors.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise, above tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 µF in parallel with 0.001 µF.
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB track inductance.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTION



Note 1: As an option, instead of a hard-wired connection, an inductor (L1) can be substituted between VDD and AVDD to improve ADC noise rejection. The inductor impedance should be less than 1Ω and the inductor capacity greater than 10 mA.

2: VDD/AVDD pins may be powered by either an external power supply or by the 3.3V VREG output.

3: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

Where:

$$f = \frac{F_{CNV}}{2}$$ (i.e., ADC conversion rate/2)

$$f = \frac{1}{(2\pi\sqrt{LC})}$$

$$L = \left(\frac{1}{(2\pi f\sqrt{C})}\right)^2$$

2.2.1 BULK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a bulk capacitor for integrated circuits, including DSCs, to supply a local power source. The value of the bulk capacitor should be determined based on the trace resistance that connects the power supply source to the device and the maximum current drawn by the device in the application. In other words, select the bulk capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

2.3 Master Clear (MCLR) Pin

The MCLR pin provides two specific device functions:

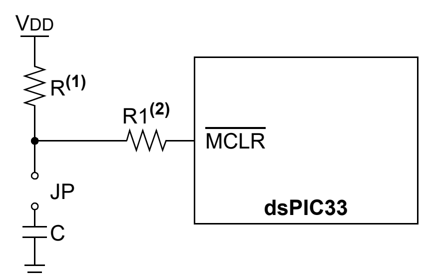
- Device Reset
- Device Programming and Debugging

During device programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the MCLR pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R and C will need to be adjusted based on the application and PCB requirements.

For example, as shown in Figure 2-2, it is recommended that the capacitor, C, be isolated from the MCLR pin during programming and debugging operations.

Place the components, as shown in Figure 2-2, within one-quarter inch (6 mm) from the MCLR pin.

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



Note 1: R ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the MCLR pin VIH and VIL specifications are met.

2: R1 ≤ 470Ω will limit any current flowing into MCLR from the external capacitor, C, in the event of MCLR pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the MCLR pin VIH and VIL specifications are met.

2.4 ICSP Pins

The PGCx and PGDx pins are used for ICSP and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ohms, not to exceed 100 Ohms.

Pull-up resistors, series diodes and capacitors on the PGCx and PGDx pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits and pin Voltage Input High (V_{IH}) and Voltage Input Low (V_{IL}) requirements.

Ensure that the “Communication Channel Select” (i.e., PGCx/PGDx pins) programmed into the device matches the physical connections for the ICSP to MPLAB® debugger tool.

For more information on the MPLAB programmer/ debugger connection requirements, refer to the Microchip website.

2.5 External Oscillator Pins

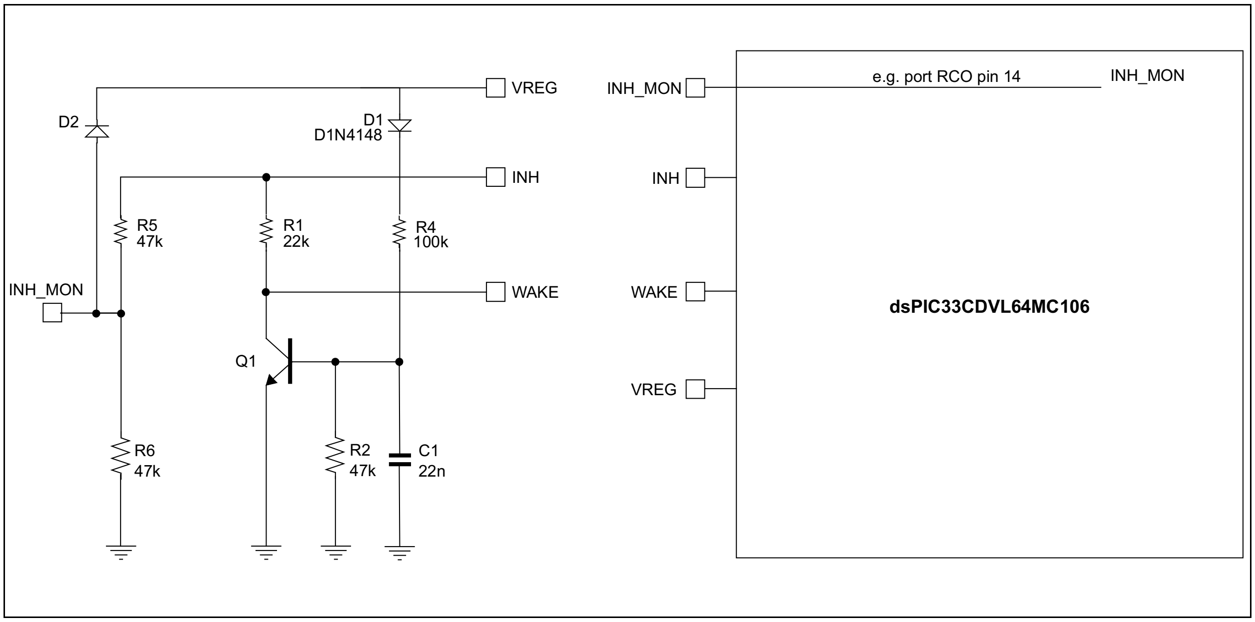
When the Primary Oscillator (POSC) circuit is used to connect a crystal oscillator, special care and consideration is required to ensure proper operation. The POSC circuit should be tested across the environmental conditions that the end product is intended to be used. The load capacitors specified in the crystal oscillator data sheet can be used as a starting point, however, the parasitic capacitance from the PCB traces can affect the circuit and the values may need to be altered to ensure proper start-up and operation.

Excessive trace length and other physical interaction can lead to poor signal quality. Poorly tuned oscillator circuits can have reduced amplitude, incorrect frequency (runt pulses), distorted waveforms and long start-up times that may result in unpredictable application behavior, such as instruction misexecution, illegal op code fetch, etc. Ensure that the crystal oscillator circuit is at full amplitude and correct frequency before the system begins to execute code. In planning the application’s routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator do not have high frequencies, short rise and fall times and other similar noise. For further information on the Primary Oscillator see **Section 9.4 “Internal Fast RC (FRC) Oscillator”**.

2.6 MOSFET Gate Driver Sleep Mode Requirements

In the case of utilizing the VREG output of the MOSFET Gate Driver module to power the host processor, and relying on LIN Transceiver BUS activity to generate a WAKE-up event if the MOSFET Gate Driver has been put to Sleep mode, additional circuitry (Figure 2-3) is required to ensure the timing requirements for a WAKE event are met. The circuit in Figure 2-3 delays any incoming rising edge on the WAKE pin >= t_SLEEP + tWAIT_SETUP in Table 35-1.

FIGURE 2-3: WAKE EXTENDER



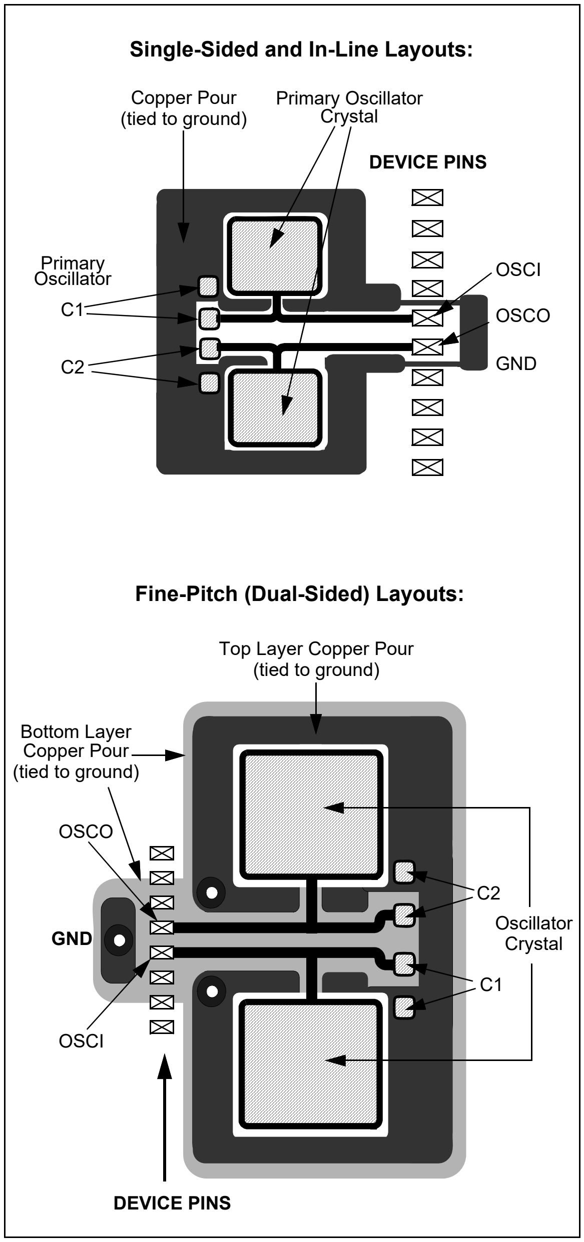
2.7 External Oscillator Layout Guidance

Use best practices during PCB layout to ensure robust start-up and operation. The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. If using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is shown in Figure 2-4.

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the Microchip website (www.microchip.com):

- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*
- *AN1798, "Crystal Selection for Low-Power Secondary Oscillator*

FIGURE 2-4: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



2.8 Oscillator Value Conditions on Device Start-up

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to a certain frequency (see **Section 9.0 "Oscillator with High-Frequency PLL"**) to comply with device PLL start-up conditions. This means that if the external oscillator frequency is outside this range, the application must start up in the FRC mode first. The default PLL settings after a POR with an oscillator frequency outside this range will violate the device operating speed.

Once the device powers up, the application firmware can initialize the PLL SFRs, CLKDIV and PLLFBD to a suitable value, and then perform a clock switch to the Oscillator + PLL clock source. Note that clock switching must be enabled in the device Configuration Word.

2.9 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a Logic Low state.

Alternatively, connect a 1k to 10k resistor between VSS and unused pins, and drive the output to logic low.

3.0 CPU

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Enhanced CPU"** (www.microchip.com/DS70005158).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33CDVL64MC106 family CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set, including significant support for Digital Signal Processing (DSP). The CPU has a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M x 24 bits of user program memory space.

An instruction prefetch mechanism helps maintain throughput and provides predictable execution. Most instructions execute in a single-cycle effective execution rate, with the exception of instructions that change the program flow, the double-word move (`MOV.D`) instruction, PSV accesses and the table instructions. Overhead-free program loop constructs are supported using the `DO` and `REPEAT` instructions, both of which are interruptible at any point.

3.1 Registers

The dsPIC33CDVL64MC106 devices have sixteen, 16-bit Working registers in the programmer's model. Each of the Working registers can act as a Data, Address or Address Offset register. The 16th Working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

In addition, the dsPIC33CDVL64MC106 devices include four Alternate Working register sets, which consist of W0 through W14. The Alternate Working registers can be made persistent to help reduce the saving and restoring of register content during Interrupt Service Routines (ISRs). The Alternate Working registers can be assigned to a specific Interrupt Priority Level (IPL1 through IPL6) by configuring the CTXTx[2:0] bits in the FALTREG Configuration register. The Alternate Working registers can also be accessed manually by using the `CTXTSWP` instruction. The CCTXI[2:0] and MCTXI[2:0] bits in the CTXTSTAT register can be used to identify the current, and most recent, manually selected Working register sets.

3.2 Instruction Set

The instruction set for the dsPIC33CDVL64MC106 family has two classes of instructions: the MCU class of instructions and the DSP class of instructions. These two instruction classes are seamlessly integrated into the architecture and execute from a single execution unit. The instruction set includes many addressing modes and was designed for optimum C compiler efficiency.

3.3 Data Space Addressing

The base Data Space can be addressed as up to 4K words or 8 Kbytes, and is split into two blocks, referred to as X and Y data memory. Each memory block has its own independent Address Generation Unit (AGU). The MCU class of instructions operates solely through the X memory AGU, which accesses the entire memory map as one linear Data Space. Certain DSP instructions operate through the X and Y AGUs to support dual operand reads, which split the data address space into two parts. The X and Y Data Space boundary is device-specific.

The upper 32 Kbytes of the Data Space memory map can optionally be mapped into Program Space (PS) at any 16K program word boundary. The Program-to-Data Space mapping feature, known as Program Space Visibility (PSV), lets any instruction access Program Space as if it were Data Space. Refer to **"Data Memory"** (www.microchip.com/DS70595) for more details on PSV and table accesses.

On the dsPIC33CDVL64MC106 devices, overhead-free circular buffers (Modulo Addressing) are supported in both X and Y address spaces. The Modulo Addressing removes the software boundary checking overhead for DSP algorithms. The X AGU Circular Addressing can be used with any of the MCU class of instructions. The X AGU also supports Bit-Reversed Addressing to greatly simplify input or output data re-ordering for radix-2 FFT algorithms.

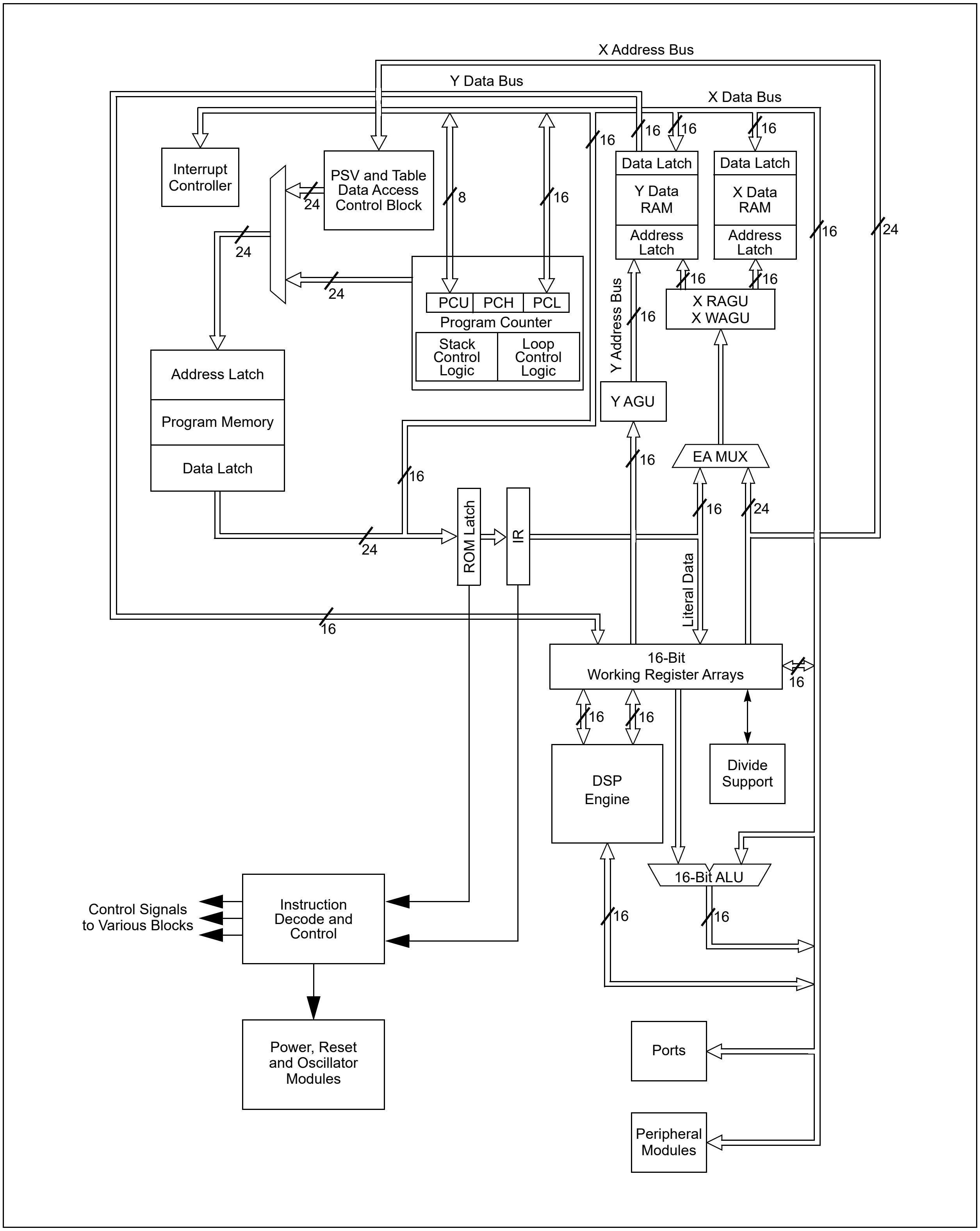
3.4 Addressing Modes

The CPU supports these addressing modes:

- Inherent (no operand)
- Relative
- Literal
- Memory Direct
- Register Direct
- Register Indirect

Each instruction is associated with a predefined addressing mode group, depending upon its functional requirements. As many as six addressing modes are supported for each instruction.

FIGURE 3-1: dsPIC33CDVL64MC106 FAMILY CPU BLOCK DIAGRAM



3.4.1 PROGRAMMER'S MODEL

The programmer's model for the dsPIC33CDVL64MC106 devices is shown in Figure 3-2. All registers in the programmer's model are memory-mapped and can be manipulated directly by instructions. Table 3-1 lists a description of each register.

In addition to the registers contained in the programmer's model, the dsPIC33CDVL64MC106 devices contain control registers for Modulo Addressing, Bit-Reversed Addressing and interrupts. These registers are described in subsequent sections of this document.

All registers associated with the programmer's model are memory-mapped, as shown in Figure 3-2.

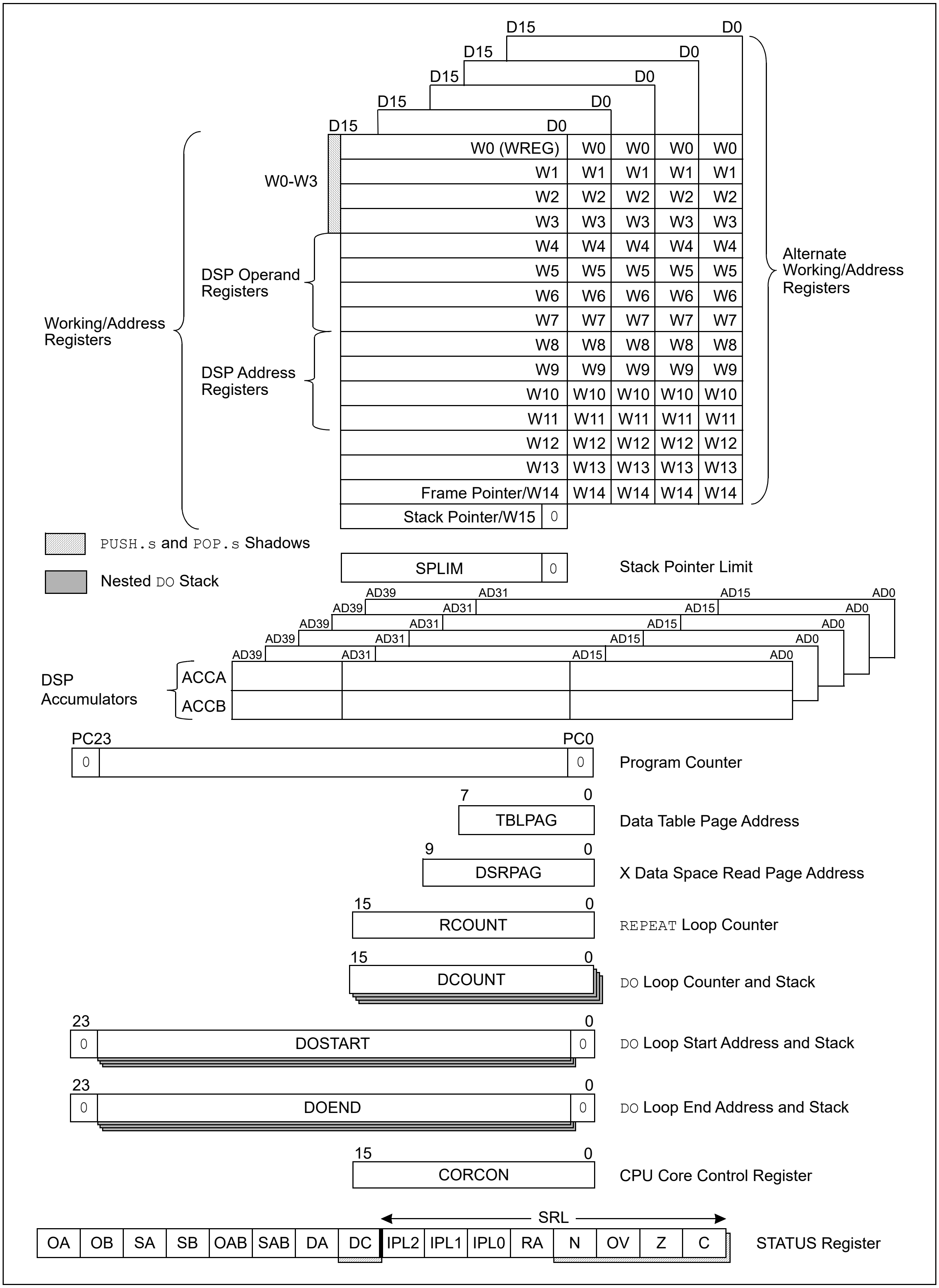
TABLE 3-1: PROGRAMMER'S MODEL REGISTER DESCRIPTIONS

Register(s) Name	Description
W0 through W15[(1)]	Working Register Array
W0 through W14[(1)]	Alternate Working Register Array 1
W0 through W14[(1)]	Alternate Working Register Array 2
W0 through W14[(1)]	Alternate Working Register Array 3
W0 through W14[(1)]	Alternate Working Register Array 4
ACCA, ACCB	40-Bit DSP Accumulators (Additional Four Alternate Accumulators)
PC	23-Bit Program Counter
SR	ALU and DSP Engine STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
DSRPAG	Extended Data Space (EDS) Read Page Register
RCOUNT	`REPEAT` Loop Counter Register
DCOUNT	`DO` Loop Counter Register
DOSTARTH, DOSTARTL[(2)]	`DO` Loop Start Address Register (High and Low)
DOENDH, DOENDL	`DO` Loop End Address Register (High and Low)
CORCON	Contains DSP Engine, `DO` Loop Control and Trap Status bits

Note 1: Memory-mapped W0 through W14 represent the value of the register in the currently active CPU context.

2: The DOSTARTH and DOSTARTL registers are read-only.

FIGURE 3-2: PROGRAMMER'S MODEL



3.4.2 CPU RESOURCES

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

3.4.2.1 Key Resources

- **"Enhanced CPU"** (www.microchip.com/DS70005158)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

3.4.3 CPU CONTROL REGISTERS

REGISTER 3-1: SR: CPU STATUS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/C-0	R/C-0	R-0	R/W-0
OA	OB	SA(3)	SB(3)	OAB	SAB	DA	DC
bit 15							bit 8

R/W-0(2)	R/W-0(2)	R/W-0(2)	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2(1)	IPL1(1)	IPL0(1)	RA	N	OV	Z	C
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’= Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **OA:** Accumulator A Overflow Status bit
1 = Accumulator A has overflowed
0 = Accumulator A has not overflowed

bit 14 **OB:** Accumulator B Overflow Status bit
1 = Accumulator B has overflowed
0 = Accumulator B has not overflowed

bit 13 **SA:** Accumulator A Saturation ‘Sticky’ Status bit(3)
1 = Accumulator A is saturated or has been saturated at some time
0 = Accumulator A is not saturated

bit 12 **SB:** Accumulator B Saturation ‘Sticky’ Status bit(3)
1 = Accumulator B is saturated or has been saturated at some time
0 = Accumulator B is not saturated

bit 11 **OAB:** OA || OB Combined Accumulator Overflow Status bit
1 = Accumulator A or B has overflowed
0 = Neither Accumulator A or B has overflowed

bit 10 **SAB:** SA || SB Combined Accumulator ‘Sticky’ Status bit
1 = Accumulator A or B is saturated or has been saturated at some time
0 = Neither Accumulator A or B is saturated

bit 9 **DA:** DO Loop Active bit
1 = DO loop is in progress
0 = DO loop is not in progress

bit 8 **DC:** MCU ALU Half Carry/$\overline{\text{Borrow}}$ bit
1 = A carry-out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry-out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred

Note 1: The IPL[2:0] bits are concatenated with the IPL[3] bit (CORCON[3]) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL, if IPL[3] = 1. User interrupts are disabled when IPL[3] = 1.

2: The IPL[2:0] Status bits are read-only when the NSTDIS bit (INTCON1[15]) = 1.

3: A data write to the SR register can modify the SA and SB bits by either a data write to SA and SB or by clearing the SAB bit. To avoid a possible SA or SB bit write race condition, the SA and SB bits should not be modified using bit operations.

REGISTER 3-1: SR: CPU STATUS REGISTER (CONTINUED)

bit 7-5 **IPL[2:0]:** CPU Interrupt Priority Level Status bits(1,2)
111 = CPU Interrupt Priority Level is 7 (15); user interrupts are disabled
110 = CPU Interrupt Priority Level is 6 (14)
101 = CPU Interrupt Priority Level is 5 (13)
100 = CPU Interrupt Priority Level is 4 (12)
011 = CPU Interrupt Priority Level is 3 (11)
010 = CPU Interrupt Priority Level is 2 (10)
001 = CPU Interrupt Priority Level is 1 (9)
000 = CPU Interrupt Priority Level is 0 (8)

bit 4 **RA:** REPEAT Loop Active bit
1 = REPEAT loop is in progress
0 = REPEAT loop is not in progress

bit 3 **N:** MCU ALU Negative bit
1 = Result was negative
0 = Result was non-negative (zero or positive)

bit 2 **OV:** MCU ALU Overflow bit
This bit is used for signed arithmetic (two's complement). It indicates an overflow of the magnitude that causes the sign bit to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

bit 1 **Z:** MCU ALU Zero bit
1 = An operation that affects the Z bit has set it at some time in the past
0 = The most recent operation that affects the Z bit has cleared it (i.e., a non-zero result)

bit 0 **C:** MCU ALU Carry/$\overline{\text{Borrow}}$ bit
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

Note 1: The IPL[2:0] bits are concatenated with the IPL[3] bit (CORCON[3]) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL, if IPL[3] = 1. User interrupts are disabled when IPL[3] = 1.
2: The IPL[2:0] Status bits are read-only when the NSTDIS bit (INTCON1[15]) = 1.
3: A data write to the SR register can modify the SA and SB bits by either a data write to SA and SB or by clearing the SAB bit. To avoid a possible SA or SB bit write race condition, the SA and SB bits should not be modified using bit operations.

REGISTER 3-2: CORCON: CORE CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
VAR	—	US1	US0	EDT[(1)]	DL2	DL1	DL0
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3[(2)]	SFA	RND	IF
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **VAR:** Variable Exception Processing Latency Control bit
1 = Variable exception processing is enabled
0 = Fixed exception processing is enabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13-12 **US[1:0]:** DSP Multiply Unsigned/Signed Control bits
11 = Reserved
10 = DSP engine multiplies are mixed sign
01 = DSP engine multiplies are unsigned
00 = DSP engine multiplies are signed

bit 11 **EDT:** Early DO Loop Termination Control bit[(1)]
1 = Terminates executing DO loop at the end of the current loop iteration
0 = No effect

bit 10-8 **DL[2:0]:** DO Loop Nesting Level Status bits
111 = Seven DO loops are active
...
001 = One DO loop is active
000 = Zero DO loops are active

bit 7 **SATA:** ACCA Saturation Enable bit
1 = Accumulator A saturation is enabled
0 = Accumulator A saturation is disabled

bit 6 **SATB:** ACCB Saturation Enable bit
1 = Accumulator B saturation is enabled
0 = Accumulator B saturation is disabled

bit 5 **SATDW:** Data Space Write from DSP Engine Saturation Enable bit
1 = Data Space write saturation is enabled
0 = Data Space write saturation is disabled

bit 4 **ACCSAT:** Accumulator Saturation Mode Select bit
1 = 9.31 saturation (super saturation)
0 = 1.31 saturation (normal saturation)

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit 3[(2)]
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

Note 1: This bit is always read as ‘0’.
2: The IPL3 bit is concatenated with the IPL[2:0] bits (SR[7:5]) to form the CPU Interrupt Priority Level.

REGISTER 3-2: CORCON: CORE CONTROL REGISTER (CONTINUED)

bit 2 **SFA:** Stack Frame Active Status bit
1 = Stack frame is active; W14 and W15 address 0x0000 to 0xFFFF, regardless of DSRPAG
0 = Stack frame is not active; W14 and W15 address the base Data Space

bit 1 **RND:** Rounding Mode Select bit
1 = Biased (conventional) rounding is enabled
0 = Unbiased (convergent) rounding is enabled

bit 0 **IF:** Integer or Fractional Multiplier Mode Select bit
1 = Integer mode is enabled for DSP multiply
0 = Fractional mode is enabled for DSP multiply

Note 1: This bit is always read as '0'.
2: The IPL3 bit is concatenated with the IPL[2:0] bits (SR[7:5]) to form the CPU Interrupt Priority Level.

REGISTER 3-3: CTXTSTAT: CPU W REGISTER CONTEXT STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	CCTXI[2:0]		
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	MCTXI[2:0]		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **CCTXI[2:0]:** Current (W Register) Context Identifier bits
111 = Reserved
...
100 = Alternate Working Register Set 4 is currently in use
011 = Alternate Working Register Set 3 is currently in use
010 = Alternate Working Register Set 2 is currently in use
001 = Alternate Working Register Set 1 is currently in use
000 = Default Working Register set is currently in use

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **MCTXI[2:0]:** Manual (W Register) Context Identifier bits
111 = Reserved
...
100 = Alternate Working Register Set 4 was most recently manually selected
011 = Alternate Working Register Set 3 was most recently manually selected
010 = Alternate Working Register Set 2 was most recently manually selected
001 = Alternate Working Register Set 1 was most recently manually selected
000 = Default Working Register set was most recently manually selected

REGISTER 3-4: MSTRPR: EDS BUS MASTER PRIORITY CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	DMAPR	—	—	—	—	NVMPR
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5 **DMAPR:** Modify DMA Controller Bus Master Priority Relative to CPU bit
1 = Raises DMA Controller bus Master priority to above that of the CPU
0 = No change to DMA Controller bus Master priority

bit 4-1 **Unimplemented:** Read as ‘0’

bit 0 **NVMPR:** Modify NVM Controller Bus Master Priority Relative to CPU bit
1 = Raises NVM Controller bus Master priority to above that of the CPU
0 = No change to NVM Controller bus Master priority

3.4.4 ARITHMETIC LOGIC UNIT (ALU)

The dsPIC33CDVL64MC106 ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. Depending on the operation, the ALU can affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

Refer to the *"16-Bit MCU and DSC Programmer's Reference Manual"* (www.microchip.com/DS70000157) for information on the SR bits affected by each instruction.

The core CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.4.4.1 Multiplier

Using the high-speed, 17-bit x 17-bit multiplier, the ALU supports unsigned, signed or mixed-sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit signed x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.4.4.2 Divider

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 32-bit signed/16-bit signed divide
- 32-bit unsigned/16-bit unsigned divide
- 16-bit signed/16-bit signed divide
- 16-bit unsigned/16-bit unsigned divide

The 16-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn) and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute. There are additional instructions: `DIV2` and `DIVF2`. Divide instructions will complete in six cycles.

3.4.5 DSP ENGINE

The DSP engine consists of a high-speed 17-bit x 17-bit multiplier, a 40-bit barrel shifter and a 40-bit adder/subtracter (with two target accumulators, round and saturation logic).

The DSP engine can also perform inherent accumulator-to-accumulator operations that require no additional data. These instructions are, `ADD`, `SUB`, `NEG`, `MIN` and `MAX`.

The DSP engine has options selected through bits in the CPU Core Control register (CORCON), as listed below:

- Fractional or integer DSP multiply (IF)
- Signed, unsigned or mixed-sign DSP multiply (USx)
- Conventional or convergent rounding (RND)
- Automatic saturation on/off for ACCA (SATA)
- Automatic saturation on/off for ACCB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation mode selection (ACCSAT)

TABLE 3-2: DSP INSTRUCTIONS SUMMARY

Instruction	Algebraic Operation	ACC Write-Back
CLR	$A = 0$	Yes
ED	$A = (x - y)^2$	No
EDAC	$A = A + (x - y)^2$	No
MAC	$A = A + (x \bullet y)$	Yes
MAC	$A = A + x^2$	No
MOVSAC	No change in A	Yes
MPY	$A = x \bullet y$	No
MPY	$A = x^2$	No
MPY.N	$A = -x \bullet y$	No
MSC	$A = A - x \bullet y$	Yes

4.0 MEMORY ORGANIZATION

Note:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"dsPIC33/PIC24 Program Memory"** (www.microchip.com/DS70000613).

The dsPIC33CDVL64MC106 architecture features separate program and data memory spaces, and buses. This architecture also allows the direct access of program memory from the Data Space (DS) during code execution.

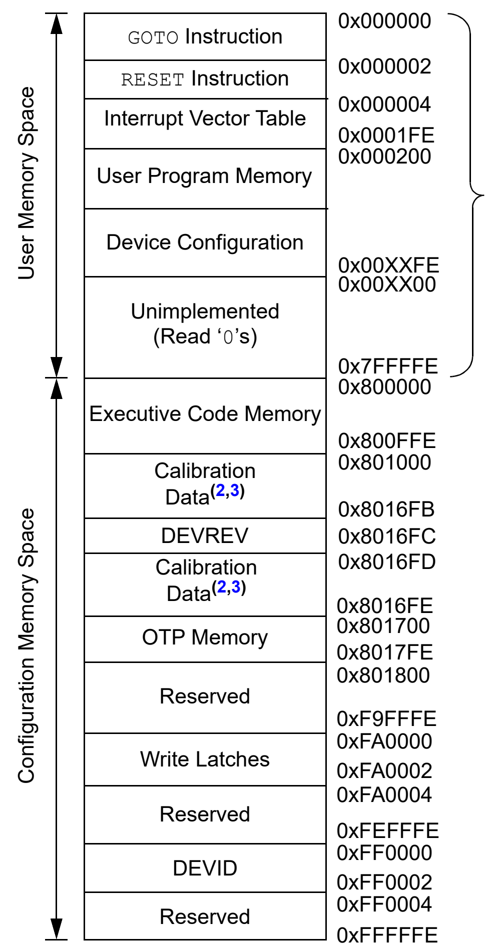
4.1 Program Address Space

The program address memory space of the dsPIC33CDVL64MC106 devices is 4M instructions. The space is addressable by a 24-bit value derived either from the 23-bit PC during program execution, or from table operation or Data Space remapping, as described in **Section 4.5.5 "Interfacing Program and Data Memory Spaces"**.

User application access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFF). The exception is the use of `TBLRD` operations, which use TBLPAG[7] to permit access to calibration data and Device ID sections of the configuration memory space.

The code memory map for the dsPIC33CDVL64MC106 devices is shown in Figure 4-1.

FIGURE 4-1: CODE MEMORY MAP FOR dsPIC33CDVL64MC106 DEVICES(1)



Note 1: Memory areas are not shown to scale.
2: Calibration data area must be maintained during programming.
3: Calibration data area includes UDID and ICSP™ Write Inhibit registers locations.

4.1.1 PROGRAM MEMORY ORGANIZATION

The program memory space is organized in word-addressable blocks. Although it is treated as 24 bits wide, it is more appropriate to think of each address of the program memory as a lower and upper word, with the upper byte of the upper word being unimplemented. The lower word always has an even address, while the upper word has an odd address (Figure 4-2).

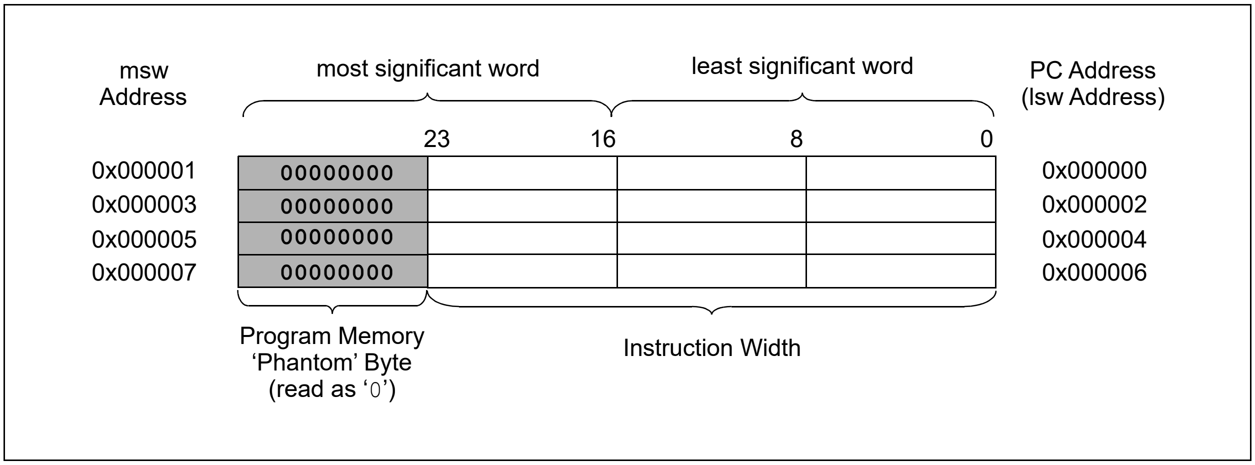
Program memory addresses are always word-aligned on the lower word, and addresses are incremented or decremented, by two, during code execution. This arrangement provides compatibility with data memory space addressing and makes data in the program memory space accessible.

4.1.2 INTERRUPT AND TRAP VECTORS

The dsPIC33CDVL64MC106 devices reserve the addresses between 0x000000 and 0x000200 for hard-coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A `GOTO` instruction is programmed by the user application at address, 0x000000, of Flash memory, with the actual address for the start of code at address, 0x000002, of Flash memory.

A more detailed discussion of the Interrupt Vector Tables (IVTs) is provided in **Section 7.0 "Interrupt Controller"**.

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.1.3 UNIQUE DEVICE IDENTIFIER (UDID)

The dsPIC33CDVL64MC106 family is individually encoded during final manufacturing with a Unique Device Identifier or UDID. The UDID cannot be erased by a bulk erase command or any other user-accessible means. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is a requirement. It may also be used by the application manufacturer for any number of things that may require unique identification, such as:

- Tracking the device
- Unique serial number
- Unique security key

The UDID comprises five 24-bit program words. When taken together, these fields form a unique 120-bit identifier.

The UDID is stored in five read-only locations, located between 0x801200 and 0x801208 in the device configuration space. Table 4-1 lists the addresses of the identifier words and shows their contents.

TABLE 4-1: UDID ADDRESSES

UDID	Address	Description
UDID1	0x801200	UDID Word 1
UDID2	0x801202	UDID Word 2
UDID3	0x801204	UDID Word 3
UDID4	0x801206	UDID Word 4
UDID5	0x801208	UDID Word 5

4.2 Data Address Space

The dsPIC33CDVL64MC106 CPU has a separate 16-bit wide data memory space. The Data Space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory map is shown in Figure 4-3.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the Data Space. This arrangement gives a base Data Space address range of 64 Kbytes or 32K words.

The lower half of the data memory space (i.e., when EA[15] = `0`) is used for implemented memory addresses, while the upper half (EA[15] = `1`) is reserved for the Program Space Visibility (PSV).

The dsPIC33CDVL64MC106 devices implement up to 16 Kbytes of data memory. If an EA points to a location outside of this area, an all-zero word or byte is returned.

4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data are aligned in data memory and registers as 16-bit words, but all Data Space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® MCU devices and improve Data Space memory usage efficiency, the dsPIC33CDVL64MC106 instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] results in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

A data byte read, reads the complete word that contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register that matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap is generated. If the error occurred on a read, the instruction underway is completed. If the error occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the LSB; the MSB is not modified.

A Sign-Extend (`SE`) instruction is provided to allow user applications to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, user applications can clear the MSB of any W register by executing a Zero-Extend (`ZE`) instruction on the appropriate address.

4.2.3 SFR SPACE

The first 4 Kbytes of the Near Data Space, from 0x0000 to 0x0FFF, is primarily occupied by Special Function Registers (SFRs). These are used by the dsPIC33CDVL64MC106 core and peripheral modules for controlling the operation of the device.

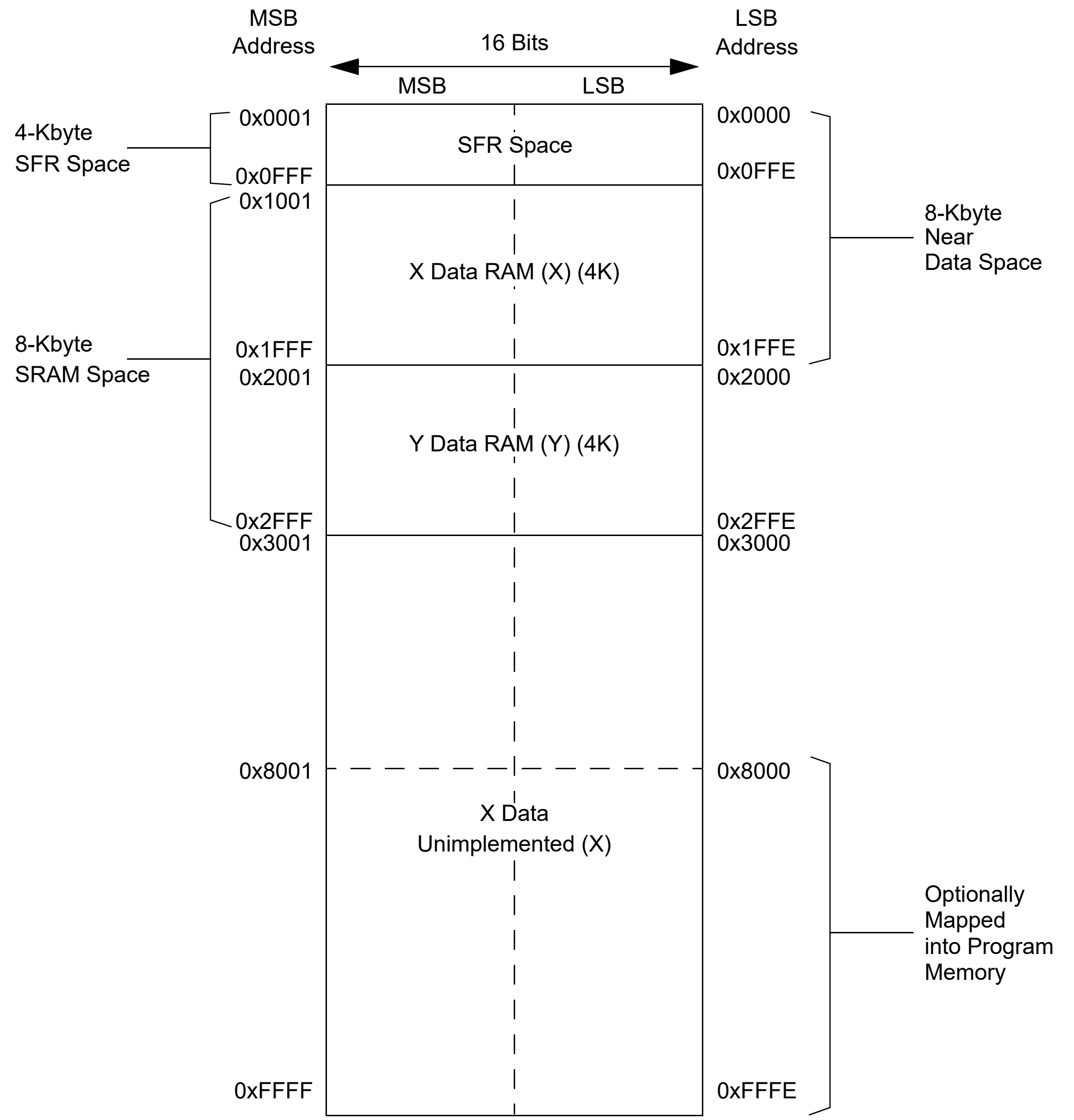
SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as ‘0’.

Note:	The actual set of peripheral features and interrupts varies by the device. Refer to the corresponding device tables and pinout diagrams for device-specific information.

4.2.4 NEAR DATA SPACE

The 8-Kbyte area, between 0x0000 and 0x1FFF, is referred to as the Near Data Space. Locations in this space are directly addressable through a 13-bit absolute address field within all memory direct instructions. Additionally, the whole Data Space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a Working register as an Address Pointer.

FIGURE 4-3: DATA MEMORY MAP FOR dsPIC33CDVL64MC106 FAMILY



Note: Memory areas are not shown to scale.

4.2.5 X AND Y DATA SPACES

The dsPIC33CDVL64MC106 core has two Data Spaces: X and Y. These Data Spaces can be considered either separate (for some DSP instructions) or as one unified linear address range (for MCU instructions). The Data Spaces are accessed using two Address Generation Units (AGUs) and separate data paths. This feature allows certain instructions to concurrently fetch two words from RAM, thereby enabling efficient execution of DSP algorithms, such as Finite Impulse Response (FIR) filtering and Fast Fourier Transform (FFT).

The X Data Space is used by all instructions and supports all addressing modes. X Data Space has separate read and write data buses. The X read data bus is the read data path for all instructions that view Data Space as combined X and Y address space. It is also the X data prefetch path for the dual operand DSP instructions (`MAC` class).

The Y Data Space is used in concert with the X Data Space by the `MAC` class of instructions (`CLR`, `ED`, `EDAC`, `MAC`, `MOVSAC`, `MPY`, `MPY.N` and `MSC`) to provide two concurrent data read paths.

Both the X and Y Data Spaces support Modulo Addressing mode for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing mode is only supported for writes to X Data Space.

All data memory writes, including in DSP instructions, view Data Space as combined X and Y address space. The boundary between the X and Y Data Spaces is device-dependent and is not user-programmable.

4.3 BIST Overview

The dsPIC33CDVL64MC106 devices feature a data memory Built-In Self-Test (BIST) that has the option to be run at start-up or run time. The memory test checks that all memory locations are functional and provides a pass/fail status of the RAM that can be used by software to take action if needed. If a failure is reported, the specific location(s) are not identified.

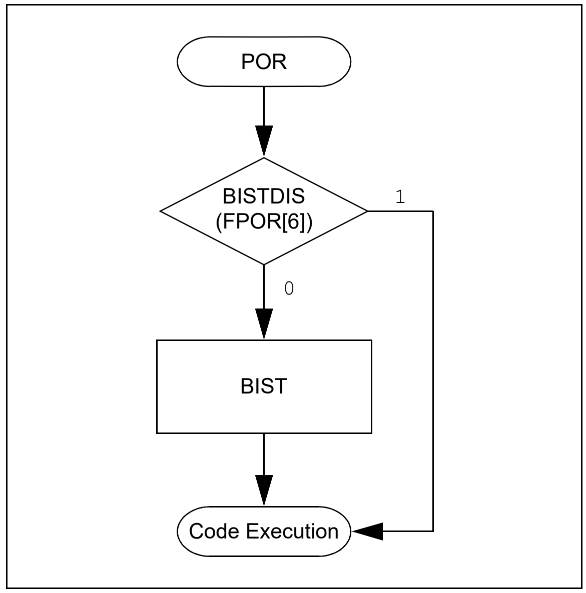
The MBISTCON register (Register 4-1) contains control and status bits for BIST operation. The MBISTDONE bit (MBISTCON[7]) indicates if a BIST was run since the last Reset and the MBISTSTAT bit (MBISTCON[4]) provides the pass/fail result.

The BIST feature operates with a clock of FRC+PLL with PLL settings forced by hardware to result in a 125 MHz clock rate, at both start-up and run time.

4.3.1 BIST AT START-UP

The BIST can be configured to automatically run on a POR-type Reset, as shown in Figure 4-4. By default, when BISTDIS (FPOR[6]) = `1`, the BIST is disabled and will not be part of device start-up. If the BISTDIS bit is cleared during device programming, the BIST will run after all Configuration registers have been loaded and before code execution begins.

FIGURE 4-4: BIST FLOWCHART



4.3.2 BIST AT RUN TIME

A BIST test can be requested to run on subsequent device Resets at any time.

A BIST will corrupt all of the RAM contents, including the Stack Pointer, and requires a subsequent Reset. The system should be prepared for a Reset before a BIST is performed. The BIST is invoked by setting the MBISTEN bit (MBISTCON[0]) and executing a Reset. The MBISTCON register is protected against accidental writes and requires an unlock sequence prior to writing. Only one bit can be set per unlock sequence. The procedure for a run-time BIST is as follows:

1. Execute the unlock sequence by consecutively writing 0x55 and 0xAA to the NVMKEY register.
2. Write 0x0001 to the MBISTCON SFR.
3. Execute a software `RESET` command.
4. Verify a Software Reset has occurred by reading SWR (RCON[6]) (optional).
5. Verify that the MBISTDONE bit is set.
6. Take action depending on test result indicated by MBISTSTAT.

4.3.3 FAULT SIMULATION

A mechanism is available to simulate a BIST failure to allow testing of Fault handling software. When the FLTINJ bit is set during a run-time BIST, the MBISTSTAT bit will be set regardless of the test result. The procedure for a BIST Fault simulation is as follows:

1. Execute the unlock sequence by consecutively writing 0x55 and 0xAA to the NVMKEY register.
2. Set the MBISTEN bit (MBISTCON[0]).
3. Execute 2nd unlock sequence by consecutively writing 0x55 and 0xAA to the NVMKEY register.
4. Set the FLTINJ bit (MBISTCON[8]).
5. Execute a software `RESET` command.
6. Verify the MBISTDONE, MBISTSTAT and FLTINJ bits are all set.

REGISTER 4-1: MBISTCON: MBIST CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0[(1)]
—	—	—	—	—	—	—	FLTINJ
bit 15							bit 8

R/W/HS-0[(1)]	U-0	U-0	R-0	U-0	U-0	U-0	R/W/HC-0[(2)]
MBISTDONE	—	—	MBISTSTAT	—	—	—	MBISTEN
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HC = Hardware Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **FLTINJ:** MBIST Fault Inject Control bit[(1)]
- 1 = The MBIST test will complete and sets MBISTSTAT = 1, simulating an SRAM test failure
- 0 = The MBIST test will execute normally

bit 7 **MBISTDONE:** MBIST Done Status bit[(1)]
- 1 = An MBIST operation has been executed
- 0 = No MBIST operation has occurred on the last Reset sequence

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **MBISTSTAT:** MBIST Status bit
- 1 = The last MBIST failed
- 0 = The last MBIST passed; all memory may not have been tested

bit 3-1 **Unimplemented:** Read as ‘0’

bit 0 **MBISTEN:** MBIST Enable bit[(2)]
- 1 = MBIST test is armed; an MBIST test will execute at the next device Reset
- 0 = MBIST test is disarmed

Note 1: HW resets only on a true POR Reset.
2: This bit will self-clear when the MBIST test is complete.

4.4 Memory Resources

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

4.4.1 KEY RESOURCES

- **“dsPIC33/PIC24 Program Memory”** (www.microchip.com/DS70000613)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

4.5 SFR Maps

The following tables show the dsPIC33CDVL64MC106 SFR names, addresses and Reset values. These tables contain all registers applicable to the dsPIC33CDVL64MC106 devices. Not all registers are present on all device variants. Refer to Table 1, Table 2 and Table 3 for peripheral availability. Table 8-1 details port availability for the different package options.

TABLE 4-2: SFR BLOCK 000h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
Core			CORCON	044	0-00000000100000	CRCXORL	0B4	000000000000000-
WREG0	000	0000000000000000	MODCON	046	00--000000000000	CRCXORH	0B6	0000000000000000
WREG1	002	0000000000000000	XMODSRT	048	0000000000000000	CRCDATL	0B8	0000000000000000
WREG2	004	0000000000000000	XMODEND	04A	0000000000000001	CRCDATH	0BA	0000000000000000
WREG3	006	0000000000000000	YMODSRT	04C	0000000000000000	CRCWDATL	0BC	0000000000000000
WREG4	008	0000000000000000	YMODEND	04E	0000000000000001	CRCWDATH	0BE	0000000000000000
WREG5	00A	0000000000000000	XBREV	050	0000000000000000	**CLC**		
WREG6	00C	0000000000000000	DISICNT	052	--00000000000000	CLC1CONL	0C0	0---00--000--000
WREG7	00E	0000000000000000	TBLPAG	054	--------00000000	CLC1CONH	0C2	------------0000
WREG8	010	0000000000000000	YPAG	056	--------00000001	CLC1SEL	0C4	-000-000-000-000
WREG9	012	0000000000000000	MSTRPR	058	----------0----0	CLC1GLSL	0C8	0000000000000000
WREG10	014	0000000000000000	CTXTSTAT	05A	-----000-----000	CLC1GLSH	0CA	0000000000000000
WREG11	016	0000000000000000	**DMT**			CLC2CONL	0CC	0---00--000--000
WREG12	018	0000000000000000	DMTCON	05C	0---------------	CLC2CONH	0CE	------------0000
WREG13	01A	0000000000000000	DMTPRECLR	060	00000000--------	CLC2SEL	0D0	-000-000-000-000
WREG14	01C	0000000000000000	DMTCLR	064	--------00000000	CLC2GLSL	0D4	0000000000000000
WREG15	01E	0000000000000000	DMTSTAT	068	--------000----0	CLC2GLSH	0D6	0000000000000000
SPLIM	020	0000000000000000	DMTCNTL	06C	0000000000000000	CLC3CONL	0D8	0---00--000--000
ACCAL	022	0000000000000000	DMTCNTH	06E	0000000000000000	CLC3CONH	0DA	------------0000
ACCAH	024	0000000000000000	DMTHOLDREG	070	0000000000000000	CLC3SEL	0DC	-000-000-000-000
ACCAU	026	0000000000000000	DMTPSCNTL	074	0000000000000000	CLC3GLSL	0E0	0000000000000000
ACCBL	028	0000000000000000	DMTPSCNTH	076	0000000000000000	CLC3GLSH	0E2	0000000000000000
ACCBH	02A	0000000000000000	DMTPSINTVL	078	0000000000000000	CLC4CONL	0E4	0---00--000--000
ACCBU	02C	0000000000000000	DMTPSINTVH	07A	0000000000000000	CLC4CONH	0E6	------------0000
PCL	02E	0000000000000000	**SENT**			CLC4SEL	0E8	-000-000-000-000
PCH	030	--------00000000	SENT1CON1	080	0-0-000000-0-000	CLC4GLSL	0EC	0000000000000000
DSRPAG	032	------0000000001	SENT1CON2	084	0000000000000000	CLC4GLSH	0EE	0000000000000000
DSWPAG	034	-------000000001	SENT1CON3	088	0000000000000000	**ECC**		
RCOUNT	036	0000000000000000	SENT1STAT	08C	--------00000000	ECCCONL	0F0	---------------0
DCOUNT	038	0000000000000000	SENT1SYNC	090	0000000000000000	ECCCONH	0F2	0000000000000000
DOSTARTL	03A	0000000000000000	SENT1DATL	094	0000000000000000	ECCADDRL	0F4	0000000000000000
DOSTARTH	03C	---------0000000	SENT1DATH	096	0000000000000000	ECCADDRH	0F6	--------00000000
DOENDL	03E	0000000000000000	**CRC**			ECCSTATL	0F8	0000000000000000
DOENDH	040	---------0000000	CRCCONL	0B0	0-000000010000--	ECCSTATH	0FA	------0000000000
SR	042	0000000000000000	CRCCONH	0B2	---00000---00000			

Legend: x = unknown or indeterminate value; “-” = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-3: SFR BLOCK 100h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
Timers			POS1CNTH	14E	0000000000000000	INT1HLDH	162	0000000000000000
T1CON	100	0-0000000-00-00-	POS1HLDL	150	0000000000000000	INDX1CNTL	164	0000000000000000
TMR1	104	0000000000000000	POS1HLD	152	0000000000000000	INDX1CNTH	166	0000000000000000
PR1	108	0000000000000000	VEL1CNT	154	0000000000000000	INDX1HLDL	168	0000000000000000
QEI			VEL1CNTH	156	0000000000000000	INDX1HLD	16A	0000000000000000
QEI1CON	140	0-000000-0000000	VEL1HLDL	158	0000000000000000	QEI1GECL/ QEI1ICL	16C	0000000000000000
QEI1IOC	144	000000000000xxxx	VEL1HLD	15A	0000000000000000	QEI1GECH/ QEI1ICH	16E	0000000000000000
QEI1IOCH	146	---------------0	INT1TMRL	15C	0000000000000000	QEI1LECL	170	0000000000000000
QEI1STAT	148	--00000000000000	INT1TMRH	15E	0000000000000000	QEI1LECH	172	0000000000000000
POS1CNTL	14C	0000000000000000	INT1HLDL	160	0000000000000000			

Legend: x = unknown or indeterminate value; “-” = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-4: SFR BLOCK 200h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
I2C1			U1SCCON	258	----------00000-	SPI1STATL	2B4	---00--0001-1-00
I2C1CONL	200	0-01000000000000	U1SCINT	25A	--00-000--00-000	SPI1STATH	2B6	--000000--000000
I2C1CONH	202	---------0000000	U1INT	25C	--------00---0--	SPI1BUFL	2B8	0000000000000000
I2C1STAT	204	000--0000000000	U2MODE	260	0-000-0000000000	SPI1BUFH	2BA	0000000000000000
I2C1ADD	208	------0000000000	U2MODEH	262	00--000000000000	SPI1BRGL	2BC	---xxxxxxxxxxxxx
I2C1MSK	20C	------0000000000	U2STA	264	0000000010000000	SPI1IMSKL	2C0	---00--0000-0-00
I2C1BRG	210	0000000000000000	U2STAH	266	-000-00000101110	SPI1IMSKH	2C2	0-0000000-000000
I2C1TRN	214	--------11111111	U2BRG	268	0000000000000000	SPI1URDTL	2C4	0000000000000000
I2C1RCV	218	--------00000000	U2BRGH	26A	------------0000	SPI1URDTH	2C6	0000000000000000
UART1 and UART2			U2RXREG	26C	--------xxxxxxxx	SPI2CON1L	2C8	0-00000000000000
U1MODE	238	0-000-0000000000	U2TXREG	270	x-------xxxxxxxx	SPI2CON1H	2CA	0000000000000000
U1MODEH	23A	00--000000000000	U2P1	274	-------000000000	SPI2CON2L	2CC	-----------00000
U1STA	23C	0000000010000000	U2P2	276	-------000000000	SPI2STATL	2D0	---00--0001-1-00
U1STAH	23E	-000-00000101110	U2P3	278	0000000000000000	SPI2STATH	2D2	--000000--000000
U1BRG	240	0000000000000000	U2P3H	27A	--------00000000	SPI2BUFL	2D4	0000000000000000
U1BRGH	242	------------0000	U2TXCHK	27C	--------00000000	SPI2BUFH	2D6	0000000000000000
U1RXREG	244	--------xxxxxxxx	U2RXCHK	27E	--------00000000	SPI2BRGL	2D8	---xxxxxxxxxxxxx
U1TXREG	248	x-------xxxxxxxx	U2SCCON	280	----------00000-	SPI2IMSKL	2DC	---00--0000-0-00
U1P1	24C	-------000000000	U2SCINT	282	--00-000--00-000	SPI2IMSKH	2DE	0-0000000-000000
U1P2	24E	-------000000000	U2INT	284	--------00---0--	SPI2URDTL	2E0	0000000000000000
U1P3	250	0000000000000000	**SPI**			SPI2URDTH	2E2	0000000000000000
U1P3H	252	--------00000000	SPI1CON1L	2AC	0-00000000000000			
U1TXCHK	254	--------00000000	SPI1CON1H	2AE	0000000000000000			
U1RXCHK	256	--------00000000	SPI1CON2L	2B0	-----------00000			

Legend: x = unknown or indeterminate value; “-” = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-5: SFR BLOCK 300h-400h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
High-Speed PWM			PG1TRIGB	356	0000000000000000	PG3FFPCIH	3AE	0000-00000000000
PCLKCON	300	-------0--00--00	PG1TRIGC	358	0000000000000000	PG3SPCIL	3B0	0000000000000000
FSCL	302	0000000000000000	PG1DTL	35A	-----00000000000	PG3SPCIH	3B2	0000-00000000000
FSMINPER	304	0000000000000000	PG1DTH	35C	-----00000000000	PG3LEBL	3B4	0000000000000---
MPHASE	306	0000000000000000	PG1CAP	35E	0000000000000000	PG3LEBH	3B6	-----000----0000
MDC	308	0000000000000000	PG2CONL	360	0----000---00000	PG3PHASE	3B8	0000000000000000
MPER	30A	0000000000000000	PG2CONH	362	000-000000--0000	PG3DC	3BA	0000000000000000
LFSR	30C	-000000000000000	PG2STAT	364	0000000000000000	PG3DCA	3BC	--------00000000
CMBTRIGL	30E	--------00000000	PG2IOCONL	366	0000000000000000	PG3PER	3BE	0000000000000000
CMBTRIGH	310	--------00000000	PG2IOCONH	368	-000---0--000000	PG3TRIGA	3C0	0000000000000000
LOGCONA	312	000000000000-000	PG2EVTL	36A	00000000---00000	PG3TRIGB	3C2	0000000000000000
LOGCONB	314	000000000000-000	PG2EVTH	36C	0000--0000000000	PG3TRIGC	3C4	0000000000000000
LOGCONC	316	000000000000-000	PG2FPCIL	36E	0000000000000000	PG3DTL	3C6	-----00000000000
LOGCOND	318	000000000000-000	PG2FPCIH	370	0000-00000000000	PG3DTH	3C8	-----00000000000
LOGCONE	31A	000000000000-000	PG2CLPCIL	372	0000000000000000	PG3CAP	3CA	0000000000000000
LOGCONF	31C	000000000000-000	PG2CLPCIH	374	0000-00000000000	PG4CONL	3CC	0----000---00000
PWMEVTA	31E	0000----0000-000	PG2FFPCIL	376	0000000000000000	PG4CONH	3CE	000-000000--0000
PWMEVTB	320	0000----0000-000	PG2FFPCIH	378	0000-00000000000	PG4STAT	3D0	0000000000000000
PWMEVTC	322	0000----0000-000	PG2SPCIL	37A	0000000000000000	PG4IOCONL	3D2	0000000000000000
PWMEVTD	324	0000----0000-000	PG2SPCIH	37C	0000-00000000000	PG4IOCONH	3D4	-000---0--000000
PWMEVTE	326	0000----0000-000	PG2LEBL	37E	0000000000000---	PG4EVTL	3D6	00000000---00000
PWMEVTF	328	0000----0000-000	PG2LEBH	380	-----000----0000	PG4EVTH	3D8	0000--0000000000
PG1CONL	32A	0----000---00000	PG2PHASE	382	0000000000000000	PG4FPCIL	3DA	0000000000000000
PG1CONH	32C	000-000000--0000	PG2DC	384	0000000000000000	PG4FPCIH	3DC	0000-00000000000
PG1STAT	32E	0000000000000000	PG2DCA	386	--------00000000	PG4CLPCIL	3DE	0000000000000000
PG1IOCONL	330	0000000000000000	PG2PER	388	0000000000000000	PG4CLPCIH	3E0	0000-00000000000
PG1IOCONH	332	-000---0--000000	PG2TRIGA	38A	0000000000000000	PG4FFPCIL	3E2	0000000000000000
PG1EVTL	334	00000000---00000	PG2TRIGB	38C	0000000000000000	PG4FFPCIH	3E4	0000-00000000000
PG1EVTH	336	0000--0000000000	PG2TRIGC	38E	0000000000000000	PG4SPCIL	3E6	0000000000000000
PG1FPCIL	338	0000000000000000	PG2DTL	390	-----00000000000	PG4SPCIH	3E8	0000-00000000---
PG1FPCIH	33A	0000-00000000000	PG2DTH	392	-----00000000000	PG4LEBL	3EA	0000000000000---
PG1CLPCIL	33C	0000000000000000	PG2CAP	394	0000000000000000	PG4LEBH	3EC	-----000----0000
PG1CLPCIH	33E	0000-00000000000	PG3CONL	396	0----000---00000	PG4PHASE	3EE	0000000000000000
PG1FFPCIL	340	0000000000000000	PG3CONH	398	000-000000--0000	PG4DC	3F0	0000000000000000
PG1FFPCIH	342	0000-00000000000	PG3STAT	39A	0000000000000000	PG4DCA	3F2	--------00000000
PG1SPCIL	344	0000000000000000	PG3IOCONL	39C	0000000000000000	PG4PER	3F4	0000000000000000
PG1SPCIH	346	0000-00000000000	PG3IOCONH	39E	-000---0--000000	PG4TRIGA	3F6	0000000000000000
PG1LEBL	348	0000000000000---	PG3EVTL	3A0	00000000---00000	PG4TRIGB	3F8	0000000000000000
PG1LEBH	34A	-----000----0000	PG3EVTH	3A2	0000--0000000000	PG4TRIGC	3FA	0000000000000000
PG1PHASE	34C	0000000000000000	PG3FPCIL	3A4	0000000000000000	PG4DTL	3FC	-----00000000000
PG1DC	34E	0000000000000000	PG3FPCIH	3A6	0000-00000000000	PG4DTH	3FE	-----00000000000
PG1DCA	350	--------00000000	PG3CLPCIL	3A8	0000000000000000	PG4CAP	400	0000000000000000
PG1PER	352	0000000000000000	PG3CLPCIH	3AA	0000-00000000000			
PG1TRIGA	354	0000000000000000	PG3FFPCIL	3AC	0000000000000000			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-6: SFR BLOCK 800h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
Interrupts			IPC5	84A	-100-----100-100	IPC42	894	-100-100-100----
IFS0	800	0000000000-00000	IPC6	84C	-100-100-----100	IPC43	896	-100-100-100-100
IFS1	802	-00000-00-000000	IPC7	84E	-----100-100-100	IPC44	898	-100-100-100-100
IFS2	804	--0---00---00---	IPC8	850	-100------------	IPC45	89A	-------------100
IFS3	806	0----000----0000	IPC9	852	-------------100	IPC47	89E	-100-100-100----
IFS4	808	--0-0----0000--0	IPC10	854	---------100-100	INTCON1	8C0	0000000000-0000-
IFS5	80A	000000--0000000-	IPC11	856	---------100----	INTCON2	8C2	100----0----0000
IFS6	80C	---0000000000000	IPC12	858	-100-100-100-100	INTCON3	8C4	0------0---0----
IFS7	80E	0000000000000---	IPC14	85C	-----100-100-100	INTCON4	8C6	--------------00
IFS10	814	0000000---------	IPC15	85E	-100------------	INTTREG	8C8	--0-000000000000
IFS11	816	000--------00000	IPC16	860	-100---------100	**Flash**		
IEC0	820	0000000000-00000	IPC17	862	-----100-100-100	NVMCON	8D0	0000--00----0000
IEC1	822	-00000-00-000000	IPC18	864	-100------------	NVMADR	8D2	xxxxxxxxxxxxxxxx
IEC2	824	--0---00---00---	IPC19	866	---------100----	NVMADRU	8D4	--------xxxxxxxx
IEC3	826	0----000----0000	IPC20	868	-100-100-100----	NVMKEY	8D6	--------00000000
IEC4	828	--0-0----0000--0	IPC21	86A	-100-100-100-100	NVMSRCADRL	8D8	0000000000000000
IEC5	82A	000000--0000000-	IPC22	86C	-100-100--------	NVMSRCADRH	8DA	--------00000000
IEC6	82C	---0000000000000	IPC23	86E	-100-100-100-100	**Op Amp**		
IEC7	82E	0000000000000---	IPC24	870	-100-100-100-100	AMPCON1L	8DC	0------------000
IEC10	834	0000000---------	IPC25	872	-100-100-100-100	AMPCON1H	8DE	-------------000
IEC11	836	000--------00000	IPC26	874	-100-100-100-100	**CBG**		
IPC0	840	-100-100-100-100	IPC27	876	-------------100	BIASCON	8F0	0-----------0000
IPC1	842	-100-100-----100	IPC28	878	-100------------	IBIASCONL	8F4	--000000--000000
IPC2	844	-100-100-100-100	IPC29	87A	-100-100-100-100	IBIASCONH	8F6	--000000--000000
IPC3	846	-100-100-100-100	IPC30	87C	-100-100-100-100			
IPC4	848	-100-100-100-100	IPC31	87E	-100-100-100-100			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-7: SFR BLOCK 900h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
PTG			**CCP**			CCP2BUFH	996	0000000000000000
PTGCST	900	0-00-000000---00	CCP1CON1L	950	0-00000000000000	CCP3CON1L	998	0-00000000000000
PTGCON	902	000000000000-000	CCP1CON1H	952	00--000000000000	CCP3CON1H	99A	00--000000000000
PTGBTE	904	0000000000000000	CCP1CON2L	954	00-0----00000000	CCP3CON2L	99C	00-0----00000000
PTGBTEH	906	0000000000000000	CCP1CON2H	956	0-------00-00000	CCP3CON2H	99E	0-------00-00000
PTGHOLD	908	0000000000000000	CCP1CON3L	958	----------000000	CCP3CON3H	9A2	0000------0-00--
PTGT0LIM	90C	0000000000000000	CCP1CON3H	95A	0000------0-00--	CCP3STATL	9A4	-----0--01100000
PTGT1LIM	910	0000000000000000	CCP1STATL	95C	-----0--01100000	CCP3STATH	9A6	-----------00000
PTGSDLIM	914	0000000000000000	CCP1STATH	95E	-----------00000	CCP3TMRL	9A8	0000000000000000
PTGC0LIM	918	0000000000000000	CCP1TMRL	960	0000000000000000	CCP3TMRH	9AA	0000000000000000
PTGC1LIM	91C	0000000000000000	CCP1TMRH	962	0000000000000000	CCP3PRL	9AC	1111111111111111
PTGADJ	920	0000000000000000	CCP1PRL	964	1111111111111111	CCP3PRH	9AE	1111111111111111
PTGL0	924	0000000000000000	CCP1PRH	966	1111111111111111	CCP3RA	9B0	0000000000000000
PTGQPTR	928	-----------00000	CCP1RA	968	0000000000000000	CCP3RB	9B4	0000000000000000
PTGQUE0	930	0000000000000000	CCP1RB	96C	0000000000000000	CCP3BUFL	9B8	0000000000000000
PTGQUE1	932	0000000000000000	CCP1BUFL	970	0000000000000000	CCP3BUFH	9BA	0000000000000000
PTGQUE2	934	0000000000000000	CCP1BUFH	972	0000000000000000	CCP4CON1L	9BC	0-00000000000000
PTGQUE3	936	0000000000000000	CCP2CON1L	974	0-00000000000000	CCP4CON1H	9BE	00--000000000000
PTGQUE4	938	0000000000000000	CCP2CON1H	976	00--000000000000	CCP4CON2L	9C0	00-0----00000000
PTGQUE5	93A	0000000000000000	CCP2CON2L	978	00-0----00000000	CCP4CON2H	9C2	0-------00-00000
PTGQUE6	93C	0000000000000000	CCP2CON2H	97A	0-------00-00000	CCP4CON3H	9C6	0000------0-00--
PTGQUE7	93E	0000000000000000	CCP2CON3H	97E	0000------0-00--	CCP4STATL	9C8	-----0--01100000
PTGQUE8	940	0000000000000000	CCP2STATL	980	-----0--01100000	CCP4STATH	9CA	-----------00000
PTGQUE9	942	0000000000000000	CCP2STATH	982	-----------00000	CCP4TMRL	9CC	0000000000000000
PTGQUE10	944	0000000000000000	CCP2TMRL	984	0000000000000000	CCP4TMRH	9CE	0000000000000000
PTGQUE11	946	0000000000000000	CCP2TMRH	986	0000000000000000	CCP4PRL	9D0	1111111111111111
PTGQUE12	948	0000000000000000	CCP2PRL	988	1111111111111111	CCP4PRH	9D2	1111111111111111
PTGQUE13	94A	0000000000000000	CCP2PRH	98A	1111111111111111	CCP4RA	9D4	0000000000000000
PTGQUE14	94C	0000000000000000	CCP2RA	98C	0000000000000000	CCP4RB	9D8	0000000000000000
PTGQUE15	94E	0000000000000000	CCP2RB	990	0000000000000000	CCP4BUFL	9DC	0000000000000000
			CCP2BUFL	994	0000000000000000	CCP4BUFH	9DE	0000000000000000

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-8: SFR BLOCK A00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
DMA			DMACNT0	ACC	0000000000000001	DMADST2	ADE	0000000000000000
DMACON	ABC	0-0------------0	DMACH1	ACE	-----00000000000	DMACNT2	AE0	0000000000000001
DMABUF	ABE	0000000000000000	DMAINT1	AD0	0000000000000--0	DMACH3	AE2	-----00000000000
DMAL	AC0	0000000000000000	DMASRC1	AD2	0000000000000000	DMAINT3	AE4	0000000000000--0
DMAH	AC2	0000000000000000	DMADST1	AD4	0000000000000000	DMASRC3	AE6	0000000000000000
DMACH0	AC4	-----00000000000	DMACNT1	AD6	0000000000000001	DMADST3	AE8	0000000000000000
DMAINT0	AC6	0000000000000--0	DMACH2	AD8	-----00000000000	DMACNT3	AEA	0000000000000001
DMASRC0	AC8	0000000000000000	DMAINT2	ADA	0000000000000--0			
DMADST0	ACA	0000000000000000	DMASRC2	ADC	0000000000000000			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-9: SFR BLOCK B00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
ADC			ADCMP1ENL	B40	0000000000000000	ADTRIG0L	B80	---00000---00000
ADCON1L	B00	0-0-------------	ADCMP1ENH	B42	--------------00	ADTRIG0H	B82	---00000---00000
ADCON1H	B02	--------011-----	ADCMP1LO	B44	0000000000000000	ADTRIG1L	B84	---00000---00000
ADCON2L	B04	00-0-000-0000000	ADCMP1HI	B46	0000000000000000	ADTRIG1H	B86	---00000---00000
ADCON2H	B06	00----0000000000	ADCMP2ENL	B48	0000000000000000	ADTRIG2L	B88	---00000---00000
ADCON3L	B08	0000000000000000	ADCMP2ENH	B4A	--------------00	ADTRIG2H	B8A	---00000---00000
ADCON3H	B0A	000000000-------	ADCMP2LO	B4C	0000000000000000	ADTRIG3L	B8C	---00000---00000
ADMOD0L	B10	0000000000000000	ADCMP2HI	B4E	0000000000000000	ADTRIG3H	B8E	---00000---00000
ADMOD0H	B12	0000000000000000	ADCMP3ENL	B50	0000000000000000	ADTRIG4L	B90	---00000---00000
ADMOD1L	B14	------------0000	ADCMP3ENH	B52	--------------00	ADCMP0CON	BA0	---0000000000000
ADIEL	B20	0000000000000000	ADCMP3LO	B54	0000000000000000	ADCMP1CON	BA4	---0000000000000
ADIEH	B22	--------------00	ADCMP3HI	B56	0000000000000000	ADCMP2CON	BA8	---0000000000000
ADCSSL	B28	0000000000000000	ADFL0DAT	B68	0000000000000000	ADCMP3CON	BAC	---0000000000000
ADCSSH	B2A	0000000000000000	ADFL0CON	B6A	00000000---00000	ADLVLTRGL	BD0	0000000000000000
ADSTATL	B30	0000000000000000	ADFL1DAT	B6C	0000000000000000	ADLVLTRGH	BD2	--------------00
ADSTATH	B32	--------------00	ADFL1CON	B6E	00000000---00000	ADEIEL	BF0	0000000000000000
ADCMP0ENL	B38	0000000000000000	ADFL2DAT	B70	0000000000000000	ADEIEH	BF2	--------------00
ADCMP0ENH	B3A	--------------00	ADFL2CON	B72	00000000---00000	ADEISTATL	BF8	0000000000000000
ADCMP0LO	B3C	0000000000000000	ADFL3DAT	B74	0000000000000000	ADEISTATH	BFA	--------------00
ADCMP0HI	B3E	0000000000000000	ADFL3CON	B76	00000000---00000			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-10: SFR BLOCK C00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
ADC (Continued)			ADCBUF9	C1E	0000000000000000	DACCTRL2H	C86	------0010001010
ADCON5L	C00	0-------0-------	ADCBUF10	C20	0000000000000000	DAC1CONL	C88	000--000x0000000
ADCON5H	C02	----xxxx0-------	ADCBUF11	C22	0000000000000000	DAC1CONH	C8A	------0000000000
ADCBUF0	C0C	0000000000000000	ADCBUF12	C24	0000000000000000	DAC1DATL	C8C	----000000000000
ADCBUF1	C0E	0000000000000000	ADCBUF13	C26	0000000000000000	DAC1DATH	C8E	----000000000000
ADCBUF2	C10	0000000000000000	ADCBUF14	C28	0000000000000000	SLP1CONL	C90	0000000000000000
ADCBUF3	C12	0000000000000000	ADCBUF15	C2A	0000000000000000	SLP1CONH	C92	0---000---------
ADCBUF4	C14	0000000000000000	ADCBUF16	C2C	0000000000000000	SLP1DAT	C94	0000000000000000
ADCBUF5	C16	0000000000000000	ADCBUF17	C2E	0000000000000000	VREGCON	CFC	0-------------00
ADCBUF6	C18	0000000000000000	**DAC**					
ADCBUF7	C1A	0000000000000000	DACCTRL1L	C80	0-0-----0000-000			
ADCBUF8	C1C	0000000000000000	DACCTRL2L	C84	------0001010101			

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-11: SFR BLOCK D00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
PPS			RPINR21	D2E	0000000000000000	RPOR3	D86	--000000--000000
RPCON	D00	----0-----------	RPINR22	D30	0000000000000000	RPOR4	D88	--000000--000000
RPINR0	D04	00000000--------	RPINR23	D32	--------00000000	RPOR5	D8A	--000000--000000
RPINR1	D06	0000000000000000	RPINR27	D3A	0000000000000000	RPOR6	D8C	--000000--000000
RPINR2	D08	00000000--------	RPINR37	D4E	0000000000000000	RPOR7	D8E	--000000--000000
RPINR3	D0A	0000000000000000	RPINR38	D50	--------00000000	RPOR8	D90	--000000--000000
RPINR4	D0C	0000000000000000	RPINR42	D58	0000000000000000	RPOR9	D92	--000000--000000
RPINR5	D0E	0000000000000000	RPINR43	D5A	0000000000000000	RPOR10	D94	--000000--000000
RPINR6	D10	0000000000000000	RPINR44	D5C	0000000000000000	RPOR11	D96	--000000--000000
RPINR11	D1A	0000000000000000	RPINR45	D5E	00000000--------	RPOR12	D98	--000000--000000
RPINR12	D1C	0000000000000000	RPINR46	D60	0000000000000000	RPOR13	D9A	--000000--000000
RPINR13	D1E	0000000000000000	RPINR47	D62	0000000000000000	RPOR14	D9C	--000000--000000
RPINR14	D20	0000000000000000	RPINR48	D64	0000000000000000	RPOR16	DA0	--000000--------
RPINR15	D22	0000000000000000	RPINR49	D66	0000000000000000	RPOR17	DA2	--000000--000000
RPINR18	D28	0000000000000000	RPOR0	D80	--000000--000000	RPOR18	DA4	--000000--000000
RPINR19	D2A	0000000000000000	RPOR1	D82	--000000--000000	RPOR19	DA6	--000000--000000
RPINR20	D2C	0000000000000000	RPOR2	D84	--000000--000000			

Legend: x = unknown or indeterminate value; "–" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-12: SFR BLOCK E00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
I/O Ports			ODCB	E24	0000000000000000	CNSTATC	E4A	--00000000000000
ANSELA	E00	-----------11111	CNPUB	E26	0000000000000000	CNEN1C	E4C	--00000000000000
TRISA	E02	-----------11111	CNPDB	E28	0000000000000000	CNFC	E4E	--00000000000000
PORTA	E04	-----------00000	CNCONB	E2A	0---0-----------	ANSELD	E54	--1--1----------
LATA	E06	-----------xxxxx	CNEN0B	E2C	0000000000000000	TRISD	E56	--1--1-1------1-
ODCA	E08	-----------00000	CNSTATB	E2E	0000000000000000	PORTD	E58	--0--0-0------0-
CNPUA	E0A	-----------00000	CNEN1B	E30	0000000000000000	LATD	E5A	--x--x-x------x-
CNPDA	E0C	-----------00000	CNFB	E32	0000000000000000	ODCD	E5C	--0--0-0------0-
CNCONA	E0E	0---0-----------	ANSELC	E38	--------11--1111	CNPUD	E5E	--0--0-0------0-
CNEN0A	E10	-----------00000	TRISC	E3A	--11111111111111	CNPDD	E60	0000000000000000
CNSTATA	E12	-----------00000	PORTC	E3C	--00000000000000	CNCOND	E62	0---0-----------
CNEN1A	E14	-----------00000	LATC	E3E	--xxxxxxxxxxxxxx	CNEN0D	E64	--0--0-0------0-
CNFA	E16	-----------00000	ODCC	E40	--00000000000000	CNSTATD	E66	--0--0-0------0-
ANSELB	E1C	------111--11111	CNPUC	E42	--00000000000000	CNEN1D	E68	--0--0-0------0-
TRISB	E1E	1111111111111111	CNPDC	E44	--00000000000000	CNFD	E6A	--0--0-0------0-
PORTB	E20	0000000000000000	CNCONC	E46	0---0-----------	**Memory BIST**		
LATB	E22	xxxxxxxxxxxxxxxx	CNEN0C	E48	--00000000000000	MBISTCON	EFC	-------00--0---0

Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address values are in hexadecimal. Reset values are in binary.

TABLE 4-13: SFR BLOCK F00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
UART3			U3SCINT	F22	--00-000--00-000	PMD4	FAA	------------0---
U3MODE	F00	0-000-0000000000	U3INT	F24	--------00---0--	PMD6	FAE	----0000--------
U3MODEH	F02	00--000000000000	**Reset and Oscillator**			PMD7	FB0	-------0----0---
U3STA	F04	0000000010000000	RCON	F80	00----0000-00011	PMD8	FB2	--0-0--0--00000-
U3STAH	F06	-000-00000101110	OSCCON	F84	-000-yyy0-0-0--0	**WDT**		
U3BRG	F08	0000000000000000	CLKDIV	F86	00110000--000001	WDTCONL	FB4	0--0000000000000
U3BRGH	F0A	------------0000	PLLFBD	F88	----000010010110	WDTCONH	FB6	0000000000000000
U3RXREG	F0C	--------xxxxxxxx	PLLDIV	F8A	------00-011-001	**Reference Clock Output**		
U3TXREG	F10	x-------xxxxxxxx	OSCTUN	F8C	----------000000	REFOCONL	FB8	0-000-00----0000
U3P1	F14	-------000000000	DCOTUN	F9C	----------000000	REFOCONH	FBA	-000000000000000
U3P2	F16	-------000000000	DCOCON	F9E	--0-xxxx--------	REFOTRIMH	FBE	000000000-------
U3P3	F18	0000000000000000	**PMD**			**Programmer/Debugger**		
U3P3H	F1A	--------00000000	PMDCONL	FA0	----0-----------	VISI	FCC	0000000000000000
U3TXCHK	F1C	--------00000000	PMD1	FA4	----000-00000--0	APPO	FD2	0000000000000000
U3RXCHK	F1E	--------00000000	PMD2	FA6	------------0000	APPI	FD4	0000000000000000
U3SCCON	F20	----------00000-	PMD3	FA8	--------0---0---	APPS	FD6	-----------00000

Legend: x = unknown or indeterminate value; "-" =unimplemented bits; y = value set by Configuration bits. Address values are in hexadecimal. Reset values are in binary.

4.5.1 PAGED MEMORY SCHEME

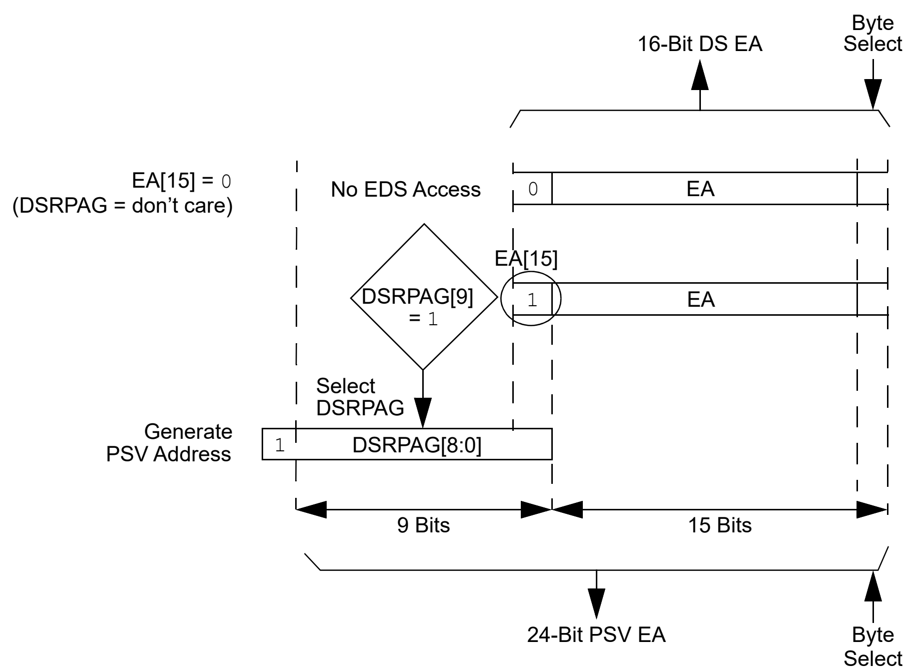
The dsPIC33CDVL64MC106 architecture extends the available Data Space through a paging scheme, which allows the available Data Space to be accessed using MOV instructions in a linear fashion for pre- and post-modified Effective Addresses (EAs). The upper half of the base Data Space address is used in conjunction with the Data Space Read Page (DSRPAG) register to form the Program Space Visibility (PSV) address.

The Data Space Read Page (DSRPAG) register is located in the SFR space. Construction of the PSV address is shown in Figure 4-5. When DSRPAG[9] = 1 and the base address bit, EA[15] = 1, the DSRPAG[8:0] bits are concatenated onto EA[14:0] to form the 24-bit PSV read address.

The paged memory scheme provides access to multiple 32-Kbyte windows in the PSV memory. The Data Space Read Page (DSRPAG) register, in combination with the upper half of the Data Space address, can provide up to 8 Mbytes of PSV address space. The paged data memory space is shown in Figure 4-6.

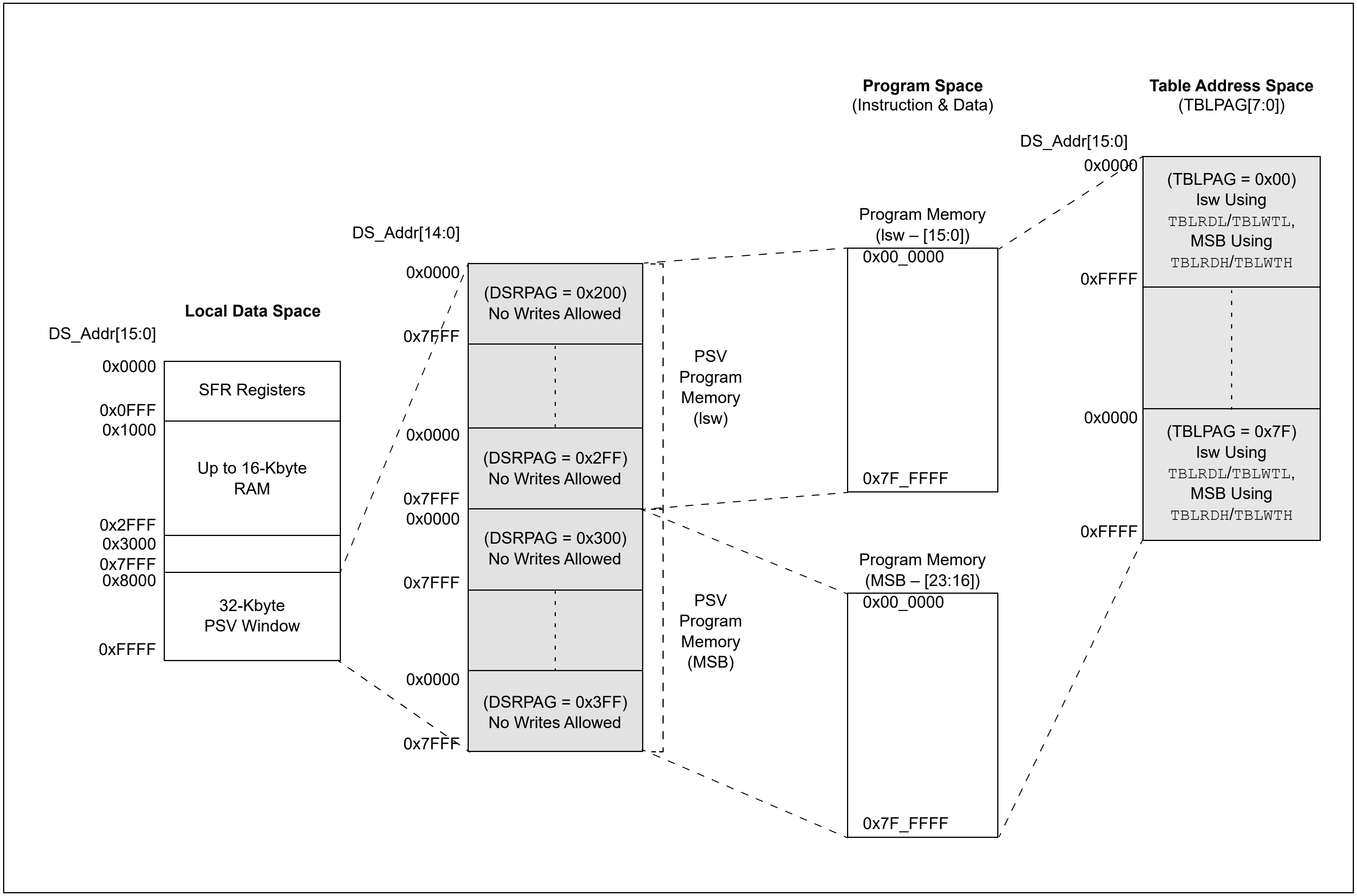
The Program Space (PS) can be accessed with a DSRPAG of 0x200 or greater. Only reads from PS are supported using the DSRPAG.

FIGURE 4-5: PROGRAM SPACE VISIBILITY (PSV) READ ADDRESS GENERATION



Note: DS read access when DSRPAG = 0x000 will force an address error trap.

FIGURE 4-6: PAGED DATA MEMORY SPACE



When a PSV page overflow or underflow occurs, EA[15] is cleared as a result of the register indirect EA calculation. An overflow or underflow of the EA in the PSV pages can occur at the page boundaries when:

- The initial address, prior to modification, addresses the PSV page.
- The EA calculation uses Pre- or Post-Modified Register Indirect Addressing; however, this does not include Register Offset Addressing.

In general, when an overflow is detected, the DSRPAG register is incremented and the EA[15] bit is set to keep the base address within the PSV window. When an underflow is detected, the DSRPAG register is decremented and the EA[15] bit is set to keep the base address within the PSV window. This creates a linear PSV address space, but only when using Register Indirect Addressing modes.

Exceptions to the operation described above arise when entering and exiting the boundaries of Page 0 and PSV spaces. Table 4-14 lists the effects of overflow and underflow scenarios at different boundaries.

In the following cases, when overflow or underflow occurs, the EA[15] bit is set and the DSRPAG is not modified; therefore, the EA will wrap to the beginning of the current page:

- Register Indirect with Register Offset Addressing
- Modulo Addressing
- Bit-Reversed Addressing

TABLE 4-14: OVERFLOW AND UNDERFLOW SCENARIOS AT PAGE 0 AND PSV SPACE BOUNDARIES(2,3,4)

O/U, R/W	Operation	Before			After		
		DSRPAG	DS EA[15]	Page Description	DSRPAG	DS EA[15]	Page Description
O, Read	`[++Wn]` or `[Wn++]`	DSRPAG = 0x2FF	1	PSV: Last lsw page	DSRPAG = 0x300	1	PSV: First MSB page
O, Read		DSRPAG = 0x3FF	1	PSV: Last MSB page	DSRPAG = 0x3FF	0	See Note 1
U, Read	`[--Wn]` or `[Wn--]`	DSRPAG = 0x001	1	PSV page	DSRPAG = 0x001	0	See Note 1
U, Read		DSRPAG = 0x200	1	PSV: First lsw page	DSRPAG = 0x200	0	See Note 1
U, Read		DSRPAG = 0x300	1	PSV: First MSB page	DSRPAG = 0x2FF	1	PSV: Last lsw page

Legend: O = Overflow, U = Underflow, R = Read, W = Write

Note 1: The Register Indirect Addressing now addresses a location in the base Data Space (0x0000-0x8000).
2: An EDS access, with DSRPAG = 0x000, will generate an address error trap.
3: Only reads from PS are supported using DSRPAG.
4: Pseudolinear Addressing is not supported for large offsets.

4.5.1.1 Extended X Data Space

The lower portion of the base address space range, between 0x0000 and 0x7FFF, is always accessible, regardless of the contents of the Data Space Read Page register. It is indirectly addressable through the register indirect instructions. It can be regarded as being located in the default EDS Page 0 (i.e., EDS address range of 0x000000 to 0x007FFF with the base address bit, EA[15] = 0, for this address range). However, Page 0 cannot be accessed through the upper 32 Kbytes, 0x8000 to 0xFFFF, of base Data Space in combination with DSRPAG = 0x000. Consequently, DSRPAG is initialized to 0x001 at Reset.

Note 1: DSRPAG should not be used to access Page 0. An EDS access with DSRPAG set to 0x000 will generate an address error trap.

2: Clearing the DSRPAG in software has no effect.

The remaining PSV pages are only accessible using the DSRPAG register in combination with the upper 32 Kbytes, 0x8000 to 0xFFFF, of the base address, where the base address bit, EA[15] = 1.

4.5.1.2 Software Stack

The W15 register serves as a dedicated Software Stack Pointer (SSP), and is automatically modified by exception processing, subroutine calls and returns; however, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies reading, writing and manipulating the Stack Pointer (for example, creating stack frames).

Note: To protect against misaligned stack accesses, W15[0] is fixed to '0' by the hardware.

W15 is initialized to 0x1000 during all Resets. This address ensures that the SSP points to valid RAM in the dsPIC33CDVL64MC106 devices and permits stack availability for non-maskable trap exceptions. These can occur before the SSP is initialized by the user software. You can reprogram the SSP during initialization to any location within Data Space.

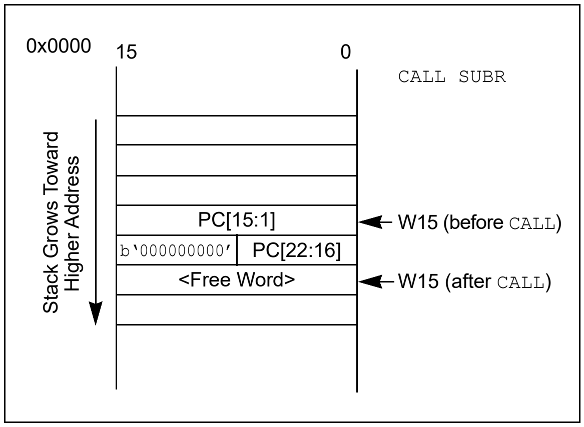
The Software Stack Pointer always points to the first available free word and fills the software stack, working from lower toward higher addresses. Figure 4-7 illustrates how it pre-decrements for a stack pop (read) and post-increments for a stack push (writes).

When the PC is pushed onto the stack, PC[15:0] are pushed onto the first available stack word, then PC[22:16] are pushed into the second available stack location. For a PC push during any `CALL` instruction, the MSB of the PC is zero-extended before the push, as shown in Figure 4-7. During exception processing, the MSB of the PC is concatenated with the lower eight bits of the CPU STATUS Register, SR. This allows the contents of SRL to be preserved automatically during interrupt processing.

Note 1: To maintain system Stack Pointer (W15) coherency, W15 is never subject to (EDS) paging, and is therefore, restricted to an address range of 0x0000 to 0xFFFF. The same applies to the W14 when used as a Stack Frame Pointer (SFA = 1).

2: As the stack can be placed in, and can access X and Y spaces, care must be taken regarding its use, particularly with regard to local automatic variables in a C development environment

FIGURE 4-7: CALL STACK FRAME



4.5.2 INSTRUCTION ADDRESSING MODES

The addressing modes shown in Table 4-15 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the `MAC` class of instructions differ from those in the other instruction types.

4.5.2.1 File Register Instructions

Most file register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (Near Data Space). Most file register instructions employ a Working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same file register or WREG (with the exception of the `MUL` instruction), which writes the result to a register or register pair. The `MOV` instruction allows additional flexibility and can access the entire Data Space.

4.5.2.2 MCU Instructions

The three-operand MCU instructions are of the form:

```
Operand 3 = Operand 1 <function> Operand 2
```

where `Operand 1` is always a Working register (that is, the addressing mode can only be Register Direct), which is referred to as Wb. `Operand 2` can be a W register fetched from data memory or a 5-bit literal. The result location can either be a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-Modified
- Register Indirect Pre-Modified
- 5-Bit or 10-Bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions can support different subsets of these addressing modes.

TABLE 4-15: FUNDAMENTAL ADDRESSING MODES SUPPORTED

Addressing Mode	Description
File Register Direct	The address of the file register is specified explicitly.
Register Direct	The contents of a register are accessed directly.
Register Indirect	The contents of Wn form the Effective Address (EA).
Register Indirect Post-Modified	The contents of Wn form the EA. Wn is post-modified (incremented or decremented) by a constant value.
Register Indirect Pre-Modified	Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA.
Register Indirect with Register Offset (Register Indexed)	The sum of Wn and Wb forms the EA.
Register Indirect with Literal Offset	The sum of Wn and a literal forms the EA.

4.5.2.3 Move and Accumulator Instructions

Move instructions, and the DSP accumulator class of instructions, provide a greater degree of addressing flexibility than other instructions. In addition to the addressing modes supported by most MCU instructions, move and accumulator instructions also support Register Indirect with Register Offset Addressing mode, also referred to as Register Indexed mode.

Note:	For the `MOV` instructions, the addressing mode specified in the instruction can differ for the source and destination EA. However, the 4-bit Wb (Register Offset) field is shared by both source and destination (but typically only used by one).

In summary, the following addressing modes are supported by move and accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-Modified
- Register Indirect Pre-Modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-Bit Literal
- 16-Bit Literal

Note:	Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

4.5.2.4 `MAC` Instructions

The dual source operand DSP instructions (`CLR`, `ED`, `EDAC`, `MAC`, `MPY`, `MPY.N`, `MOVSAC` and `MSC`), also referred to as `MAC` instructions, use a simplified set of addressing modes to allow the user application to effectively manipulate the Data Pointers through register indirect tables.

The two-source operand prefetch registers must be members of the set {W8, W9, W10, W11}. For data reads, W8 and W9 are always directed to the X RAGU, and W10 and W11 are always directed to the Y AGU. The Effective Addresses generated (before and after modification) must therefore, be valid addresses within X Data Space for W8 and W9, and Y Data Space for W10 and W11.

Note:	Register Indirect with Register Offset Addressing mode is available only for W9 (in X space) and W11 (in Y space).

In summary, the following addressing modes are supported by the `MAC` class of instructions:

- Register Indirect
- Register Indirect Post-Modified by 2
- Register Indirect Post-Modified by 4
- Register Indirect Post-Modified by 6
- Register Indirect with Register Offset (Indexed)

4.5.2.5 Other Instructions

Besides the addressing modes outlined previously, some instructions use literal constants of various sizes. For example, `BRA` (branch) instructions use 16-bit signed literals to specify the branch destination directly, whereas the `DISI` instruction uses a 14-bit unsigned literal field. In some instructions, such as `ULNK`, the source of an operand or result is implied by the opcode itself. Certain operations, such as a `NOP`, do not have any operands.

4.5.3 MODULO ADDRESSING

Modulo Addressing mode is a method of providing an automated means to support circular data buffers using hardware. The objective is to remove the need for software to perform data address boundary checks when executing tightly looped code, as is typical in many DSP algorithms.

Modulo Addressing can operate in either Data or Program Space (since the Data Pointer mechanism is essentially the same for both). One circular buffer can be supported in each of the X (which also provides the pointers into Program Space) and Y Data Spaces. Modulo Addressing can operate on any W Register Pointer. However, it is not advisable to use W14 or W15 for Modulo Addressing since these two registers are used as the Stack Frame Pointer and Stack Pointer, respectively.

In general, any particular circular buffer can be configured to operate in only one direction, as there are certain restrictions on the buffer start address (for incrementing buffers) or end address (for decrementing buffers), based upon the direction of the buffer.

The only exception to the usage restrictions is for buffers that have a power-of-two length. As these buffers satisfy the start and end address criteria, they can operate in a Bidirectional mode (that is, address boundary checks are performed on both the lower and upper address boundaries).

4.5.3.1 Start and End Address

The Modulo Addressing scheme requires that a starting and ending address be specified and loaded into the 16-bit Modulo Buffer Address registers: XMODSRT, XMODEND, YMODSRT and YMODEND (see Table 4-2).

Note:	Y space Modulo Addressing EA calculations assume word-sized data (LSb of every EA is always clear).

The length of a circular buffer is not directly specified. It is determined by the difference between the corresponding start and end addresses. The maximum possible length of the circular buffer is 32K words (64 Kbytes).

4.5.3.2 W Address Register Selection

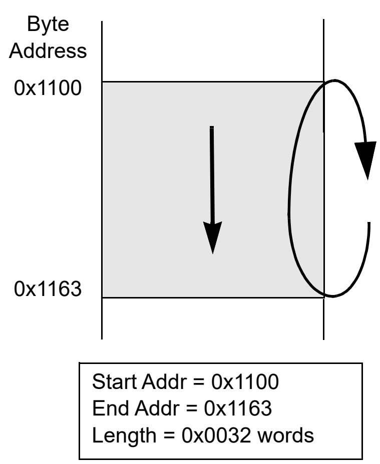
The Modulo and Bit-Reversed Addressing Control register, MODCON[15:0], contains enable flags, as well as a W register field to specify the W Address registers. The XWM and YWM fields select the registers that operate with Modulo Addressing:

- If XWM = `1111`, X RAGU and X WAGU Modulo Addressing is disabled.
- If YWM = `1111`, Y AGU Modulo Addressing is disabled.

The X Address Space Pointer W (XWM) register, to which Modulo Addressing is to be applied, is stored in MODCON[3:0] (see Table 4-2). Modulo Addressing is enabled for X Data Space when XWM is set to any value other than '`1111`' and the XMODEN bit is set (MODCON[15]).

The Y Address Space Pointer W (YWM) register, to which Modulo Addressing is to be applied, is stored in MODCON[7:4]. Modulo Addressing is enabled for Y Data Space when YWM is set to any value other than '`1111`' and the YMODEN bit (MODCON[14]) is set.

FIGURE 4-8: MODULO ADDRESSING OPERATION EXAMPLE




```
MOV    #0x1100, W0
MOV    W0, XMODSRT    ;set modulo start address
MOV    #0x1163, W0
MOV    W0, MODEND     ;set modulo end address
MOV    #0x8001, W0
MOV    W0, MODCON     ;enable W1, X AGU for modulo

MOV    #0x0000, W0    ;W0 holds buffer fill value

MOV    #0x1110, W1    ;point W1 to buffer

DO     AGAIN, #0x31   ;fill the 50 buffer locations
MOV    W0, [W1++]     ;fill the next location
AGAIN: INC W0, W0     ;increment the fill value
```

4.5.3.3 Modulo Addressing Applicability

Modulo Addressing can be applied to the Effective Address (EA) calculation associated with any W register. Address boundaries check for addresses equal to:

- The upper boundary addresses for incrementing buffers
- The lower boundary addresses for decrementing buffers

It is important to realize that the address boundaries check for addresses less than, or greater than, the upper (for incrementing buffers) and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes can, therefore, jump beyond boundaries and still be adjusted correctly.

> **Note:** The modulo corrected Effective Address is written back to the register only when Pre-Modify or Post-Modify Addressing mode is used to compute the Effective Address. When an address offset (such as [W7 + W2]) is used, Modulo Addressing correction is performed, but the contents of the register remain unchanged.

4.5.4 BIT-REVERSED ADDRESSING

Bit-Reversed Addressing mode is intended to simplify data reordering for radix-2 FFT algorithms. It is supported by the X AGU for data writes only.

The modifier, which can be a constant value or register contents, is regarded as having its bit order reversed. The address source and destination are kept in normal order. Thus, the only operand requiring reversal is the modifier.

4.5.4.1 Bit-Reversed Addressing Implementation

Bit-Reversed Addressing mode is enabled in any of these situations:

- BWMx bits (W register selection) in the MODCON register are any value other than '`1111`' (the stack cannot be accessed using Bit-Reversed Addressing).
- The BREN bit is set in the XBREV register.
- The addressing mode used is Register Indirect with Pre-Increment or Post-Increment.

If the length of a bit-reversed buffer is $M = 2^N$ bytes, the last 'N' bits of the data buffer start address must be zeros.

XB[14:0] is the Bit-Reversed Addressing modifier, or 'pivot point', which is typically a constant. In the case of an FFT computation, its value is equal to half of the FFT data buffer size.

> **Note:** All bit-reversed EA calculations assume word-sized data (LSb of every EA is always clear). The XB value is scaled accordingly to generate compatible (byte) addresses.

When enabled, Bit-Reversed Addressing is executed only for Register Indirect with Pre-Increment or Post-Increment Addressing and word-sized data writes. It does not function for any other addressing mode or for byte-sized data and normal addresses are generated instead. When Bit-Reversed Addressing is active, the W Address Pointer is always added to the address modifier (XB) and the offset associated with the Register Indirect Addressing mode is ignored. In addition, as word-sized data are a requirement, the LSb of the EA is ignored (and always clear).

> **Note:** Modulo Addressing and Bit-Reversed Addressing can be enabled simultaneously using the same W register, but Bit-Reversed Addressing operation will always take precedence for data writes when enabled.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV[15]) bit, a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the Bit-Reversed Pointer.

FIGURE 4-9: BIT-REVERSED ADDRESSING EXAMPLE

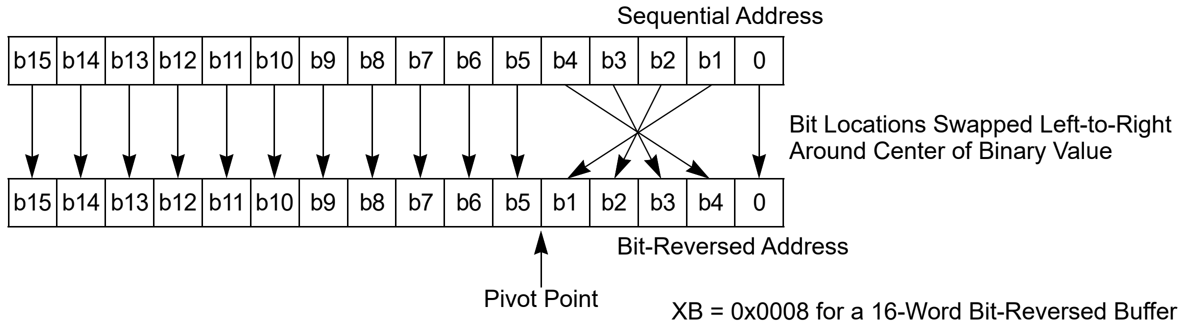


TABLE 4-16: BIT-REVERSED ADDRESSING SEQUENCE (16-ENTRY)

Normal Address					Bit-Reversed Address				
A3	A2	A1	A0	Decimal	A3	A2	A1	A0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	8
0	0	1	0	2	0	1	0	0	4
0	0	1	1	3	1	1	0	0	12
0	1	0	0	4	0	0	1	0	2
0	1	0	1	5	1	0	1	0	10
0	1	1	0	6	0	1	1	0	6
0	1	1	1	7	1	1	1	0	14
1	0	0	0	8	0	0	0	1	1
1	0	0	1	9	1	0	0	1	9
1	0	1	0	10	0	1	0	1	5
1	0	1	1	11	1	1	0	1	13
1	1	0	0	12	0	0	1	1	3
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	0	1	1	1	7
1	1	1	1	15	1	1	1	1	15

4.5.5 INTERFACING PROGRAM AND DATA MEMORY SPACES

The dsPIC33CDVL64MC106 architecture uses a 24-bit wide Program Space (PS) and a 16-bit wide Data Space (DS). The architecture is also a modified Harvard scheme, meaning that data can also be present in the Program Space. To use these data successfully, they must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the architecture of the dsPIC33CDVL64MC106 devices provides two methods by which Program Space can be accessed during operation:

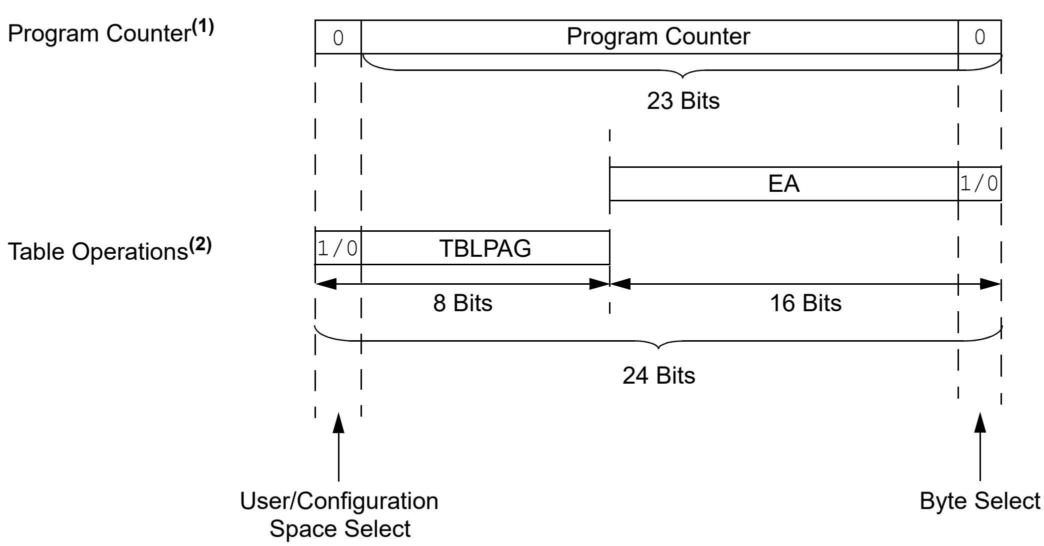
- Using table instructions to access individual bytes or words anywhere in the Program Space
- Remapping a portion of the Program Space into the Data Space (Program Space Visibility)

Table instructions allow an application to read small areas of the program memory. This capability makes the method ideal for accessing data tables that need to be updated periodically. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data. The application can only access the least significant word of the program word.

TABLE 4-17: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		[23]	**[22:16]**	**[15]**	**[14:1]**	**[0]**
Instruction Access (Code Execution)	User	0	PC[22:1]			0
		0xxx xxxx xxxx xxxx xxxx xxx0				
TBLRD (Byte/Word Read)	User	TBLPAG[7:0]		Data EA[15:0]		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	Configuration	TBLPAG[7:0]		Data EA[15:0]		
		1xxx xxxx		xxxx xxxx xxxx xxxx		

FIGURE 4-10: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note 1: The Least Significant bit (LSb) of Program Space addresses is always fixed as '0' to maintain word alignment of data in the Program and Data Spaces.

2: Table operations are not required to be word-aligned. Table Read operations are permitted in the configuration memory space.

4.5.5.1 Data Access from Program Memory Using Table Instructions

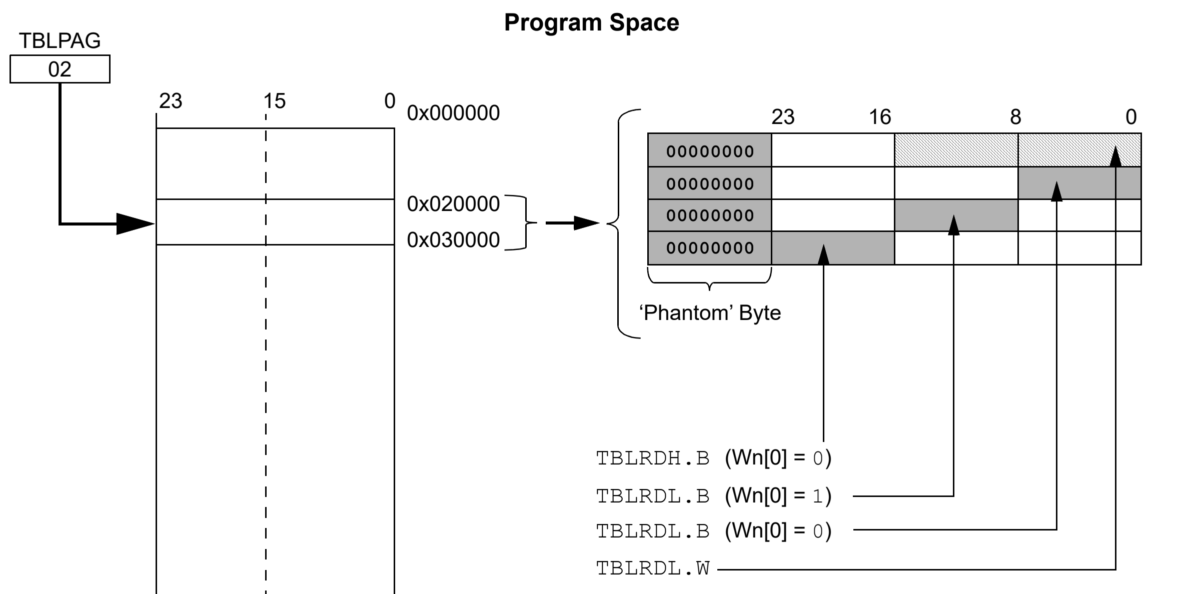
The `TBLRDL` instruction offers a direct method of reading the lower word of any address within the Program Space without going through Data Space. The `TBLRDH` instruction is the only method to read the upper eight bits of a Program Space word as data.

This allows program memory addresses to directly map to Data Space addresses. Program memory can thus be regarded as two 16-bit wide word address spaces, residing side by side, each with the same address range. `TBLRDL` accesses the space that contains the least significant data word. `TBLRDH` accesses the space that contains the upper data byte.

Two table instructions are provided to read byte or word-sized (16-bit) data from Program Space. Both function as either byte or word operations.

- `TBLRDL` (Table Read Low):
  - In Word mode, this instruction maps the lower word of the Program Space location (P[15:0]) to a data address (D[15:0])
  - In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when Byte Select is ‘`1`’; the lower byte is selected when it is ‘`0`’.
- `TBLRDH` (Table Read High):
  - In Word mode, this instruction maps the entire upper word of a program address (P[23:16]) to a data address. The ‘phantom’ byte (D[15:8]) is always ‘`0`’.
  - In Byte mode, this instruction maps the upper or lower byte of the program word to D[7:0] of the data address in the `TBLRDL` instruction. The data are always ‘`0`’ when the upper ‘phantom’ byte is selected (Byte Select = `1`).

FIGURE 4-11: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



The address for the table operation is determined by the data EA within the page defined by the TBLPAG register.
Only read operations are shown; write operations are also valid in the user memory area.

5.0 FLASH PROGRAM MEMORY

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Flash Programming"** (www.microchip.com/DS70000609).

The dsPIC33CDVL64MC106 devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire V_{DD} range.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™) programming capability
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)
- Run-Time Self-Programming (RTSP)

ICSP allows for a dsPIC33CDVL64MC106 device to be serially programmed while in the end application circuit. This is done with a Programming Clock and Programming Data (PGCx/PGDx) line, and three other lines for power (V_{DD}), ground (V_{SS}) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

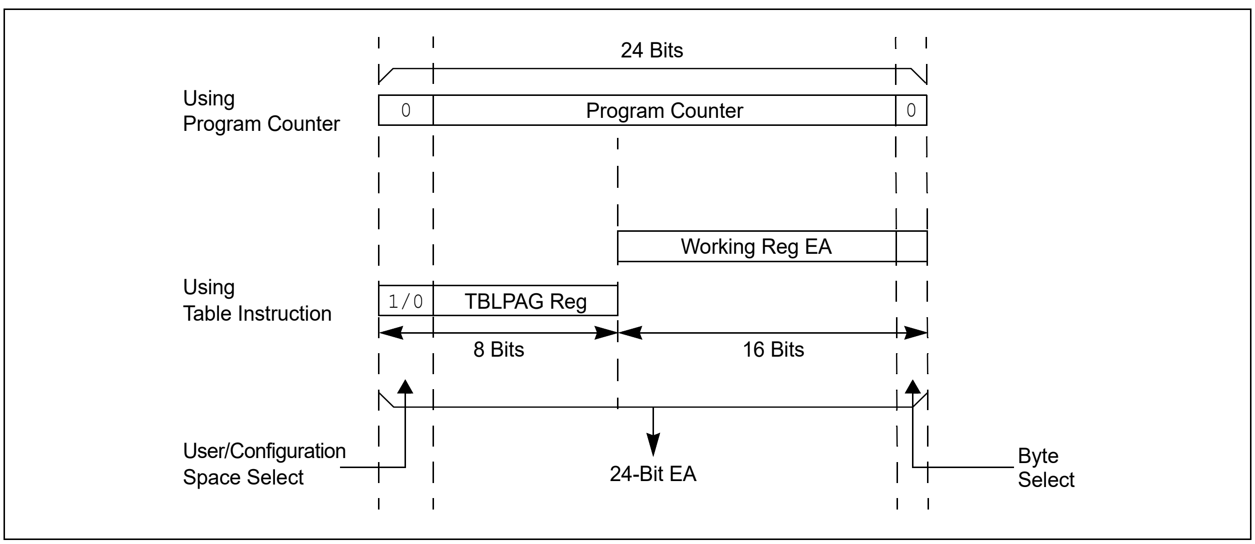
Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the Programming Executive, to manage the programming process. Using an SPI data frame format, the Programming Executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

RTSP is accomplished using `TBLRD` (Table Read) and `TBLWT` (Table Write) instructions. With RTSP, the user application can write program memory data by double program memory words or by blocks ('rows') of 128 instructions (256 addressable bytes). RTSP can erase program memory in blocks or 'pages' of 1024 instructions (2048 addressable bytes) at a time.

5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using bits[7:0] of the TBLPAG register and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 5-1. The `TBLRDL` and `TBLWTL` instructions are used to read or write to bits[15:0] of program memory. `TBLRDL` and `TBLWTL` can access program memory in both Word and Byte modes. The `TBLRDH` and `TBLWTH` instructions are used to read or write to bits[23:16] of program memory. `TBLRDH` and `TBLWTH` can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.2 RTSP Operation

The dsPIC33CDVL64MC106 Flash program memory array is organized into rows of 128 instructions or 384 bytes. RTSP allows the user application to erase a single page (eight rows or 1024 instructions) of memory at a time and to program one row at a time. It is possible to program two instructions at a time as well.

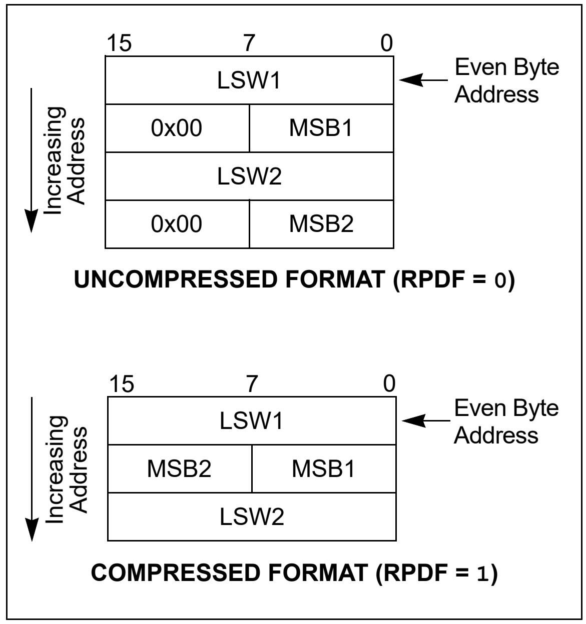
The page erase and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 3072 bytes and 384 bytes, respectively. Table 33-17 in **Section 33.0 "Electrical Characteristics"** lists the typical erase and programming times. To write into the Flash memory, it is necessary to erase the page that contains the desired address of the location the user wants to change.

Row programming is performed by loading 384 bytes into data memory and then loading the address of the first byte in that row into the NVMSRCADRL/H register pair. Once the write has been initiated, the device will automatically load the write latches, and increment the NVMSRCADRL/H and the NVMADR/U registers until all bytes have been programmed. The RPDF bit (NVMCON[9]) selects the format of the stored data in RAM to be either compressed or uncompressed. See Figure 5-2 for data formatting. Compressed data help to reduce the amount of required RAM by using the upper byte of the second word for the MSB of the second instruction.

The basic sequence for RTSP word programming is to use the `TBLWTL` and `TBLWTH` instructions to load two of the 24-bit instructions into the write latches found in configuration memory space. Refer to Figure 4-1 for write latch addresses. Programming is performed by unlocking and setting the control bits in the NVMCON register.

All erase and program operations may optionally use the NVM interrupt to signal the successful completion of the operation.

FIGURE 5-2: UNCOMPRESSED/ COMPRESSED FORMAT



A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. The processor stalls (waits) until the programming operation is finished. Setting the WR bit (NVMCON[15]) starts the operation and the WR bit is automatically cleared when the operation is finished. The WR bit is protected against an accidental write. To set this bit, 0x55 and 0xAA values must be written sequentially into the NVMKEY register. After the programming command (WR bit = `1`) has been executed, the user application must wait until programming is complete (WR bit = `0`). The two instructions following the start of the programming sequence should be `NOP`s.

Note:	MPLAB® XC16 provides a built-in C language function, including the unlocking sequence to set the WR bit in the NVMCON register: `__builtin_write_NVM()`

5.3 Program Flash Memory Control Registers

Six SFRs are used to write and erase the Program Flash Memory: NVMCON, NVMKEY, NVMADR/U and NVMSRCADRL/H.

The NVMCON register (Register 5-1) selects the operation to be performed (page erase, word/row program, Inactive Partition erase) and initiates the program or erase cycle.

NVMKEY (Register 5-4) is a write-only register that is used for write protection. To start a programming or erase sequence, the user application must consecutively write 0x55 and 0xAA to the NVMKEY register.

There are two NVM Address registers: NVMADRU and NVMADR. These two registers, when concatenated, form the 24-bit Effective Address (EA) of the selected word/row for programming operations, or the selected page for erase operations. The NVMADRU register is used to hold the upper eight bits of the EA, while the NVMADR register is used to hold the lower 16 bits of the EA.

For row programming operation, data to be written to Program Flash Memory are written into data memory space (RAM) at an address defined by the NVMSRCADRL/H register pair (location of first element in row programming data).

REGISTER 5-1: NVMCON: NONVOLATILE MEMORY (NVM) CONTROL REGISTER

R/SO-0[1,6]	R/W-0[1]	R/W-0[1]	R/W-0	U-0	U-0	R/W-0	R/C-0
WR	WREN	WRERR	NVMSIDL[2]	—	—	RPDF	URERR
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0[1]	R/W-0[1]	R/W-0[1]	R/W-0[1]
—	—	—	—	NVMOP[3:0][3,4]			
bit 7							bit 0

Legend:	C = Clearable bit	SO = Settable Only bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **WR:** Write Control bit[1,6]
1 = Initiates a Flash memory program or erase operation; the operation is self-timed and the bit is cleared by hardware once the operation is complete
0 = Program or erase operation is complete and inactive

bit 14 **WREN:** Write Enable bit[1]
1 = Enables Flash program/erase operations
0 = Inhibits Flash program/erase operations

bit 13 **WRERR:** Write Sequence Error Flag bit[1]
1 = An improper program or erase sequence attempt, or termination has occurred (bit is set automatically on any set attempt of the WR bit)
0 = The program or erase operation completed normally

bit 12 **NVMSIDL:** NVM Stop in Idle Control bit[2]
1 = Flash voltage regulator goes into Standby mode during Idle mode
0 = Flash voltage regulator is active during Idle mode

bit 11-10 **Unimplemented:** Read as ‘0’

bit 9 **RPDF:** Row Programming Data Format bit
1 = Row data to be stored in RAM are in compressed format
0 = Row data to be stored in RAM are in uncompressed format

bit 8 **URERR:** Row Programming Data Underrun Error bit
1 = Indicates row programming operation has been terminated
0 = No data underrun error is detected

bit 7-4 **Unimplemented:** Read as ‘0’

Note 1: These bits can only be reset on a POR.
2: If this bit is set, there will be minimal power savings (I_{IDLE}), and upon exiting Idle mode, there is a delay (T_{VREG}) before Flash memory becomes operational.
3: All other combinations of NVMOP[3:0] are unimplemented.
4: Execution of the PWRSAV instruction is ignored while any of the NVM operations are in progress.
5: Two adjacent words on a 4-word boundary are programmed during execution of this operation.
6: An unlock sequence is required to write to this bit (see **Section 5.2 “RTSP Operation”**).

REGISTER 5-1: NVMCON: NONVOLATILE MEMORY (NVM) CONTROL REGISTER (CONTINUED)

bit 3-0 **NVMOP[3:0]:** NVM Operation Select bits(1,3,4)

1111 = Reserved
1110 = User memory bulk erase operation
1101 = Reserved
1100 = Reserved
1011 = Reserved
1010 = Reserved
1001 = Reserved
1000 = Reserved
0111 = Reserved
0101 = Reserved
0100 = Reserved
0011 = Memory page erase operation
0010 = Memory row program operation
0001 = Memory double-word program operation(5)
0000 = Reserved

Note 1: These bits can only be reset on a POR.

2: If this bit is set, there will be minimal power savings (IIDLE), and upon exiting Idle mode, there is a delay (TVREG) before Flash memory becomes operational.

3: All other combinations of NVMOP[3:0] are unimplemented.

4: Execution of the PWRSAV instruction is ignored while any of the NVM operations are in progress.

5: Two adjacent words on a 4-word boundary are programmed during execution of this operation.

6: An unlock sequence is required to write to this bit (see **Section 5.2 "RTSP Operation"**).

REGISTER 5-2: NVMADR: NONVOLATILE MEMORY LOWER ADDRESS REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADR[15:8]							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **NVMADR[15:0]:** Nonvolatile Memory Lower Write Address bits
Selects the lower 16 bits of the location to program or erase in Program Flash Memory. This register may be read or written to by the user application.

REGISTER 5-3: NVMADRU: NONVOLATILE MEMORY UPPER ADDRESS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADRU[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **NVMADRU[23:16]:** Nonvolatile Memory Upper Write Address bits
Selects the upper eight bits of the location to program or erase in Program Flash Memory. This register may be read or written to by the user application.

REGISTER 5-4: NVMKEY: NONVOLATILE MEMORY KEY REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **NVMKEY[7:0]:** NVM Key Register bits (write-only)

REGISTER 5-5: NVMSRCADRL: NVM SOURCE DATA ADDRESS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADR[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **NVMSRCADR[15:0]:** NVM Source Data Address bits
The RAM address of the data to be programmed into Flash when the NVMOP[3:0] bits are set to row programming.

REGISTER 5-6: NVMSRCADRH: NVM SOURCE DATA ADDRESS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADR[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **NVMSRCADR[23:16]:** NVM Source Data Address bits
The RAM address of the data to be programmed into Flash when the NVMOP[3:0] bits are set to row programming.

5.4 Error Correcting Code (ECC)

In order to improve program memory performance and durability, these devices include Error Correcting Code (ECC) functionality as an integral part of the Flash memory controller. ECC can determine the presence of single-bit errors in program data, including which bit is in error, and correct the data automatically without user intervention. ECC cannot be disabled.

When data are written to program memory, ECC generates a 7-bit Hamming code parity value for every two (24-bit) instruction words. The data are stored in blocks of 48 data bits and seven parity bits; parity data are not memory-mapped and are inaccessible. When the data are read back, the ECC calculates the parity on them and compares it to the previously stored parity value. If a parity mismatch occurs, there are two possible outcomes:

- Single-bit error has occurred and has been automatically corrected on readback.
- Double-bit error has occurred and the read data are not changed.

Single-bit error occurrence can be identified by the state of the ECCSBEIF (IFS0[13]) bit. An interrupt can be generated when the corresponding interrupt enable bit is set, ECCSBEIE (IEC0[13]). The ECCSTATL register contains the parity information for single-bit errors. The SECOUT[7:0] bits field contains the expected calculated SEC parity and the SECIN[7:0] bits contain the actual value from a Flash read operation. The SECSYNDx bits (ECCSTATH[7:0]) indicate the bit position of the single-bit error within the 48-bit pair of instruction words. When no error is present, SECINx equals SECOUTx and SECSYNDx is zero. The ECCSTATL and ECCSTATH registers will only update and be valid when an error has occurred, or when included Fault injection is enabled, and an ECCADDR match occurs.

Double-bit errors result in a generic hard trap. The ECCDBE bit (INTCON4[1]) bit will be set to identify the source of the hard trap. If no Interrupt Service Routine is implemented for the hard trap, a device Reset will also occur. The ECCSTATH register contains double-bit error status information. The DEDOUT bit is the expected calculated DED parity and DEDIN is the actual value from a Flash read operation. When no error is present, DEDIN equals DEDOUT.

5.4.1 ECC FAULT INJECTION

To test Fault handling, an EEC error can be generated. Both single and double-bit errors can be generated in both the read and write data paths. Read path Fault injection first reads the Flash data and then modifies them prior to entering the ECC logic. Write path Fault injection modifies the actual data prior to them being written into the target Flash and will cause an EEC error on a subsequent Flash read. The following procedure is used to inject a Fault:

1. Load the Flash target address into the ECCADDR register.
2. Select 1st Fault bit determined by FLT1PTRx (ECCCONH[7:0]). The target bit is inverted to create the Fault.
3. If a double Fault is desired, select the 2nd Fault bit determined by FLT2PTRx (ECCCONH[15:8]), otherwise set to all '1's.
4. Write the NVMKEY unlock sequence (see **Section 5.3 "Program Flash Memory Control Registers"**).
5. Enable the ECC Fault injection logic by setting the FLTINJ bit (ECCCONL[0]).
6. Perform a read or write to the Flash target address.

5.4.2 ECC CONTROL REGISTERS

REGISTER 5-7: ECCCONL: ECC FAULT INJECTION CONFIGURATION REGISTER LOW

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	FLTINJ
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **Unimplemented:** Read as ‘0’

bit 0 **FLTINJ:** Fault Injection Sequence Enable bit
1 = Enabled
0 = Disabled

REGISTER 5-8: ECCCONH: ECC FAULT INJECTION CONFIGURATION REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLT2PTR[7:0]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLT1PTR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **FLT2PTR[7:0]:** ECC Fault Injection Bit Pointer 2 bits
11111111-00111000 = No Fault injection occurs
00110111 = Fault injection (bit inversion) occurs on bit 55 of ECC bit order
•
•
•
00000001 = Fault injection (bit inversion) occurs on bit 1 of ECC bit order
00000000 = Fault injection (bit inversion) occurs on bit 0 of ECC bit order

bit 7-0 **FLT1PTR[7:0]:** ECC Fault Injection Bit Pointer 1 bits
11111111-00111000 = No Fault injection occurs
00110111 = Fault injection occurs on bit 55 of ECC bit order
•
•
•
00000001 = Fault injection occurs on bit 1 of ECC bit order
00000000 = Fault injection occurs on bit 0 of ECC bit order

REGISTER 5-9: ECCADDRL: ECC FAULT INJECT ADDRESS COMPARE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCADDR[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCADDR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **ECCADDR[15:0]:** ECC Fault Injection NVM Address Match Compare bits

REGISTER 5-10: ECCADDRH: ECC FAULT INJECT ADDRESS COMPARE REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCADDR[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’
bit 7-0 **ECCADDR[23:16]:** ECC Fault Injection NVM Address Match Compare bits

REGISTER 5-11: ECCSTATL: ECC SYSTEM STATUS DISPLAY REGISTER LOW

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECOUT[7:0]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECIN[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **SECOUT[7:0]:** Calculated Single Error Correction Parity Value bits

bit 7-0 **SECIN[7:0]:** Read Single Error Correction Parity Value bits
SECIN[7:0] bits are the actual parity value of a Flash read operation.

REGISTER 5-12: ECCSTATH: ECC SYSTEM STATUS DISPLAY REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	DEDOUT	DEDIN
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECSYND[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9 **DEDOUT:** Calculated Dual Bit Error Detection Parity bit

bit 8 **DEDIN:** Read Dual Bit Error Detection Parity bit
DEDIN is the actual parity value of a Flash read operation.

bit 7-0 **SECSYND[7:0]:** Calculated ECC Syndrome Value bits
Indicates the bit location that contains the error.

5.5 Flash OTP by ICSP™ Write Inhibit

ICSP Write Inhibit is an access restriction feature, that when activated, restricts all of Flash memory. Once activated, ICSP Write Inhibit permanently prevents ICSP Flash programming and erase operations, and cannot be deactivated. This feature is intended to prevent alteration of Flash memory contents, with behavior similar to One-Time-Programmable (OTP) devices.

RTSP, including erase and programming operations, is not restricted when ICSP Write Inhibit is activated; however, code to perform these actions must be programmed into the device before ICSP Write Inhibit is activated. This allows for a bootloader-type application to alter Flash contents with ICSP Write Inhibit activated.

Entry into ICSP and Enhanced ICSP modes is not affected by ICSP Write Inhibit. In these modes, it will continue to be possible to read configuration memory space and any user memory space regions which are not code-protected. With ICSP writes inhibited, an attempt to set WR (NVMCON[15]) = `1` will maintain WR = `0`, and instead, set WRERR (NVMCON[13]) = `1`. All Enhanced ICSP erase and programming commands will have no effect with self-checked programming commands returning a FAIL response opcode (PASS if the destination already exactly matched the requested programming data).

Once ICSP Write Inhibit is activated, it is not possible for a device executing in Debug mode to erase/write Flash, nor can a debug tool switch the device to Production mode. ICSP Write Inhibit should therefore only be activated on devices programmed for production.

5.5.1 ACTIVATING FLASH OTP BY ICSP WRITE INHIBIT

Note: It is not possible to deactivate ICSP Write Inhibit.

ICSP Write Inhibit is activated by executing a pair of NVMCON double-word programming commands to save two 16-bit activation values in the configuration memory space. The target NVM addresses and values required for activation are shown in Table 5-1. Once both addresses contain their activation values, ICSP Write Inhibit will take permanent effect on the next device Reset.

Only the lower 16 data bits stored at the activation addresses are evaluated; the upper eight bits and second 24-bit word written by the double-word programming should be written as ‘`0`’s. The addresses can be programmed in any order and also during separate ICSP/Enhanced ICSP/RTSP sessions, but any attempt to program an incorrect 16-bit value or use a row programming operation to program the values will be aborted without altering the existing data.

TABLE 5-1: ICSP™ WRITE INHIBIT ACTIVATION ADDRESSES AND DATA

	Configuration Memory Address	ICSP™ Write Inhibit Activation Value
Write Lock 1	0x801028	`0x006D63`
Write Lock 2	0x80102C	`0x006870`

NOTES:

6.0 RESETS

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Reset"** (www.microchip.com/DS70602).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices.

The Reset module combines all Reset sources and controls the device Master Reset Signal, $\overline{\text{SYSRST}}$. The following is a list of device Reset sources:

- POR: Power-on Reset
- BOR: Brown-out Reset
- $\overline{\text{MCLR}}$: Master Clear Pin Reset
- SWR: `RESET` Instruction
- WDTO: Watchdog Timer Time-out Reset
- CM: Configuration Mismatch Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Condition Device Reset
  - Illegal Opcode Reset
  - Uninitialized W Register Reset
  - Security Reset

A simplified block diagram of the Reset module is shown in Figure 6-1.

Any active source of Reset will make the $\overline{\text{SYSRST}}$ signal active. On system Reset, some of the registers associated with the CPU and peripherals are forced to a known Reset state and some are unaffected.

> **Note:** Refer to the specific peripheral section or **Section 4.0 "Memory Organization"** of this manual for register Reset states.

All types of device Reset set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1).

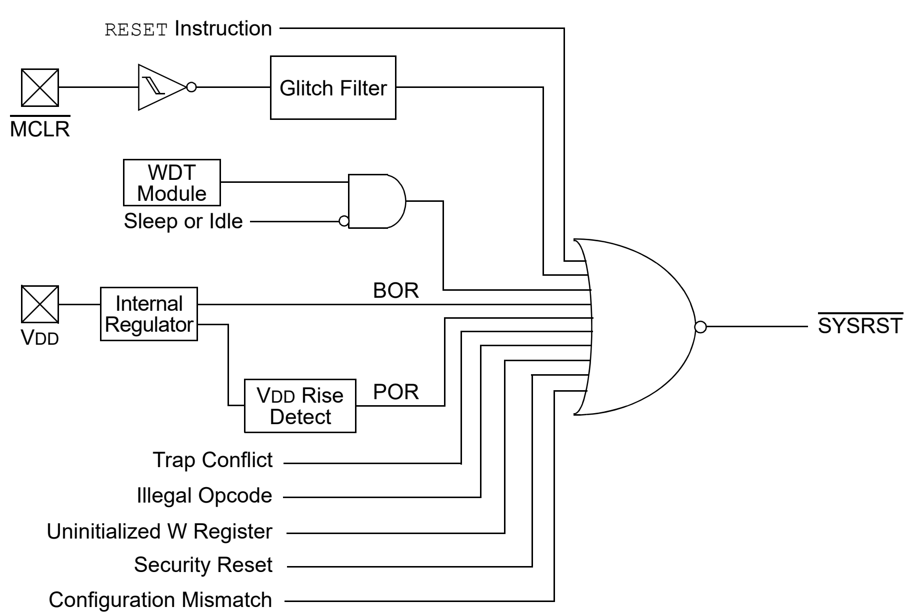
A POR clears all the bits, except for the BOR and POR bits (RCON[1:0]) that are set. The user application can set or clear any bit, at any time, during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software does not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer and device power-saving states. The function of these bits is discussed in other sections of this manual.

> **Note:** The status bits in the RCON register should be cleared after they are read so that the next RCON register value after a device Reset is meaningful.

For all Resets, the default clock source is determined by the FNOSC[2:0] bits in the FOSCSEL Configuration register. The value of the FNOSCx bits is loaded into the NOSC[2:0] (OSCCON[10:8]) bits on Reset, which in turn, initializes the system clock.

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



6.1 Reset Resources

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

6.1.1 KEY RESOURCES

- **"Reset"** (www.microchip.com/DS70602)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

REGISTER 6-1: RCON: RESET CONTROL REGISTER[1]

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	VREGS
bit 15							bit 8

R/W-0	R/W-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR	SWR	—	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **TRAPR:** Trap Reset Flag bit
1 = A Trap Conflict Reset has occurred
0 = A Trap Conflict Reset has not occurred

bit 14 **IOPUWR:** Illegal Opcode or Uninitialized W Register Access Reset Flag bit
1 = An illegal opcode detection, an illegal address mode or Uninitialized W register used as an Address Pointer caused a Reset
0 = An illegal opcode or Uninitialized W Register Reset has not occurred

bit 13-10 **Unimplemented:** Read as '0'

bit 9 **CM:** Configuration Mismatch Flag bit
1 = A Configuration Mismatch Reset has occurred.
0 = A Configuration Mismatch Reset has not occurred

bit 8 **VREGS:** Voltage Regulator Standby During Sleep bit
1 = Voltage regulator is active during Sleep
0 = Voltage regulator goes into Standby mode during Sleep

bit 7 **EXTR:** External Reset ($\overline{\text{MCLR}}$) Pin bit
1 = A Master Clear (pin) Reset has occurred
0 = A Master Clear (pin) Reset has not occurred

bit 6 **SWR:** Software RESET (Instruction) Flag bit
1 = A RESET instruction has been executed
0 = A RESET instruction has not been executed

bit 5 **Reserved:** Read as '0'

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
1 = WDT time-out has occurred
0 = WDT time-out has not occurred

bit 3 **SLEEP:** Wake-up from Sleep Flag bit
1 = Device has been in Sleep mode
0 = Device has not been in Sleep mode

bit 2 **IDLE:** Wake-up from Idle Flag bit
1 = Device has been in Idle mode
0 = Device has not been in Idle mode

bit 1 **BOR:** Brown-out Reset Flag bit
1 = A Brown-out Reset has occurred
0 = A Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit
1 = A Power-on Reset has occurred
0 = A Power-on Reset has not occurred

Note 1: All of the Reset status bits can be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

NOTES:

7.0 INTERRUPT CONTROLLER

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Interrupts"** (www.microchip.com/DS70000600).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices.

The dsPIC33CDVL64MC106 interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the dsPIC33CDVL64MC106 CPU.

The interrupt controller has the following features:

- Six Processor Exceptions and Software Traps
- Seven User-Selectable Priority Levels
- Interrupt Vector Table (IVT) with a Unique Vector for each Interrupt or Exception Source
- Fixed Priority within a Specified User Priority Level
- Fixed Interrupt Entry and Return Latencies
- Alternate Interrupt Vector Table (AIVT) for Debug Support

7.1 Interrupt Vector Table

The dsPIC33CDVL64MC106 Interrupt Vector Table (IVT), shown in Figure 7-1, resides in program memory, starting at location, 000004h. The IVT contains six non-maskable trap vectors and up to 246 sources of interrupts. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

Interrupt vectors are prioritized in terms of their natural priority. This priority is linked to their position in the vector table. Lower addresses generally have a higher natural priority. For example, the interrupt associated with Vector 0 takes priority over interrupts at any other vector address.

7.1.1 ALTERNATE INTERRUPT VECTOR TABLE

The Alternate Interrupt Vector Table (AIVT), shown in Figure 7-2, is available only when the Boot Segment (BS) is defined and the AIVT has been enabled. To enable the Alternate Interrupt Vector Table, the Configuration bits, BSEN and AIVTDIS in the FSEC register, must be programmed, and the AIVTEN bit must be set (INTCON2[8] = `1`). When the AIVT is enabled, all interrupt and exception processes use the alternate vectors instead of the default vectors. The AIVT begins at the start of the last page of the Boot Segment, defined by BSLIM[12:0]. The second half of the page is no longer usable space. The Boot Segment must be at least two pages to enable the AIVT.

> **Note:** Although the Boot Segment must be enabled in order to enable the AIVT, application code does not need to be present inside of the Boot Segment. The AIVT (and IVT) will inherit the Boot Segment code protection.

The AIVT supports debugging by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time.

7.2 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. The dsPIC33CDVL64MC106 family clears its registers in response to a Reset, which forces the PC to zero. The device then begins program execution at location, 0x000000. A `GOTO` instruction at the Reset address can redirect program execution to the appropriate start-up routine.

> **Note:** Any unimplemented or unused vector locations in the IVT should be programmed with the address of a default interrupt handler routine that contains a `RESET` instruction.

FIGURE 7-1: dsPIC33CDVL64MC106 FAMILY INTERRUPT VECTOR TABLE

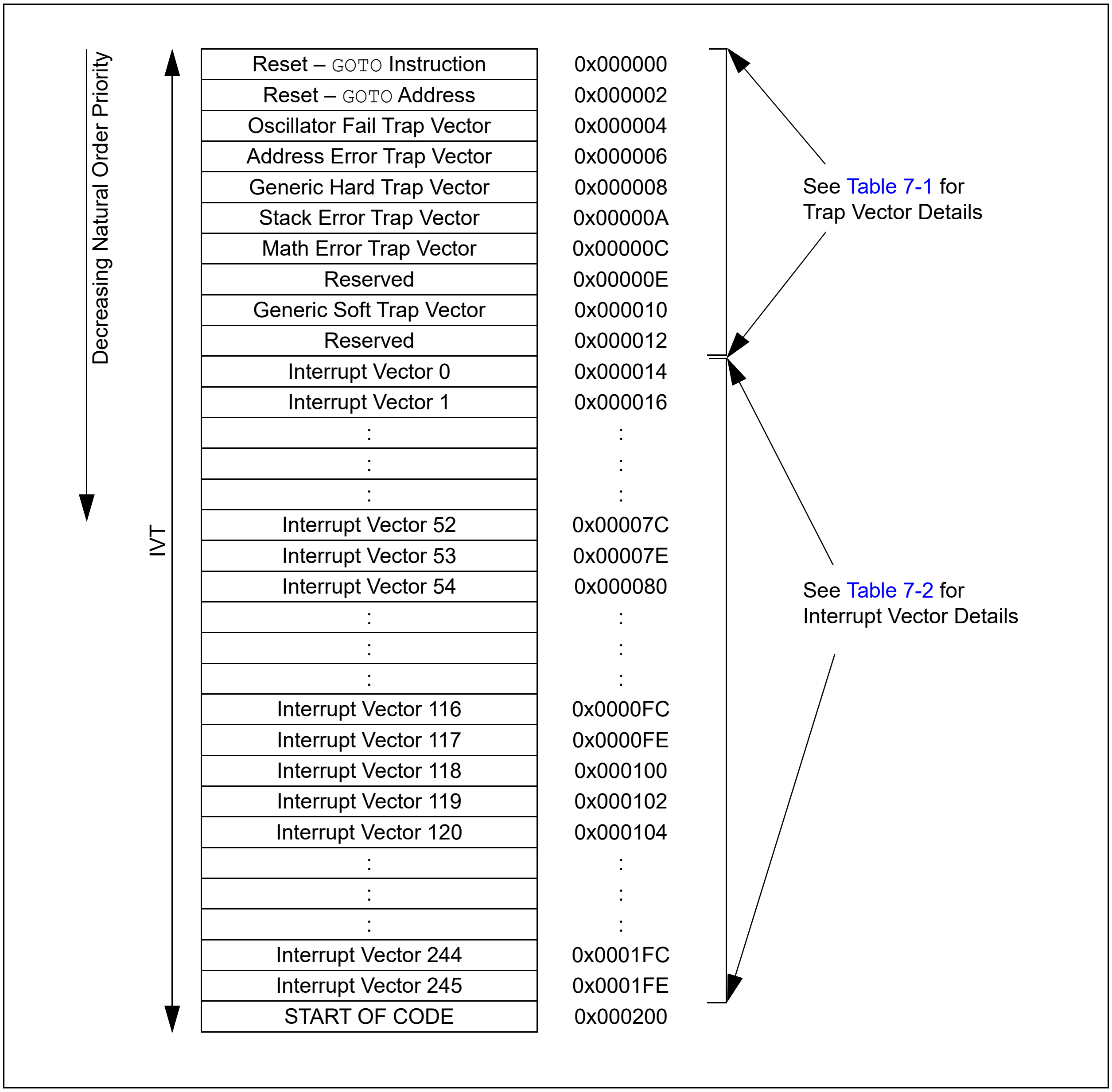


FIGURE 7-2: dsPIC33CDVL64MC106 FAMILY ALTERNATE INTERRUPT VECTOR TABLE

Decreasing Natural Order Priority

AIVT

Reserved	BSLIM[12:0][1] + 0x000000
Reserved	BSLIM[12:0][1] + 0x000002
Oscillator Fail Trap Vector	BSLIM[12:0][1] + 0x000004
Address Error Trap Vector	BSLIM[12:0][1] + 0x000006
Generic Hard Trap Vector	BSLIM[12:0][1] + 0x000008
Stack Error Trap Vector	BSLIM[12:0][1] + 0x00000A
Math Error Trap Vector	BSLIM[12:0][1] + 0x00000C
Reserved	BSLIM[12:0][1] + 0x00000E
Generic Soft Trap Vector	BSLIM[12:0][1] + 0x000010
Reserved	BSLIM[12:0][1] + 0x000012
Interrupt Vector 0	BSLIM[12:0][1] + 0x000014
Interrupt Vector 1	BSLIM[12:0][1] + 0x000016
:	:
:	:
:	:
Interrupt Vector 52	BSLIM[12:0][1] + 0x00007C
Interrupt Vector 53	BSLIM[12:0][1] + 0x00007E
Interrupt Vector 54	BSLIM[12:0][1] + 0x000080
:	:
:	:
:	:
Interrupt Vector 116	BSLIM[12:0][1] + 0x0000FC
Interrupt Vector 117	BSLIM[12:0][1] + 0x0000FE
Interrupt Vector 118	BSLIM[12:0][1] + 0x000100
Interrupt Vector 119	BSLIM[12:0][1] + 0x000102
Interrupt Vector 120	BSLIM[12:0][1] + 0x000104
:	:
:	:
:	:
Interrupt Vector 244	BSLIM[12:0][1] + 0x0001FC
Interrupt Vector 245	BSLIM[12:0][1] + 0x0001FE

See Table 7-1 for Trap Vector Details

See Table 7-2 for Interrupt Vector Details

Note 1: The address depends on the size of the Boot Segment defined by BSLIM[12:0]: [($\overline{\text{BSLIM[12:0]}}$ – 1) x 0x800] + Offset.

TABLE 7-1: TRAP VECTOR DETAILS

Trap Description	MPLAB® XC16 Trap ISR Name	IVT Address	Trap Bit Location			Priority
			Interrupt Flag	Type	Enable	
Oscillator Failure	_OscillatorFail	0x000004	INTCON1[1]	—	—	15
Address Error	_AddressError	0x000006	INTCON1[3]	—	—	14
ECC Double-Bit Error	_HardTrapError	0x000008	INTCON4[1]	—	—	13
Software Generated Trap	_HardTrapError	0x000008	INTCON4[0]	—	INTCON2[13]	13
Stack Error	_StackError	0x00000A	INTCON1[2]	—	—	12
Overflow Accumulator A	_MathError	0x00000C	INTCON1[4]	INTCON1[14]	INTCON1[10]	11
Overflow Accumulator B	_MathError	0x00000C	INTCON1[4]	INTCON1[13]	INTCON1[9]	11
Catastrophic Overflow Accumulator A	_MathError	0x00000C	INTCON1[4]	INTCON1[12]	INTCON1[8]	11
Catastrophic Overflow Accumulator B	_MathError	0x00000C	INTCON1[4]	INTCON1[11]	INTCON1[8]	11
Shift Accumulator Error	_MathError	0x00000C	INTCON1[4]	INTCON1[7]	INTCON1[8]	11
Divide-by-Zero Error	_MathError	0x00000C	INTCON1[4]	INTCON1[6]	INTCON1[8]	11
Reserved	Reserved	0x00000E	—	—	—	—
NVM Address Error	_SoftTrapError	0x000010	INTCON3[8]	—	—	9
DMA Address Error	_SoftTrapError	0x000010	INTCON3[5]	—	—	9
DO Stack Overflow	_SoftTrapError	0x000010	INTCON3[4]	—	—	9
Reserved	Reserved	0x000012	—	—	—	—

TABLE 7-2: INTERRUPT VECTOR DETAILS

Interrupt Source	MPLAB® XC16 ISR Name	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
					Flag	Enable	Priority
External Interrupt 0	_INT0Interrupt	8	0	0x000014	IFS0[0]	IEC0[0]	IPC0[2:0]
Timer1	_T1Interrupt	9	1	0x000016	IFS0[1]	IEC0[1]	IPC0[6:4]
Change Notice Interrupt A	_CNAInterrupt	10	2	0x000018	IFS0[2]	IEC0[2]	IPC0[10:8]
Change Notice Interrupt B	_CNBInterrupt	11	3	0x00001A	IFS0[3]	IEC0[3]	IPC0[14:12]
DMA Channel 0	_DMA0Interrupt	12	4	0x00001C	IFS0[4]	IEC0[4]	IPC1[2:0]
Reserved	Reserved	13	5	0x00001E	—	—	—
Input Capture/Output Compare 1	_CCP1Interrupt	14	6	0x000020	IFS0[6]	IEC0[6]	IPC1[10:8]
CCP1 Timer	_CCT1Interrupt	15	7	0x000022	IFS0[7]	IEC0[7]	IPC1[14:12]
DMA Channel 1	_DMA1Interrupt	16	8	0x000024	IFS0[8]	IEC0[8]	IPC2[2:0]
SPI1 Receiver	_SPI1RXInterrupt	17	9	0x000026	IFS0[9]	IEC0[9]	IPC2[6:4]
SPI1 Transmitter	_SPI1TXInterrupt	18	10	0x000028	IFS0[10]	IEC0[10]	IPC2[10:8]
UART1 Receiver	_U1RXInterrupt	19	11	0x00002A	IFS0[11]	IEC0[11]	IPC2[14:12]
UART1 Transmitter	_U1TXInterrupt	20	12	0x00002C	IFS0[12]	IEC0[12]	IPC3[2:0]
ECC Single-Bit Error	_ECCSBEInterrupt	21	13	0x00002E	IFS0[13]	IEC0[13]	IPC3[6:4]
NVM Write Complete	_NVMInterrupt	22	14	0x000030	IFS0[14]	IEC0[14]	IPC3[10:8]
External Interrupt 1	_INT1Interrupt	23	15	0x000032	IFS0[15]	IEC0[15]	IPC3[14:12]
I2C1 Slave Event	_SI2C1Interrupt	24	16	0x000034	IFS1[0]	IEC1[0]	IPC4[2:0]
I2C1 Master Event	_MI2C1Interrupt	25	17	0x000036	IFS1[1]	IEC1[1]	IPC4[6:4]
DMA Channel 2	_DMA2Interrupt	26	18	0x000038	IFS1[2]	IEC1[2]	IPC4[10:8]
Change Notice Interrupt C	_CNCInterrupt	27	19	0x00003A	IFS1[3]	IEC1[3]	IPC4[14:12]
External Interrupt 2	_INT2Interrupt	28	20	0x00003C	IFS1[4]	IEC1[4]	IPC5[2:0]
DMA Channel 3	_DMA3Interrupt	29	21	0x00003E	IFS1[5]	IEC1[5]	IPC5[6:4]
Reserved	Reserved	30	22	0x000040	—	—	—
Input Capture/Output Compare 2	_CCP2Interrupt	31	23	0x000042	IFS1[7]	IEC1[7]	IPC5[14:12]
CCP2 Timer	_CCT2Interrupt	32	24	0x000044	IFS1[8]	IEC1[8]	IPC6[2:0]
Reserved	Reserved	33	25	0x000046	—	—	—
External Interrupt 3	_INT3Interrupt	34	26	0x000048	IFS1[10]	IEC1[10]	IPC6[10:8]
U2RX – UART2 Receiver	_U2RXInterrupt	35	27	0x00004A	IFS1[11]	IEC1[11]	IPC6[14:12]
U2TX – UART2 Transmitter	_U2TXInterrupt	36	28	0x00004C	IFS1[12]	IEC1[12]	IPC7[2:0]
SPI2 Receiver	_SPI2RXInterrupt	37	29	0x00004E	IFS1[13]	IEC1[13]	IPC7[6:4]
SPI2 Transmitter	_SPI2TXInterrupt	38	30	0x000050	IFS1[14]	IEC1[14]	IPC7[10:8]
Reserved	Reserved	39-42	31-34	0x000052-0x000058	—	—	—
Input Capture/Output Compare 3	_CCP3Interrupt	43	35	0x00005A	IFS2[3]	IEC2[3]	IPC8[14:12]
CCP3 Timer	_CCT3Interrupt	44	36	0x00005C	IFS2[4]	IEC2[4]	IPC9[2:0]
Reserved	Reserved	45-47	37-39	0x00005E-0x000062	—	—	—
Input Capture/Output Compare 4	_CCP4Interrupt	48	40	0x000064	IFS2[8]	IEC2[8]	IPC10[2:0]
CCP4 Timer	_CCT4Interrupt	49	41	0x000066	IFS2[9]	IEC2[9]	IPC10[6:4]
Reserved	Reserved	50-52	42-44	0x000068-0x00006C	—	—	—
Deadman Timer	_DMTInterrupt	53	45	0x00006E	IFS2[13]	IEC2[13]	IPC11[6:4]
Reserved	Reserved	54-55	46-47	0x000070-0x000072	—	—	—
QEI Position Counter Compare	_QEI1Interrupt	56	48	0x000074	IFS3[0]	IEC3[0]	IPC12[2:0]
UART1 Error	_U1EInterrupt	57	49	0x000076	IFS3[1]	IEC3[1]	IPC12[6:4]
UART2 Error	_U2EInterrupt	58	50	0x000078	IFS3[2]	IEC3[2]	IPC12[10:8]
CRC Generator	_CRCInterrupt	59	51	0x00007A	IFS3[3]	IEC3[3]	IPC12[14:12]

TABLE 7-2: INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	MPLAB® XC16 ISR Name	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
					Flag	Enable	Priority
Reserved	Reserved	60-63	52-55	0x00007C-0x000082	—	—	—
UART3 Error	_U3EInterrupt	64	56	0x000084	IFS3[8]	IEC3[8]	IPC14[2:0]
UART3 Receiver	_U3RXInterrupt	65	57	0x000086	IFS3[9]	IEC3[9]	IPC14[6:4]
UART3 Transmitter	_U3TXInterrupt	66	58	0x000088	IFS3[10]	IEC3[10]	IPC14[10:8]
Reserved	Reserved	67-68	59-60	0x00008A-0x00008C	—	—	—
In-Circuit Debugger	_ICDInterrupt	69	61	0x00008E	IFS3[13]	IEC3[13]	IPC15[6:4]
Reserved	Reserved	70	62	0x000090	—	—	—
PTG Step	_PTGSTEPInterrupt	71	63	0x000092	IFS3[15]	IEC3[15]	IPC15[14:12]
I2C1 Bus Collision	_I2C1BCInterrupt	72	64	0x000094	IFS4[0]	IEC4[0]	IPC16[2:0]
Reserved	Reserved	73-74	65-66	0x000096-0x000098	—	—	—
PWM Generator 1	_PWM1Interrupt	75	67	0x00009A	IFS4[3]	IEC4[3]	IPC16[14:12]
PWM Generator 2	_PWM2Interrupt	76	68	0x00009C	IFS4[4]	IEC4[4]	IPC17[2:0]
PWM Generator 3	_PWM3Interrupt	77	69	0x00009E	IFS4[5]	IEC4[5]	IPC17[6:4]
PWM Generator 4	_PWM4Interrupt	78	70	0x0000A0	IFS4[6]	IEC4[6]	IPC17[10:8]
Reserved	Reserved	79-82	71-74	0x0000A2-0x0000A8	—	—	—
Change Notice D	_CNDInterrupt	83	75	0x0000AA	IFS4[11]	IEC4[11]	IPC18[14:12]
Reserved	Reserved	84	76	0x0000AC	—	—	—
Comparator 1	_CMP1Interrupt	85	77	0x0000AE	IFS4[13]	IEC4[13]	IPC19[6:4]
Reserved	Reserved	86-88	78-80	0x0000B0-0x0000B4	—	—	—
PTG Watchdog Timer Time-out	_PTGWDTInterrupt	89	81	0x0000B6	IFS5[1]	IEC5[1]	IPC20[6:4]
PTG Trigger 0	_PTG0Interrupt	90	82	0x0000B8	IFS5[2]	IEC5[2]	IPC20[10:8]
PTG Trigger 1	_PTG1Interrupt	91	83	0x0000BA	IFS5[3]	IEC5[3]	IPC20[14:12]
PTG Trigger 2	_PTG2Interrupt	92	84	0x0000BC	IFS5[4]	IEC5[4]	IPC21[2:0]
PTG Trigger 3	_PTG3Interrupt	93	85	0x0000BE	IFS5[5]	IEC5[6]	IPC21[6:4]
SENT1 TX/RX	_SENT1Interrupt	94	86	0x0000C0	IFS5[6]	IEC5[6]	IPC21[10:8]
SENT1 Error	_SENT1EInterrupt	95	87	0x0000C2	IFS5[7]	IEC5[7]	IPC21[14:12]
Reserved	Reserved	96-97	88-89	0x0000C4-0x0000C6	—	—	—
ADC Global Interrupt	_ADCInterrupt	98	90	0x0000C8	IFS5[10]	IEC5[10]	IPC22[10:8]
ADC AN0 Interrupt	_ADCAN0Interrupt	99	91	0x0000CA	IFS5[11]	IEC5[11]	IPC22[14:12]
ADC AN1 Interrupt	_ADCAN1Interrupt	100	92	0x0000CC	IFS5[12]	IEC5[12]	IPC23[2:0]
ADC AN2 Interrupt	_ADCAN2Interrupt	101	93	0x0000CE	IFS5[13]	IEC5[13]	IPC23[6:4]
ADC AN3 Interrupt	_ADCAN3Interrupt	102	94	0x0000D0	IFS5[14]	IEC5[14]	IPC23[10:8]
ADC AN4 Interrupt	_ADCAN4Interrupt	103	95	0x0000D2	IFS5[15]	IEC5[15]	IPC23[14:12]
ADC AN5 Interrupt	_ADCAN5Interrupt	104	96	0x0000D4	IFS6[0]	IEC6[0]	IPC24[2:0]
ADC AN6 Interrupt	_ADCAN6Interrupt	105	97	0x0000D6	IFS6[1]	IEC6[1]	IPC24[6:4]
ADC AN7 Interrupt	_ADCAN7Interrupt	106	98	0x0000D8	IFS6[2]	IEC6[2]	IPC24[10:8]
ADC AN8 Interrupt	_ADCAN8Interrupt	107	99	0x0000DA	IFS6[3]	IEC6[3]	IPC24[14:12]
ADC AN9 Interrupt	_ADCAN9Interrupt	108	100	0x0000DC	IFS6[4]	IEC6[4]	IPC25[2:0]
ADC AN10 Interrupt	_ADCAN10Interrupt	109	101	0x0000DE	IFS6[5]	IEC6[5]	IPC25[6:4]
ADC AN11 Interrupt	_ADCAN11Interrupt	110	102	0x0000E0	IFS6[6]	IEC6[6]	IPC25[10:8]
ADC AN12 Interrupt	_ADCAN12Interrupt	111	103	0x0000E2	IFS6[7]	IEC6[7]	IPC25[14:12]
ADC AN13 Interrupt	_ADCAN13Interrupt	112	104	0x0000E4	IFS6[8]	IEC6[8]	IPC26[2:0]
ADC AN14 Interrupt	_ADCAN14Interrupt	113	105	0x0000E6	IFS6[9]	IEC6[9]	IPC26[6:4]
ADC AN15 Interrupt	_ADCAN15Interrupt	114	106	0x0000E8	IFS6[10]	IEC6[10]	IPC26[10:8]
Reserved	Reserved	109-114	101-106	0x0000DE-0x0000E8	—	—	—
ADC AN16 Interrupt	_ADCAN16Interrupt	115	107	0x0000EA	IFS6[11]	IEC6[11]	IPC26[14:12]
ADC AN17 Interrupt	_ADCAN17Interrupt	116	108	0x0000EC	IFS6[12]	IEC6[12]	IPC27[2:0]
Reserved	Reserved	117-123	109-115	0x0000EE-0x0000FA	—	—	—

TABLE 7-2: INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	MPLAB® XC16 ISR Name	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
					Flag	Enable	Priority
ADC Digital Comparator 0	_ADCMP0Interrupt	124	116	0x0000FC	IFS7[4]	IEC7[4]	IPC29[2:0]
ADC Digital Comparator 1	_ADCMP1Interrupt	125	117	0x0000FE	IFS7[5]	IEC7[5]	IPC29[6:4]
ADC Digital Comparator 2	_ADCMP2Interrupt	126	118	0x000100	IFS7[6]	IEC7[6]	IPC29[10:8]
ADC Digital Comparator 3	_ADCMP3Interrupt	127	119	0x000102	IFS7[7]	IEC7[7]	IPC29[14:12]
ADC Oversample Filter 0	_ADFLTR0Interrupt	128	120	0x000104	IFS7[8]	IEC7[8]	IPC30[2:0]
ADC Oversample Filter 1	_ADFLTR1Interrupt	129	121	0x000106	IFS7[9]	IEC7[9]	IPC30[6:4]
ADC Oversample Filter 2	_ADFLTR2Interrupt	130	122	0x000108	IFS7[10]	IEC7[10]	IPC30[10:8]
ADC Oversample Filter 3	_ADFLTR3Interrupt	131	123	0x00010A	IFS7[11]	IEC7[11]	IPC30[14:12]
CLC1 Positive Edge	_CLC1PInterrupt	132	124	0x00010C	IFS7[12]	IEC7[12]	IPC31[2:0]
CLC2 Positive Edge	_CLC2PInterrupt	133	125	0x00010E	IFS7[13]	IEC7[13]	IPC31[6:4]
SPI1 Error	_SPI1Interrupt	134	126	0x000110	IFS7[14]	IEC7[14]	IPC31[10:8]
SPI2 Error	_SPI2Interrupt	135	127	0x000112	IFS7[15]	IEC7[15]	IPC31[14:12]
Reserved	Reserved	136-176	128-168	0x000114-0x000164	—	—	—
PEVTA – PWM Event A	_PEVTAInterrupt	177	169	0x000166	IFS10[9]	IEC10[9]	IPC42[6:4]
PEVTB – PWM Event B	_PEVTBInterrupt	178	170	0x000168	IFS10[10]	IEC10[10]	IPC42[10:8]
PEVTC – PWM Event C	_PEVTCInterrupt	179	171	0x00016A	IFS10[11]	IEC10[11]	IPC42[14:12]
PEVTD – PWM Event D	_PEVTDInterrupt	180	172	0x00016C	IFS10[12]	IEC10[12]	IPC43[2:0]
PEVTE – PWM Event E	_PEVTEInterrupt	181	173	0x00016E	IFS10[13]	IEC10[13]	IPC43[6:4]
PEVTF – PWM Event F	_PEVTFInterrupt	182	174	0x000170	IFS10[14]	IEC10[14]	IPC43[10:8]
CLC3 Positive Edge	_CLC3PInterrupt	183	175	0x000172	IFS10[15]	IEC10[15]	IPC43[14:12]
CLC4 Positive Edge	_CLC4PInterrupt	184	176	0x000174	IFS11[0]	IEC11[0]	IPC44[2:0]
CLC1 Negative Edge	_CLC1NInterrupt	185	177	0x000176	IFS11[1]	IEC11[1]	IPC44[6:4]
CLC2 Negative Edge	_CLC2NInterrupt	186	178	0x000178	IFS11[2]	IEC11[2]	IPC44[10:8]
CLC3 Negative Edge	_CLC3NInterrupt	187	179	0x00017A	IFS11[3]	IEC11[3]	IPC44[14:]12]
CLC4 Negative Edge	_CLC4NInterrupt	188	180	0x00017C	IFS11[4]	IEC11[4]	IPC45[2:0]
Reserved	Reserved	189-196	181-188	0x0017E-0x0018C	—	—	—
UART1 Event	_U1EVTInterrupt	197	189	0x00018E	IFS11[13]	IEC11[13]	IPC47[6:4]
UART2 Event	_U2EVTInterrupt	198	190	0x000190	IFS11[14]	IEC11[14]	IPC47[12:8]
UART3 Event	_U3EVTInterrupt	199	191	0x000192	IFS11[15]	IEC11[15]	IPC47[14:12]
Reserved	Reserved	200-255	192-247	0x000194-0x0001FE	—	—	—

TABLE 7-3: INTERRUPT FLAG REGISTERS

Register	Address	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IFS0	800h	INT1IF	NVMIF	ECCSBEIF	U1TXIF	U1RXIF	SPI1TXIF	SPI1RXIF	DMA1IF	CCT1IF	CCP1IF	—	DMA0IF	CNBIF	CNAIF	T1IF	INT0IF
IFS1	802h	—	SPI2TXIF	SPI2RXIF	U2TXIF	U2RXIF	INT3IF	—	CCT2IF	CCP2IF	—	DMA3IF	INT2IF	CNCIF	DMA2IF	MI2C1IF	SI2C1IF
IFS2	804h	—	—	DMTIF	—	—	—	CCT4IF	CCP4IF	—	—	—	CCT3IF	CCP3IF	—	—	—
IFS3	806h	PTGSTEPIF	—	ICDIF	—	—	U3TXIF	U3RXIF	U3EIF	—	—	—	—	CRCIF	U2EIF	U1EIF	QEI1IF
IFS4	808h	—	—	CMP1IF	—	CNDIF	—	—	—	—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	I2C1BCIF
IFS5	80Ah	ADCAN4IF	ADCAN3IF	ADCAN2IF	ADCAN1IF	ADCAN0IF	ADCIF	—	—	SENT1EIF	SENT1IF	PTG3IF	PTG2IF	PTG1IF	PTG0IF	PTGWDTIF	—
IFS6	80Ch	—	—	—	ADCAN17IF	ADCAN16IF	ADCAN15IF	ADCAN14IF	ADCAN13IF	ADCAN12IF	ADCAN11IF	ADCAN10IF	ADCAN9IF	ADCAN8IF	ADCAN7IF	ADCAN6IF	ADCAN5IF
IFS7	80Eh	SPI2GIF	SPI1GIF	CLC2PIF	CLC1PIF	ADFLTR3IF	ADFLTR2IF	ADFLTR1IF	ADFLTR0IF	ADCMP3IF	ADCMP2IF	ADCMP1IF	ADCMP0IF	—	—	—	—
IFS10	814h	CLC3PIF	PEVTFIF	PEVTEIF	PEVTDIF	PEVTCIF	PEVTBIF	PEVTAIF	—	—	—	—	—	—	—	—	—
IFS11	816h	U3EVTIF	U2EVTIF	U1EVTIF	—	—	—	—	—	—	—	—	CLC4NIF	CLC3NIF	CLC2NIF	CLC1NIF	CLC4PIF

Legend: — = Unimplemented.

TABLE 7-4: INTERRUPT ENABLE REGISTERS

Register	Address	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEC0	820h	INT1IE	NVMIE	ECCSBEIE	U1TXIE	U1RXIE	SPI1TXIE	SPI1RXIE	DMA1IE	CCT1IE	CCP1IE	—	DMA0IE	CNBIE	CNAIE	T1IE	INT0IE
IEC1	822h	—	SPI2TXIE	SPI2RXIE	U2TXIE	U2RXIE	INT3IE	—	CCT2IE	CCP2IE	—	DMA3IE	INT2IE	CNCIE	DMA2IE	MI2C1IE	SI2C1IE
IEC2	824h	—	—	DMTIE	—	—	—	CCT4IE	CCP4IE	—	—	—	CCT3IE	CCP3IE	—	—	—
IEC3	826h	PTGSTEPIE	—	ICDIE	—	—	U3TXIE	U3RXIE	U3EIE	—	—	—	—	CRCIE	U2EIE	U1EIE	QEI1IE
IEC4	828h	—	—	CMP1IE	—	CNDIE	—	—	—	—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	I2C1BCIE
IEC5	82Ah	ADCAN4IE	ADCAN3IE	ADCAN2IE	ADCAN1IE	ADCAN0IE	ADCIE	—	—	SENT1EIE	SENT1IE	PTG3IE	PTG2IE	PTG1IE	PTG0IE	PTGWDTIE	—
IEC6	82Ch	—	—	—	ADCAN17IE	ADCAN16IE	ADCAN15IE	ADCAN14IE	ADCAN13IE	ADCAN12IE	ADCAN11IE	ADCAN10IE	ADCAN9IE	ADCAN8IE	ADCAN7IE	ADCAN6IE	ADCAN5IE
IEC7	82Eh	SPI2GIE	SPI1GIE	CLC2PIE	CLC1PIE	ADFLTR3IE	ADFLTR2IE	ADFLTR1IE	ADFLTR0IE	ADCMP3IE	ADCMP2IE	ADCMP1IE	ADCMP0IE	—	—	—	—
IEC10	834h	CLC3PIE	PEVTFIE	PEVTEIE	PEVTDIE	PEVTCIE	PEVTBIE	PEVTAIE	—	—	—	—	—	—	—	—	—
IEC11	836h	U3EVTIE	U2EVTIE	U1EVTIE	—	—	—	—	—	—	—	—	CLC4NIE	CLC3NIE	CLC2NIE	CLC1NIE	CLC4PIE

Legend: — = Unimplemented.

TABLE 7-5: INTERRUPT PRIORITY REGISTERS

Register	Address	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPC0	840h	—	CNBIP2	CNBIP1	CNBIP0	—	CNAIP2	CNAIP1	CNAIP0	—	T1IP2	T1IP1	T1IP0	—	INT0IP2	INT0IP1	INT0IP0
IPC1	842h	—	CCT1IP2	CCT1IP1	CCT1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0	—	—	—	—	—	DMA0IP2	DMA0IP1	DMA0IP0
IPC2	844h	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1TXIP2	SPI1TXIP1	SPI1TXIP0	—	SPI1RXIP2	SPI1RXIP1	SPI1RXIP0	—	DMA1IP2	DMA1IP1	DMA1IP0
IPC3	846h	—	INT1IP2	INT1IP1	INT1IP0	—	NVMIP2	NVMIP1	NVMIP0	—	ECCSBEIP2	ECCSBEIP1	ECCSBEIP0	—	U1TXIP2	U1TXIP1	U1TXIP0
IPC4	848h	—	CNCIP2	CNCIP1	CNCIP0	—	DMA2IP2	DMA2IP1	DMA2IP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
IPC5	84Ah	—	CCP2IP2	CCP2IP1	CCP2IP0	—	—	—	—	—	DMA3IP2	DMA3IP1	DMA3IP20	—	INT2IP2	INT2IP1	INT2IP0
IPC6	84Ch	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	—	CCT2IP2	CCT2IP1	CCT2IP0
IPC7	84Eh	—	—	—	—	—	SPI2TXIP2	SPI2TXIP1	SPI2TXIP0	—	SPI2RXIP2	SPI2RXIP1	SPI2RXIP0	—	U2TXIP2	U2TXIP1	U2TXIP0
IPC8	850h	—	CCP3IP2	CCP3IP1	CCP3IP0	—	—	—	—	—	—	—	—	—	—	—	—
IPC9	852h	—	—	—	—	—	—	—	—	—	—	—	—	—	CCT3IP2	CCT3IP1	CCT3IP0
IPC10	854h	—	CCP5IP2	CCP5IP1	CCP5IP0	—	—	—	—	—	CCT4IP2	CCT4IP1	CCT4IP0	—	CCP4IP2	CCP4IP1	CCP4IP0
IPC11	856h	—	—	—	—	—	—	—	—	—	DMTIP2	DMTIP1	DMTIP0	—	—	—	—
IPC12	858h	—	CRCIP2	CRCIP1	CRCIP0	—	U2EIP2	U2EIP1	U2EIP0	—	U1EIP2	U1EIP1	U1EIP0	—	QEI1IP2	QEI1IP1	QEI1IP0
IPC14	85Ch	—	—	—	—	—	U3TXIP2	U3TXIP1	U3TXIP1	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3EIP2	U3EIP1	U3EIP0
IPC15	85Eh	—	PTGSTEPIP2	PTGSTEPIP1	PTGSTEPIP0	—	—	—	—	—	ICDIP2	ICDIP1	ICDIP0	—	—	—	—
IPC16	860h	—	PWM1IP2	PWM1IP1	PWM1IP0	—	—	—	—	—	—	—	—	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0
IPC17	862h	—	—	—	—	—	PWM4IP2	PWM4IP1	PWM4IP0	—	PWM3IP2	PWM3IP1	PWM3IP0	—	PWM2IP2	PWM2IP1	PWM2IP0
IPC18	864h	—	CNDIP2	CNDIP1	CNDIP0	—	—	—	—	—	—	—	—	—	—	—	—
IPC19	866h	—	—	—	—	—	—	—	—	—	CMP1IP2	CMP1IP1	CMP1IP0	—	—	—	—
IPC20	868h	—	PTG1IP2	PTG1IP1	PTG1IP0	—	PTG0IP2	PTG0IP1	PTG0IP0	—	PTGWDTIP2	PTGWDTIP1	PTGWDTIP0	—	—	—	—
IPC21	86Ah	—	SENT1EIP2	SENT1EIP1	SENT1EIP0	—	SENT1IP2	SENT1IP1	SENT1IP0	—	PTG3IP2	PTG3IP1	PTG3IP0	—	PTG2IP2	PTG2IP1	PTG2IP0
IPC22	86Ch	—	ADCAN0IP2	ADCAN0IP1	ADCAN0IP0	—	ADCIP2	ADCIP1	ADCIP0	—	—	—	—	—	—	—	—
IPC23	86Eh	—	ADCAN4IP2	ADCAN4IP1	ADCAN4IP0	—	ADCAN3IP2	ADCAN3IP1	ADCAN3IP0	—	ADCAN2IP2	ADCAN2IP1	ADCAN2IP0	—	ADCAN1IP2	ADCAN1IP1	ADCAN1IP0
IPC24	870h	—	ADCAN8IP2	ADCAN8IP1	ADCAN8IP0	—	ADCAN7IP2	ADCAN7IP1	ADCAN7IP0	—	ADCAN6IP2	ADCAN6IP1	ADCAN6IP0	—	ADCAN5IP2	ADCAN5IP1	ADCAN5IP0
IPC25	872h	—	ADCAN12IP2	ADCAN12IP1	ADCAN12IP0	—	ADCAN11IP2	ADCAN11IP1	ADCAN11IP0	—	ADCAN10IP2	ADCAN10IP1	ADCAN10IP0	—	ADCAN9IP2	ADCAN9IP1	ADCAN9IP0
IPC26	874h	—	ADCAN16IP2	ADCAN16IP2	ADCAN16IP2	—	ADCAN15IP2	ADCAN15IP1	ADCAN15IP0	—	ADCAN14IP2	ADCAN14IP1	ADCAN14IP0	—	ADCAN13IP2	ADCAN13IP1	ADCAN13IP0
IPC27	876h	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCAN17IP2	ADCAN17IP1	ADCAN17IP0
IPC29	87Ah	—	ADCMP3IP2	ADCMP3IP1	ADCMP3IP0	—	ADCMP2IP2	ADCMP2IP1	ADCMP2IP0	—	ADCMP1IP2	ADCMP1IP1	ADCMP1IP0	—	ADCMP0IP2	ADCMP0IP1	ADCMP0IP0
IPC30	87Ch	—	ADFLTR3IP2	ADFLTR3IP1	ADFLTR3IP0	—	ADFLTR2IP2	ADFLTR2IP1	ADFLTR2IP0	—	ADFLTR1IP2	ADFLTR1IP1	ADFLTR1IP0	—	ADFLTR0IP2	ADFLTR0IP1	ADFLTR0IP0
IPC31	87Eh	—	SPI2GIP0	SPI2GIP1	SPI2GIP0	—	SPI1GIP2	SPI1GIP1	SPI1GIP0	—	CLC2PIP2	CLC2PIP1	CLC2PIP0	—	CLC1PIP2	CLC1PIP1	CLC1PIP0
IPC42	894h	—	PEVTCIP2	PEVTCIP1	PEVTCIP0	—	PEVTBIP2	PEVTBIP1	PEVTBIP0	—	PEVTAIP2	PEVTAIP1	PEVTAIP0	—	—	—	—
IPC43	896h	—	CLC3PIP2	CLC3PIP1	CLC3PIP0	—	PEVTFIP2	PEVTFIP1	PEVTFIP0	—	PEVTEIP2	PEVTEIP1	PEVTEIP0	—	PEVTDIP2	PEVTDIP1	PEVTDIP0
IPC44	898h	—	CLC3NIP2	CLC3NIP1	CLC3NIP0	—	CLC2NIP2	CLC2NIP1	CLC2NIP0	—	CLC1NIP2	CLC1NIP1	CLC1NIP0	—	CLC4PIP2	CLC4PIP1	CLC4PIP0
IPC45	89Ah	—	—	—	—	—	—	—	—	—	—	—	—	—	CLC4NIP2	CLC4NIP1	CLC4NIP0
IPC47	89Eh	—	U3EVTIP2	U3EVTIP1	U3EVTIP0	—	U2EVTIP2	U2EVTIP1	U2EVTIP0	—	U1EVTIP2	U1EVTIP1	U1EVTIP0	—	—	—	—

Legend: — = Unimplemented.

7.3 Interrupt Resources

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

7.3.1 KEY RESOURCES

- **"Interrupts"** (www.microchip.com/DS70000600)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

7.4 Interrupt Control and Status Registers

The dsPIC33CDVL64MC106 devices implement the following registers for the interrupt controller:

- INTCON1
- INTCON2
- INTCON3
- INTCON4
- INTTREG

7.4.1 INTCON1 THROUGH INTCON4

Global interrupt control functions are controlled from INTCON1, INTCON2, INTCON3 and INTCON4.

INTCON1 contains the Interrupt Nesting Disable bit (NSTDIS), as well as the control and status flags for the processor trap sources.

The INTCON2 register controls external interrupt request signal behavior, contains the Global Interrupt Enable bit (GIE) and the Alternate Interrupt Vector Table Enable bit (AIVTEN).

INTCON3 contains the status flags for the `DO` stack overflow status trap sources.

The INTCON4 register contains the Software Generated Hard Trap Status bit (SGHT).

7.4.2 IFSx

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit, which is set by the respective peripherals or external signal and is cleared via software.

7.4.3 IECx

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

7.4.4 IPCx

The IPCx registers are used to set the Interrupt Priority Level (IPL) for each source of interrupt. Each user interrupt source can be assigned to one of seven priority levels.

7.4.5 INTTREG

The INTTREG register contains the associated interrupt vector number and the new CPU Interrupt Priority Level, which are latched into the Vector Number (VECNUM[7:0]) and Interrupt Level bits (ILR[3:0]) fields in the INTTREG register. The new Interrupt Priority Level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence as they are listed in Table 7-2. For example, INT0 (External Interrupt 0) is shown as having Vector Number 8 and a natural order priority of 0. Thus, the INT0IF bit is found in IFS0[0], the INT0IE bit in IEC0[0] and the INT0IP[2:0] bits in the first position of IPC0 (IPC0[2:0]).

7.4.6 STATUS/CONTROL REGISTERS

Although these registers are not specifically part of the interrupt control hardware, two of the CPU Control registers contain bits that control interrupt functionality. For more information on these registers, refer to **"Enhanced CPU"** (www.microchip.com/DS70005158).

- The CPU STATUS Register, SR, contains the IPL[2:0] bits (SR[7:5]). These bits indicate the current CPU Interrupt Priority Level. The user software can change the current CPU Interrupt Priority Level by writing to the IPLx bits.
- The CORCON register contains the IPL3 bit, which together with IPL[2:0], also indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

All Interrupt registers are described in Register 7-3 through Register 7-7 in the following pages.

REGISTER 7-1: SR: CPU STATUS REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/C-0	R/C-0	R-0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8

R/W-0[3]	R/W-0[3]	R/W-0[3]	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[2]	IPL1[2]	IPL0[2]	RA	N	OV	Z	C
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’= Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **IPL[2:0]:** CPU Interrupt Priority Level Status bits[2,3]
`111` = CPU Interrupt Priority Level is 7 (15); user interrupts are disabled
`110` = CPU Interrupt Priority Level is 6 (14)
`101` = CPU Interrupt Priority Level is 5 (13)
`100` = CPU Interrupt Priority Level is 4 (12)
`011` = CPU Interrupt Priority Level is 3 (11)
`010` = CPU Interrupt Priority Level is 2 (10)
`001` = CPU Interrupt Priority Level is 1 (9)
`000` = CPU Interrupt Priority Level is 0 (8)

Note 1: For complete register details, see Register 3-1.
2: The IPL[2:0] bits are concatenated with the IPL[3] bit (CORCON[3]) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL, if IPL[3] = `1`. User interrupts are disabled when IPL[3] = `1`.
3: The IPL[2:0] Status bits are read-only when the NSTDIS bit (INTCON1[15]) = `1`.

REGISTER 7-2: CORCON: CORE CONTROL REGISTER[1]

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
VAR	—	US1	US0	EDT	DL2	DL1	DL0
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3[2]	SFA	RND	IF
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’= Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **VAR:** Variable Exception Processing Latency Control bit
1 = Variable exception processing latency is enabled
0 = Fixed exception processing latency is enabled

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit 3[2]
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

Note 1: For complete register details, see Register 3-2.
2: The IPL3 bit is concatenated with the IPL[2:0] bits (SR[7:5]) to form the CPU Interrupt Priority Level.

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **NSTDIS:** Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14 **OVAERR:** Accumulator A Overflow Trap Flag bit
1 = Trap was caused by an overflow of Accumulator A
0 = Trap was not caused by an overflow of Accumulator A

bit 13 **OVBERR:** Accumulator B Overflow Trap Flag bit
1 = Trap was caused by an overflow of Accumulator B
0 = Trap was not caused by an overflow of Accumulator B

bit 12 **COVAERR:** Accumulator A Catastrophic Overflow Trap Flag bit
1 = Trap was caused by a catastrophic overflow of Accumulator A
0 = Trap was not caused by a catastrophic overflow of Accumulator A

bit 11 **COVBERR:** Accumulator B Catastrophic Overflow Trap Flag bit
1 = Trap was caused by a catastrophic overflow of Accumulator B
0 = Trap was not caused by a catastrophic overflow of Accumulator B

bit 10 **OVATE:** Accumulator A Overflow Trap Enable bit
1 = Trap overflow of Accumulator A is enabled
0 = Trap is disabled

bit 9 **OVBTE:** Accumulator B Overflow Trap Enable bit
1 = Trap overflow of Accumulator B is enabled
0 = Trap is disabled

bit 8 **COVTE:** Catastrophic Overflow Trap Enable bit
1 = Trap catastrophic overflow of Accumulator A or B is enabled
0 = Trap is disabled

bit 7 **SFTACERR:** Shift Accumulator Error Status bit
1 = Math error trap was caused by an invalid accumulator shift
0 = Math error trap was not caused by an invalid accumulator shift

bit 6 **DIV0ERR:** Divide-by-Zero Error Status bit
1 = Math error trap was caused by a divide-by-zero
0 = Math error trap was not caused by a divide-by-zero

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **MATHERR:** Math Error Status bit
1 = Math error trap has occurred
0 = Math error trap has not occurred

bit 3 **ADDRERR:** Address Error Trap Status bit
1 = Address error trap has occurred
0 = Address error trap has not occurred

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1 (CONTINUED)

bit 2	**STKERR:** Stack Error Trap Status bit 1 = Stack error trap has occurred 0 = Stack error trap has not occurred
bit 1	**OSCFAIL:** Oscillator Failure Trap Status bit 1 = Oscillator failure trap has occurred 0 = Oscillator failure trap has not occurred
bit 0	**Unimplemented:** Read as ‘0’

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
GIE	DISI	SWTRAP	—	—	—	—	AIVTEN
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	INT3EP	INT2EP	INT1EP	INT0EP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **GIE:** Global Interrupt Enable bit
1 = Interrupts and associated IE bits are enabled
0 = Interrupts are disabled, but traps are still enabled

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13 **SWTRAP:** Software Trap Status bit
1 = Software trap is enabled
0 = Software trap is disabled

bit 12-9 **Unimplemented:** Read as ‘0’

bit 8 **AIVTEN:** Alternate Interrupt Vector Table Enable bit
1 = Uses Alternate Interrupt Vector Table
0 = Uses standard Interrupt Vector Table

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **INT3EP:** External Interrupt 3 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 7-5: INTCON3: INTERRUPT CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	NAE
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	—	DAE	DOOVR	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **NAE:** NVM Address Error Soft Trap Status bit
1 = NVM address error soft trap has occurred
0 = NVM address error soft trap has not occurred

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **DAE:** DMA Address Error Soft Trap Status bit
1 = DMA address error trap has occurred
0 = DMA address error trap has not occurred

bit 4 **DOOVR:** DO Stack Overflow Soft Trap Status bit
1 = DO stack overflow soft trap has occurred
0 = DO stack overflow soft trap has not occurred

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 7-6: INTCON4: INTERRUPT CONTROL REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	ECCDBE	SGHT
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1 **ECCDBE:** ECC Double-Bit Error Trap bit
1 = ECC double-bit error trap has occurred
0 = ECC double-bit error trap has not occurred

bit 0 **SGHT:** Software Generated Hard Trap Status bit
1 = Software generated hard trap has occurred
0 = Software generated hard trap has not occurred

REGISTER 7-7: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER

U-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
—	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
VECNUM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **VHOLD:** Vector Number Capture Enable bit
1 = VECNUM[7:0] bits read current value of vector number encoding tree (i.e., highest priority pending interrupt)
0 = Vector number latched into VECNUM[7:0] at Interrupt Acknowledge and retained until next IACK

bit 12 **Unimplemented:** Read as ‘0’

bit 11-8 **ILR[3:0]:** New CPU Interrupt Priority Level bits
1111 = CPU Interrupt Priority Level is 15
. . .
0001 = CPU Interrupt Priority Level is 1
0000 = CPU Interrupt Priority Level is 0

bit 7-0 **VECNUM[7:0]:** Vector Number of Pending Interrupt bits
11111111 = 255, Reserved; do not use
. . .
00001001 = 9, T1 – Timer1 interrupt
00001000 = 8, INT0 – External Interrupt 0
00000111 = 7, Reserved; do not use
00000110 = 6, Generic soft error trap
00000101 = 5, Reserved; do not use
00000100 = 4, Math error trap
00000011 = 3, Stack error trap
00000010 = 2, Generic hard trap
00000001 = 1, Address error trap
00000000 = 0, Oscillator fail trap

8.0 I/O PORTS

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"I/O Ports with Edge Detect"** (www.microchip.com/DS70005322).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices.

Many of the device pins are shared among the peripherals and the Parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity. The PORT registers are located in the SFR.

Some of the key features of the I/O ports are:

- Individual Output Pin Open-Drain Enable/Disable
- Individual Input Pin Weak Pull-up and Pull-Down
- Monitor Selective Inputs and Generate Interrupt when Change in Pin State is Detected
- Operation during Sleep and Idle Modes

8.1 Parallel I/O (PIO) Ports

All port pins have 12 registers directly associated with their operation as digital I/Os. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input.

All port pins are defined as inputs after a Reset. Reads from the latch (LATx), read the latch. Writes to the latch, write the latch. Reads from the port (PORTx), read the port pins, while writes to the port pins, write the latch. Any bit and its associated data and control registers that are not valid for a particular device are disabled. This means the corresponding LATx and TRISx registers, and the port pin are read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs. Table 8-1 shows the pin availability. Table 8-2 shows the 5V input tolerant pins across this device.

TABLE 8-1: PIN AND ANSELx AVAILABILITY

Device	Rx15	Rx14	Rx13	Rx12	Rx11	Rx10	Rx9	Rx8	Rx7	Rx6	Rx5	Rx4	Rx3	Rx2	Rx1	Rx0
PORTA																
dsPIC33CDV64MC106	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
dsPIC33CDVL64MC106	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
ANSELA	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
PORTB																
dsPIC33CDV64MC106	X(1)	X(1)	X(1)	X(1)	X(1)	X(1)	X	X	X	X	X	X	X	X	X	X
dsPIC33CDVL64MC106	X(1)	X(1)	X(1)	X(1)	X(1)	X(1)	X	X	X	X	X	X	X	X	X	X
ANSELB	—	—	—	—	—	—	X	X	X	—	—	X	X	X	X	X
PORTC																
dsPIC33CDV64MC106	—	—	X(1)	X(1)	X	X	X	X	X	X	X	X	X	X	X	X
dsPIC33CDVL64MC106	—	—	X(1)	X(1)	X(1)	X(1)	X	X	X	X	X(1)	X	X	X	X	X
ANSELC	—	—	—	—	—	—	—	—	X	X	—	—	X	X	X	X
PORTD																
dsPIC33CDV64MC106	—	—	X	—	—	X	—	X	—	—	—	—	—	—	X(1)	—
dsPIC33CDVL64MC106	—	—	X	—	—	X	—	X	—	—	—	—	—	—	X(1)	—
ANSELD	—	—	X	—	—	X	—	—	—	—	—	—	—	—	—	—

Note 1: Pin is connected to device interconnect and may not be externally available. See Table 1 for more information.

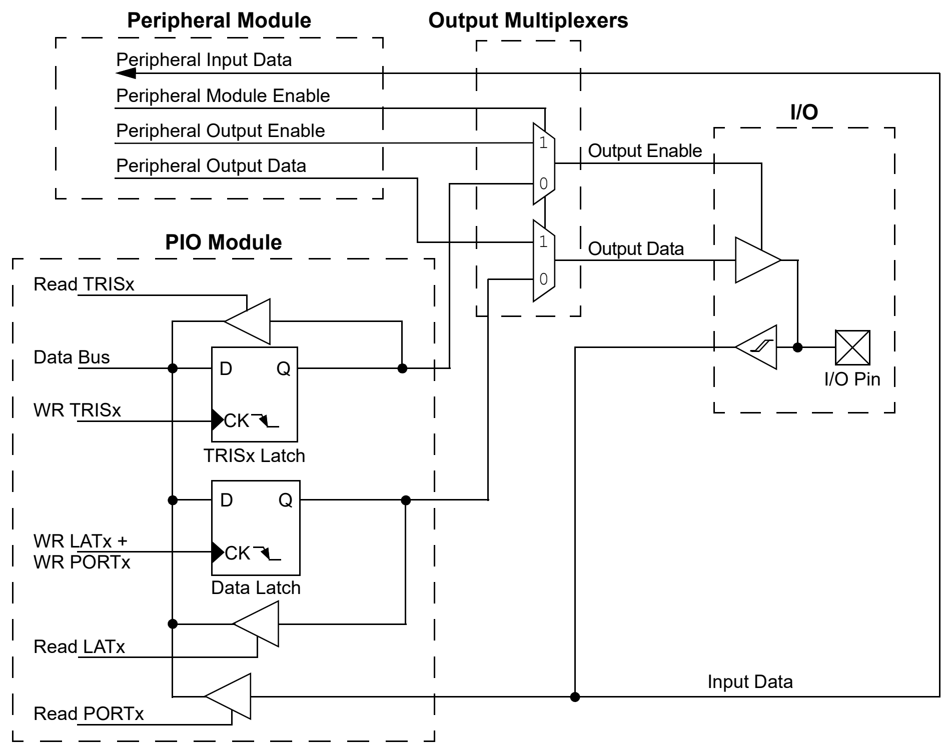
TABLE 8-2: 5V INPUT TOLERANT PORTS

PORTA	—	—	—	—	—	—	—	—	—	—	—	RA4	RA3	RA2	RA1	RA0
PORTB	RB15	RB14	RB13	RB12	RB11	RB10(1)	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
PORTC	—	—	RC13	RC12	RC11	RC10	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
PORTD	—	—	RD13	—	—	RD10	—	RD8	—	—	—	—	—	—	RD1	—

Legend: Shaded pins are up to 5.5 VDC input tolerant.

Note 1: A pull-up resistor is connected to this pin during programming or when JTAG is enabled in the Configuration bits; this limits the maximum voltage on this pin to 3.6V. If JTAG is disabled, the maximum voltage on this pin can reach 5.5V.

FIGURE 8-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



8.1.1 OPEN-DRAIN CONFIGURATION

In addition to the PORTx, LATx and TRISx registers for data control, port pins can also be individually configured for either digital or open-drain output. This is controlled by the Open-Drain Enable for PORTx register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs, other than VDD, by using external pull-up resistors. The maximum open-drain voltage allowed on any pin is the same as the maximum VIH specification for that particular pin.

8.2 Configuring Analog and Digital Port Pins

The ANSELx registers control the operation of the analog port pins. The port pins that are to function as analog inputs must have their corresponding ANSELx and TRISx bits set. In order to use port pins for I/O functionality with digital modules, such as timers, UARTs, etc., the corresponding ANSELx bit must be cleared.

The ANSELx registers have a default value of 0xFFFF; therefore, all pins that share analog functions are analog (not digital) by default.

Pins with analog functions affected by the ANSELx registers are listed with a buffer type of analog in the Pinout I/O Descriptions (see Table 1-1).

If the TRISx bit is cleared (output) while the ANSELx bit is set, the digital output level (VOH or VOL) is converted by an analog peripheral, such as the ADC module or comparator module.

When the PORTx register is read, all pins configured as analog input channels are read as cleared (a low level).

Pins configured as digital inputs do not convert an analog input. Analog levels on any pin, defined as a digital input (including the ANx pins), can cause the input buffer to consume current that exceeds the device specifications.

8.2.1 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a `NOP`.

8.3 Control Registers

The following registers are in the PORT module:

- Register 8-1: ANSELx (one per port)
- Register 8-2: TRISx (one per port)
- Register 8-3: PORTx (one per port)
- Register 8-4: LATx (one per port)
- Register 8-5: ODCx (one per port)
- Register 8-6: CNPUx (one per port)
- Register 8-7: CNPDx (one per port)
- Register 8-8: CNCONx (one per port – optional)
- Register 8-9: CNEN0x (one per port)
- Register 8-10: CNSTATx (one per port – optional)
- Register 8-11: CNEN1x (one per port)
- Register 8-12: CNFx (one per port)

REGISTER 8-1: ANSELx: ANALOG SELECT FOR PORTx REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSELx[15:8]							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSELx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **ANSELx[15:0]:** Analog Select for PORTx bits
1 = Analog input is enabled and digital input is disabled on the PORTx[n] pin
0 = Analog input is disabled and digital input is enabled on the PORTx[n] pin

REGISTER 8-2: TRISx: OUTPUT ENABLE FOR PORTx REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx[15:8]							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **TRISx[15:0]:** Output Enable for PORTx bits
1 = LATx[n] is not driven on the PORTx[n] pin
0 = LATx[n] is driven on the PORTx[n] pin

REGISTER 8-3: PORTx: INPUT DATA FOR PORTx REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx[15:8]							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PORTx[15:0]:** PORTx Data Input Value bits

REGISTER 8-4: LATx: OUTPUT DATA FOR PORTx REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx[15:8]							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **LATx[15:0]:** PORTx Data Output Value bits

REGISTER 8-5: ODCx: OPEN-DRAIN ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ODCx[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ODCx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **ODCx[15:0]:** PORTx Open-Drain Enable bits
1 = Open-drain is enabled on the PORTx pin
0 = Open-drain is disabled on the PORTx pin

REGISTER 8-6: CNPUx: CHANGE NOTIFICATION PULL-UP ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPUx[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPUx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNPUx[15:0]:** Change Notification Pull-up Enable for PORTx bits
1 = The pull-up for PORTx[n] is enabled – takes precedence over the pull-down selection
0 = The pull-up for PORTx[n] is disabled

REGISTER 8-7: CNPDx: CHANGE NOTIFICATION PULL-DOWN ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPDx[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPDx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNPDx[15:0]:** Change Notification Pull-Down Enable for PORTx bits
1 = The pull-down for PORTx[n] is enabled (if the pull-up for PORTx[n] is not enabled)
0 = The pull-down for PORTx[n] is disabled

REGISTER 8-8: CNCONx: CHANGE NOTIFICATION CONTROL FOR PORTx REGISTER

R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
ON	—	—	—	CNSTYLE	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ON:** Change Notification (CN) Control for PORTx On bit
1 = CN is enabled
0 = CN is disabled

bit 14-12 **Unimplemented:** Read as ‘0’

bit 11 **CNSTYLE:** Change Notification Style Selection bit
1 = Edge style (detects edge transitions, CNFx[15:0] bits are used for a Change Notification event)
0 = Mismatch style (detects change from last port read, CNSTATx[15:0] bits are used for a Change Notification event)

bit 10-0 **Unimplemented:** Read as ‘0’

REGISTER 8-9: CNEN0x: CHANGE NOTIFICATION INTERRUPT ENABLE FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN0x[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN0x[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNEN0x[15:0]:** Change Notification Interrupt Enable for PORTx bits
1 = Interrupt-on-change (from the last read value) is enabled for PORTx[n]
0 = Interrupt-on-change is disabled for PORTx[n]

REGISTER 8-10: CNSTATx: CHANGE NOTIFICATION INTERRUPT STATUS FOR PORTx REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CNSTATx[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CNSTATx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNSTATx[15:0]:** Change Notification Interrupt Status for PORTx bits
<u>When CNSTYLE (CNCONx[11]) = `0`:</u>
`1` = Change occurred on PORTx[n] since last read of PORTx[n]
`0` = Change did not occur on PORTx[n] since last read of PORTx[n]

REGISTER 8-11: CNEN1x: CHANGE NOTIFICATION INTERRUPT EDGE SELECT FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN1x[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN1x[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNEN1x[15:0]:** Change Notification Interrupt Edge Select for PORTx bits

REGISTER 8-12: CNFx: CHANGE NOTIFICATION INTERRUPT FLAG FOR PORTx REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNFx[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNFx[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CNFx[15:0]:** Change Notification Interrupt Flag for PORTx bits
When CNSTYLE (CNCONx[11]) = 1:
1 = An enabled edge event occurred on the PORTx[n] pin
0 = An enabled edge event did not occur on the PORTx[n] pin

8.4 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the dsPIC33CDVL64MC106 devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature can detect input Change-of-States, even in Sleep mode, when the clocks are disabled. Every I/O port pin can be selected (enabled) for generating an interrupt request on a Change-of-State. Five control registers are associated with the Change Notification (CN) functionality of each I/O port. To enable the Change Notification feature for the port, the ON bit (CNCONx[15]) must be set.

The CNEN0x and CNEN1x registers contain the CN interrupt enable control bits for each of the input pins. The setting of these bits enables a CN interrupt for the corresponding pins. Also, these bits, in combination with the CNSTYLE bit (CNCONx[11]), define a type of transition when the interrupt is generated. Possible CN event options are listed in Table 8-3.

TABLE 8-3: CHANGE NOTIFICATION EVENT OPTIONS

CNSTYLE Bit (CNCONx[11])	CNEN1x Bit	CNEN0x Bit	Change Notification Event Description
0	Does not matter	0	Disabled
0	Does not matter	1	Detects a mismatch between the last read state and the current state of the pin
1	0	0	Disabled
1	0	1	Detects a positive transition only (from '0' to '1')
1	1	0	Detects a negative transition only (from '1' to '0')
1	1	1	Detects both positive and negative transitions

The CNSTATx register indicates whether a change occurred on the corresponding pin since the last read of the PORTx bit. In addition to the CNSTATx register, the CNFx register is implemented for each port. This register contains flags for Change Notification events. These flags are set if the valid transition edge, selected in the CNEN0x and CNEN1x registers, is detected. CNFx stores the occurrence of the event. CNFx bits must be cleared in software to get the next Change Notification interrupt. The CN interrupt is generated only for the I/Os configured as inputs (corresponding TRISx bits must be set).

Note: Pull-ups and pull-downs on Input Change Notification pins should always be disabled when the port pin is configured as a digital output.

8.5 Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features, while minimizing the conflict of features on I/O pins. The challenge is even greater on low pin count devices. In an application where more than one peripheral needs to be assigned to a single pin, inconvenient work arounds in application code, or a complete redesign, may be the only option.

Peripheral Pin Select configuration provides an alternative to these choices by enabling peripheral set selection and placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the device to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select configuration feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of most digital peripherals to any one of these I/O pins. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

8.5.1 AVAILABLE PINS

The number of available pins is dependent on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the label, "RPn", in their full pin designation, where "n" is the remappable pin number. "RP" is used to designate pins that support both remappable input and output functions.

8.5.2 AVAILABLE PERIPHERALS

The peripherals managed by the Peripheral Pin Select are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer-related peripherals (input capture and output compare) and interrupt-on-change inputs.

In comparison, some digital only peripheral modules are never included in the Peripheral Pin Select feature. This is because the peripheral's function requires special I/O circuitry on a specific port and cannot be easily connected to multiple pins. One example includes I^2C modules. A similar requirement excludes all modules with analog inputs, such as the A/D Converter (ADC).

A key difference between remappable and non-remappable peripherals is that remappable peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non-remappable peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

When a remappable peripheral is active on a given I/O pin, it takes priority over all other digital I/Os and digital communication peripherals associated with the pin. Priority is given regardless of the type of peripheral that is mapped. Remappable peripherals never take priority over any analog functions associated with the pin.

8.5.3 CONTROLLING CONFIGURATION CHANGES

Because peripheral mapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. The dsPIC33CDVL64MC106 devices have implemented the control register lock sequence.

After a Reset, writes to the RPINRx and RPORx registers are allowed, but they can be disabled by setting the IOLOCK bit (RPCON[11]). Attempted writes with the IOLOCK bit set will appear to execute normally, but the contents of the registers will remain unchanged. Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes. To set or clear IOLOCK, the NVMKEY unlock sequence must be executed:

1. Write 0x55 to NVMKEY.
2. Write 0xAA to NVMKEY.
3. Clear (or set) IOLOCK as a single operation.

Note: MPLAB® XC16 compiler provides a built-in C language function for unlocking and modifying the RPCON register: `__builtin_write_RPCON(value);` For more information, see the XC16 compiler help files.

8.5.4 INPUT MAPPING

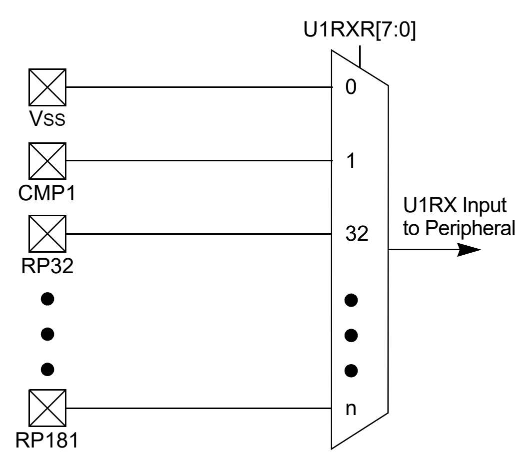
The inputs of the Peripheral Pin Select options are mapped on the basis of the peripheral. That is, a control register associated with a peripheral dictates the pin it will be mapped to. The RPINRx registers are used to configure peripheral input mapping. Each register contains sets of 8-bit fields, with each set associated with one of the remappable peripherals. Programming a given peripheral's bit field with an appropriate 8-bit index value maps the RPn pin with the corresponding value, or internal signal, to that peripheral. See Table 8-4 for a list of available inputs.

For example, Figure 8-2 illustrates remappable pin selection for the U1RX input.

EXAMPLE 8-1: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS

```
//
*******************************************
// Unlock Registers
//*****************************************
__builtin_write_RPCON(0x0000);
//*****************************************
// Configure Input Functions (See Table 8-5)
// Assign U1Rx To Pin RP35
//*************************
_U1RXR = 35;
// Assign U1CTS To Pin RP36
//*************************
_U1CTSR = 36;
//*****************************************
// Configure Output Functions (See Table 8-7)
//*****************************************
// Assign U1Tx To Pin RP37
//*************************
_RP37R = 1;
//*************************
// Assign U1RTS To Pin RP38
//*************************
_RP38R = 2;
//*****************************************
// Lock Registers
//*****************************************
__builtin_write_RPCON(0x0800);
```

FIGURE 8-2: REMAPPABLE INPUT FOR U1RX



Note: For input only, Peripheral Pin Select functionality does not have priority over TRISx settings. Therefore, when configuring an RPn pin for input, the corresponding bit in the TRISx register must also be configured for input (set to '1'). Physical connection to a pin can be made through RP32 through RP77. There are internal signals and virtual pins that can be connected to an input. Table 8-4 shows the details of the input assignment.

TABLE 8-4: REMAPPABLE PIN INPUTS

RPINRx[15:8] or RPINRx[7:0]	Function	Available on Ports
0	Vss	Internal
1	Comparator 1	Internal
2	Comparator 2	Internal
3	Comparator 3	Internal
4-5	RP4-RP5	Reserved
6	PTG Trigger 26	Internal
7	PTG Trigger 27	Internal
8-10	RP8-RP10	Reserved
11	PWM Event Out C	Internal
12	PWM Event Out D	Internal
13	PWM Event Out E	Internal
14-31	RP14-RP31	Reserved
32	RP32	Port Pin RB0
33	RP33	Port Pin RB1
34	RP34	Port Pin RB2
35	RP35	Port Pin RB3
36	RP36	Port Pin RB4
37	RP37	Port Pin RB5
38	RP38	Port Pin RB6
39	RP39	Port Pin RB7
40	RP40	Port Pin RB8
41	RP41	Port Pin RB9
42	RP42	Port Pin RB10
43	RP43	Port Pin RB11
44	RP44	Port Pin RB12
45	RP45	Port Pin RB13
46	RP46	Port Pin RB14
47	RP47	Port Pin RB15
48	RP48	Port Pin RC0
49	RP49	Port Pin RC1
50	RP50	Port Pin RC2
51	RP51	Port Pin RC3
52	RP52	Port Pin RC4
53	RP53	Port Pin RC5
54	RP54	Port Pin RC6
55	RP55	Port Pin RC7
56	RP56	Port Pin RC8
57	RP57	Port Pin RC9
58	RP58	Port Pin RC10
59	RP59	Port Pin RC11
60	RP60	Port Pin RC12
61	RP61	Port Pin RC13

TABLE 8-4: REMAPPABLE PIN INPUTS (CONTINUED)

RPINRx[15:8] or RPINRx[7:0]	Function	Available on Ports
62-64	RP62-RP64	Reserved
65	RP65	Port Pin RD1
66-71	RP66-RP71	Reserved
72	RP72	Port Pin RD8
73	RP73	Reserved
74	RP74	Port Pin RD10
75-76	RP75-RP76	Reserved
77	RP77	Port Pin RD13
78-167	RP78-RP167	Reserved
168	DAC pwm_req_on	Internal
169	DAC1 pwm_req_off	Internal
170-175	RP170-175	Reserved
176	RP176	Virtual RPV0
177	RP177	Virtual RPV1
178	RP178	Virtual RPV2
179	RP179	Virtual RPV3
180	RP180	Virtual RPV4
181	RP181	Virtual RPV5

8.5.5 VIRTUAL CONNECTIONS

The dsPIC33CDVL64MC106 devices support six virtual RPn pins (RP176-RP181), which are identical in functionality to all other RPn pins, with the exception of pinouts. These six pins are internal to the devices and are not connected to a physical device pin.

These pins provide a simple way for inter-peripheral connection without utilizing a physical pin. For example, the output of the analog comparator can be connected to RP176 and the PWM Fault input can be configured for RP176 as well. This configuration allows the analog comparator to trigger PWM Faults without the use of an actual physical pin on the device.

TABLE 8-5: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)

Input Name[1]	Function Name	Register	Register Bits
External Interrupt 1	INT1	RPINR0	INT1R[7:0]
External Interrupt 2	INT2	RPINR1	INT2R[7:0]
External Interrupt 3	INT3	RPINR1	INT3R[7:0]
Timer1 External Clock	T1CK	RPINR2	T1CK[7:0]
SCCP Timer1	TCKI1	RPINR3	TCKI1R[7:0]
SCCP Capture 1	ICM1	RPINR3	ICM1R[7:0]
SCCP Timer2	TCKI2	RPINR4	TCKI2R[7:0]
SCCP Capture 2	ICM2	RPINR4	ICM2R[7:0]
SCCP Timer3	TCKI3	RPINR5	TCKI3R[7:0]
SCCP Capture 3	ICM3	RPINR5	ICM3R[7:0]
SCCP Timer4	TCKI4	RPINR6	TCKI4R[7:0]
SCCP Capture 4	ICM4	RPINR6	ICM4R[7:0]
SCCP Fault A	OCFA	RPINR11	OCFAR[7:0]
SCCP Fault B	OCFB	RPINR11	OCFBR[7:0]
PWM PCI Input 8	PCI8	RPINR12	PCI8R[7:0]
PWM PCI Input 9	PCI9	RPINR12	PCI9R[7:0]
PWM PCI Input 10	PCI10	RPINR13	PCI10R[7:0]
PWM PCI Input 11	PCI11	RPINR13	PCI11R[7:0]
QEI1 Input A	QEIA1	RPINR14	QEIA1R[7:0]
QEI1 Input B	QEIB1	RPINR14	QEIB1R[7:0]
QEI1 Index 1 Input	QEINDX1	RPINR15	QEINDX1R[7:0]
QEI1 Home 1 Input	QEIHOM1	RPINR15	QEIHOM1R[7:0]
UART1 Receive	U1RX	RPINR18	U1RXR[7:0]
UART1 Data-Set-Ready	$\overline{\text{U1DSR}}$	RPINR18	U1DSRR[7:0]
UART2 Receive	U2RX	RPINR19	U2RXR[7:0]
UART2 Data-Set-Ready	$\overline{\text{U2DSR}}$	RPINR19	U2DSRR[7:0]
SPI1 Data Input	SDI1	RPINR20	SDI1R[7:0]
SPI1 Clock Input	SCK1IN	RPINR20	SCK1R[7:0]
SPI1 Slave Select	$\overline{\text{SS1}}$	RPINR21	SS1R[7:0]
Reference Clock Input	REFCLKI	RPINR21	REFOIR[7:0]
SPI2 Data Input	SDI2	RPINR22	SDI2R[7:0]
SPI2 Clock Input	SCK2IN	RPINR22	SCK2R[7:0]
SPI2 Slave Select	$\overline{\text{SS2}}$	RPINR23	SS2R[7:0]
UART3 Receive	U3RX	RPINR27	U3RXR[7:0]
UART3 Data-Set-Ready	$\overline{\text{U3DSR}}$	RPINR27	U3DSRR[7:0]
SCCP Fault C	OCFC	RPINR37	OCFCR[7:0]
PWM PCI Input 17	PCI17	RPINR37	PCI17R[7:0]
PWM PCI Input 18	PCI18	RPINR38	PCI18R[7:0]
PWM PCI Input 12	PCI12	RPINR42	PCI12R[7:0]
PWM PCI Input 13	PCI13	RPINR42	PCI13R[7:0]
PWM PCI Input 14	PCI14	RPINR43	PCI14R[7:0]
PWM PCI Input 15	PCI15	RPINR43	PCI15R[7:0]
PWM PCI Input 16	PCI16	RPINR44	PCI16R[7:0]

Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

TABLE 8-5: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION) (CONTINUED)

Input Name[1]	Function Name	Register	Register Bits
SENT1 Input	SENT1	RPINR44	SENT1R[7:0]
CLC Input A	CLCINA	RPINR45	CLCINAR[7:0]
CLC Input B	CLCINB	RPINR46	CLCINBR[7:0]
CLC Input C	CLCINC	RPINR46	CLCINCR[7:0]
CLC Input D	CLCIND	RPINR47	CLCINDR[7:0]
ADC Trigger Input (ADTRIG31)	ADCTRG	RPINR47	ADCTRGR[7:0]
SCCP Fault D	OCFD	RPINR48	OCFDR[7:0]
UART1 Clear-to-Send	$\overline{\text{U1CTS}}$	RPINR48	U1CTSR[7:0]
UART2 Clear-to-Send	$\overline{\text{U2CTS}}$	RPINR49	U2CTSR[7:0]
UART3 Clear-to-Send	$\overline{\text{U3CTS}}$	RPINR49	U3CTSR[7:0]

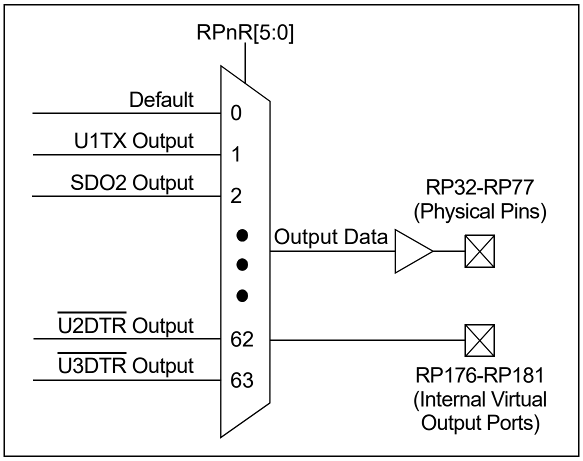
Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

8.5.6 OUTPUT MAPPING

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains sets of 6-bit fields, with each set associated with one RPn pin (see Register 8-43 through Register 8-62). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 8-7 and Figure 8-3).

A null output is associated with the output register Reset value of '0'. This is done to ensure that remappable outputs remain disconnected from all output pins by default.

FIGURE 8-3: MULTIPLEXING REMAPPABLE OUTPUTS FOR RPn



Note 1:	There are six virtual output ports which are not connected to any I/O ports (RP176-RP181). These virtual ports can be accessed by RPOR17, RPOR18 and RPOR19.

8.5.7 MAPPING LIMITATIONS

The control schema of the peripheral select pins is not limited to a small range of fixed peripheral configurations. There are no mutual or hardware-enforced lockouts between any of the peripheral mapping SFRs. Literally, any combination of peripheral mappings, across any or all of the RPn pins, is possible. This includes both many-to-one and one-to-many mappings of peripheral inputs, and outputs to pins. While such mappings may be technically possible from a configuration point of view, they may not be supportable from an electrical point of view (see Table 8-6).

TABLE 8-6: REMAPPABLE OUTPUT PIN REGISTERS

Register	RP Pin	I/O Port
RPOR0[5:0]	RP32	Port Pin RB0
RPOR0[13:8]	RP33	Port Pin RB1
RPOR1[5:0]	RP34	Port Pin RB2
RPOR1[13:8]	RP35	Port Pin RB3
RPOR2[5:0]	RP36	Port Pin RB4
RPOR2[13:8]	RP37	Port Pin RB5
RPOR3[5:0]	RP38	Port Pin RB6
RPOR3[13:8]	RP39	Port Pin RB7
RPOR4[5:0]	RP40	Port Pin RB8
RPOR4[13:8]	RP41	Port Pin RB9
RPOR5[5:0]	RP42	Port Pin RB10
RPOR5[13:8]	RP43	Port Pin RB11
RPOR6[5:0]	RP44	Port Pin RB12
RPOR6[13:8]	RP45	Port Pin RB13
RPOR7[5:0]	RP46	Port Pin RB14
RPOR7[13:8]	RP47	Port Pin RB15
RPOR8[5:0]	RP48	Port Pin RC0
RPOR8[13:8]	RP49	Port Pin RC1
RPOR9[5:0]	RP50	Port Pin RC2
RPOR9[13:8]	RP51	Port Pin RC3
RPOR10[5:0]	RP52	Port Pin RC4
RPOR10[13:8]	RP53	Port Pin RC5
RPOR11[5:0]	RP54	Port Pin RC6
RPOR11[13:8]	RP55	Port Pin RC7
RPOR12[5:0]	RP56	Port Pin RC8
RPOR12[13:8]	RP57	Port Pin RC9
RPOR13[5:0]	RP58	Port Pin RC10
RPOR13[13:8]	RP59	Port Pin RC11
RPOR14[5:0]	RP60	Port Pin RC12
RPOR14[13:8]	RP61	Port Pin RC13
RPOR15[5:0]	RP65	Port Pin RD1
RPOR15[13:8]	RP72	Port Pin RD8
RPOR16[5:0]	RP74	Port Pin RD10
RPOR16[13:8]	RP77	Port Pin RD13
RPOR17[5:0]	RP176	Virtual Pin RPV0
RPOR17[13:8]	RP177	Virtual Pin RPV1
RPOR18[5:0]	RP178	Virtual Pin RPV2
RPOR18[13:8]	RP179	Virtual Pin RPV3
RPOR19[5:0]	RP180	Virtual Pin RPV4
RPOR19[13:8]	RP181	Virtual Pin RPV5

TABLE 8-7: OUTPUT SELECTION FOR REMAPPABLE PINS (RPn)

Function	RPnR[5:0]	Output Name
Not Connected	0	Not Connected
U1TX	1	RPn tied to UART1 Transmit
$\overline{\text{U1RTS}}$	2	RPn tied to UART1 Request-to-Send
U2TX	3	RPn tied to UART2 Transmit
$\overline{\text{U2RTS}}$	4	RPn tied to UART2 Request-to-Send
SDO1	5	RPn tied to SPI1 Data Output
SCK1	6	RPn tied to SPI1 Clock Output
$\overline{\text{SS1}}$	7	RPn tied to SPI1 Slave Select
SDO2	8	RPn tied to SPI2 Data Output
SCK2	9	RPn tied to SPI2 Clock Output
$\overline{\text{SS2}}$	10	RPn tied to SPI2 Slave Select
REFCLKO	14	RPn tied to Reference Clock Output
OCM1A	15	RPn tied to SCCP1 Output
OCM2A	16	RPn tied to SCCP2 Output
OCM3A	17	RPn tied to SCCP3 Output
OCM4A	18	RPn tied to SCCP4 Output
CMP1	23	RPn tied to Comparator 1 Output
U3TX	27	RPn tied to UART3 Transmit
$\overline{\text{U3RTS}}$	28	RPn tied to UART3 Request-to-Send
PWM4H	34	RPn tied to PWM4H Output
PWM4L	35	RPn tied to PWM4L Output
PWMEA	36	RPn tied to PWM Event A Output
PWMEB	37	RPn tied to PWM Event B Output
QEICMP1	38	RPn tied to QEI1 Comparator Output
CLC1OUT	40	RPn tied to CLC1 Output
CLC2OUT	41	RPn tied to CLC2 Output
PWMEC	44	RPn tied to PWM Event C Output
PWMED	45	RPn tied to PWM Event D Output
PTGTRG24	46	RPn tied to PTG Trigger Output 24
PTGTRG25	47	RPn tied to PTG Trigger Output 25
SENT1OUT	48	RPn tied to SENT1 Output
CLC3OUT	59	RPn tied to CLC4 Output
CLC4OUT	60	RPn tied to CLC4 Output
$\overline{\text{U1DTR}}$	61	RPn tied to UART1 DTR
$\overline{\text{U2DTR}}$	62	RPn tied to UART2 DTR
$\overline{\text{U3DTR}}$	63	RPn tied to UART3 DTR

8.5.8 I/O HELPFUL TIPS

1. In some cases, certain pins, as defined in Table 33-15 under "Injection Current", have internal protection diodes to VDD and VSS. The term, "Injection Current", is also referred to as "Clamp Current". On designated pins, with sufficient external current-limiting precautions by the user, I/O pin input voltages are allowed to be greater or lesser than the data sheet absolute maximum ratings, with respect to the VSS and VDD supplies. Note that when the user application forward biases either of the high or low-side internal input clamp diodes, that the resulting current being injected into the device that is clamped internally by the VDD and VSS power rails, may affect the ADC accuracy by four to six counts.
2. I/O pins that are shared with any analog input pin (i.e., ANx) are always analog pins, by default, after any Reset. Consequently, configuring a pin as an analog input pin automatically disables the digital input pin buffer and any attempt to read the digital input level by reading PORTx or LATx will always return a '0', regardless of the digital logic level on the pin. To use a pin as a digital I/O pin on a shared ANx pin, the user application needs to configure the Analog Select for PORTx registers in the I/O ports module (i.e., ANSELx) by setting the appropriate bit that corresponds to that I/O port pin to a '0'.

Note:	Although it is not possible to use a digital input pin when its analog function is enabled, it is possible to use the digital I/O output function, TRISx = 0x0, while the analog function is also enabled. However, this is not recommended, particularly if the analog input is connected to an external analog voltage source, which would create signal contention between the analog signal and the output pin driver.

3. Most I/O pins have multiple functions. Referring to the device pin diagrams in this data sheet, the priorities of the functions allocated to any pins are indicated by reading the pin name, from left-to-right. The left most function name takes precedence over any function to its right in the naming convention. For example: AN16/T2CK/T7CK/RC1; this indicates that AN16 is the highest priority in this example and will supersede all other functions to its right in the list. Those other functions to its right, even if enabled, would not work as long as any other function to its left was enabled. This rule applies to all of the functions listed for a given pin.
4. Each pin has an internal weak pull-up resistor and pull-down resistor that can be configured using the CNPUx and CNPDx registers, respectively. These resistors eliminate the need for external resistors in certain applications. The internal pull-up is up to ~(VDD – 0.8), not VDD. This value is still above the minimum VIH of CMOS and TTL devices.
5. When driving LEDs directly, the I/O pin can source or sink more current than what is specified in the VOH/IOH and VOL/IOL DC characteristics specification. The respective IOH and IOL current rating only applies to maintaining the corresponding output at or above the VOH, and at or below the VOL levels. However, for LEDs, unlike digital inputs of an externally connected device, they are not governed by the same minimum VIH/VIL levels. An I/O pin output can safely sink or source any current less than that listed in the Absolute Maximum Ratings in **Section 33.0 "Electrical Characteristics"** of this data sheet. For example:

   VOH = 2.4V @ IOH = -8 mA and VDD = 3.3V

   The maximum output current sourced by any 8 mA I/O pin = 12 mA.

   LED source current < 12 mA is technically permitted.

6. The Peripheral Pin Select (PPS) pin mapping rules are as follows:
   a) Only one "output" function can be active on a given pin at any time, regardless if it is a dedicated or remappable function (one pin, one output).
   b) It is possible to assign a "remappable output" function to multiple pins and externally short or tie them together for increased current drive.
   c) If any "dedicated output" function is enabled on a pin, it will take precedence over any remappable "output" function.
   d) If any "dedicated digital" (input or output) function is enabled on a pin, any number of "input" remappable functions can be mapped to the same pin.
   e) If any "dedicated analog" function(s) are enabled on a given pin, "digital input(s)" of any kind will all be disabled, although a single "digital output", at the user's cautionary discretion, can be enabled and active as long as there is no signal contention with an external analog input signal. For example, it is possible for the ADC to convert the digital output logic level, or to toggle a digital output on a comparator or ADC input, provided there is no external analog input, such as for a Built-In Self-Test (BIST).
   f) Any number of "input" remappable functions can be mapped to the same pin(s) at the same time, including to any pin with a single output from either a dedicated or remappable "output".
   g) The TRISx registers control *only* the digital I/O output buffer. Any other dedicated or remappable active "output" will automatically override the TRISx setting. The TRISx register *does not* control the digital logic "input" buffer. Remappable digital "inputs" do not automatically override TRISx settings, which means that the TRISx bit must be set to input for pins with only remappable input function(s) assigned.
   h) All analog pins are enabled by default after any Reset and the corresponding digital input buffer on the pin has been disabled. Only the Analog Select for PORTx (ANSELx) registers control the digital input buffer, *not* the TRISx register. The user must disable the analog function on a pin using the Analog Select for PORTx registers in order to use any "digital input(s)" on a corresponding pin, no exceptions.

8.5.9 I/O PORTS RESOURCES

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

8.5.9.1 Key Resources

- **"I/O Ports with Edge Detect"** (www.microchip.com/DS70005322)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

TABLE 8-8: PORTA REGISTER SUMMARY

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELA	—	—	—	—	—	—	—	—	—	—	—	ANSELA[4:0]				
TRISA	—	—	—	—	—	—	—	—	—	—	—	TRISA[4:0]				
PORTA	—	—	—	—	—	—	—	—	—	—	—	RA[4:0]				
LATA	—	—	—	—	—	—	—	—	—	—	—	LATA[4:0]				
ODCA	—	—	—	—	—	—	—	—	—	—	—	ODCA[4:0]				
CNPUA	—	—	—	—	—	—	—	—	—	—	—	CNPUA[4:0]				
CNPDA	—	—	—	—	—	—	—	—	—	—	—	CNPDA[4:0]				
CNCONA	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0A	—	—	—	—	—	—	—	—	—	—	—	CNEN0A[4:0]				
CNSTATA	—	—	—	—	—	—	—	—	—	—	—	CNSTATA[4:0]				
CNEN1A	—	—	—	—	—	—	—	—	—	—	—	CNEN1A[4:0]				
CNFA	—	—	—	—	—	—	—	—	—	—	—	CNFA[4:0]				

TABLE 8-9: PORTB REGISTER SUMMARY

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELB	—	—	—	—	—	—	ANSELB[9:7]			—	—	ANSELB[4:0]				
TRISB	TRISB[15:0]															
PORTB	RB[15:0]															
LATB	LATB[15:0]															
ODCB	ODCB[15:0]															
CNPUB	CNPUB[15:0]															
CNPDB	CNPDB[15:0]															
CNCONB	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0B	CNEN0[15:0]															
CNSTATB	CNSTATB[15:0]															
CNEN1B	CNEN1B[15:0]															
CNFB	CNFB[15:0]															

TABLE 8-10: PORTC REGISTER SUMMARY

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELC	—	—	—	—	—	—	—	—	ANSELC[7:6]		—	—	ANSELC[3:0]			
TRISC	—	—	TRISC[13:0]													
PORTC	—	—	RC[13:0]													
LATC	—	—	LATC[13:0]													
ODCC	—	—	ODCC[13:0]													
CNPUC	—	—	CNPUC[13:0]													
CNPDC	—	—	CNPDC[13:0]													
CNCONC	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0C	—	—	CNEN0C[13:0]													
CNSTATC	—	—	CNSTATC[13:0]													
CNEN1C	—	—	CNEN1C[13:0]													
CNFC	—	—	CNFC[13:0]													

TABLE 8-11: PORTD REGISTER SUMMARY

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELD	—	—	ANSELD13	—	—	ANSELD10	—	—	—	—	—	—	—	—	—	—
TRISD	—	—	TRISD13	—	—	TRISD10	—	TRISD8	—	—	—	—	—	—	TRISD1	—
PORTD	—	—	RD13	—	—	RD10	—	RD8	—	—	—	—	—	—	RD1	—
LATD	—	—	LATD13	—	—	LATD10	—	LATD8	—	—	—	—	—	—	LATD1	—
ODCD	—	—	ODCD13	—	—	ODCD10	—	ODCD8	—	—	—	—	—	—	ODCD1	—
CNPUD	—	—	CNPUD13	—	—	CNPUD10	—	CNPUD8	—	—	—	—	—	—	CNPUD1	—
CNPDD	—	—	CNPDD13	—	—	CNPDD10	—	CNPDD8	—	—	—	—	—	—	CNPDD1	—
CNCOND	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0D	—	—	CNEN0D13	—	—	CNEN0D10	—	CNEN0D8	—	—	—	—	—	—	CNEN0D1	—
CNSTATD	—	—	CNSTATD13	—	—	CNSTATD10	—	CNSTATD8	—	—	—	—	—	—	CNSTATD1	—
CNEN1D	—	—	CNEN1D13	—	—	CNEN1D10	—	CNEN1D8	—	—	—	—	—	—	CNEN1D1	—
CNFD	—	—	CNFD13	—	—	CNFD10	—	CNFD8							CNFD1	—

8.5.10 PERIPHERAL PIN SELECT REGISTERS

REGISTER 8-13: RPCON: PERIPHERAL REMAPPING CONFIGURATION REGISTER[(1)]

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	IOLOCK	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11 **IOLOCK:** Peripheral Remapping Register Lock bit
1 = All Peripheral Remapping registers are locked and cannot be written
0 = All Peripheral Remapping registers are unlocked and can be written

bit 10-0 **Unimplemented:** Read as ‘0’

Note 1: Writing to this register needs an unlock sequence.

REGISTER 8-14: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **INT1R[7:0]:** Assign External Interrupt 1 (INT1) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 8-15: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT3R7	INT3R6	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **INT3R[7:0]:** Assign External Interrupt 3 (INT3) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **INT2R[7:0]:** Assign External Interrupt 2 (INT2) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-16: RPINR2: PERIPHERAL PIN SELECT INPUT REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1CKR7	T1CKR6	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **T1CKR[7:0]:** Assign Timer1 External Clock (T1CK) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 8-17: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM1R7	ICM1R6	ICM1R5	ICM1R4	ICM1R3	ICM1R2	ICM1R1	ICM1R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TCKI1R7	TCKI1R6	TCKI1R5	TCKI1R4	TCKI1R3	TCKI1R2	TCKI1R1	TCKI1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **ICM1R[7:0]:** Assign SCCP Capture 1 (ICM1) Input to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **TCKI1[7:0]:** Assign SCCP Timer1 (TCKI1) Input to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-18: RPINR4: PERIPHERAL PIN SELECT INPUT REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM2R7	ICM2R6	ICM2R5	ICM2R4	ICM2R3	ICM2R2	ICM2R1	ICM2R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TCKI2R7	TCKI2R6	TCKI2R5	TCKI2R4	TCKI2R3	TCKI2R2	TCKI2R1	TCKI2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **ICM2R[7:0]:** Assign SCCP Capture 2 (ICM2) Input to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **TCKI2R[7:0]:** Assign SCCP Timer2 (TCKI2) Input to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-19: RPINR5: PERIPHERAL PIN SELECT INPUT REGISTER 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM3R7	ICM3R6	ICM3R5	ICM3R4	ICM3R3	ICM3R2	ICM3R1	ICM3R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TCKI3R7	TCKI3R6	TCKI3R5	TCKI3R4	TCKI3R3	TCKI3R2	TCKI3R1	TCKI3R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **ICM3R[7:0]:** Assign SCCP Capture 3 (ICM3) Input to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **TCKI3R[7:0]:** Assign SCCP Timer3 (TCKI3) Input to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-20: RPINR6: PERIPHERAL PIN SELECT INPUT REGISTER 6

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM4R7	ICM4R6	ICM4R5	ICM4R4	ICM4R3	ICM4R2	ICM4R1	ICM4R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TCKI4R7	TCKI4R6	TCKI4R5	TCKI4R4	TCKI4R3	TCKI4R2	TCKI4R1	TCKI4R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **ICM4R[7:0]:** Assign SCCP Capture 4 (ICM4) Input to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **TCKI4R[7:0]:** Assign SCCP Timer4 (TCKI4) Input to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-21: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFBR7	OCFBR6	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFAR7	OCFAR6	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **OCFBR[7:0]:** Assign xCCP Fault B (OCFB) Input to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **OCFAR[7:0]:** Assign xCCP Fault A (OCFA) Input to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-22: RPINR12: PERIPHERAL PIN SELECT INPUT REGISTER 12

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI9R7	PCI9R6	PCI9R5	PCI9R4	PCI9R3	PCI9R2	PCI9R1	PCI9R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI8R7	PCI8R6	PCI8R5	PCI8R4	PCI8R3	PCI8R2	PCI8R1	PCI8R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **PCI9R[7:0]:** Assign PWM Input 9 (PCI9) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **PCI8R[7:0]:** Assign PWM Input 8 (PCI8) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-23: RPINR13: PERIPHERAL PIN SELECT INPUT REGISTER 13

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI11R7	PCI11R6	PCI11R5	PCI11R4	PCI11R3	PCI11R2	PCI11R1	PCI11R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI10R7	PCI10R6	PCI10R5	PCI10R4	PCI10R3	PCI10R2	PCI10R1	PCI10R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **PCI11R[7:0]:** Assign PWM Input 11 (PCI11) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **PCI10R[7:0]:** Assign PWM Input 10 (PCI10) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-24: RPINR14: PERIPHERAL PIN SELECT INPUT REGISTER 14

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIB1R7	QEIB1R6	QEIB1R5	QEIB1R4	QEIB1R3	QEIB1R2	QEIB1R1	QEIB1R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIA1R7	QEIA1R6	QEIA1R5	QEIA1R4	QEIA1R3	QEIA1R2	QEIA1R1	QEIA1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **QEIB1R[7:0]:** Assign QEI1 Input B (QEIB1) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **QEIA1R[7:0]:** Assign QEI1 Input A (QEIA1) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-25: RPINR15: PERIPHERAL PIN SELECT INPUT REGISTER 15

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIHOM1R7	QEIHOM1R6	QEIHOM1R5	QEIHOM1R4	QEIHOM1R3	QEIHOM1R2	QEIHOM1R1	QEIHOM1R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEINDX1R7	QEINDX1R6	QEINDX1R5	QEINDX1R4	QEINDX1R3	QEINDX1R2	QEINDX1R1	QEINDX1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **QEIHOM1R[7:0]:** Assign QEI1 Home 1 Input (QEIHOM1) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **QEINDX1R[7:0]:** Assign QEI1 Index 1 Input (QEINDX1) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-26: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1DSRR7	U1DSRR6	U1DSRR5	U1DSRR4	U1DSRR3	U1DSRR2	U1DSRR1	U1DSRR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1RXR7	U1RXR6	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **U1DSRR[7:0]:** Assign UART1 Data-Set-Ready ($\overline{\text{U1DSR}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U1RXR[7:0]:** Assign UART1 Receive (U1RX) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-27: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U2DSRR7	U2DSRR6	U2DSRR5	U2DSRR4	U2DSRR3	U2DSRR2	U2DSRR1	U2DSRR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U2RXR7	U2RXR6	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **U2DSRR[7:0]:** Assign UART2 Data-Set-Ready ($\overline{\text{U2DSR}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U2RXR[7:0]:** Assign UART2 Receive (U2RX) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-28: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK1R7	SCK1R6	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDI1R7	SDI1R6	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **SCK1R[7:0]:** Assign SPI1 Clock Input (SCK1IN) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **SDI1R[7:0]:** Assign SPI1 Data Input (SDI1) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-29: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
REFOIR7	REFOIR6	REFOIR5	REFOIR4	REFOIR3	REFOIR2	REFOIR1	REFOIR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SS1R7	SS1R6	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **REFOIR[7:0]:** Assign Reference Clock Input (REFCLKI) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **SS1R[7:0]:** Assign SPI1 Slave Select ($\overline{SS1}$) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-30: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK2R7	SCK2R6	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDI2R7	SDI2R6	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **SCK2R[7:0]:** Assign SPI2 Clock Input (SCK2IN) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **SDI2R[7:0]:** Assign SPI2 Data Input (SDI2) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-31: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SS2R7	SS2R6	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **SS2R[7:0]:** Assign SPI2 Slave Select ($\overline{SS2}$) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-32: RPINR27: PERIPHERAL PIN SELECT INPUT REGISTER 27

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3DSRR7	U3DSRR6	U3DSRR5	U3DSRR4	U3DSRR3	U3DSRR2	U3DSRR1	U3DSRR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3RXR7	U3RXR6	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **U3DSRR[7:0]:** Assign UART3 Data-Set-Ready ($\overline{U3DSR}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U3RXR[7:0]:** Assign UART3 Receive (U3RX) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-33: RPINR37: PERIPHERAL PIN SELECT INPUT REGISTER 37

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI17R7	PCI17R6	PCI17R5	PCI17R4	PCI17R3	PCI17R2	PCI17R1	PCI17R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFCR7	OCFCR6	OCFCR5	OCFCR4	OCFCR3	OCFCR2	OCFCR1	OCFCR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **PCI17R[7:0]:** Assign PWM Input 17 (PCI17) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **OCFCR[7:0]:** Assign xCCP Fault C (OCFC) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-34: RPINR38: PERIPHERAL PIN SELECT INPUT REGISTER 38

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI18R7	PCI18R6	PCI18R5	PCI18R4	PCI18R3	PCI18R2	PCI18R1	PCI18R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **PCI18R[7:0]:** Assign PWM Input 18 (PCI18) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-35: RPINR42: PERIPHERAL PIN SELECT INPUT REGISTER 42

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI13R7	PCI13R6	PCI13R5	PCI13R4	PCI13R3	PCI13R2	PCI13R1	PCI13R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI12R7	PCI12R6	PCI12R5	PCI12R4	PCI12R3	PCI12R2	PCI12R1	PCI12R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **PCI13R[7:0]:** Assign PWM Input 13 (PCI13) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **PCI12R[7:0]:** Assign PWM Input 12 (PCI12) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-36: RPINR43: PERIPHERAL PIN SELECT INPUT REGISTER 43

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI15R7	PCI15R6	PCI15R5	PCI15R4	PCI15R3	PCI15R2	PCI15R1	PCI15R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI14R7	PCI14R6	PCI14R5	PCI14R4	PCI14R3	PCI14R2	PCI14R1	PCI14R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **PCI15R[7:0]:** Assign PWM Input 15 (PCI15) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **PCI14R[7:0]:** Assign PWM Input 14 (PCI14) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-37: RPINR44: PERIPHERAL PIN SELECT INPUT REGISTER 44

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SENT1R7	SENT1R6	SENT1R5	SENT1R4	SENT1R3	SENT1R2	SENT1R1	SENT1R0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI16R7	PCI16R6	PCI16R5	PCI16R4	PCI16R3	PCI16R2	PCI16R1	PCI16R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **SENT1R[7:0]:** Assign SENT1 Input (SENT1) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **PCI16[7:0]:** Assign PWM Input 16 (PCI16) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-38: RPINR45: PERIPHERAL PIN SELECT INPUT REGISTER 45

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINAR7	CLCINAR6	CLCINAR5	CLCINAR4	CLCINAR3	CLCINAR2	CLCINAR1	CLCINAR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **CLCINAR[7:0]:** Assign CLC Input A (CLCINA) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 8-39: RPINR46: PERIPHERAL PIN SELECT INPUT REGISTER 46

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINCR7	CLCINCR6	CLCINCR5	CLCINCR4	CLCINCR3	CLCINCR2	CLCINCR1	CLCINCR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINBR7	CLCINBR6	CLCINBR5	CLCINBR4	CLCINBR3	CLCINBR2	CLCINBR1	CLCINBR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **CLCINCR[7:0]:** Assign CLC Input C (CLCINC) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **CLCINBR[7:0]:** Assign CLC Input B (CLCINB) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-40: RPINR47: PERIPHERAL PIN SELECT INPUT REGISTER 47

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCTRGR7	ADCTRGR6	ADCTRGR5	ADCTRGR4	ADCTRGR3	ADCTRGR2	ADCTRGR1	ADCTRGR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINDR7	CLCINDR6	CLCINDR5	CLCINDR4	CLCINDR3	CLCINDR2	CLCINDR1	CLCINDR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **ADCTRGR[7:0]:** Assign ADC Trigger Input (ADCTRG) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **CLCINDR[7:0]:** Assign CLC Input D (CLCIND) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-41: RPINR48: PERIPHERAL PIN SELECT INPUT REGISTER 48

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1CTSR7	U1CTSR6	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFDR7	OCFDR6	OCFDR5	OCFDR4	OCFDR3	OCFDR2	OCFDR1	OCFDR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **U1CTSR[7:0]:** Assign UART1 Clear-to-Send ($\overline{\text{U1CTS}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **OCFDR[7:0]:** Assign xCCP Fault D (OCFD) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-42: RPINR49: PERIPHERAL PIN SELECT INPUT REGISTER 49

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3CTSR7	U3CTSR6	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U2CTSR7	U2CTSR6	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **U3CTSR[7:0]:** Assign UART3 Clear-to-Send ($\overline{\text{U3CTS}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U2CTSR[7:0]:** Assign UART2 Clear-to-Send ($\overline{\text{U2CTS}}$) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-43: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP33R5	RP33R4	RP33R3	RP33R2	RP33R1	RP33R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP32R5	RP32R4	RP32R3	RP32R2	RP32R1	RP32R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP33R[5:0]:** Peripheral Output Function is Assigned to RP33 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP32R[5:0]:** Peripheral Output Function is Assigned to RP32 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-44: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP35R5	RP35R4	RP35R3	RP35R2	RP35R1	RP35R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP34R5	RP34R4	RP34R3	RP34R2	RP34R1	RP34R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP35R[5:0]:** Peripheral Output Function is Assigned to RP35 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP34R[5:0]:** Peripheral Output Function is Assigned to RP34 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-45: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP37R[5:0]:** Peripheral Output Function is Assigned to RP37 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP36R[5:0]:** Peripheral Output Function is Assigned to RP36 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-46: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP38R5	RP38R5	RP38R5	RP38R5	RP38R5	RP38R5
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP39R[5:0]:** Peripheral Output Function is Assigned to RP39 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP38R[5:0]:** Peripheral Output Function is Assigned to RP38 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-47: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP41R[5:0]:** Peripheral Output Function is Assigned to RP41 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP40R[5:0]:** Peripheral Output Function is Assigned to RP40 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-48: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP43R[5:0]:** Peripheral Output Function is Assigned to RP43 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP42R[5:0]:** Peripheral Output Function is Assigned to RP42 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-49: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP45R[5:0]:** Peripheral Output Function is Assigned to RP45 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP44R[5:0]:** Peripheral Output Function is Assigned to RP44 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-50: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP47R[5:0]:** Peripheral Output Function is Assigned to RP47 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP46R[5:0]:** Peripheral Output Function is Assigned to RP46 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-51: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP49R5	RP49R4	RP49R3	RP49R2	RP49R1	RP49R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP48R5	RP48R4	RP48R3	RP48R2	RP48R1	RP48R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14	**Unimplemented:** Read as ‘0’
bit 13-8	**RP49R[5:0]:** Peripheral Output Function is Assigned to RP49 Output Pin bits (see Table 8-7 for peripheral function numbers)
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**RP48R[5:0]:** Peripheral Output Function is Assigned to RP48 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-52: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP51R5	RP51R4	RP51R3	RP51R2	RP51R1	RP51R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP50R5	RP50R4	RP50R3	RP50R2	RP50R1	RP50R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14	**Unimplemented:** Read as ‘0’
bit 13-8	**RP51R[5:0]:** Peripheral Output Function is Assigned to RP51 Output Pin bits (see Table 8-7 for peripheral function numbers)
bit 7-6	**Unimplemented:** Read as ‘0’
bit 5-0	**RP50R[5:0]:** Peripheral Output Function is Assigned to RP50 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-53: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP53R5	RP53R4	RP53R3	RP53R2	RP53R1	RP53R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP52R5	RP52R4	RP52R3	RP52R2	RP52R1	RP52R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP53[5:0]:** Peripheral Output Function is Assigned to RP53 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP52R[5:0]:** Peripheral Output Function is Assigned to RP52 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-54: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP55R[5:0]:** Peripheral Output Function is Assigned to RP55 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP54R[5:0]:** Peripheral Output Function is Assigned to RP54 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-55: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP57R[5:0]:** Peripheral Output Function is Assigned to RP57 Output Pin bits
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP56R[5:0]:** Peripheral Output Function is Assigned to RP56 Output Pin bits
(see Table 8-7 for peripheral function numbers)

REGISTER 8-56: RPOR13: PERIPHERAL PIN SELECT OUTPUT REGISTER 13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP59R5	RP59R4	RP59R3	RP59R2	RP59R1	RP59R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP58R5	RP58R4	RP58R3	RP58R2	RP58R1	RP58R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP59R[5:0]:** Peripheral Output Function is Assigned to RP59 Output Pin bits
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP58R[5:0]:** Peripheral Output Function is Assigned to RP58 Output Pin bits
(see Table 8-7 for peripheral function numbers)

REGISTER 8-57: RPOR14: PERIPHERAL PIN SELECT OUTPUT REGISTER 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP61R5	RP61R4	RP61R3	RP61R2	RP61R1	RP61R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP60R5	RP60R4	RP60R3	RP60R2	RP60R1	RP60R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP61R[5:0]:** Peripheral Output Function is Assigned to RP61 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP60R[5:0]:** Peripheral Output Function is Assigned to RP60 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-58: RPOR15: PERIPHERAL PIN SELECT OUTPUT REGISTER 15

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP72R5	RP72R4	RP72R3	RP72R2	RP72R1	RP72R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP65R5	RP65R4	RP65R3	RP65R2	RP65R1	RP65R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP72R[5:0]:** Peripheral Output Function is Assigned to RP72 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP65R[5:0]:** Peripheral Output Function is Assigned to RP65 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-59: RPOR16: PERIPHERAL PIN SELECT OUTPUT REGISTER 16

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP77R5	RP77R4	RP77R3	RP77R2	RP77R1	RP77R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP74R5	RP74R4	RP74R3	RP74R2	RP74R1	RP74R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP77R[5:0]:** Peripheral Output Function is Assigned to RP77 Output Pin bits
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP74R[5:0]:** Peripheral Output Function is Assigned to RP74 Output Pin bits
(see Table 8-7 for peripheral function numbers)

REGISTER 8-60: RPOR17: PERIPHERAL PIN SELECT OUTPUT REGISTER 17

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP177R5[(1)]	RP177R4[(1)]	RP177R3[(1)]	RP177R2[(1)]	RP177R1[(1)]	RP177R0[(1)]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP176R5[(1)]	RP176R4[(1)]	RP176R3[(1)]	RP176R2[(1)]	RP176R1[(1)]	RP176R0[(1)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP177R[5:0]:** Peripheral Output Function is Assigned to RP177 Output Pin bits[(1)]
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP176R[5:0]:** Peripheral Output Function is Assigned to RP176 Output Pin bits[(1)]
(see Table 8-7 for peripheral function numbers)

Note 1: These are virtual output ports.

REGISTER 8-61: RPOR18: PERIPHERAL PIN SELECT OUTPUT REGISTER 18

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP179R5[(1)]	RP179R4[(1)]	RP179R3[(1)]	RP179R2[(1)]	RP179R1[(1)]	RP179R0[(1)]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP178R5[(1)]	RP178R4[(1)]	RP178R3[(1)]	RP178R2[(1)]	RP178R1[(1)]	RP178R0[(1)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP179R[5:0]:** Peripheral Output Function is Assigned to RP179 Output Pin bits[(1)]
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP178R[5:0]:** Peripheral Output Function is Assigned to RP178 Output Pin bits[(1)]
(see Table 8-7 for peripheral function numbers)

Note 1: These are virtual output ports.

REGISTER 8-62: RPOR19: PERIPHERAL PIN SELECT OUTPUT REGISTER 19

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP181R5[(1)]	RP181R4[(1)]	RP181R3[(1)]	RP181R2[(1)]	RP181R1[(1)]	RP181R0[(1)]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP180R5[(1)]	RP180R4[(1)]	RP180R3[(1)]	RP180R2[(1)]	RP180R1[(1)]	RP180R0[(1)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP181R[5:0]:** Peripheral Output Function is Assigned to RP181 Output Pin bits
(see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP180R[5:0]:** Peripheral Output Function is Assigned to RP180 Output Pin bits
(see Table 8-7 for peripheral function numbers)

Note 1: These are virtual output ports.

TABLE 8-12: PPS INPUT CONTROL REGISTERS

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RPCON	—	—	—	—	IOLOCK	—	—	—	—	—	—	—	—	—	—	—
RPINR0	INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—
RPINR1	INT3R7	INT3R6	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
RPINR2	T1CKR7	T1CKR6	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0	—	—	—	—	—	—	—	—
RPINR3	ICM1R7	ICM1R6	ICM1R5	ICM1R4	ICM1R3	ICM1R2	ICM1R1	ICM1R0	TCKI1R7	TCKI1R6	TCKI1R5	TCKI1R4	TCKI1R3	TCKI1R2	TCKI1R1	TCKI1R0
RPINR4	ICM2R7	ICM2R6	ICM2R5	ICM2R4	ICM2R3	ICM2R2	ICM2R1	ICM2R0	TCKI2R7	TCKI2R6	TCKI2R5	TCKI2R4	TCKI2R3	TCKI2R2	TCKI2R1	TCKI2R0
RPINR5	ICM3R7	ICM3R6	ICM3R5	ICM3R4	ICM3R3	ICM3R2	ICM3R1	ICM3R0	TCKI3R7	TCKI3R6	TCKI3R5	TCKI3R4	TCKI3R3	TCKI3R2	TCKI3R1	TCKI3R0
RPINR6	ICM4R7	ICM4R6	ICM4R5	ICM4R4	ICM4R3	ICM4R2	ICM4R1	ICM4R0	TCKI4R7	TCKI4R	TCKI4R5	TCKI4R4	TCKI4R3	TCKI4R2	TCKI4R1	TCKI4R0
RPINR11	OCFBR7	OCFBR6	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	OCFAR7	OCFAR6	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
RPINR12	PCI9R7	PCI9R6	PCI9R5	PCI9R4	PCI9R3	PCI9R2	PCI9R1	PCI9R0	PCI8R7	PCI8R6	PCI8R5	PCI8R4	PCI8R3	PCI8R2	PCI8R1	PCI8R0
RPINR13	PCI11R7	PCI11R6	PCI11R5	PCI11R4	PCI11R3	PCI11R2	PCI11R1	PCI11R0	PCI10R7	PCI10R6	PCI10R5	PCI10R4	PCI10R3	PCI10R2	PCI10R1	PCI10R0
RPINR14	QEIB1R7	QEIB1R6	QEIB1R5	QEIB1R4	QEIB1R3	QEIB1R2	QEIB1R1	QEIB1R0	QEIA1R7	QEIA1R6	QEIA1R5	QEIA1R4	QEIA1R3	QEIA1R2	QEIA1R1	QEIA1R0
RPINR15	QEIHOM1R7	QEIHOM1R6	QEIHOM1R5	QEIHOM1R4	QEIHOM1R3	QEIHOM1R2	QEIHOM1R1	QEIHOM1R0	QEINDX1R7	QEINDX1R6	QEINDX1R5	QEINDX1R4	QEINDX1R3	QEINDX1R2	QEINDX1R1	QEINDX1R0
RPINR18	U1DSRR7	U1DSRR6	U1DSRR5	U1DSRR4	U1DSRR3	U1DSRR2	U1DSRR1	U1DSRR0	U1RXR7	U1RXR6	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
RPINR19	U2DSRR7	U2DSRR6	U2DSRR5	U2DSRR4	U2DSRR3	U2DSRR2	U2DSRR1	U2DSRR0	U2RXR7	U2RXR6	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
RPINR20	SCK1R7	SCK1R6	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	SDI1R7	SDI1R6	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
RPINR21	REFOIR7	REFOIR6	REFOIR5	REFOIR4	REFOIR3	REFOIR2	REFOIR1	REFOIR0	SS1R7	SS1R6	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
RPINR22	SCK2R7	SCK2R6	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	SDI2R7	SDI2R6	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
RPINR23	—	—	—	—	—	—	—	—	SS2R7	SS2R6	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
RPINR27	U3DSRR7	U3DSRR6	U3DSRR5	U3DSRR4	U3DSRR3	U3DSRR2	U3DSRR1	U3DSRR0	U3RXR7	U3RXR6	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
RPINR37	PCI17R7	PCI17R6	PCI17R5	PCI17R4	PCI17R3	PCI17R2	PCI17R1	PCI17R0	OCFCR7	OCFCR6	OCFCR5	OCFCR4	OCFCR3	OCFCR2	OCFCR1	OCFCR0
RPINR38	—	—	—	—	—	—	—	—	PCI18R7	PCI18R6	PCI18R5	PCI18R4	PCI18R3	PCI18R2	PCI18R1	PCI18R0
RPINR42	PCI13R7	PCI13R6	PCI13R5	PCI13R4	PCI13R3	PCI13R2	PCI13R1	PCI13R0	PCI12R7	PCI12R6	PCI12R5	PCI12R4	PCI12R3	PCI12R2	PCI12R1	PCI12R0
RPINR43	PCI15R7	PCI15R6	PCI15R5	PCI15R4	PCI15R3	PCI15R2	PCI15R1	PCI15R0	PCI14R7	PCI14R6	PCI14R5	PCI14R4	PCI14R3	PCI14R2	PCI14R1	PCI14R0
RPINR44	SENT1R7	SENT1R6	SENT1R5	SENT1R4	SENT1R3	SENT1R2	SENT1R1	SENT1R0	PCI16R7	PCI16R6	PCI16R5	PCI16R4	PCI16R3	PCI16R2	PCI16R1	PCI16R0
RPINR45	CLCINAR7	CLCINAR6	CLCINAR5	CLCINAR4	CLCINAR3	CLCINAR2	CLCINAR1	CLCINAR0	—	—	—	—	—	—	—	—
RPINR46	CLCINCR7	CLCINCR6	CLCINCR5	CLCINCR4	CLCINCR3	CLCINCR2	CLCINCR1	CLCINCR0	CLCINBR7	CLCINBR6	CLCINBR5	CLCINBR4	CLCINBR3	CLCINBR2	CLCINBR1	CLCINBR0
RPINR47	ADCTRGR7	ADCTRGR6	ADCTRGR5	ADCTRGR4	ADCTRGR3	ADCTRGR2	ADCTRGR1	ADCTRGR0	CLCINDR7	CLCINDR6	CLCINDR5	CLCINDR4	CLCINDR3	CLCINDR2	CLCINDR1	CLCINDR0
RPINR48	U1CTSR7	U1CTSR6	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	OCFDR7	OCFDR6	OCFDR5	OCFDR4	OCFDR3	OCFDR2	OCFDR1	OCFDR0
RPINR49	U3CTSR7	U3CTSR6	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	U2CTSR7	U2CTSR6	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0

TABLE 8-13: PPS OUTPUT CONTROL REGISTERS

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RPOR0	—	—	RP33R5	RP33R4	RP33R3	RP33R2	RP33R1	RP33R0	—	—	RP32R5	RP32R4	RP32R3	RP32R2	RP32R1	RP32R0
RPOR1	—	—	RP35R5	RP35R4	RP35R3	RP35R2	RP35R1	RP35R0	—	—	RP34R5	RP34R4	RP34R3	RP34R2	RP34R1	RP34R0
RPOR2	—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0	—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
RPOR3	—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0	—	—	RP38R5	RP38R4	RP38R3	RP38R2	RP38R1	RP38R0
RPOR4	—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0	—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0
RPOR5	—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0	—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0
RPOR6	—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0	—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
RPOR7	—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0	—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
RPOR8	—	—	RP49R5	RP49R4	RP49R3	RP49R2	RP49R1	RP49R0	—	—	RP48R5	RP48R4	RP48R3	RP48R2	RP48R1	RP48R0
RPOR9	—	—	RP51R5	RP51R4	RP51R3	RP51R2	RP51R1	RP51R0	—	—	RP50R5	RP50R4	RP50R3	RP50R2	RP50R1	RP50R0
RPOR10	—	—	RP53R5	RP53R4	RP53R3	RP53R2	RP53R1	RP53R0	—	—	RP52R5	RP52R4	RP52R3	RP52R2	RP52R1	RP52R0
RPOR11	—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0	—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0
RPOR12	—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0	—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0
RPOR13	—	—	RP59R5	RP59R4	RP59R3	RP59R2	RP59R1	RP59R0	—	—	RP58R5	RP58R4	RP58R3	RP58R2	RP58R1	RP58R0
RPOR14	—	—	RP61R5	RP61R4	RP61R3	RP61R2	RP61R1	RP61R0	—	—	RP60R5	RP60R4	RP60R3	RP60R2	RP60R1	RP60R0
RPOR15	—	—	RP72R5	RP72R4	RP72R3	RP72R2	RP72R1	RP72R0	—	—	RP65R5	RP65R4	RP65R3	RP65R2	RP65R1	RP65R0
RPOR16	—	—	RP77R5	RP77R4	RP77R3	RP77R2	RP77R1	RP77R0	—	—	RP74R5	RP74R4	RP74R3	RP74R2	RP74R1	RP74R0
RPOR17	—	—	RP177R5	RP177R4	RP177R3	RP177R2	RP177R1	RP177R0	—	—	RP176R5	RP176R4	RP176R3	RP176R2	RP176R1	RP176R0
RPOR18	—	—	RP179R5	RP179R4	RP179R3	RP179R2	RP179R1	RP179R0	—	—	RP178R5	RP178R4	RP178R3	RP178R2	RP178R1	RP178R0
RPOR19	—	—	RP181R5	RP181R4	RP181R3	RP181R2	RP181R1	RP181R0	—	—	RP180R5	RP180R4	RP180R3	RP180R2	RP180R1	RP180R0

NOTES:

9.0 OSCILLATOR WITH HIGH-FREQUENCY PLL

Note 1: This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Oscillator Module with High-Speed PLL"**(www.microchip.com/DS70005255).

The dsPIC33CDVL64MC106 family oscillator with high-frequency PLL includes these characteristics:

- On-Chip Phase-Locked Loop (PLL) to Boost Internal Operating Frequency on Select Internal and External Oscillator Sources
- Doze Mode for System Power Savings
- Scalable Reference Clock Output (REFCLKO)
- On-the-Fly Clock Switching between Various Clock Sources
- Fail-Safe Clock Monitoring (FSCM) that Detects Clock Failure and Permits Safe Application Recovery or Shutdown

A block diagram of the dsPIC33CDVL64MC106 oscillator system is shown in Figure 9-1.

FIGURE 9-1: dsPIC33CDVL64MC106 FAMILY CORE CLOCK SOURCES BLOCK DIAGRAM

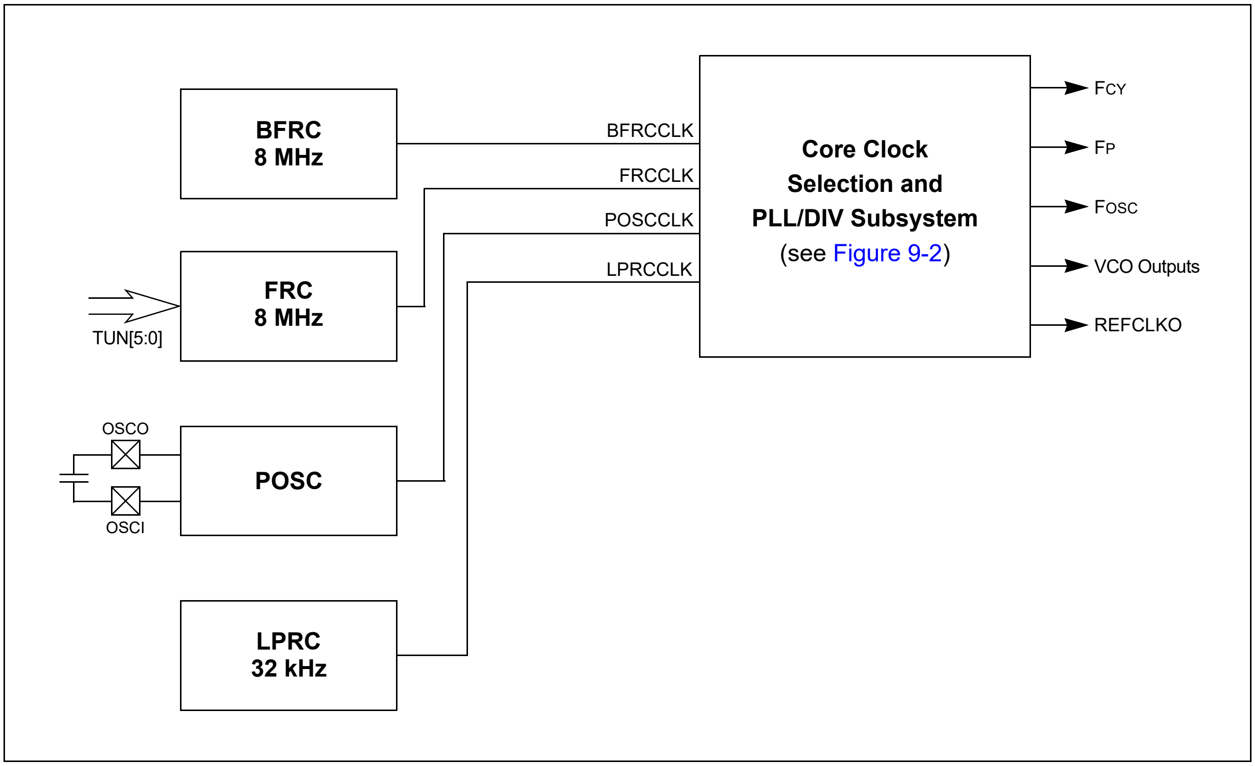
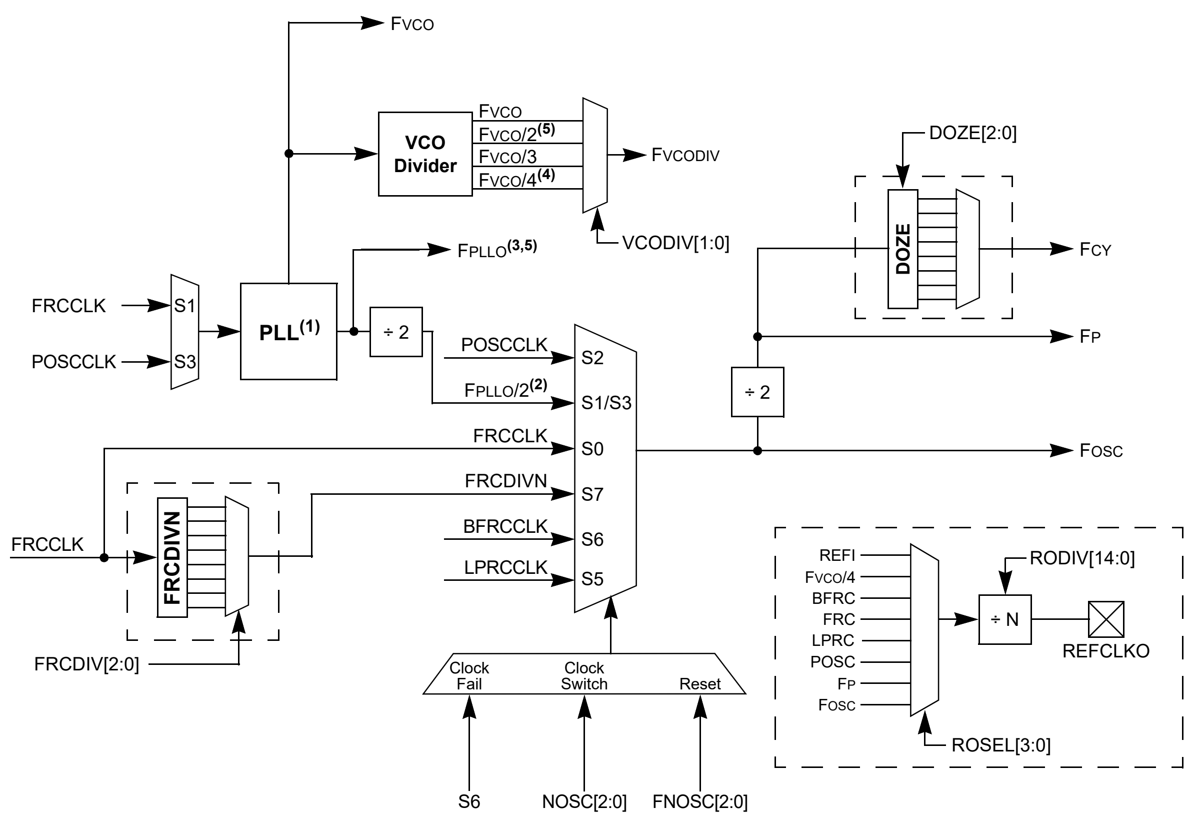


FIGURE 9-2: dsPIC33CDVL64MC106 FAMILY CORE OSCILLATOR SUBSYSTEM



Note 1: See Figure 9-3 for details of the PLL module.
2: XTPLL, HSPLL, ECPLL, FRCPLL (FPLLO).
3: Clock option for PWM.
4: Clock option for ADC.
5: Clock option for DAC.

9.1 Primary PLL

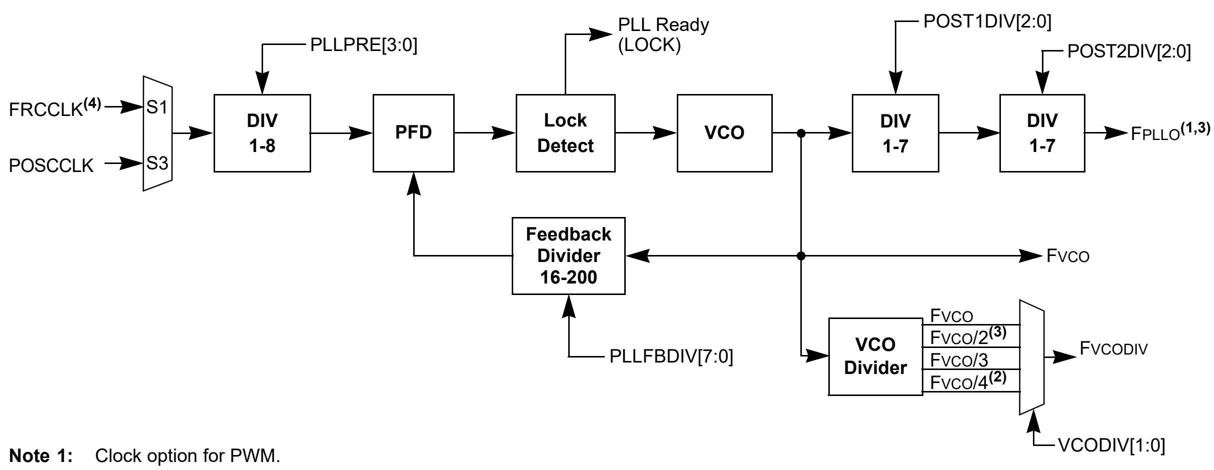
The Primary Oscillator and internal FRC Oscillator sources can optionally use an on-chip PLL to obtain higher operating speeds. Figure 9-3 illustrates a block diagram of the PLL module.

For PLL operation, the following requirements must be met at all times without exception:

- The PLL Input Frequency (F_{PLLI}) must be in the range of 8 MHz to 64 MHz.
- The PFD Input Frequency (F_{PFD}) must be in the range of 8 MHz to (F_{VCO}/16) MHz.

The VCO Output Frequency (F_{VCO}) must be in the range of 400 MHz to 1600 MHz.

FIGURE 9-3: PLL AND VCO DETAIL



Note 1: Clock option for PWM.
2: Clock option for ADC.
3: Clock option for DAC.
4: PLL source is always FRC unless FNOSC is the Primary Oscillator with PLL.

Equation 9-1 provides the relationship between the PLL Input Frequency (F_{PLLI}) and VCO Output Frequency (F_{VCO}).

EQUATION 9-1: FVCO CALCULATION

$$F_{VCO} = F_{PLLI} \times \left(\frac{M}{N1}\right) = F_{PLLI} \times \left(\frac{PLLFBDIV[7:0]}{PLLPRE[3:0]}\right)$$

Equation 9-2 provides the relationship between the PLL Input Frequency (F_{PLLI}) and PLL Output Frequency (F_{PLLO}).

EQUATION 9-2: FPLLO CALCULATION

$$F_{PLLO} = F_{PLLI} \times \left(\frac{M}{N1 \times N2 \times N3}\right) = F_{PLLI} \times \left(\frac{PLLFBDIV[7:0]}{PLLPRE[3:0] \times POST1DIV[2:0] \times POST2DIV[2:0]}\right)$$

Where:

$M = PLLFBDIV[7:0]$
$N1 = PLLPRE[3:0]$
$N2 = POST1DIV[2:0]$
$N3 = POST2DIV[2:0]$

Note: The PLL Phase Detector Input Divider Select (PLLPREx) bits and the PLL Feedback Divider (PLLFBDIVx) bits should not be changed when operating in PLL mode. Therefore, the user must start in either a non-PLL mode or clock switch to a non-PLL mode (e.g., internal FRC Oscillator) to make any necessary changes and then clock switch to the desired PLL mode.

It is not permitted to directly clock switch from one PLL clock source to a different PLL clock source. The user would need to transition between PLL clock sources with a clock switch to a non-PLL clock source.

Example 9-1 illustrates code for using the PLL (50 MIPS) with the Primary Oscillator.

EXAMPLE 9-1: CODE EXAMPLE FOR USING PLL (50 MIPS) WITH PRIMARY OSCILLATOR (POSC)

```
//code example for 50 MIPS system clock using POSC with 10 MHz external crystal

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// Enable Clock Switching and Configure POSC in XT mode
_FOSC(FCKSM_CSECMD & POSCMD_XT);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 100;      // M = 100
    PLLDIVbits.POST1DIV = 5;        // N2=5
    PLLDIVbits.POST2DIV = 1;        // N3=1

    // Initiate Clock Switch to Primary Oscillator with PLL (NOSC=0b011)
    __builtin_write_OSCCONH(0x03);
    __builtin_write_OSCCONL(OSCCON | 0x01);

    // Wait for Clock switch to occur
    while (OSCCONbits.OSWEN!= 0);

    // Wait for PLL to lock
    while (OSCCONbits.LOCK!= 1);
}
```

Example 9-2 illustrates code for using the PLL with an 8 MHz internal FRC.

EXAMPLE 9-2: CODE EXAMPLE FOR USING PLL (50 MIPS) WITH 8 MHz INTERNAL FRC

```
//code example for 50 MIPS system clock using 8MHz FRC

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// Enable Clock Switching
_FOSC(FCKSM_CSECMD);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 125;      // M = 125
    PLLDIVbits.POST1DIV = 5;        // N2=5
    PLLDIVbits.POST2DIV = 1;        // N3=1

    // Initiate Clock Switch to FRC with PLL (NOSC=0b001)
    __builtin_write_OSCCONH(0x01);
    __builtin_write_OSCCONL(OSCCON | 0x01);

    // Wait for Clock switch to occur
    while (OSCCONbits.OSWEN!= 0);

    // Wait for PLL to lock
    while (OSCCONbits.LOCK!= 1);
}
```

9.2 CPU Clocking

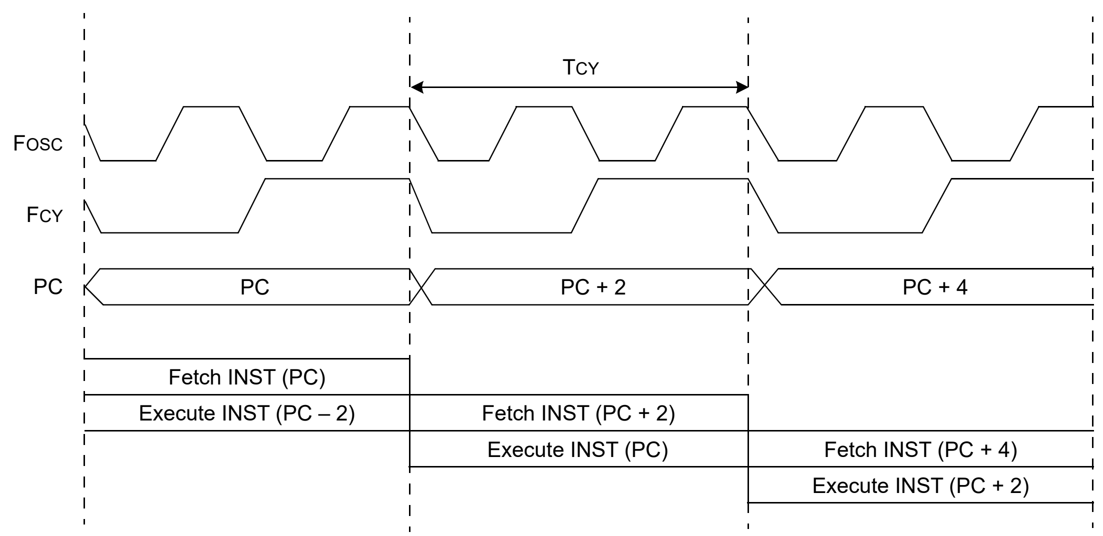
The dsPIC33CDVL64MC106 devices can be configured to use any of the following clock configurations:

- Primary Oscillator (POSC) on the OSCI and OSCO Pins
- Internal Fast RC Oscillator (FRC) with Optional Clock Divider
- Internal Low-Power RC Oscillator (LPRC)
- Primary Oscillator with PLL (ECPLL, HSPLL, XTPLL)
- Internal Fast RC Oscillator with PLL (FRCPLL)
- Internal Backup Fast RC Oscillator (BFRC)

The system clock source is divided by two to produce the internal instruction cycle clock. In this document, the instruction cycle clock is denoted by FCY. The timing diagram in Figure 9-4 illustrates the relationship between the system clock (FOSC), the instruction cycle clock (FCY) and the Program Counter (PC).

The internal instruction cycle clock (FCY) can be output on the OSCO I/O pin if the Primary Oscillator mode (POSCMD[1:0]) is not configured as HS/XT. For more information, see **Section 9.0 "Oscillator with High-Frequency PLL"**.

FIGURE 9-4: CLOCK AND INSTRUCTION CYCLE TIMING



9.3 Primary Oscillator (POSC)

The dsPIC33CDVL64MC106 family features a Primary Oscillator (POSC) and it is available on the OSCI and OSCO pins. This connection enables an external crystal (or ceramic resonator) to provide the clock to the device. The Primary Oscillator provides three modes of operation:

- Medium Speed Oscillator (XT Mode):
  The XT mode is a Medium Gain, Medium Frequency mode used to work with crystal frequencies of 3.5 MHz to 10 MHz.
- High-Speed Oscillator (HS Mode):
  The HS mode is a High-Gain, High-Frequency mode used to work with crystal frequencies of 10 MHz to 32 MHz.
- External Clock Source Operation (EC Mode):
  If the on-chip oscillator is not used, the EC mode allows the internal oscillator to be bypassed. The device clocks are generated from an external source (0 MHz to up to 64 MHz) and input on the OSCI pin.

9.3.1 PRIMARY OSCILLATOR PIN FUNCTIONALITY

The Primary Oscillator pins (OSCI and OSCO) can be used for other functions when the primary oscillator is not being used. The POSCMD<1:0> Configuration bits in the Oscillator Configuration register (FOSC<1:0>) determine the oscillator pin function. The OSCIOFNC bit (FOSC<2>) determines the OSCO/CLKO pin function. By default, the CLKO function is active and the pin will output a clock frequency of Fcy. A clock signal is present on the OSCO/CLKO pin when the device is unprogrammed or during the programming sequence. Care should be taken when the OSCO/CLKO pin is used to drive other circuitry.

9.4 Internal Fast RC (FRC) Oscillator

The dsPIC33CDVL64MC106 devices contain one instance of the internal Fast RC (FRC) Oscillator and it provides a nominal 8 MHz clock without requiring an external crystal or ceramic resonator, which results in system cost savings for applications that do not require a precise clock reference.

9.5 Low-Power RC (LPRC) Oscillator

The dsPIC33CDVL64MC106 family devices contain one instance of the Low-Power RC (LPRC) Oscillator, which provides a nominal clock frequency of 32 kHz, and is the clock source for the Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM) circuits in the clock subsystem. The LPRC Oscillator is shut off in Sleep mode.

The LPRC Oscillator remains enabled under these conditions:

- The FSCM is enabled.
- The WDT is enabled.
- The LPRC Oscillator is selected as the system clock.

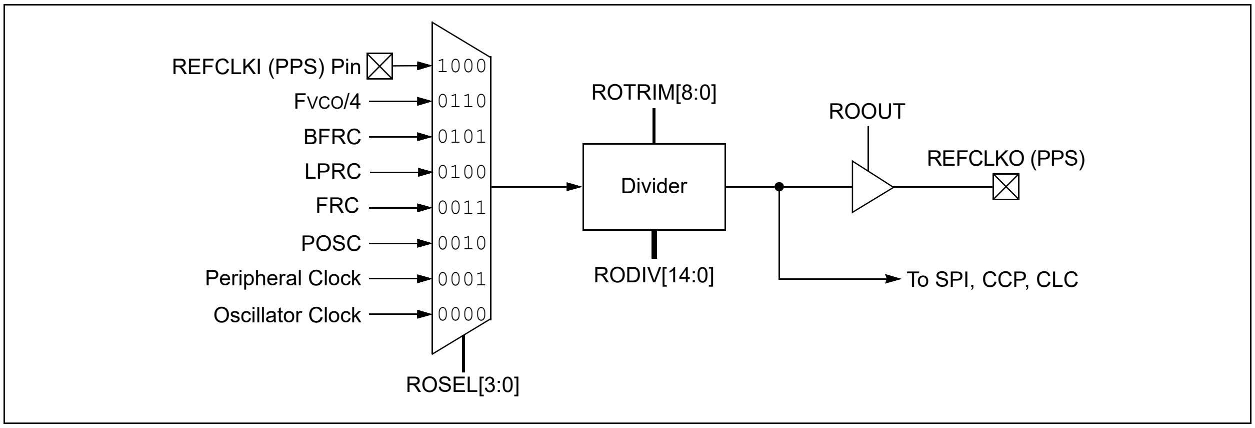
9.6 Backup Internal Fast RC (BFRC) Oscillator

The oscillator block provides a stable reference clock source for the Fail-Safe Clock Monitor (FSCM). When FSCM is enabled in the FCKSM[1:0] Configuration bits (FOSC[7:6]), it constantly monitors the main clock source against a reference signal from the 8 MHz Backup Internal Fast RC (BFRC) Oscillator. In case of a clock failure, the Fail-Safe Clock Monitor switches the clock to the BFRC Oscillator, allowing for continued low-speed operation or a safe application shutdown.

9.7 Reference Clock Output

In addition to the CLKO output (FOSC/2), the dsPIC33CDVL64MC106 devices can be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock sub-multiples to drive external devices in the application. CLKO is enabled by Configuration bit, OSCIOFNC, and is independent of the REFCLKO reference clock. REFCLKO is mappable to any I/O pin that has mapped output capability. Refer to Table 8-7 for more information. The Reference Clock Output module block diagram is shown in Figure 9-5.

FIGURE 9-5: REFERENCE CLOCK GENERATOR



This reference clock output is controlled by the REFOCONL and REFOCONH registers. Setting the ROEN bit (REFOCONL[15]) makes the clock signal available on the REFCLKO pin. The RODIV[14:0] bits (REFOCONH[14:0]) and ROTRIM[8:0] bits (REFOTRIMH[15:7]) enable the selection of different clock divider options. The formula for determining the final frequency output is shown in Equation 9-3. The ROSWEN bit (REFOCONL[9]) indicates that the clock divider has been successfully switched. In order to switch the REFCLKO divider, the user should ensure that this bit reads as '0'. Write the updated values to the RODIV[14:0] or ROTRIM[8:0] bits, set the ROSWEN bit and then wait until it is cleared before assuming that the REFCLKO clock is valid.

EQUATION 9-3: CALCULATING FREQUENCY OUTPUT

$$F_{REFOUT} = \frac{F_{REFIN}}{2 \cdot (RODIV[14:0] + ROTRIM[8:0]/512)}$$

Where: F_{REFOUT} = Output Frequency
F_{REFIN} = Input Frequency
When $RODIV[14:0] = 0$, the output clock is the same as the input clock.

The ROSEL[3:0] bits (REFOCONL[3:0]) determine which clock source is used for the reference clock output. The ROSLP bit (REFOCONL[11]) determines if the reference source is available on REFCLKO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSLP bit must be set and the clock selected by the ROSEL[3:0] bits must be enabled for operation during Sleep mode, if possible. Clearing the ROSEL[3:0] bits allows the reference output frequency to change, as the system clock changes, during any clock switches. The ROOUT bit enables/disables the reference clock output on the REFCLKO pin.

The ROACTIV bit (REFOCONL[8]) indicates that the module is active; it can be cleared by disabling the module (setting ROEN to '0'). The user must not change the reference clock source, or adjust the divider when the ROACTIV bit indicates that the module is active. To avoid glitches, the user should not disable the module until the ROACTIV bit is '1'.

9.8 Oscillator Configuration

The oscillator system has both Configuration registers and SFRs to configure, control and monitor the system. The FOSCSEL and FOSC Configuration registers (Register 30-4 and Register 30-5, respectively) are used for initial setup.

Table 9-1 lists the configuration settings that select the device's oscillator source and operating mode at a Power-on Reset (POR).

TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Source	Oscillator Mode	FNOSC[2:0] Value	POSCMD[1:0] Value
S0	Fast RC Oscillator (FRC)(1)	000	xx
S1	Fast RC Oscillator with PLL (FRCPLL)(1)	001	xx
S2	Primary Oscillator (EC)(1)	010	00
S2	Primary Oscillator (XT)	010	01
S2	Primary Oscillator (HS)	010	10
S3	Primary Oscillator with PLL (ECPLL)(1)	011	00
S3	Primary Oscillator with PLL (XTPLL)	011	01
S3	Primary Oscillator with PLL (HSPLL)	011	10
S4	Reserved	100	xx
S5	Low-Power RC Oscillator (LPRC)(1)	101	xx
S6	Backup FRC (BFRC)(1)	110	xx
S7	Fast RC Oscillator with ÷ N Divider (FRCDIVN)(1,2)	111	xx

Note 1: The OSCO pin function is determined by the OSCIOFNC Configuration bit.
2: This is the default oscillator mode for an unprogrammed (erased) device.

9.9 OSCCON Unlock Sequence

The OSCCON register is protected against unintended writes through a lock mechanism. The upper and lower bytes of OSCCON have their own unlock sequence, and both must be used when writing to both bytes of the register. Before OSCCON can be written to, the following unlock sequence must be used:

1. Execute the unlock sequence for the OSCCON high byte.
   In two back-to-back instructions:
   - Write 0x78 to OSCCON[15:8]
   - Write 0x9A to OSCCON[15:8]
2. In the instruction immediately following the unlock sequence, the OSCCON[15:8] bits can be modified.
3. Execute the unlock sequence for the OSCCON low byte.
   In two back-to-back instructions:
   - Write 0x46 to OSCCON[7:0]
   - Write 0x57 to OSCCON[7:0]
4. In the instruction immediately following the unlock sequence, the OSCCON[7:0] bits can be modified.

Note: MPLAB® XC16 provides a built-in C language function, including the unlocking sequence to modify high and low bytes in the OSCCON register:
`__builtin_write_OSCCONH(value)`
`__builtin_write_OSCCONL(value)`

9.10 Oscillator Control Registers

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER[1]

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC2	COSC1	COSC0	—	NOSC2[2]	NOSC1[2]	NOSC0[2]
bit 15							bit 8

R/W-0	U-0	R-0	U-0	R-0	U-0	U-0	R/W-0
CLKLOCK	—	LOCK	—	CF[3]	—	—	OSWEN
bit 7							bit 0

Legend:	y = Value set from Configuration bits on POR		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **COSC[2:0]:** Current Oscillator Selection bits (read-only)
111 = Fast RC Oscillator (FRC) with Divide-by-n (FRCDIVN)
110 = Backup FRC (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved – default to FRC
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **NOSC[2:0]:** New Oscillator Selection bits[2]
111 = Fast RC Oscillator (FRC) with Divide-by-n (FRCDIVN)
110 = Backup FRC (BFRC)
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved – default to FRC
011 = Primary Oscillator (XT, HS, EC) with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
1 = If (FCKSM0 = 1), then clock and PLL configurations are locked; if (FCKSM0 = 0), then clock and PLL configurations may be modified
0 = Clock and PLL selections are not locked, configurations may be modified

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **LOCK:** PLL Lock Status bit (read-only)
1 = Indicates that PLL is in lock or PLL start-up timer is satisfied
0 = Indicates that PLL is out of lock, start-up timer is in progress or PLL is disabled

bit 4 **Unimplemented:** Read as ‘0’

Note 1: Writes to this register require an unlock sequence (see **Section 9.9 “OSCCON Unlock Sequence”**).
2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.
3: This bit should only be cleared in software.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER[1] (CONTINUED)

bit 3 **CF:** Clock Fail Detect bit[3]
1 = FSCM has detected a clock failure
0 = FSCM has not detected a clock failure

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **OSWEN:** Oscillator Switch Enable bit
1 = Requests oscillator switch to the selection specified by the NOSC[2:0] bits
0 = Oscillator switch is complete

Note 1: Writes to this register require an unlock sequence (see **Section 9.9 "OSCCON Unlock Sequence"**).
2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transitional clock source between the two PLL modes.
3: This bit should only be cleared in software.

REGISTER 9-2: CLKDIV: CLOCK DIVIDER REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE2[1]	DOZE1[1]	DOZE0[1]	DOZEN[2,3]	FRCDIV2	FRCDIV1	FRCDIV0
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	—	PLLPRE[3:0][4]			
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ROI:** Recover on Interrupt bit
1 = Interrupts will clear the DOZEN bit and the processor clock, and the Peripheral Clock ratio is set to 1:1
0 = Interrupts have no effect on the DOZEN bit

bit 14-12 **DOZE[2:0]:** Processor Clock Reduction Select bits[1]
111 = FP divided by 128
110 = FP divided by 64
101 = FP divided by 32
100 = FP divided by 16
011 = FP divided by 8 (default)
010 = FP divided by 4
001 = FP divided by 2
000 = FP divided by 1

bit 11 **DOZEN:** Doze Mode Enable bit[2,3]
1 = DOZE[2:0] field specifies the ratio between the Peripheral Clocks and the processor clocks
0 = Processor clock and Peripheral Clock ratio is forced to 1:1

bit 10-8 **FRCDIV[2:0]:** Internal Fast RC Oscillator Postscaler bits
111 = FRC divided by 256
110 = FRC divided by 64
101 = FRC divided by 32
100 = FRC divided by 16
011 = FRC divided by 8
010 = FRC divided by 4
001 = FRC divided by 2
000 = FRC divided by 1 (default)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **Reserved:** Read as '0'

Note 1: The DOZE[2:0] bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE[2:0] are ignored.
2: This bit is cleared when the ROI bit is set and an interrupt occurs.
3: The DOZEN bit cannot be set if DOZE[2:0] = 000. If DOZE[2:0] = 000, any attempt by user software to set the DOZEN bit is ignored.
4: PLLPRE[3:0] may be updated while the PLL is operating, but the VCO may overshoot.

REGISTER 9-2: CLKDIV: CLOCK DIVIDER REGISTER (CONTINUED)

bit 3-0 **PLLPRE[3:0]:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)[4]
`11111` = Reserved
. . .
`1001` = Reserved
`1000` = Input divided by 8
`0111` = Input divided by 7
`0110` = Input divided by 6
`0101` = Input divided by 5
`0100` = Input divided by 4
`0011` = Input divided by 3
`0010` = Input divided by 2
`0001` = Input divided by 1 (power-on default selection)
`0000` = Reserved

Note 1: The DOZE[2:0] bits can only be written to when the DOZEN bit is clear. If DOZEN = `1`, any writes to DOZE[2:0] are ignored.
2: This bit is cleared when the ROI bit is set and an interrupt occurs.
3: The DOZEN bit cannot be set if DOZE[2:0] = `000`. If DOZE[2:0] = `000`, any attempt by user software to set the DOZEN bit is ignored.
4: PLLPRE[3:0] may be updated while the PLL is operating, but the VCO may overshoot.

REGISTER 9-3: PLLFBD: PLL FEEDBACK DIVIDER REGISTER

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0
PLLFBDIV[7:0]							
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-8 **Reserved:** Maintain as ‘0’

bit 7-0 **PLLFBDIV[7:0]:** PLL Feedback Divider bits (also denoted as ‘M’, PLL multiplier)
11111111 = Reserved
. . .
11001000 = 200 Maximum(1)
. . .
10010110 = 150 (default)
. . .
00010000 = 16 Minimum(1)
. . .
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The default power-on feedback divider is 150 (decimal) with an 8 MHz FRC input clock. The VCO frequency is 1.2 GHz.

REGISTER 9-4: OSCTUN: FRC OSCILLATOR TUNING REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN[5:0]					
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5-0 **TUN[5:0]:** FRC Oscillator Tuning bits

011111 = Maximum frequency deviation of +1.45%
011110 = Center frequency + 1.40%
. . .
000001 = Center frequency + 0.047%
000000 = Center frequency (8.00 MHz nominal)
111111 = Center frequency – 0.047%
. . .
100001 = Center frequency – 1.45%
100000 = Minimum frequency deviation of -1.50%

REGISTER 9-5: PLLDIV: PLL OUTPUT DIVIDER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	VCODIV[1:0]	
bit 15							bit 8

U-0	R/W-0	R/W-1	R/W-1	U-0	R/W-0	R/W-0	R/W-1
—	POST1DIV2[1,2]	POST1DIV1[1,2]	POST1DIV0[1,2]	—	POST2DIV2[1,2]	POST2DIV1[1,2]	POST2DIV0[1,2]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-8 **VCODIV[1:0]:** PLL VCO Output Divider Select bits
11 = Fvco
10 = Fvco/2
01 = Fvco/3
00 = Fvco/4

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **POST1DIV[2:0]:** PLL Output Divider #1 Ratio bits[1,2]
POST1DIV[2:0] can have a valid value, from 1 to 7 (POST1DIVx value should be greater than or equal to the POST2DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **POST2DIV[2:0]:** PLL Output Divider #2 Ratio bits[1,2]
POST2DIV[2:0] can have a valid value, from 1 to 7 (POST2DIVx value should be less than or equal to the POST1DIVx value). The POST1DIVx divider is designed to operate at higher clock rates than the POST2DIVx divider.

Note 1: The POST1DIVx and POST2DIVx divider values must not be changed while the PLL is operating.
2: The default values for POST1DIVx and POST2DIVx are 4 and 1, respectively, yielding a 150 MHz system source clock.

REGISTER 9-6: REFOCONL: REFERENCE CLOCK CONTROL LOW REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	HC/R/W-0	HSC/R-0
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL[3:0]			
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ROEN:** Reference Clock Enable bit
1 = Reference Oscillator is enabled on the REFCLKO pin
0 = Reference Oscillator is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ROSIDL:** Reference Clock Stop in Idle bit
1 = Reference Oscillator is disabled in Idle mode
0 = Reference Oscillator continues to run in Idle mode

bit 12 **ROOUT:** Reference Clock Output Enable bit
1 = Reference clock external output is enabled and available on the REFCLKO pin
0 = Reference clock external output is disabled

bit 11 **ROSLP:** Reference Clock Stop in Sleep bit
1 = Reference Oscillator continues to run in Sleep modes
0 = Reference Oscillator is disabled in Sleep modes

bit 10 **Unimplemented:** Read as ‘0’

bit 9 **ROSWEN:** Reference Clock Switch Request and Status bit
1 = Clock divider change (requested by changes to RODIVx) is requested or is in progress (set in software, cleared by hardware upon completion)
0 = Clock divider change has completed or is not pending

bit 8 **ROACTIV:** Reference Clock Status bit
1 = Reference clock is active; do not change clock source
0 = Reference clock is stopped; clock source and configuration may be safely changed

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **ROSEL[3:0]:** Reference Clock Source Select bits
1111 = Reserved
... = Reserved
1000 = Reserved
0111 = REFI pin
0110 = FVCO/4
0101 = BFRC
0100 = LPRC
0011 = FRC
0010 = Primary Oscillator
0001 = Peripheral Clock (FP)
0000 = System clock (FOSC)

REGISTER 9-7: REFOCONH: REFERENCE CLOCK CONTROL HIGH REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV[14:8]						
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RODIV[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-0 **RODIV[14:0]:** Reference Clock Integer Divider Select bits

Divider for the selected input clock source is two times the selected value.
111 1111 1111 1111 = Base clock value divided by 65,534 (2 * 7FFFh)
111 1111 1111 1110 = Base clock value divided by 65,532 (2 * 7FFEh)
111 1111 1111 1101 = Base clock value divided by 65,530 (2 * 7FFDh)
. . .
000 0000 0000 0010 = Base clock value divided by 4 (2 * 2)
000 0000 0000 0001 = Base clock value divided by 2 (2 * 1)
000 0000 0000 0000 = Base clock value

REGISTER 9-8: REFOTRIMH: REFERENCE OSCILLATOR TRIM REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROTRIM[15:8]							
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ROTRIM	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **ROTRIM[8:0]:** REFO Trim bits

These bits provide a fractional additive to the RODIV[14:0] value for the 1/2 period of the REFO clock.
000000000 = 0/512 (0.0 divisor added to the RODIV[14:0] value)
000000001 = 1/512 (0.001953125 divisor added to the RODIV[14:0] value)
000000010 = 2/512 (0.00390625 divisor added to the RODIV[14:0] value)
. . .
100000000 = 256/512 (0.5000 divisor added to the RODIV[14:0] value)
. . .
111111110 = 510/512 (0.99609375 divisor added to the RODIV[14:0] value)
111111111 = 511/512 (0.998046875 divisor added to the RODIV[14:0] value)

bit 6-0 **Unimplemented:** Read as ‘0’

10.0 DIRECT MEMORY ACCESS (DMA) CONTROLLER

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. For more information, refer to **"Direct Memory Access Controller (DMA)"** (www.microchip.com/DS30009742).

The Direct Memory Access (DMA) Controller is designed to service high data throughput peripherals operating on the SFR bus, allowing them to access data memory directly and alleviating the need for CPU-intensive management. By allowing these data-intensive peripherals to share their own data path, the main data bus is also deloaded, resulting in additional power savings.

The DMA Controller functions both as a peripheral and a direct extension of the CPU. It is located on the microcontroller data bus, between the CPU and DMA-enabled peripherals, with direct access to SRAM. This partitions the SFR bus into two buses, allowing the DMA Controller access to the DMA-capable peripherals located on the new DMA SFR bus. The controller serves as an Initiator device on the DMA SFR bus, controlling data flow from DMA-capable peripherals.

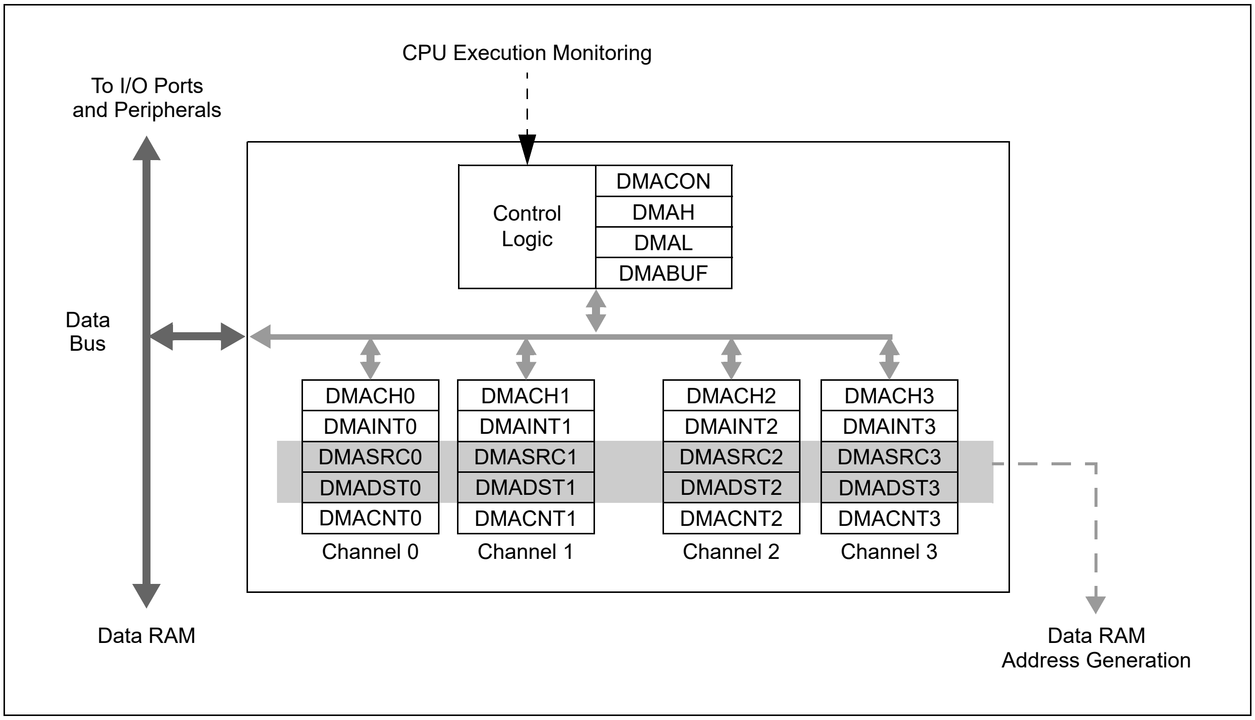
The controller also monitors CPU instruction processing directly, allowing it to be aware of when the CPU requires access to peripherals on the DMA bus and automatically relinquishing control to the CPU as needed. This increases the effective bandwidth for handling data without DMA operations causing a processor Stall. This makes the controller essentially transparent to the user.

The DMA Controller has these features:

- Four Independently Programmable Channels
- Concurrent Operation with the CPU (no DMA caused Wait states)
- DMA Bus Arbitration
- Five Programmable Address Modes
- Four Programmable Transfer Modes
- Four Flexible Internal Data Transfer modes
- Byte or Word Support for Data Transfer
- 16-Bit Source and Destination Address Register for each Channel, Dynamically Updated and Reloadable
- 16-Bit Transaction Count Register, Dynamically Updated and Reloadable
- Upper and Lower Address Limit Registers
- Counter Half-Full Level Interrupt
- Software Triggered Transfer
- Null Write mode for Symmetric Buffer Operations

A simplified block diagram of the DMA Controller is shown if Figure 10-1.

FIGURE 10-1: DMA FUNCTIONAL BLOCK DIAGRAM



10.1 Summary of DMA Operations

The DMA Controller is capable of moving data between addresses according to a number of different parameters. Each of these parameters can be independently configured for any transaction. In addition, any or all of the DMA channels can independently perform a different transaction at the same time. Transactions are classified by these parameters:

- Source and Destination (SFRs and data RAM)
- Data Size (byte or word)
- Trigger Source
- Transfer Mode (One-Shot, Repeated or Continuous)
- Addressing Modes (Fixed Address or Address Blocks with or without Address Increment/Decrement)

In addition, the DMA Controller provides channel priority arbitration for all channels.

10.1.1 SOURCE AND DESTINATION

Using the DMA Controller, data may be moved between any two addresses in the Data Space. The SFR space (0000h to 0FFFh) or the data RAM space (1000h to 2FFFh) can serve as either the source or the destination. Data can be moved between these areas in either direction or between addresses in either area. The four different combinations are shown in Figure 10-2.

If it is necessary to protect areas of data RAM, the DMA Controller allows the user to set upper and lower address boundaries for operations in the Data Space above the SFR space. The boundaries are set by the DMAH and DMAL Limit registers. If a DMA channel attempts an operation outside of the address boundaries, the transaction is terminated and an interrupt is generated.

10.1.2 DATA SIZE

The DMA Controller can handle both 8-bit and 16-bit transactions. Size is user-selectable using the SIZE bit (DMACHn[1]). By default, each channel is configured for word-sized transactions. When byte-sized transactions are chosen, the LSB of the source and/or destination address determines if the data represent the upper or lower byte of the data RAM location.

10.1.3 TRIGGER SOURCE

The DMA Controller can use 82 of the device's interrupt sources to initiate a transaction. The DMA trigger sources occur in reverse order from their natural interrupt priority and are shown in Table 10-1.

Since the source and destination addresses for any transaction can be programmed independently of the trigger source, the DMA Controller can use any trigger to perform an operation on any peripheral. This also allows DMA channels to be cascaded to perform more complex transfer operations.

10.1.4 TRANSFER MODE

The DMA Controller supports four types of data transfers, based on the volume of data to be moved for each trigger.

- One-Shot: A single transaction occurs for each trigger.
- Continuous: A series of back-to-back transactions occur for each trigger; the number of transactions is determined by the DMACNTn transaction counter.
- Repeated One-Shot: A single transaction is performed repeatedly, once per trigger, until the DMA channel is disabled.
- Repeated Continuous: A series of transactions are performed repeatedly, one cycle per trigger, until the DMA channel is disabled.

All transfer modes allow the option to have the source and destination addresses, and counter value, automatically reloaded after the completion of a transaction.

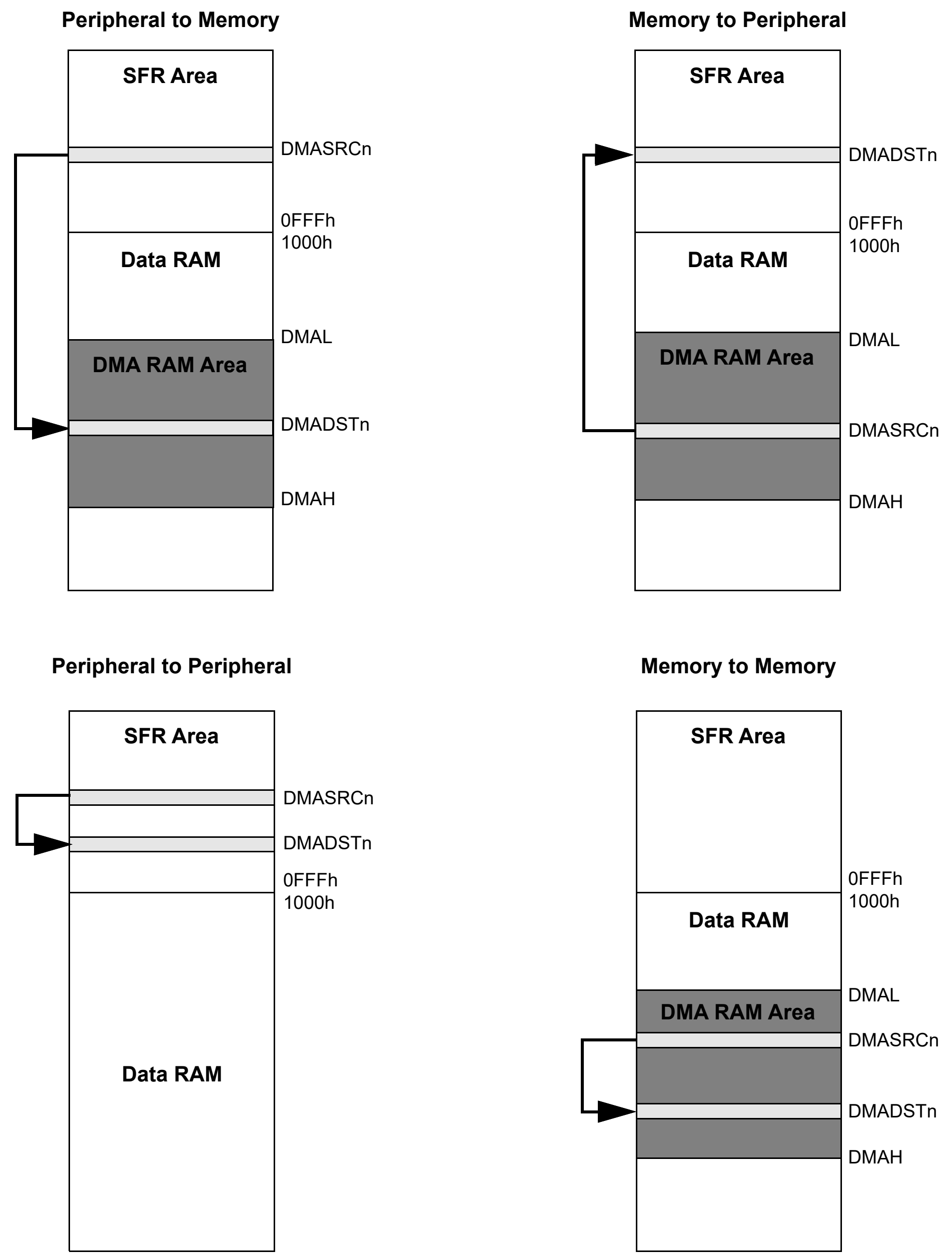
10.1.5 ADDRESSING MODES

The DMA Controller also supports transfers between single addresses or address ranges. The four basic options are:

- Fixed-to-Fixed: Between two constant addresses
- Fixed-to-Block: From a constant source address to a range of destination addresses
- Block-to-Fixed: From a range of source addresses to a single, constant destination address
- Block-to-Block: From a range of source addresses to a range of destination addresses

The option to select auto-increment or auto-decrement of source and/or destination addresses is available for Block Addressing modes.

FIGURE 10-2: TYPES OF DMA DATA TRANSFERS



Note: Relative sizes of memory areas are not shown to scale.

10.1.6 CHANNEL PRIORITY

Each DMA channel functions independently of the others, but also competes with the others for access to the data and DMA buses. When access collisions occur, the DMA Controller arbitrates between the channels using a user-selectable priority scheme. Two schemes are available:

- Round Robin: When two or more channels collide, the lower numbered channel receives priority on the first collision. On subsequent collisions, the higher numbered channels each receive priority based on their channel number.
- Fixed: When two or more channels collide, the lowest numbered channel always receives priority, regardless of past history; however, any channel being actively processed is not available for an immediate retrigger. If a higher priority channel is continually requesting service, it will be scheduled for service after the next lower priority channel with a pending request.

10.2 Typical Setup

To set up a DMA channel for a basic data transfer:

1. Enable the DMA Controller (DMAEN = 1) and select an appropriate channel priority scheme by setting or clearing PRSSEL.
2. Program DMAH and DMAL with appropriate upper and lower address boundaries for data RAM operations.
3. Select the DMA channel to be used and disable its operation (CHEN = 0).
4. Program the appropriate source and destination addresses for the transaction into the channel's DMASRCn and DMADSTn registers.
5. Program the DMACNTn register for the number of triggers per transfer (One-Shot or Continuous modes) or the number of words (bytes) to be transferred (Repeated modes).
6. Set or clear the SIZE bit to select the data size.
7. Program the TRMODE[1:0] bits to select the Data Transfer mode.
8. Program the SAMODE[1:0] and DAMODE[1:0] bits to select the addressing mode.
9. Enable the DMA channel by setting CHEN.
10. Enable the trigger source interrupt.

10.3 Peripheral Module Disable

The channels of the DMA Controller can be individually powered down using the Peripheral Module Disable (PMD) registers.

10.4 Registers

The DMA Controller uses a number of registers to control its operation. The number of registers depends on the number of channels implemented for a particular device.

There are always four module-level registers (one control and three buffer/address):

- DMACON: DMA Engine Control Register (Register 10-1)
- DMAH and DMAL: DMA High and Low Address Limit Registers
- DMABUF: DMA Transfer Data Buffer

Each of the DMA channels implements five registers (two control and three buffer/address):

- DMACHn: DMA Channel n Control Register (Register 10-2)
- DMAINTn: DMA Channel n Interrupt Register (Register 10-3)
- DMASRCn: DMA Data Source Address for Channel n Register
- DMADSTn: DMA Data Destination Address for Channel n Register
- DMACNTn: DMA Transaction Counter for Channel n Register

For dsPIC33CDVL64MC106 devices, there are a total of 34 registers.

REGISTER 10-1: DMACON: DMA ENGINE CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
DMAEN	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PRSSEL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **DMAEN:** DMA Module Enable bit
1 = Enables module
0 = Disables module and terminates all active DMA operation(s)

bit 14-1 **Unimplemented:** Read as ‘0’

bit 0 **PRSSEL:** Channel Priority Scheme Selection bit
1 = Round robin scheme
0 = Fixed priority scheme

REGISTER 10-2: DMACHn: DMA CHANNEL n CONTROL REGISTER

U-0	U-0	U-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NULLW	RELOAD(1)	CHREQ(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **Reserved:** Maintain as ‘0’

bit 11 **Unimplemented:** Read as ‘0’

bit 10 **NULLW:** Null Write Mode bit
1 = A dummy write is initiated to DMASRCn for every write to DMADSTn
0 = No dummy write is initiated

bit 9 **RELOAD:** Address and Count Reload bit(1)
1 = DMASRCn, DMADSTn and DMACNTn registers are reloaded to their previous values upon the start of the next operation
0 = DMASRCn, DMADSTn and DMACNTn are not reloaded on the start of the next operation(2)

bit 8 **CHREQ:** DMA Channel Software Request bit(3)
1 = A DMA request is initiated by software; automatically cleared upon completion of a DMA transfer
0 = No DMA request is pending

bit 7-6 **SAMODE[1:0]:** Source Address Mode Selection bits
11 = Reserved
10 = DMASRCn is decremented based on the SIZE bit after a transfer completion
01 = DMASRCn is incremented based on the SIZE bit after a transfer completion
00 = DMASRCn remains unchanged after a transfer completion

bit 5-4 **DAMODE[1:0]:** Destination Address Mode Selection bits
11 = Reserved
10 = DMADSTn is decremented based on the SIZE bit after a transfer completion
01 = DMADSTn is incremented based on the SIZE bit after a transfer completion
00 = DMADSTn remains unchanged after a transfer completion

bit 3-2 **TRMODE[1:0]:** Transfer Mode Selection bits
11 = Repeated Continuous
10 = Continuous
01 = Repeated One-Shot
00 = One-Shot

bit 1 **SIZE:** Data Size Selection bit
1 = Byte (8-bit)
0 = Word (16-bit)

bit 0 **CHEN:** DMA Channel Enable bit
1 = The corresponding channel is enabled
0 = The corresponding channel is disabled

Note 1: Only the original DMACNTn is required to be stored to recover the original DMASRCn and DMADSTn values.
2: DMACNTn will always be reloaded in Repeated mode transfers, regardless of the state of the RELOAD bit.
3: The number of transfers executed while CHREQ is set depends on the configuration of TRMODE[1:0].

REGISTER 10-3: DMAINTn: DMA CHANNEL n INTERRUPT REGISTER

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DBUFWF[1]	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
HIGHIF[1,2]	LOWIF[1,2]	DONEIF[1]	HALFIF[1]	OVRUNIF[1]	—	—	HALFEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **DBUFWF:** DMA Buffered Data Write Flag bit[1]
1 = The content of the DMA buffer has not been written to the location specified in DMADSTn or DMASRCn in Null Write mode
0 = The content of the DMA buffer has been written to the location specified in DMADSTn or DMASRCn in Null Write mode

bit 14-8 **CHSEL[6:0]:** DMA Channel Trigger Selection bits
See Table 10-1 for a complete list.

bit 7 **HIGHIF:** DMA High Address Limit Interrupt Flag bit[1,2]
1 = The DMA channel has attempted to access an address higher than DMAH or the upper limit of the data RAM space
0 = The DMA channel has not invoked the high address limit interrupt

bit 6 **LOWIF:** DMA Low Address Limit Interrupt Flag bit[1,2]
1 = The DMA channel has attempted to access the DMA SFR address lower than DMAL, but above the SFR range (07FFh)
0 = The DMA channel has not invoked the low address limit interrupt

bit 5 **DONEIF:** DMA Complete Operation Interrupt Flag bit[1]
If CHEN = 1:
1 = The previous DMA session has ended with completion
0 = The current DMA session has not yet completed
If CHEN = 0:
1 = The previous DMA session has ended with completion
0 = The previous DMA session has ended without completion

bit 4 **HALFIF:** DMA 50% Watermark Level Interrupt Flag bit[1]
1 = DMACNTn has reached the halfway point to 0000h
0 = DMACNTn has not reached the halfway point

bit 3 **OVRUNIF:** DMA Channel Overrun Flag bit[1]
1 = The DMA channel is triggered while it is still completing the operation based on the previous trigger
0 = The overrun condition has not occurred

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **HALFEN:** Halfway Completion Watermark bit
1 = Interrupts are invoked when DMACNTn has reached its halfway point and at completion
0 = An interrupt is invoked only at the completion of the transfer

Note 1: Setting these flags in software does not generate an interrupt.
2: Testing for address limit violations (DMASRCn or DMADSTn is either greater than DMAH or less than DMAL) is NOT done before the actual access.

TABLE 10-1: DMA CHANNEL TRIGGER SOURCES

CHSEL[6:0]		Trigger (Interrupt)	CHSEL[6:0]		Trigger (Interrupt)	CHSEL[6:0]		Trigger (Interrupt)
0	00h	INT0 – External Interrupt 0	33	21h	(Reserved, do not use)	66	42h	AD1FLTR3 – Oversample Filter 3
1	01h	SCCP1 IC/OC	34	22h	(Reserved, do not use)	67	43h	AD1FLTR4 – Oversample Filter 4
2	02h	SPI1 Receiver	35	23h	(Reserved, do not use)	68	44h	CLC1 Positive Edge Interrupt
3	03h	SPI1 Transmitter	36	24h	PWM Event C	69	45h	CLC2 Positive Edge Interrupt
4	04h	UART1 Receiver	37	25h	SENT1 TX/RX	70	46h	SPI1 – Fault Interrupt
5	05h	UART1 Transmitter	38	26h	(Reserved, do not use)	71	47h	SPI2 – Fault Interrupt
6	06h	ECC Single-Bit Error	39	27h	ADC Common Interrupt	72	48h	(Reserved, do not use)
7	07h	NVM Write Complete	40	28h	ADC Done AN0	. . .	. . .	(Reserved, do not use)
8	08h	INT1 – External Interrupt 1	41	29h	ADC Done AN1	86	56h	(Reserved, do not use)
9	09h	SI2C1 – I2C1 Client Event	42	2Ah	ADC Done AN2	87	57h	PWM Event D
10	0Ah	MI2C1 – I2C1 Host Event	43	2Bh	ADC Done AN3	88	58h	PWM Event E
11	0Bh	INT2 – External Interrupt 2	44	2Ch	ADC Done AN4	89	59h	PWM Event F
12	0Ch	SCCP2 Interrupt	45	2Dh	ADC Done AN5	90	5Ah	(Reserved, do not use)
13	0Dh	INT3 – External Interrupt 3	46	2Eh	ADC Done AN6	91	5Bh	(Reserved, do not use)
14	0Eh	UART2 Receiver	47	2Fh	ADC Done AN7	92	5Ch	(Reserved, do not use)
15	0Fh	UART2 Transmitter	48	30h	ADC Done AN8	93	5Dh	(Reserved, do not use)
16	10h	SPI2 Receiver	49	31h	ADC Done AN9	94	5Eh	(Reserved, do not use)
17	11h	SPI2 Transmitter	50	32h	ADC Done AN10	95	5Fh	(Reserved, do not use)
18	12h	SCCP3 Interrupt	51	33h	ADC Done AN11	96	60h	CLC3 Positive Edge Interrupt
19	13h	(Reserved, do not use)	52	34h	ADC Done AN12	97	61h	CLC4 Positive Edge Interrupt
20	14h	(Reserved, do not use)	53	35h	ADC Done AN13	98	62h	(Reserved, do not use)
21	15h	SCCP4 Interrupt	54	36h	ADC Done AN14	99	63h	(Reserved, do not use)
22	16h	(Reserved, do not use)	55	37h	ADC Done AN15	100	64h	(Reserved, do not use)
23	17h	(Reserved, do not use)	56	38h	ADC Done AN16	101	65h	(Reserved, do not use)
24	18h	CRC Generator Interrupt	57	39h	ADC Done AN17	102	66h	(Reserved, do not use)
25	19h	PWM Event A	58	3Ah	(Reserved, do not use)	103	67h	(Reserved, do not use)
26	1Ah	(Reserved, do not use)	59	3Bh	(Reserved, do not use)	104	68h	UART3 Receiver
27	1Bh	PWM Event B	60	3Ch	(Reserved, do not use)	105	69h	UART3 Transmitter
28	1Ch	PWM Generator 1	61	3Dh	(Reserved, do not use)	106	6Ah	(Reserved, do not use)
29	1Dh	PWM Generator 2	62	3Eh	(Reserved, do not use)	. . .	. . .	(Reserved, do not use)
30	1Eh	PWM Generator 3	63	3Fh	(Reserved, do not use)	127	7Fh	(Reserved, do not use)
31	1Fh	PWM Generator 4	64	40h	AD1FLTR1 – Oversample Filter 1			
32	20h	(Reserved, do not use)	65	41h	AD1FLTR2 – Oversample Filter 2			

NOTES:

11.0 HIGH-RESOLUTION PWM WITH FINE EDGE PLACEMENT

Note 1: This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"High-Resolution PWM with Fine Edge Placement"** (www.microchip.com/DS70005320).

The High-Resolution PWM (HRPWM) module is a Pulse-Width Modulation (PWM) module to support both motor control and power supply applications. This flexible module provides features to support many types of Motor Control (MC) and Power Control (PC) applications, including:

- AC-to-DC Converters
- DC-to-DC Converters
- AC and DC Motors: BLDC, PMSM, ACIM, SRM, etc.
- Inverters
- Battery Chargers
- Digital Lighting
- Power Factor Correction (PFC)

Note: The Fine Edge Placement feature is not available in this family of devices.

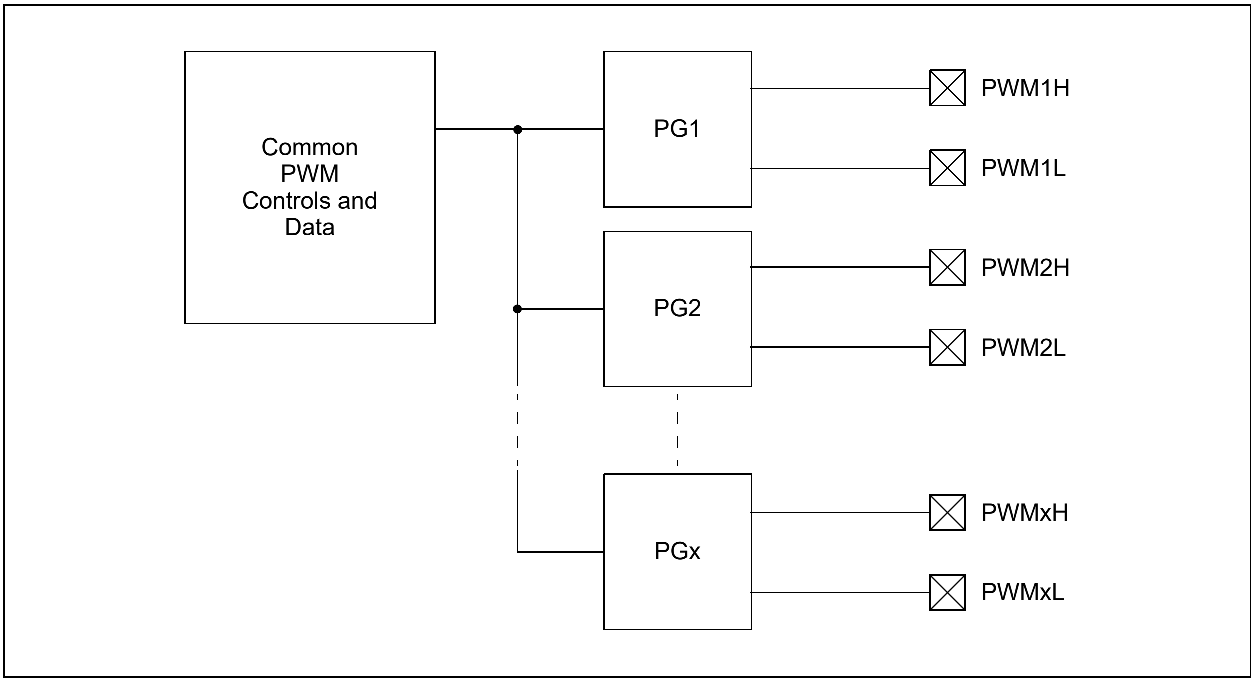
11.1 Features

- Four Independent PWM Generators, each with Dual Outputs
- Operating Modes:
  - Independent Edge mode
  - Variable Phase PWM mode
  - Center-Aligned mode
  - Double Update Center-Aligned mode
  - Dual Edge Center-Aligned mode
  - Dual PWM mode
- Output Modes:
  - Complementary
  - Independent
  - Push-Pull
- Dead-Time Generator
- Leading-Edge Blanking (LEB)
- Output Override for Fault Handling
- Flexible Period/Duty Cycle Updating Options
- Programmable Control Inputs (PCI)
- Advanced Triggering Options
- Six Combinatorial Logic Outputs
- Six PWM Event Outputs

11.2 Architecture Overview

The PWM module consists of a common set of controls and features, and multiple instantiations of PWM Generators (PGs). Each PWM Generator can be independently configured or multiple PWM Generators can be used to achieve complex multiphase systems. PWM Generators can also be used to implement sophisticated triggering, protection and logic functions. A high-level block diagram is shown in Figure 11-1.

FIGURE 11-1: PWM HIGH-LEVEL BLOCK DIAGRAM



11.3 PWM4H Output on PPS

All devices support the capability to output a PWM4H signal via PPS on to any "RPn" pin. If PWM4H PPS output functions are used on 48-pin devices that also have a fixed RP65/PWM4H/RD1 pin, the output signal will be present on both the dedicated and "RPn" pins. The PWM4L/H Output Port Enable bits, PENH and PENL (PG4IOCONH[3:2]), control both dedicated and PPS pins together; it is not possible to disable the dedicated pin and use only PPS.

Given the natural priority of the "RPn" functions above that of the PWM, it is possible to use the PPS output functions on the dedicated RP65/PWM4H/RD1 pin while the PWM4H signal is routed to other pins via PPS.

11.4 Write Restrictions

The LOCK bit (PCLKCON[8]) may be set in software to block writes to certain registers. For more information, refer to **"High-Resolution PWM with Fine Edge Placement"** (www.microchip.com/DS70005320).

The following lock/unlock sequence is required to set or clear the LOCK bit:

1. Write 0x55 to NVMKEY.
2. Write 0xAA to NVMKEY.
3. Clear (or set) the LOCK bit (PCLKCON[8]) as a single operation.

In general, modifications to configuration controls should not be done while the module is running, as indicated by the ON bit (PGxCONL[15]) being set.

11.5 Control Registers

There are two categories of Special Function Registers (SFRs) used to control the operation of the PWM module:

- Common, shared by all PWM Generators
- PWM Generator-specific

An ‘x’ in the register name denotes an instance of a PWM Generator.

A ‘y’ in the register name denotes an instance of the common function.

REGISTER 11-1: PCLKCON: PWM CLOCK CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	LOCK(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	DIVSEL1	DIVSEL0	—	—	MCLKSEL1(2)	MCLKSEL0(2)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **LOCK:** Lock bit(1)
1 = Write-protected registers and bits are locked
0 = Write-protected registers and bits are unlocked

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **DIVSEL[1:0]:** PWM Clock Divider Selection bits
11 = Divide ratio is 1:16
10 = Divide ratio is 1:8
01 = Divide ratio is 1:4
00 = Divide ratio is 1:2

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1-0 **MCLKSEL[1:0]:** PWM Master Clock Selection bits(2)
11 = FVCO/3
10 = FPLLO – Primary PLL post-divider output
01 = FVCO/2
00 = FOSC

Note 1: The LOCK bit is protected against an accidental write. To set this bit, 0x55 and 0xAA values must be written sequentially into the NVMKEY register (see **Section 11.4 “Write Restrictions”**).

2: Changing the MCLKSEL[1:0] bits while ON (PGxCONL[15]) = 1 is not recommended.

REGISTER 11-2: FSCL: FREQUENCY SCALE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSCL[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSCL[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **FSCL[15:0]:** Frequency Scale Register bits
The value in this register is added to the frequency scaling accumulator at each PWM clock. When the accumulated value exceeds the value of FSMINPER, a clock pulse is produced.

REGISTER 11-3: FSMINPER: FREQUENCY SCALING MINIMUM PERIOD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSMINPER[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSMINPER[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **FSMINPER[15:0]:** Frequency Scaling Minimum Period Register bits
This register holds the minimum clock period (maximum clock frequency) that can be produced by the frequency scaling circuit.

REGISTER 11-4: MPHASE: MASTER PHASE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPHASE[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPHASE[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **MPHASE[15:0]:** Master Phase Register bits
This register holds the phase offset value that can be shared by multiple PWM Generators.

REGISTER 11-5: MDC: MASTER DUTY CYCLE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC[15:8][(1)]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC[7:0][(1)]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **MDC[15:0]:** Master Duty Cycle Register bits[(1)]
This register holds the duty cycle value that can be shared by multiple PWM Generators.

Note 1: Duty cycle values less than ‘`0x0008`’ should not be used.

REGISTER 11-6: MPER: MASTER PERIOD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPER[15:8][1]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPER[7:0][1]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **MPER[15:0]:** Master Period Register bits[1]
This register holds the period value that can be shared by multiple PWM Generators.

Note 1: Period values less than ‘`0x0010`’ should not be used.

REGISTER 11-7: CMBTRIGL: COMBINATIONAL TRIGGER REGISTER LOW

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CTA4EN	CTA3EN	CTA2EN	CTA1EN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **CTA4EN:** Enable Trigger Output from PWM Generator #4 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger A signal
0 = Disables

bit 2 **CTA3EN:** Enable Trigger Output from PWM Generator #3 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger A signal
0 = Disables

bit 1 **CTA2EN:** Enable Trigger Output from PWM Generator #2 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger A signal
0 = Disables

bit 0 **CTA1EN:** Enable Trigger Output from PWM Generator #1 as Source for Combinational Trigger A bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger A signal
0 = Disables

REGISTER 11-8: CMBTRIGH: COMBINATIONAL TRIGGER REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CTB4EN	CTB3EN	CTB2EN	CTB1EN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **CTB4EN:** Enable Trigger Output from PWM Generator #4 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger B signal
0 = Disables

bit 2 **CTB3EN:** Enable Trigger Output from PWM Generator #3 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger B signal
0 = Disables

bit 1 **CTB2EN:** Enable Trigger Output from PWM Generator #2 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger B signal
0 = Disables

bit 0 **CTB1EN:** Enable Trigger Output from PWM Generator #1 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR’d into the Combinatorial Trigger B signal
0 = Disables

REGISTER 11-9: LOGCONy: COMBINATORIAL PWM LOGIC CONTROL REGISTER y[(2)]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWMS1y3[(1)]	PWMS1y2[(1)]	PWMS1y1[(1)]	PWMS1y0[(1)]	PWMS2y3[(1)]	PWMS2y2[(1)]	PWMS2y1[(1)]	PWMS2y0[(1)]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
S1yPOL	S2yPOL	PWMLFy1	PWMLFy0	—	PWMLFyD2[(3)]	PWMLFyD1[(3)]	PWMLFyD0[(3)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **PWMS1y[3:0]:** Combinatorial PWM Logic Source #1 Selection bits[(1)]
1111-1000 = Reserved
0111 = PWM4L
0110 = PWM4H
0101 = PWM3L
0100 = PWM3H
0011 = PWM2L
0010 = PWM2H
0001 = PWM1L
0000 = PWM1H

bit 11-8 **PWMS2y[3:0]:** Combinatorial PWM Logic Source #2 Selection bits[(1)]
1111-1000 = Reserved
0111 = PWM4L
0110 = PWM4H
0101 = PWM3L
0100 = PWM3H
0011 = PWM2L
0010 = PWM2H
0001 = PWM1L
0000 = PWM1H

bit 7 **S1yPOL:** Combinatorial PWM Logic Source #1 Polarity bit
1 = Input is inverted
0 = Input is positive logic

bit 6 **S2yPOL:** Combinatorial PWM Logic Source #2 Polarity bit
1 = Input is inverted
0 = Input is positive logic

bit 5-4 **PWMLFy[1:0]:** Combinatorial PWM Logic Function Selection bits
11 = Reserved
10 = PWMS1y ^ PWMS2y (XOR)
01 = PWMS1y & PWMS2y (AND)
00 = PWMS1y | PWMS2y (OR)

bit 3 **Unimplemented:** Read as ‘0’

Note 1: Logic function input will be connected to ‘0’ if the PWM channel is not present.
2: ‘y’ denotes a common instance (A-F).
3: Instances of y = A, C, E of LOGCONy assign logic function output to the PWMxH pin. Instances of y = B, D, F of LOGCONy assign logic function to the PWMxL pin.

REGISTER 11-9: LOGCONy: COMBINATORIAL PWM LOGIC CONTROL REGISTER y[(2)] (CONTINUED)

bit 2-0 **PWMLFyD[2:0]:** Combinatorial PWM Logic Destination Selection bits[(3)]

111-100 = Reserved
011 = Logic function is assigned to PWM4H or PWM4L pin
010 = Logic function is assigned to PWM3H or PWM3L pin
001 = Logic function is assigned to PWM2H or PWM2L pin
000 = No assignment, combinatorial PWM logic function is disabled

Note 1: Logic function input will be connected to '0' if the PWM channel is not present.

2: 'y' denotes a common instance (A-F).

3: Instances of y = A, C, E of LOGCONy assign logic function output to the PWMxH pin. Instances of y = B, D, F of LOGCONy assign logic function to the PWMxL pin.

REGISTER 11-10: PWMEVTy: PWM EVENT OUTPUT CONTROL REGISTER y[(5)]

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
EVTyOEN	EVTyPOL	EVTySTRD	EVTySYNC	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EVTySEL3	EVTySEL2	EVTySEL1	EVTySEL0	—	EVTyPGS2[(2)]	EVTyPGS1[(2)]	EVTyPGS0[(2)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **EVTyOEN:** PWM Event Output Enable bit
1 = Event output signal is output on PWMEy pin
0 = Event output signal is internal only

bit 14 **EVTyPOL:** PWM Event Output Polarity bit
1 = Event output signal is active-low
0 = Event output signal is active-high

bit 13 **EVTySTRD:** PWM Event Output Stretch Disable bit
1 = Event output signal pulse width is not stretched
0 = Event output signal is stretched to eight PWM clock cycles minimum[(1)]

bit 12 **EVTySYNC:** PWM Event Output Sync bit
1 = Event output signal is synchronized to the system clock
0 = Event output is not synchronized to the system clock
Event output signal pulse will be two system clocks when this bit is set and EVTySTRD = 1.

bit 11-8 **Unimplemented:** Read as '0'

bit 7-4 **EVTySEL[3:0]:** PWM Event Selection bits
1111-1010 = Reserved
1001 = ADC Trigger 2 signal
1000 = ADC Trigger 1 signal
0111 = STEER signal (available in Push-Pull Output modes only)[(4)]
0110 = CAHALF signal (available in Center-Aligned modes only)[(4)]
0101 = PCI Fault active output signal
0100 = PCI current limit active output signal
0011 = PCI feed-forward active output signal
0010 = PCI Sync active output signal
0001 = PWM Generator output signal[(3)]
0000 = Source is selected by the PGTRGSEL[2:0] bits

bit 3 **Unimplemented:** Read as '0'

Note 1: The event signal is stretched using peripheral_clk because different PWM Generators may be operating from different clock sources.
2: No event will be produced if the selected PWM Generator is not present.
3: This is the PWM Generator output signal prior to output mode logic and any output override logic.
4: This signal should be the PGx_clk domain signal prior to any synchronization into the system clock domain.
5: 'y' denotes a common instance (A-F).

REGISTER 11-10: PWMEVTy: PWM EVENT OUTPUT CONTROL REGISTER y[5] (CONTINUED)

bit 2-0 **EVTyPGS[2:0]:** PWM Event Source Selection bits[2]
111-100 = Reserved
011 = PWM Generator 4
. . .
000 = PWM Generator 1

Note 1: The event signal is stretched using peripheral_clk because different PWM Generators may be operating from different clock sources.
2: No event will be produced if the selected PWM Generator is not present.
3: This is the PWM Generator output signal prior to output mode logic and any output override logic.
4: This signal should be the PGx_clk domain signal prior to any synchronization into the system clock domain.
5: ‘y’ denotes a common instance (A-F).

REGISTER 11-11: LFSR: LINEAR FEEDBACK SHIFT REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	LFSR[14:8]						
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LFSR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-0 **LFSR[14:0]:** Linear Feedback Shift Register bits
A read of this register will provide a 15-bit pseudorandom value.

REGISTER 11-12: PGxCONL: PWM GENERATOR x CONTROL REGISTER LOW

R/W-0	r-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ON	—	—	—	—	TRGCNT2	TRGCNT1	TRGCNT0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CLKSEL1	CLKSEL0	MODSEL2	MODSEL1	MODSEL0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ON:** Enable bit
1 = PWM Generator is enabled
0 = PWM Generator is not enabled

bit 14 **Reserved:** Maintain as ‘0’

bit 13-11 **Unimplemented:** Read as ‘0’

bit 10-8 **TRGCNT[2:0]:** Trigger Count Select bits
111 = PWM Generator produces eight PWM cycles after triggered
110 = PWM Generator produces seven PWM cycles after triggered
101 = PWM Generator produces six PWM cycles after triggered
100 = PWM Generator produces five PWM cycles after triggered
011 = PWM Generator produces four PWM cycles after triggered
010 = PWM Generator produces three PWM cycles after triggered
001 = PWM Generator produces two PWM cycles after triggered
000 = PWM Generator produces one PWM cycle after triggered

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-3 **CLKSEL[1:0]:** Clock Selection bits
11 = PWM Generator uses Master clock scaled by frequency scaling circuit(1)
10 = PWM Generator uses Master clock divided by clock divider circuit(1)
01 = PWM Generator uses Master clock selected by the MCLKSEL[1:0] (PCLKCON[1:0]) control bits
00 = No clock selected, PWM Generator is in lowest power state (default)

bit 2-0 **MODSEL[2:0]:** Mode Selection bits
111 = Dual Edge Center-Aligned PWM mode (interrupt/register update twice per cycle)
110 = Dual Edge Center-Aligned PWM mode (interrupt/register update once per cycle)
101 = Double-Update Center-Aligned PWM mode
100 = Center-Aligned PWM mode
011 = Reserved
010 = Independent Edge PWM mode, dual output
001 = Variable Phase PWM mode
000 = Independent Edge PWM mode

Note 1: The PWM Generator time base operates from the frequency scaling circuit clock, effectively scaling the duty cycle and period of the PWM Generator output.

REGISTER 11-13: PGxCONH: PWM GENERATOR x CONTROL REGISTER HIGH

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
MDCSEL	MPERSEL	MPHSEL	—	MSTEN	UPDMOD2	UPDMOD1	UPDMOD0
bit 15							bit 8

r-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TRGMOD	—	—	SOCS3[1,2,3]	SOCS2[1,2,3]	SOCS1[1,2,3]	SOCS0[1,2,3]
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **MDCSEL:** Master Duty Cycle Register Select bit
1 = PWM Generator uses MDC register
0 = PWM Generator uses PGxDC register

bit 14 **MPERSEL:** Master Period Register Select bit
1 = PWM Generator uses MPER register
0 = PWM Generator uses PGxPER register

bit 13 **MPHSEL:** Master Phase Register Select bit
1 = PWM Generator uses MPHASE register
0 = PWM Generator uses PGxPHASE register

bit 12 **Unimplemented:** Read as ‘0’

bit 11 **MSTEN:** Master Update Enable bit
1 = PWM Generator broadcasts software set/clear of the UPDREQ status bit and EOC signal to other PWM Generators
0 = PWM Generator does not broadcast the UPDREQ status bit state or EOC signal

bit 10-8 **UPDMOD[2:0]:** PWM Buffer Update Mode Selection bits
011 = Slaved immediate update
Updates Data registers immediately, or as soon as possible, when a Master update request is received. A Master update request will be transmitted if MSTEN = 1 and UPDREQ = 1 for the requesting PWM Generator.
010 = Slaved SOC update
Updates Data registers at the start of the next cycle if a Master update request is received. A Master update request will be transmitted if MSTEN = 1 and UPDREQ = 1 for the requesting PWM Generator.
001 = Immediate update
Updates Data registers immediately, or as soon as possible, if UPDREQ = 1. The UPDATE status bit will be cleared automatically after the update occurs.
000 = SOC update
Updates Data registers at start of next PWM cycle if UPDREQ = 1. The UPDATE status bit will be cleared automatically after the update occurs.[1]

bit 7 **Reserved:** Maintain as ‘0’

Note 1: The PCI selected Sync signal is always available to be OR’d with the selected SOC signal per the SOCS[3:0] bits if the PCI Sync function is enabled.
2: The source selected by the SOCS[3:0] bits MUST operate from the same clock source as the local PWM Generator. If not, the source must be routed through the PCI Sync logic so the trigger signal may be synchronized to the PWM Generator clock domain.
3: PWM Generators are grouped into groups of four: PG1-PG4 and PG5-PG8, if available. Any generator within a group of four may be used to trigger another generator within the same group.

REGISTER 11-13: PGxCONH: PWM GENERATOR x CONTROL REGISTER HIGH (CONTINUED)

bit 6 **TRGMOD:** PWM Generator Trigger Mode Selection bit
1 = PWM Generator operates in Retriggerable mode
0 = PWM Generator operates in Single Trigger mode

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **SOCS[3:0]:** Start-of-Cycle Selection bits(1,2,3)
1111 = TRIG bit or PCI Sync function only (no hardware trigger source is selected)
1110-0101 = Reserved
0100 = Trigger output selected by PG4 PGTRGSEL[2:0] bits (PGxEVTL[2:0])
0011 = Trigger output selected by PG3 PGTRGSEL[2:0] bits (PGxEVTL[2:0])
0010 = Trigger output selected by PG2 PGTRGSEL[2:0] bits (PGxEVTL[2:0])
0001 = Trigger output selected by PG1 PGTRGSEL[2:0] bits (PGxEVTL[2:0])
0000 = Local EOC – PWM Generator is self-triggered

Note 1: The PCI selected Sync signal is always available to be OR'd with the selected SOC signal per the SOCS[3:0] bits if the PCI Sync function is enabled.

2: The source selected by the SOCS[3:0] bits MUST operate from the same clock source as the local PWM Generator. If not, the source must be routed through the PCI Sync logic so the trigger signal may be synchronized to the PWM Generator clock domain.

3: PWM Generators are grouped into groups of four: PG1-PG4 and PG5-PG8, if available. Any generator within a group of four may be used to trigger another generator within the same group.

REGISTER 11-14: PGxSTAT: PWM GENERATOR x STATUS REGISTER

HS/C-0	HS/C-0	HS/C-0	HS/C-0	R-0	R-0	R-0	R-0
SEVT	FLTEVT	CLEVT	FFEVT	SACT	FLTACT	CLACT	FFACT
bit 15							bit 8

W-0	W-0	HS/R/W-0	R-0	W-0	R-0	R-0	R-0
TRSET	TRCLR	CAP[(1)]	UPDATE	UPDREQ	STEER	CAHALF	TRIG
bit 7							bit 0

Legend:	C = Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	‘0’ = Bit is cleared	x = Bit is unknown
-n = Value at POR	‘1’ = Bit is set	U = Unimplemented bit, read as ‘0’	

bit 15 **SEVT:** PCI Sync Event bit
1 = A PCI Sync event has occurred (rising edge on PCI Sync output or PCI Sync output is high when module is enabled)
0 = No PCI Sync event has occurred

bit 14 **FLTEVT:** PCI Fault Active Status bit
1 = A Fault event has occurred (rising edge on PCI Fault output or PCI Fault output is high when module is enabled)
0 = No Fault event has occurred

bit 13 **CLEVT:** PCI Current Limit Status bit
1 = A PCI current limit event has occurred (rising edge on PCI current limit output or PCI current limit output is high when module is enabled)
0 = No PCI current limit event has occurred

bit 12 **FFEVT:** PCI Feed-Forward Active Status bit
1 = A PCI feed-forward event has occurred (rising edge on PCI feed-forward output or PCI feed-forward output is high when module is enabled)
0 = No PCI feed-forward event has occurred

bit 11 **SACT:** PCI Sync Status bit
1 = PCI Sync output is active
0 = PCI Sync output is inactive

bit 10 **FLTACT:** PCI Fault Active Status bit
1 = PCI Fault output is active
0 = PCI Fault output is inactive

bit 9 **CLACT:** PCI Current Limit Status bit
1 = PCI current limit output is active
0 = PCI current limit output is inactive

bit 8 **FFACT:** PCI Feed-Forward Active Status bit
1 = PCI feed-forward output is active
0 = PCI feed-forward output is inactive

bit 7 **TRSET:** PWM Generator Software Trigger Set bit
User software writes a ‘1’ to this bit location to trigger a PWM Generator cycle. The bit location always reads as ‘0’. The TRIG bit will indicate ‘1’ when the PWM Generator is triggered.

bit 6 **TRCLR:** PWM Generator Software Trigger Clear bit
User software writes a ‘1’ to this bit location to stop a PWM Generator cycle. The bit location always reads as ‘0’. The TRIG bit will indicate ‘0’ when the PWM Generator is not triggered.

Note 1: User software may write a ‘1’ to CAP as a request to initiate a software capture. The CAP status bit will be set when the capture event has occurred. No further captures will occur until CAP is cleared by software.

REGISTER 11-14: PGxSTAT: PWM GENERATOR x STATUS REGISTER (CONTINUED)

bit 5 **CAP:** Capture Status bit[1]
1 = PWM Generator time base value has been captured in PGxCAP
0 = No capture has occurred

bit 4 **UPDATE:** PWM Data Register Update Status bit
1 = PWM Data register update is pending – user Data registers are not writable
0 = No PWM Data register update is pending

bit 3 **UPDREQ:** PWM Data Register Update Request bit
User software writes a '1' to this bit location to request a PWM Data register update. The bit location always reads as '0'. The UPDATE status bit will indicate '1' when an update is pending.

bit 2 **STEER:** Output Steering Status bit (Push-Pull Output mode only)
1 = PWM Generator is in 2nd cycle of Push-Pull mode
0 = PWM Generator is in 1st cycle of Push-Pull mode

bit 1 **CAHALF:** Half Cycle Status bit (Center-Aligned modes only)
1 = PWM Generator is in 2nd half of time base cycle
0 = PWM Generator is in 1st half of time base cycle

bit 0 **TRIG:** PWM Trigger Status bit
1 = PWM Generator is triggered and PWM cycle is in progress
0 = No PWM cycle is in progress

Note 1: User software may write a '1' to CAP as a request to initiate a software capture. The CAP status bit will be set when the capture event has occurred. No further captures will occur until CAP is cleared by software.

REGISTER 11-15: PGxIOCONL: PWM GENERATOR x I/O CONTROL REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLMOD	SWAP	OVRENH	OVRENL	OVRDAT1	OVRDAT0	OSYNC1	OSYNC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTDAT1	FLTDAT0	CLDAT1	CLDAT0	FFDAT1	FFDAT0	DBDAT1	DBDAT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **CLMOD:** Current Limit Mode Select bit
1 = If PCI current limit is active, then the PWMxH and PWMxL output signals are inverted (bit flipping), and the CLDAT[1:0] bits are not used
0 = If PCI current limit is active, then the CLDAT[1:0] bits define the PWM output levels

bit 14 **SWAP:** Swap PWM Signals to PWMxH and PWMxL Device Pins bit
1 = The PWMxH signal is connected to the PWMxL pin and the PWMxL signal is connected to the PWMxH pin
0 = PWMxH/L signals are mapped to their respective pins

bit 13 **OVRENH:** User Override Enable for PWMxH Pin bit
1 = OVRDAT1 provides data for output on the PWMxH pin
0 = PWM Generator provides data for the PWMxH pin

bit 12 **OVRENL:** User Override Enable for PWMxL Pin bit
1 = OVRDAT0 provides data for output on the PWMxL pin
0 = PWM Generator provides data for the PWMxL pin

bit 11-10 **OVRDAT[1:0]:** Data for PWMxH/PWMxL Pins if Override is Enabled bits
If OVERENH = 1, then OVRDAT1 provides data for PWMxH.
If OVERENL = 1, then OVRDAT0 provides data for PWMxL.

bit 9-8 **OSYNC[1:0]:** User Output Override Synchronization Control bits
11 = Reserved
10 = User output overrides via the OVRENH/L and OVRDAT[1:0] bits occur when specified by the UPDMOD[2:0] bits in the PGxCONH register
01 = User output overrides via the OVRENH/L and OVRDAT[1:0] bits occur immediately (as soon as possible)
00 = User output overrides via the OVRENH/L and OVRDAT[1:0] bits are synchronized to the local PWM time base (next Start-of-Cycle)

bit 7-6 **FLTDAT[1:0]:** Data for PWMxH/PWMxL Pins if Fault Event is Active bits
If Fault is active, then FLTDAT1 provides data for PWMxH.
If Fault is active, then FLTDAT0 provides data for PWMxL.

bit 5-4 **CLDAT[1:0]:** Data for PWMxH/PWMxL Pins if Current Limit Event is Active bits
If current limit is active, then CLDAT1 provides data for PWMxH.
If current limit is active, then CLDAT0 provides data for PWMxL.

bit 3-2 **FFDAT[1:0]:** Data for PWMxH/PWMxL Pins if Feed-Forward Event is Active bits
If feed-forward is active, then FFDAT1 provides data for PWMxH.
If feed-forward is active, then FFDAT0 provides data for PWMxL.

bit 1-0 **DBDAT[1:0]:** Data for PWMxH/PWMxL Pins if Debug Mode is Active bits
If Debug mode is active and device halted, then DBDAT1 provides data for PWMxH.
If Debug mode is active and device halted, then DBDAT0 provides data for PWMxL.

REGISTER 11-16: PGxIOCONH: PWM GENERATOR x I/O CONTROL REGISTER HIGH

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
—	CAPSRC2[1]	CAPSRC1[1]	CAPSRC0[1]	—	—	—	DTCMPSEL
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMOD1	PMOD0	PENH	PENL	POLH	POLL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **CAPSRC[2:0]:** Time Base Capture Source Selection bits[1]
- 111 = Reserved
- 110 = Reserved
- 101 = Reserved
- 100 = Capture time base value at assertion of selected PCI Fault signal
- 011 = Capture time base value at assertion of selected PCI current limit signal
- 010 = Capture time base value at assertion of selected PCI feed-forward signal
- 001 = Capture time base value at assertion of selected PCI Sync signal
- 000 = No hardware source selected for time base capture – software only

bit 11-9 **Unimplemented:** Read as ‘0’

bit 8 **DTCMPSEL:** Dead-Time Compensation Select bit
- 1 = Dead-time compensation is controlled by PCI feed-forward limit logic
- 0 = Dead-time compensation is controlled by PCI Sync logic

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **PMOD[1:0]:** PWM Generator Output Mode Selection bits
- 11 = Reserved
- 10 = PWM Generator outputs operate in Push-Pull mode
- 01 = PWM Generator outputs operate in Independent mode
- 00 = PWM Generator outputs operate in Complementary mode

bit 3 **PENH:** PWMxH Output Port Enable bit
- 1 = PWM Generator controls the PWMxH output pin
- 0 = PWM Generator does not control the PWMxH output pin

bit 2 **PENL:** PWMxL Output Port Enable bit
- 1 = PWM Generator controls the PWMxL output pin
- 0 = PWM Generator does not control the PWMxL output pin

bit 1 **POLH:** PWMxH Output Polarity bit
- 1 = Output pin is active-low
- 0 = Output pin is active-high

bit 0 **POLL:** PWMxL Output Polarity bit
- 1 = Output pin is active-low
- 0 = Output pin is active-high

Note 1: A capture may be initiated in software at any time by writing a ‘1’ to CAP (PGxSTAT[5]).

REGISTER 11-17: PGxEVTL: PWM GENERATOR x EVENT REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADTR1PS4	ADTR1PS3	ADTR1PS2	ADTR1PS1	ADTR1PS0	ADTR1EN3	ADTR1EN2	ADTR1EN1
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	UPDTRG1	UPDTRG0	PGTRGSEL2[(1)]	PGTRGSEL1[(1)]	PGTRGSEL0[(1)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-11 **ADTR1PS[4:0]:** ADC Trigger 1 Postscaler Selection bits
11111 = 1:32
. . .
00010 = 1:3
00001 = 1:2
00000 = 1:1

bit 10 **ADTR1EN3:** ADC Trigger 1 Source is PGxTRIGC Compare Event Enable bit
1 = PGxTRIGC register compare event is enabled as trigger source for ADC Trigger 1
0 = PGxTRIGC register compare event is disabled as trigger source for ADC Trigger 1

bit 9 **ADTR1EN2:** ADC Trigger 1 Source is PGxTRIGB Compare Event Enable bit
1 = PGxTRIGB register compare event is enabled as trigger source for ADC Trigger 1
0 = PGxTRIGB register compare event is disabled as trigger source for ADC Trigger 1

bit 8 **ADTR1EN1:** ADC Trigger 1 Source is PGxTRIGA Compare Event Enable bit
1 = PGxTRIGA register compare event is enabled as trigger source for ADC Trigger 1
0 = PGxTRIGA register compare event is disabled as trigger source for ADC Trigger 1

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-3 **UPDTRG[1:0]:** Update Trigger Select bits
11 = A write of the PGxTRIGA register automatically sets the UPDATE bit
10 = A write of the PGxPHASE register automatically sets the UPDATE bit
01 = A write of the PGxDC register automatically sets the UPDATE bit
00 = User must set the UPDREQ bit (PGxSTAT[3]) manually

bit 2-0 **PGTRGSEL[2:0]:** PWM Generator Trigger Output Selection bits[(1)]
111 = Reserved
110 = Reserved
101 = Reserved
100 = Reserved
011 = PGxTRIGC compare event is the PWM Generator trigger
010 = PGxTRIGB compare event is the PWM Generator trigger
001 = PGxTRIGA compare event is the PWM Generator trigger
000 = EOC event is the PWM Generator trigger

Note 1: These events are derived from the internal PWM Generator time base comparison events.

REGISTER 11-18: PGxEVTH: PWM GENERATOR x EVENT REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
FLTIEN[(1)]	CLIEN[(2)]	FFIEN[(3)]	SIEN[(4)]	—	—	IEVTSEL1	IEVTSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADTR2EN3	ADTR2EN2	ADTR2EN1	ADTR1OFS4	ADTR1OFS3	ADTR1OFS2	ADTR1OFS1	ADTR1OFS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **FLTIEN:** PCI Fault Interrupt Enable bit[(1)]
1 = Fault interrupt is enabled
0 = Fault interrupt is disabled

bit 14 **CLIEN:** PCI Current Limit Interrupt Enable bit[(2)]
1 = Current limit interrupt is enabled
0 = Current limit interrupt is disabled

bit 13 **FFIEN:** PCI Feed-Forward Interrupt Enable bit[(3)]
1 = Feed-forward interrupt is enabled
0 = Feed-forward interrupt is disabled

bit 12 **SIEN:** PCI Sync Interrupt Enable bit[(4)]
1 = Sync interrupt is enabled
0 = Sync interrupt is disabled

bit 11-10 **Unimplemented:** Read as ‘0’

bit 9-8 **IEVTSEL[1:0]:** Interrupt Event Selection bits
11 = Time base interrupts are disabled (Sync, Fault, current limit and feed-forward events can be independently enabled)
10 = Interrupts CPU at ADC Trigger 1 event
01 = Interrupts CPU at TRIGA compare event
00 = Interrupts CPU at EOC

bit 7 **ADTR2EN3:** ADC Trigger 2 Source is PGxTRIGC Compare Event Enable bit
1 = PGxTRIGC register compare event is enabled as trigger source for ADC Trigger 2
0 = PGxTRIGC register compare event is disabled as trigger source for ADC Trigger 2

bit 6 **ADTR2EN2:** ADC Trigger 2 Source is PGxTRIGB Compare Event Enable bit
1 = PGxTRIGB register compare event is enabled as trigger source for ADC Trigger 2
0 = PGxTRIGB register compare event is disabled as trigger source for ADC Trigger 2

bit 5 **ADTR2EN1:** ADC Trigger 2 Source is PGxTRIGA Compare Event Enable bit
1 = PGxTRIGA register compare event is enabled as trigger source for ADC Trigger 2
0 = PGxTRIGA register compare event is disabled as trigger source for ADC Trigger 2

Note 1: An interrupt is only generated on the rising edge of the PCI Fault active signal.
2: An interrupt is only generated on the rising edge of the PCI current limit active signal.
3: An interrupt is only generated on the rising edge of the PCI feed-forward active signal.
4: An interrupt is only generated on the rising edge of the PCI Sync active signal.

REGISTER 11-18: PGxEVTH: PWM GENERATOR x EVENT REGISTER HIGH (CONTINUED)

bit 4-0 **ADTR1OFS[4:0]:** ADC Trigger 1 Offset Selection bits
11111 = Offset by 31 trigger events
. . .
00010 = Offset by 2 trigger events
00001 = Offset by 1 trigger event
00000 = No offset

Note 1: An interrupt is only generated on the rising edge of the PCI Fault active signal.
2: An interrupt is only generated on the rising edge of the PCI current limit active signal.
3: An interrupt is only generated on the rising edge of the PCI feed-forward active signal.
4: An interrupt is only generated on the rising edge of the PCI Sync active signal.

REGISTER 11-19: PGxyPCIL: PWM GENERATOR xy PCI REGISTER LOW (x = PWM GENERATOR #; y = F, CL, FF OR S)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSYNCDIS	TERM2	TERM1	TERM0	AQPS	AQSS2	AQSS1	AQSS0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWTERM	PSYNC	PPS	PSS4	PSS3	PSS2	PSS1	PSS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **TSYNCDIS:** Termination Synchronization Disable bit
1 = Termination of latched PCI occurs immediately
0 = Termination of latched PCI occurs at PWM EOC

bit 14-12 **TERM[2:0]:** Termination Event Selection bits
111 = Selects PCI Source #9
110 = Selects PCI Source #8
101 = Selects PCI Source #1 (PWM Generator output selected by the PWMPCI[2:0] bits)
100 = PGxTRIGC trigger event
011 = PGxTRIGB trigger event
010 = PGxTRIGA trigger event
001 = Auto-Terminate: Terminate when PCI source transitions from active to inactive
000 = Manual Terminate: Terminate on a write of ‘1’ to the SWTERM bit location

bit 11 **AQPS:** Acceptance Qualifier Polarity Select bit
1 = Inverted
0 = Not inverted

bit 10-8 **AQSS[2:0]:** Acceptance Qualifier Source Selection bits
111 = SWPCI control bit only (qualifier forced to ‘0’)
110 = Selects PCI Source #9
101 = Selects PCI Source #8
100 = Selects PCI Source #1 (PWM Generator output selected by the PWMPCI[2:0] bits)
011 = PWM Generator is triggered
010 = LEB is active
001 = Duty cycle is active (base PWM Generator signal)
000 = No acceptance qualifier is used (qualifier forced to ‘1’)

bit 7 **SWTERM:** PCI Software Termination bit
A write of ‘1’ to this location will produce a termination event. This bit location always reads as ‘0’.

bit 6 **PSYNC:** PCI Synchronization Control bit
1 = PCI source is synchronized to PWM EOC
0 = PCI source is not synchronized to PWM EOC

bit 5 **PPS:** PCI Polarity Select bit
1 = Inverted
0 = Not inverted

REGISTER 11-19: PGxyPCIL: PWM GENERATOR xy PCI REGISTER LOW (x = PWM GENERATOR #; y = F, CL, FF OR S) (CONTINUED)

bit 4-0 **PSS[4:0]:** PCI Source Selection bits
`11111` = CLC1
`11110-11100` = Reserved
`11011` = Comparator 1 output
`11010` = PWM Event D
`11001` = PWM Event C
`11000` = PWM Event B
`10111` = PWM Event A
`10110-10100` = Reserved
`10011` = Device pin, PCI[19]
`10010` = RPn input, PCI18R
`10001` = RPn input, PCI17R
`10000` = RPn input, PCI16R
`01111` = RPn input, PCI15R
`01110` = RPn input, PCI14R
`01101` = RPn input, PCI13R
`01100` = RPn input, PCI12R
`01011` = RPn input, PCI11R
`01010` = RPn input, PCI10R
`01001` = RPn input, PCI9R
`01000` = RPn input, PCI8R
`00100-00111` = Reserved
`00011` = Internally connected to Combo Trigger B
`00010` = Internally connected to Combo Trigger A
`00001` = Internally connected to the output of PWMPCI[2:0] MUX
`00000` = Tied to ‘0’

REGISTER 11-20: PGxyPCIH: PWM GENERATOR xy PCI REGISTER HIGH (x = PWM GENERATOR #; y = F, CL, FF OR S)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
BPEN	BPSEL2[1]	BPSEL1[1]	BPSEL0[1]	—	ACP2	ACP1	ACP0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWPCI	SWPCIM1	SWPCIM0	LATMOD	TQPS	TQSS2	TQSS1	TQSS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **BPEN:** PCI Bypass Enable bit
1 = PCI function is enabled and local PCI logic is bypassed; PWM Generator will be controlled by PCI function in the PWM Generator selected by the BPSEL[2:0] bits
0 = PCI function is not bypassed

bit 14-12 **BPSEL[2:0]:** PCI Bypass Source Selection bits[1]
111-100 = Reserved
011 = PCI control is sourced from PWM Generator 4 PCI logic when BPEN = 1
010 = PCI control is sourced from PWM Generator 3 PCI logic when BPEN = 1
001 = PCI control is sourced from PWM Generator 2 PCI logic when BPEN = 1
000 = PCI control is sourced from PWM Generator 1 PCI logic when BPEN = 1

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **ACP[2:0]:** PCI Acceptance Criteria Selection bits
111 = Reserved
110 = Reserved
101 = Latched any edge
100 = Latched rising edge
011 = Latched
010 = Any edge
001 = Rising edge
000 = Level-sensitive

bit 7 **SWPCI:** Software PCI Control bit
1 = Drives a '1' to PCI logic assigned to by the SWPCIM[1:0] control bits
0 = Drives a '0' to PCI logic assigned to by the SWPCIM[1:0] control bits

bit 6-5 **SWPCIM[1:0]:** Software PCI Control Mode bits
11 = Reserved
10 = SWPCI bit is assigned to termination qualifier logic
01 = SWPCI bit is assigned to acceptance qualifier logic
00 = SWPCI bit is assigned to PCI acceptance logic

bit 4 **LATMOD:** PCI SR Latch Mode bit
1 = SR latch is Reset-dominant in Latched Acceptance modes
0 = SR latch is Set-dominant in Latched Acceptance modes

bit 3 **TQPS:** Termination Qualifier Polarity Select bit
1 = Inverted
0 = Not inverted

Note 1: Selects '0' if selected PWM Generator is not present.

REGISTER 11-20: PGxyPCIH: PWM GENERATOR xy PCI REGISTER HIGH (x = PWM GENERATOR #; y = F, CL, FF OR S) (CONTINUED)

bit 2-0 **TQSS[2:0]:** Termination Qualifier Source Selection bits
111 = SWPCI control bit only (qualifier forced to '0')
110 = Selects PCI Source #9
101 = Selects PCI Source #8
100 = Selects PCI Source #1 (PWM Generator output selected by the PWMPCI[2:0] bits)
011 = PWM Generator is triggered
010 = LEB is active
001 = Duty cycle is active (base PWM Generator signal)
000 = No termination qualifier used (qualifier forced to '1')

Note 1: Selects '0' if selected PWM Generator is not present.

REGISTER 11-21: PGxLEBL: PWM GENERATOR x LEADING-EDGE BLANKING REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LEB[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
LEB[7:0][(1)]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **LEB[15:0]:** Leading-Edge Blanking Period bits[(1)]
Leading-Edge Blanking period. The three LSbs of the blanking time are not used, providing a blanking resolution of eight clock periods. The minimum blanking period is eight clock periods, which occurs when LEB[15:3] = 0.

Note 1: Bits[2:0] are read-only and always remain as '0'.

REGISTER 11-22: PGxLEBH: PWM GENERATOR x LEADING-EDGE BLANKING REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PWMPCI[2:0][(1)]		
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PHR	PHF	PLR	PLF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **PWMPCI[2:0]:** PWM Source for PCI Selection bits[(1)]
111-100 = Reserved
011 = PWM Generator #4 output is made available to PCI logic
010 = PWM Generator #3 output is made available to PCI logic
001 = PWM Generator #2 output is made available to PCI logic
000 = PWM Generator #1 output is made available to PCI logic

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **PHR:** PWMxH Rising Edge Trigger Enable bit
1 = Rising edge of PWMxH will trigger the LEB duration counter
0 = LEB ignores the rising edge of PWMxH

bit 2 **PHF:** PWMxH Falling Edge Trigger Enable bit
1 = Falling edge of PWMxH will trigger the LEB duration counter
0 = LEB ignores the falling edge of PWMxH

bit 1 **PLR:** PWMxL Rising Edge Trigger Enable bit
1 = Rising edge of PWMxL will trigger the LEB duration counter
0 = LEB ignores the rising edge of PWMxL

bit 0 **PLF:** PWMxL Falling Edge Trigger Enable bit
1 = Falling edge of PWMxL will trigger the LEB duration counter
0 = LEB ignores the falling edge of PWMxL

Note 1: The selected PWM Generator source does not affect the LEB counter. This source can be optionally used as a PCI input, PCI qualifier, PCI terminator or PCI terminator qualifier (see the description in Register 11-19 and Register 11-20 for more information).

REGISTER 11-23: PGxPHASE: PWM GENERATOR x PHASE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPHASE[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPHASE[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxPHASE[15:0]:** PWM Generator x Phase Register bits

REGISTER 11-24: PGxDC: PWM GENERATOR x DUTY CYCLE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxDC[15:8](1)							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxDC[7:0](1)							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxDC[15:0]:** PWM Generator x Duty Cycle Register bits(1)

Note 1: Duty cycle values less than ‘0x0008’ should not be used.

REGISTER 11-25: PGxDCA: PWM GENERATOR x DUTY CYCLE ADJUSTMENT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxDCA[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **PGxDCA[7:0]:** PWM Generator x Duty Cycle Adjustment Value bits
Depending on the state of the selected PCI source, the PGxDCA value will be added to the value in the PGxDC register to create the effective duty cycle. When the PCI source is active, PGxDCA is added.

REGISTER 11-26: PGxPER: PWM GENERATOR x PERIOD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPER[15:8][(1)]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPER[7:0][(1)]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxPER[15:0]:** PWM Generator x Period Register bits[(1)]

Note 1: Period values less than ‘0x0010’ should not be used.

REGISTER 11-27: PGxTRIGA: PWM GENERATOR x TRIGGER A REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGA[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGA[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxTRIGA[15:0]:** PWM Generator x Trigger A Register bits

REGISTER 11-28: PGxTRIGB: PWM GENERATOR x TRIGGER B REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGB[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGB[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxTRIGB[15:0]:** PWM Generator x Trigger B Register bits

REGISTER 11-29: PGxTRIGC: PWM GENERATOR x TRIGGER C REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGC[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxTRIGC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxTRIGC[15:0]:** PWM Generator x Trigger C Register bits

REGISTER 11-30: PGxDTL: PWM GENERATOR x DEAD-TIME REGISTER LOW

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DTL[10:8]		
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTL[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-0 **DTL[10:0]:** PWMxL Dead-Time Delay bits

REGISTER 11-31: PGxDTH: PWM GENERATOR x DEAD-TIME REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DTH[10:8]		
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTH[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-0 **DTH[10:0]:** PWMxH Dead-Time Delay bits

REGISTER 11-32: PGxCAP: PWM GENERATOR x CAPTURE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PGxCAP[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PGxCAP[7:0][1]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PGxCAP[15:0]:** PGx Time Base Capture bits[1]

Note 1: PGxCAP[1:0] will read as ‘0’ in Standard Resolution mode.

12.0 MOSFET GATE DRIVER MODULE

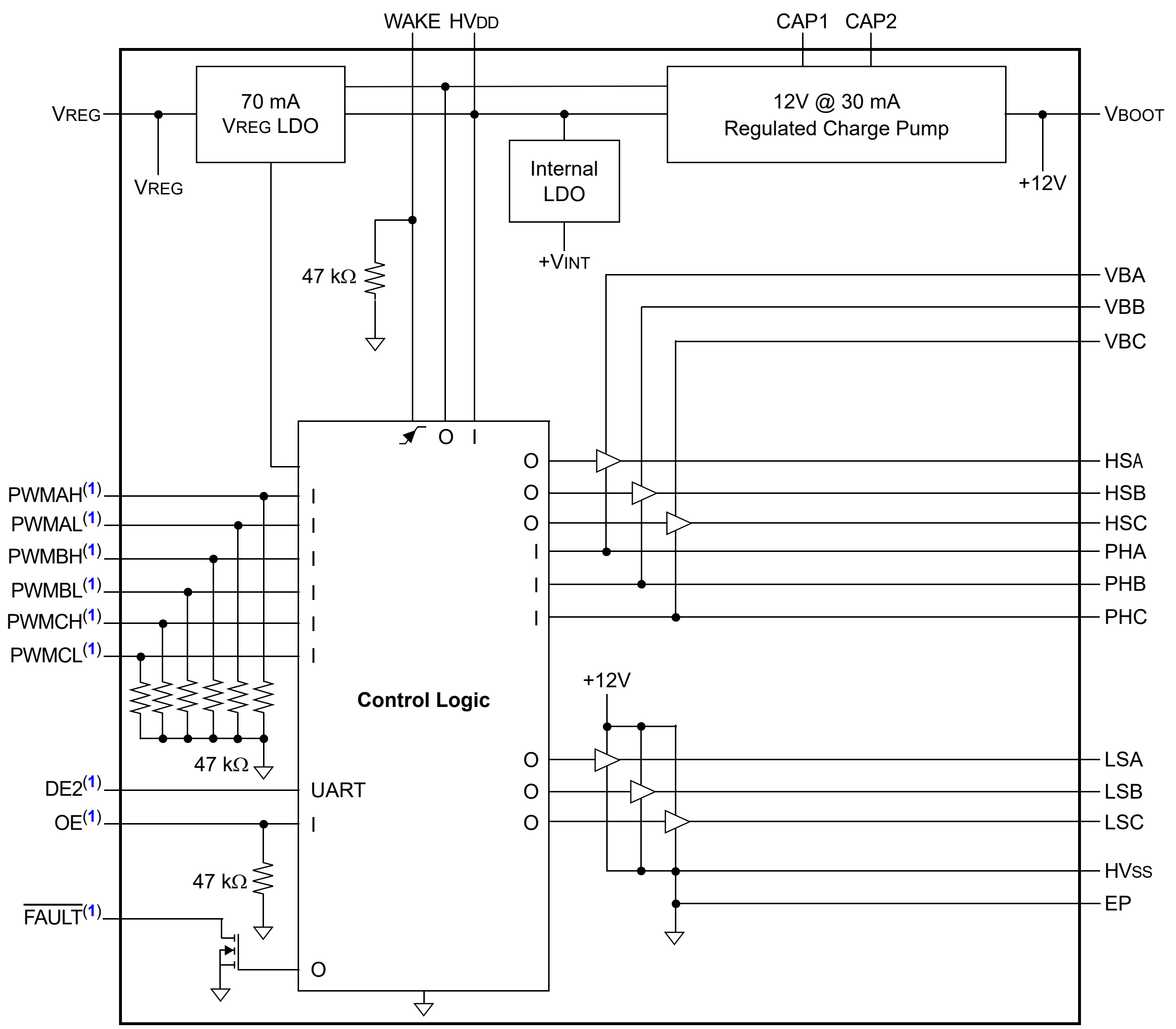
12.1 Functional Overview

The MOSFET Gate Driver module (MOSFET Driver module) incorporates a number of functions, that when paired with the host dsPIC® DSC, provides a single chip solution for controlling low-voltage motors. The MOSFET Driver module includes:

- Bias Generator:
  - +12V Low-Dropout (LDO) Linear Regulator
  - Charge Pump
  - +3.3V @ 70 mA LDO can be used to power the host dsPIC DSC
  - Input supply and temperature supervisor
- Motor Control Unit:
  - External drive for a three-phase bridge with NMOS/NMOS MOSFET pairs
- Communication Port:
  - Half-duplex UART with internal connection to the host dsPIC DSC

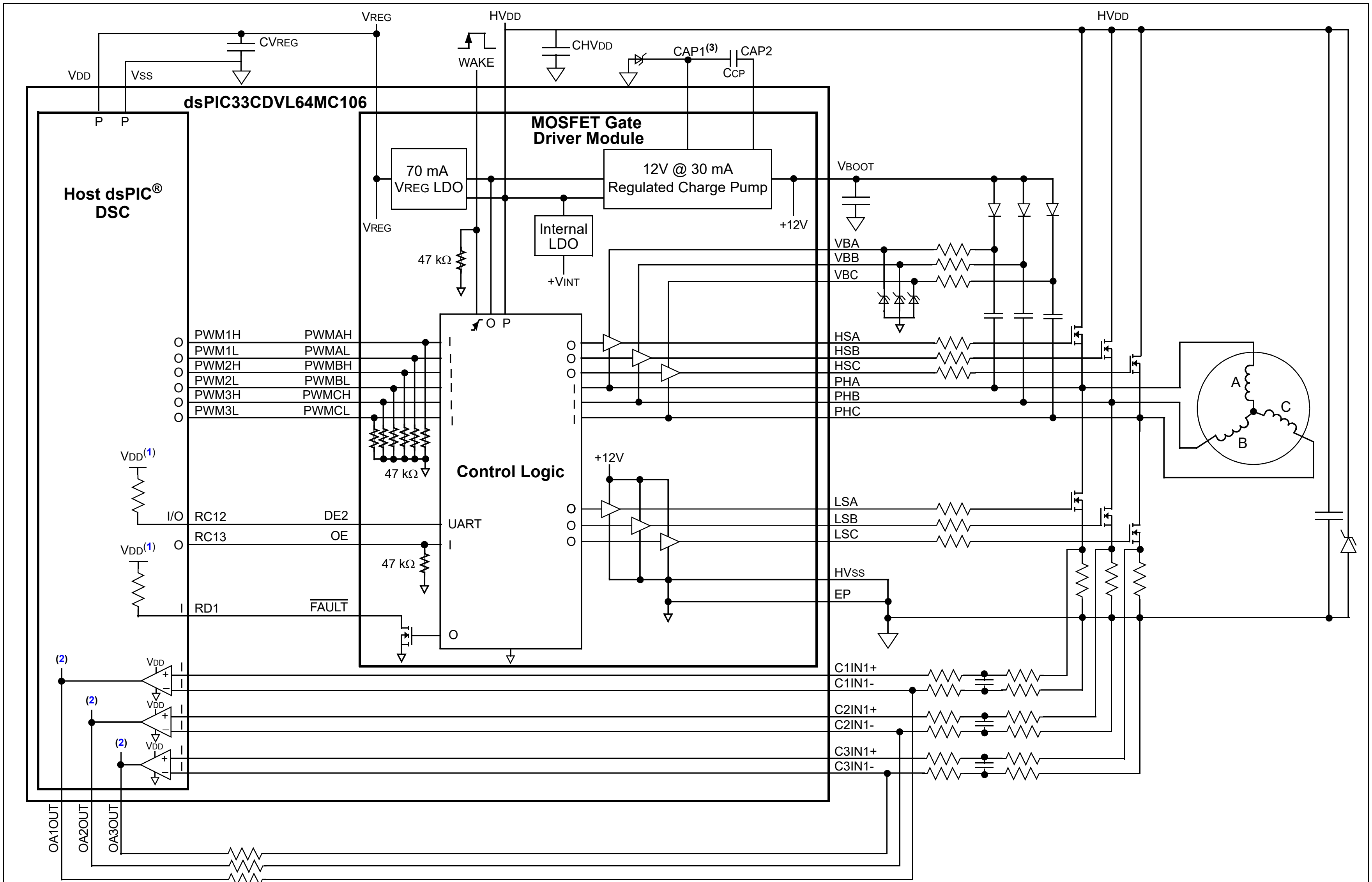
Figure 12-1 depicts the functional block diagram of the MOSFET Driver module and Figure 12-2 depicts a typical application circuit.

FIGURE 12-1: FUNCTIONAL BLOCK DIAGRAM – MOSFET GATE DRIVER MODULE



Note 1: Device interconnect is connected directly to the host microcontroller within the dsPIC33CDVL64MC106 device package.

FIGURE 12-2: MOSFET GATE DRIVER MODULE – TYPICAL MOTOR CONTROL APPLICATION CIRCUIT



Note 1: Internal pull-up resistors for RC12 and RD1 must be enabled by the user by setting the corresponding bits in the CNPUC and CNPUD registers, respectively.

2: The ADC can be configured to internally sample op amp outputs.

3: A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

12.2 Communications Port (DE2)

Open-drain communications node. The DE2 communications is a half-duplex, 9600 baud, 8-bit, no parity communications link. The open-drain DE2 pin must be pulled high by enabling the internal pull-up resistor of port pin RC12 (CNPUC[12] = 1). The pin has a minimum drive capability of 1 mA with a V_{DE2} of ≤50 mV when driving low.

12.3 Low-Side PWM Inputs (PWMAL, PWMBL, PWMCL)

Digital PWM Inputs for low-side driver control. Each input has a 47 kΩ pull-down to ground. The PWM signals may contain dead-time timing or the system may use the CFG2 Configuration register to set the dead time.

12.4 High-Side PWM Inputs (PWMAH, PWMBH, PWMCH)

Digital PWM Inputs for high-side driver control. Each input has a 47 kΩ pull-down to ground. The PWM signals may contain dead-time timing or the system may use the CFG2 Configuration register to set the dead time.

12.5 Output Enable (OE) Input

The Output Enable Input pin is used to enable/disable the output driver and the on-board functions. When OE is high, all device functions are enabled. When OE is low, the device operates in Standby or Sleep mode. When Standby mode is active, the V_{BOOT} output supply and charge pump are disabled. The high-side and low-side gate drive outputs are all set to a Low state within 100 ns of OE going low. The device transitions to Standby or Sleep mode, 1 ms after OE goes low.

The OE pin may be used to clear any hardware Faults. When a Fault occurs, the OE input may be used to clear the Fault by setting the pin low and then high again. The Fault is cleared by the rising edge of the OE signal if the hardware Fault is no longer active.

The OE pin is used to enable Sleep mode when the SLEEP bit in the CFG0 Configuration register is set to a '1'. OE must be low for a minimum of 1 ms before the transition to Standby or Sleep mode will occur. This allows time for OE to be toggled, to clear any Faults, without going into Sleep mode.

The OE pin has an internal 47 kΩ pull-down to ground.

12.6 Fault Output ($\overline{\text{FAULT}}$)

$\overline{\text{FAULT}}$ Output pin. The latched open-drain output will go low while a Fault is active. Table 12-4 shows the Faults that cause the $\overline{\text{FAULT}}$ pin to go low. The pin will stay low until the Fault is inactive and the OE pin is toggled, from low-to-high, to clear the internal Fault latch.

The $\overline{\text{FAULT}}$ pin is able to sink 1 mA of current while maintaining less than a 50 mV drop across the output.

The $\overline{\text{FAULT}}$ pin will also be active (low) upon initial power-up until the state machine completes the V_{REG} state. This may be used to signal an external host that the driver is ready.

12.7 Wake Input (WAKE)

The WAKE pin has an internal 47 kΩ pull-down to ground.

The device will awaken from Sleep mode, on the rising edge of the WAKE pin, after detecting a Low state lasting > t_{WAIT_SETUP} on the pin. The WAKE pin is capable of operating at voltage levels up to HV_{DD}.

12.8 Motor Phase Inputs (PHA, PHB, PHC)

Phase signals from the motor. These signals provide high-side N-channel MOSFET driver bias reference and Back EMF sense input. The phase signals are also used with the bootstrap capacitors to provide a high-side gate drive via the VBx inputs.

12.9 High-Side N-MOSFET Gate Driver Outputs (HSA, HSB, HSC)

High-Side N-Channel MOSFET Gate Drive signals. Connect to the gate of the external MOSFETs. A resistor and gate-to-source capacitor may be used between these pins and the MOSFET gates to limit phase node slew rate and MOSFET current.

12.10 Bootstrap Inputs (VBA, VBB, VBC)

High-side MOSFET driver bias. Connect these pins between the bootstrap charge pump diode cathode and the bootstrap charge pump capacitor. The V_{BOOT} output is used to provide the bootstrap supply voltage at the diode anodes. The phase signals are connected to the other side of the bootstrap charge pump capacitors. The bootstrap capacitors charge to V_{BOOT} when the phase signals are pulled low by the low-side drivers. When the low-side drivers turn off and the high-side drivers turn on, the phase signal is pulled to HV_{DD}, causing the bootstrap voltage to rise to HV_{DD} + 12V.

12.11 Low-Side N-MOSFET Gate Driver Outputs (LSA, LSB, LSC)

Low-Side N-Channel MOSFET Drive signals. Connect to the gate of the external MOSFETs. A resistor and gate-to-source capacitor may be used between these pins and the MOSFET gates to limit current and slew rate.

12.12 Bootstrap Supply (VBOOT)

Bootstrap Supply voltage regulator output. The VBOOT regulator output may be used to power external devices, such as Hall effect sensors or amplifiers. The regulator output requires an output capacitor for stability. The positive side of the output capacitor should be physically located as close to the VBOOT pin as is practical. A minimum capacitance of 4.7 µF is required to ensure stable operation of the VBOOT circuit. Larger capacitances may be used to increase transient performance. The VBOOT regulator is supplied by the internal charge pump when the charge pump is active. When the charge pump is inactive, the VBOOT regulator is supplied by HVDD.

The type of capacitor used may be ceramic, tantalum or aluminum electrolytic. The low-ESR characteristics of the ceramic will yield better noise and PSRR performance at high frequency.

12.13 +3.3V (VREG)

The VREG LDO may be used to power external devices, such as Hall effect sensors, amplifiers or host processors. The VREG LDO is enabled when the device is not in Sleep mode and the supply voltage is above the device shutdown voltage. The LDO requires an output capacitor for stability. The positive side of the output capacitor should be physically located as close to the VREG pin as is practical. For most applications, a minimum 4.7 µF of capacitance will ensure stable operation of the LDO circuit. Larger capacitances may be used to increase transient performance.

The type of capacitor used may be ceramic, tantalum or aluminum electrolytic. The low-ESR characteristics of the ceramic will yield better noise and PSRR performance at high frequency.

12.14 Power Supply Input (HVDD)

Connect HVDD to the main supply voltage. This voltage should be the same as the motor voltage. The driver overcurrent and Overvoltage Shutdown features are relative to the HVDD pin. When the HVDD voltage is separate from the motor voltage, the overcurrent and overvoltage protection features may not be available.

The HVDD voltage must not exceed the maximum operating limits of the device. Connect a bulk capacitor close to this pin for good load step performance and transient protection. The actual capacitance should be equal to or larger than the sum of the capacitors attached to the driver supply outputs. The attached capacitors are the VREG, VBOOT and VBx (three bootstrap capacitors), and the charge pump capacitances.

EQUATION 12-1: HVDD BULK CAPACITOR CALCULATION

$$CHV_{DD} \geq CV_{REG} + CV_{BOOT} + (3 \times CV_{BX}) + C_{CAPx}$$

The type of capacitor used may be ceramic, tantalum or aluminum electrolytic. The low-ESR characteristics of the ceramic will yield lower voltage drop, better noise and PSRR performance at high frequency.

12.15 Charge Pump Flying Capacitor (CAP1, CAP2)

Charge Pump Flying Capacitor connections. Connect the charge pump capacitor across these two pins. The charge pump flying capacitor supplies the power for the VBOOT voltage regulator when the charge pump is active.

A Schottky diode between the CAP1 pin and HVSS is recommended to ensure that the CAP1 pin absolute minimum voltage specification of -0.3V is maintained.

12.16 State Diagram

12.16.1 DE2 RECEIVE AND AUTO-BAUD SEQUENCE

FIGURE 12-3: DE2 DATA RECEPTION AND AUTO-BAUD RATE SEQUENCE (PART 1)

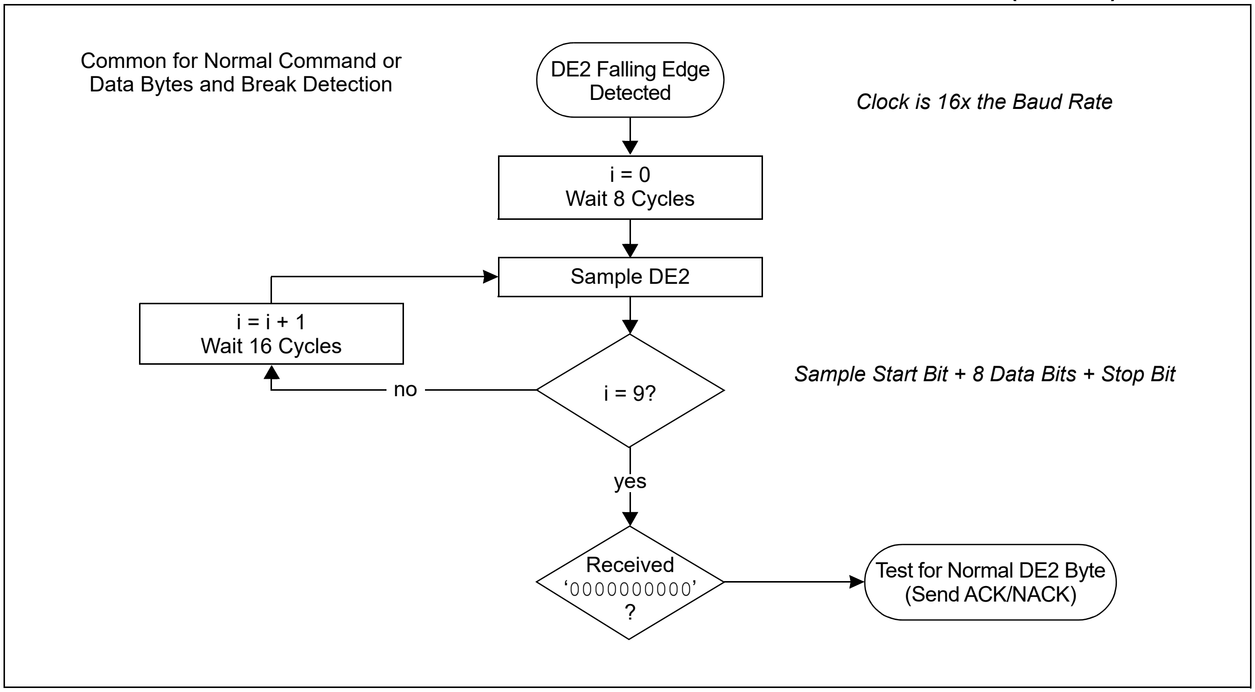
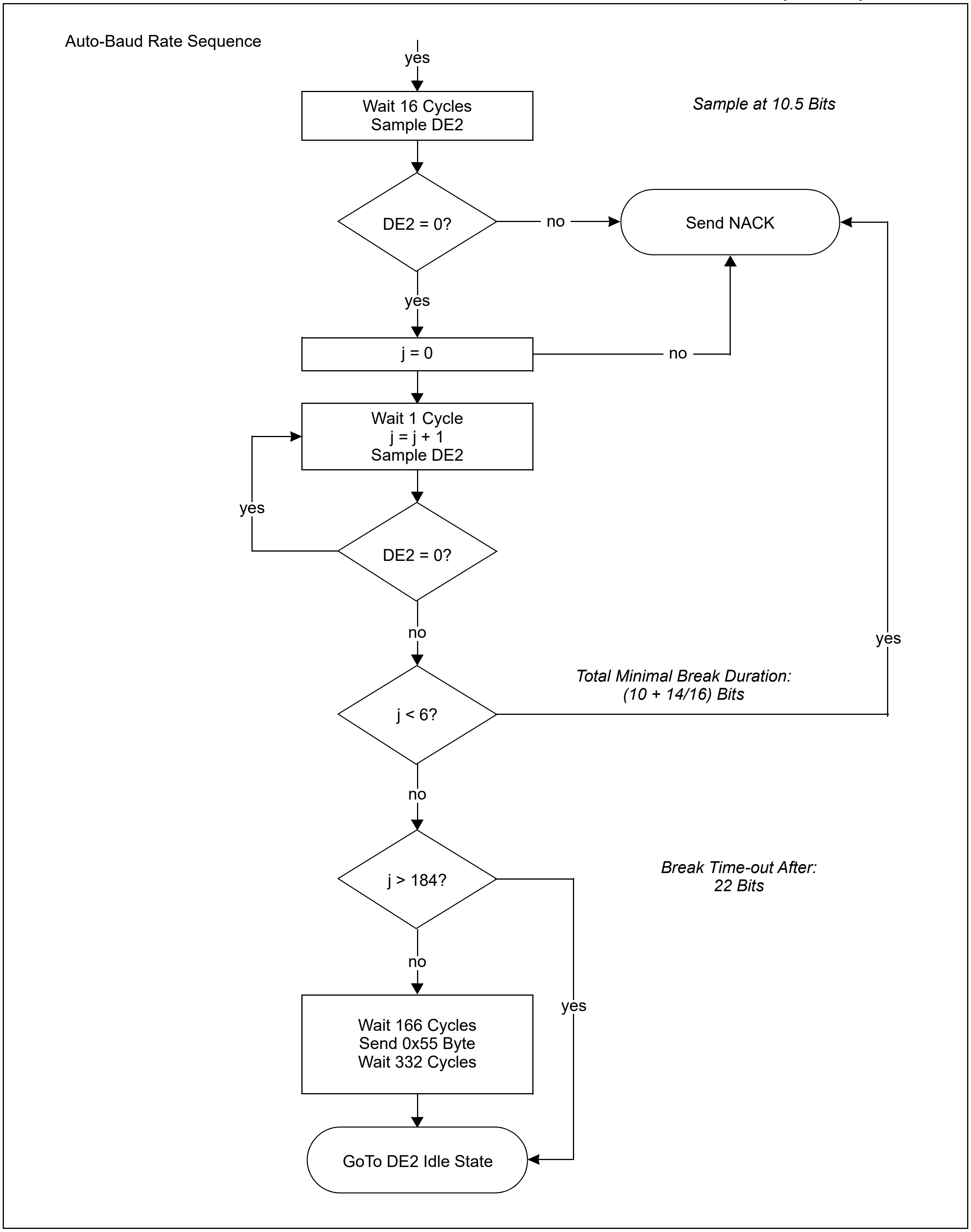


FIGURE 12-4: DE2 DATA RECEPTION AND AUTO-BAUD RATE SEQUENCE (PART 2)



12.17 Bias Generator

The internal bias generator controls several voltage rails. Two fixed output Low-Dropout linear regulators, internal bias supply LDOs and a charge pump are controlled through the bias generator. In addition, the bias generator performs supervisory functions.

12.17.1 CHARGE PUMP

An unregulated charge pump is utilized to boost the input to the VBOOT voltage regulator during low input supply voltage conditions. When the HVDD supply voltage drops below the CPSTART voltage, the charge pump is activated. When activated, 2 x HVDD is presented to the input of the VBOOT regulator. The charge pump is capable of maintaining a VBOOT output of +9V @ 15 mA for a HVDD supply voltage of 5.25V to 7V. The charge pump is capable of maintaining a VBOOT output of +12V @ 20 mA for a supply input voltage of 7V to 13.5V. The charge pump is disabled and bypassed at HVDD voltages above 13.5V, allowing an output voltage of +12V @ 30 mA.

The charge pump requires a capacitor between pins, CAP1 and CAP2. A typical charge pump capacitor should be a 0.1 µF to 1 µF ceramic capacitor.

12.17.2 VBOOT VOLTAGE REGULATOR

The VBOOT voltage regulator rail is used to supply bias voltage for the integrated 3-phase power MOSFET bridge drivers.

The regulator is capable of supplying 30 mA of external load current. The regulator has a minimum overcurrent limit of 40 mA.

The regulator gets its power from the integrated charge pump. When operating at supply voltages (HVDD) that are above +13.5V, the integrated charge pump will be disabled and the HVDD supply will power the VBOOT voltage regulator. The VBOOT regulator output may be lower than the designed voltage, while operating in the HVDD range of +12.5V to +13.0V, due to the dropout voltage of the regulator.

The VBOOT regulator requires an output capacitor, connected from VBOOT to LIN_VSS, to stabilize the internal control loop and to sustain the bootstrap capacitor energy. A minimum of 4.7 µF ceramic output capacitance is required for the VBOOT voltage regulator output; 10 µF is recommended when switching large MOSFET gate loads. The output capacitor forces a time delay between setting the OE pin high (to transition from Standby mode to Active mode) and the VBOOT regulator voltage output rising above the voltage required to set an internal VBootReady flag. The PWM inputs must not be activated while the VBOOT output is charging the output capacitors to the VBootReady voltage (typically 6.0V). The time required before allowing the PWM inputs to become active, after setting OE high to transition from Standby mode to Active mode, is dependent on output capacitance, any extra loads and supply voltage ramp-up time. The user should allow a minimum time of 0.94 ms for the VBOOT output voltage to rise above the VBOOT ready voltage. A voltage of 6V and supply current of 30 mA may be used for this delay estimation. See Equation 12-2.

EQUATION 12-2: OE PIN HIGH TO VBOOT READY

$$dt = (C \times dV)/(I)$$
$$dt = (4.7\ \mu F \times 6V)/(30\ mA)$$
$$dt = 0.94\ ms$$

There is a time-out function that allows the state machine to move from VBOOT to active after 15 ms, regardless of the VBOOT ready voltage. This time-out function prevents the driver from hanging up if the VBOOT voltage is overloaded.

There is also a capacitive voltage divider formed by the three bootstrap capacitors and the VBOOT capacitor. The VBOOT capacitor should be selected so that when the VBOOT supply is active and the bootstrap capacitors are charged, the voltage at the bootstrap capacitors will be greater than the driver Undervoltage Shutdown voltage, 4.5V. For a system with VBOOT = 12V, V_{MIN} = 4.5V and N = 3 x 1 µF CBOOTSTRAP capacitors charging at the same time, the desired CVBOOT capacitor is 1.8 µF (see Equation 12-3). Since the VBOOT supply requires a 4.7 µF capacitor, a 4.7 µF capacitor should be used. The initial voltage seen by the bootstrap capacitors using a 4.7 µF VBOOT capacitor will be 7.32V. See Equation 12-4.

EQUATION 12-3: VBOOT CAPACITOR

$$CV_{BOOT} = \frac{(N \times C_{BOOTSTRAP})}{(V_{BOOT}) \div (V_{MIN}) - 1}$$

EQUATION 12-4: BOOTSTRAP VOLTAGE

$$V_{BOOTSTRAP} = \frac{(V_{BOOT} \times CV_{BOOT})}{(CV_{BOOT} + N \times C_{BOOTSTRAP})}$$

The VBOOT output is disabled when the driver transitions to Standby or Sleep mode.

Table 12-4 shows the Faults that will also disable the VBOOT voltage regulator.

12.17.3 VREG LOW-DROPOUT (LDO) LINEAR REGULATOR

The 3.3V VREG LDO is used for internal gate control logic and can also be used to power the host dsPIC DSC.

The VREG LDO is capable of supplying 70 mA of external load current. The regulator has a minimum overcurrent limit of 80 mA. When the regulator current exceeds the overcurrent limit, the regulator will enter a True Current and Voltage Foldback mode based upon load impedance. As the load impedance decreases towards zero ohms, the regulator output current and voltage will also decrease until the final foldback current and voltage are attained.

When the regulator output voltage drops below the VREG undervoltage limit, the VREGUVF Undervoltage Fault bit will be set in the STAT1 register. The regulator will remain active during the Fault. Table 12-1 shows the registers and bits associated with Faults.

The VREG LDO will be disabled when the HVDD supply voltage Undervoltage Fault occurs. The VREG LDO will be re-enabled when the conditions in **Section 12.18.1 "Voltage Supervisor"** are met.

A minimum of 4.7 µF ceramic output capacitance is required for the VREG LDO; 10 µF is recommended to increase transient performance if supplying the host dsPIC DSC.

The VREG LDO is disabled while the system is in Sleep mode.

12.18 Supervisor

The bias generator incorporates a voltage supervisor and a temperature supervisor.

12.18.1 VOLTAGE SUPERVISOR

The voltage supervisor protects the MOSFET Gate Driver, external power MOSFETs and the host dsPIC DSC from damage due to overvoltage or undervoltage of the input supply, HVDD.

In the event of an undervoltage condition, HVDD < UVLOACT, or overvoltage condition, HVDD > UVLOACT, or VREG LDO undervoltage condition, VREG < VREGUVFACT, the gate drivers, charge pump and VBOOT regulator are switched off. The bias generator, communication port, operational amplifiers and the remainder of the motor control unit remain active. The Failure state is flagged on the FAULT pin and a DE2 status message is sent.

In extreme overvoltage conditions, HVDD > OVSHDNACT, the VREG LDO will be shut down as soon as pin OE is set to a low level. The OVSHDN status flag in the STAT0 register will be set and will remain set until the register is read by a host. The DE2 communications link will be disabled together with the VREG LDO. No Fault message will be sent to the host because the device must shut down immediately to prevent high-voltage damage. The VREG LDO will be re-enabled when the HVDD supply voltage drops below the Overvoltage Lockout value, OVLOINACT.

In the event of a severe undervoltage condition, HVDD < UVSHDNACT, the entire device will shut down except for the minimal circuitry required for a Power-on Reset recovery. A UVSHDN Fault will be set. The VREG output will be turned off and pulled low to create a "clean" shutdown of an attached host processor. The Undervoltage Shutdown condition is a Latched state. The state machine will be restarted from the Power-on Reset state when either of the following two conditions are met:

1. HVDD power is cycled.
2. HVDD rises above UVLOINACT (6.0V).

12.18.2 TEMPERATURE SUPERVISOR

An integrated temperature sensor self-protects the device circuitry. If the temperature rises above the overtemperature shutdown threshold, all device functions are turned off except for those required to send a DE2 Fault message. A Fault will be generated and a DE2 Fault message will be sent. The functions required to send the DE2 Fault message will then be shut down if pin OE is set to a low level. Active operation resumes when the temperature has cooled down below a set hysteresis value and the Fault has been cleared by toggling the OE pin from a logic low to a logic high.

It is desirable to signal the host dsPIC DSC with a warning message before the overtemperature threshold is reached. When the Thermal Warning Temperature (TWARN) set point is exceeded, the DE2 temperature warning will be sent to the host dsPIC DSC. The warning message has no effect upon driver operation. The host dsPIC DSC may then take appropriate actions to reduce the temperature rise.

12.19 Output Enable (OE)

The Output Enable (OE) pin allows the device outputs to be disabled by external control. The Output Enable pin has three modes of operation.

12.19.1 FAULT CLEARING STATE

The OE pin is used to clear any Faults and re-enable the driver. After toggling the OE pin low-to-high, the system requires a minimum time period to re-enable and start up all of the driver blocks. The start-up time is approximately 35 µs. The maximum pulse time for the high-low-high transition to clear the Faults should be less than 900 µs to prevent the system from transitioning through Standby mode. If the high-low-high transition is longer than 1 ms, the device will start up from the Standby state.

Any Fault status bits that are set will be cleared by the low-to-high transition of the OE pin, if and only if, the Fault condition has ceased to exist. If the Fault condition still exists, the active Fault status bit will remain active. No additional Fault messages will be sent for a Fault that remains active.

12.19.2 STANDBY STATE

Standby state is entered when the OE pin goes low for longer than 1 ms and the SLEEP Configuration bit is inactive. When Standby mode is entered, the following subsystems are disabled:

- High-side gate drives (HSA, HSB, HSC) forced low
- Low-side gate drives (LSA, LSB, LSC) forced low
- VBOOT LDO
- Charge pump
- Operational amplifiers if CFG0[6] = `1`
- The VREG LDO and DE2 communications stay active.

12.19.3 SLEEP MODE

Sleep mode is entered when both a `SLEEP` command is sent to the device via DE2 communications and the OE pin is low. The two conditions may occur in any order. The transition to Sleep mode occurs after the last of the two conditions occurs. The SLEEP bit in the CFG0 Configuration register indicates when the device should transition to a low-power mode. The device will operate normally until the OE pin is transitioned low by an external device. At that point in time, the SLEEP bit value determines whether the device transitions to Standby mode or low-power Sleep mode. The Supply Current (ISUP) during Sleep mode will typically be 5 µA. When Sleep mode is activated, most functions will be shut off, including the VREG LDO. Only the Power-on Reset monitor and minimal state machine will remain active to detect a wake-up event. This indicates that the host processor will be shut down if the host is using the VREG LDO regulator for power. The device will stay in the low-power Sleep mode until either of the following conditions is met:

- The WAKE pin transitions high after being in a Low state lasting longer than tWAIT_SETUP.
- Power is cycled.

When an application is utilizing the LIN Transceiver module to provide a WAKE source to the driver via LIN Bus activity (see Section 17.1.8, Wake Pin (LIN_WKIN) for details on waking LIN Transceiver), it is important to ensure the timing requirements surrounding Sleep mode are met. If a rising edge occurs prior to tSLEEP + tWAIT_SETUP elapsing, the rising edge will be ignored and the application will not awake from Sleep. Refer to Section 2.6, MOSFET Gate Driver Sleep Mode Requirements for additional circuitry which will assist in generating the appropriate delay of a WAKE input signal.

In conjunction with the additional hardware in Figure 2-3, the node labeled “INH_MON” should be utilized as an input to the dsPIC DSC to monitor the state of the LIN Transceiver module to ensure the LIN device is in Sleep mode *prior* to sending the Sleep command to the gate driver. Since INH_MON is a product of LIN_INH, it will be low when LIN_EN is set low. After INH_MON is confirmed to be in a Low state via software, the host dsPIC DSC can proceed with putting the gate driver to sleep after tSLEEP_MIN has elapsed. Figure 12-5 shows a block diagram of the recommended software flow. See sections Section 17.1.6, Enable Input Pin (LIN_EN) and Section 17.1.7, Inhibit Output Pin (LIN_INH) for details on the functionality of pins LIN_EN and LIN_INH for the LIN Transceiver.

The MOSFET Gate Driver is not required to retain configuration data while in Sleep mode. When exiting Sleep mode, the host should send a new configuration message to configure the device if the default configuration values are not desired. The same configuration sequence used during power-up may be used when exiting Sleep mode.

When activated, Sleep mode will always be entered regardless of any active Fault. This allows a transition to Sleep mode when the host is powered by the VREG LDO and the regulator is in an unreliable state. The SLEEP bit in the Configuration register will be ignored at power-up until the system has enabled the V_{REG} LDO and the V_{REG} LDO has entered regulation.

12.20 Faults

12.20.1 FAULT PIN OUTPUT ($\overline{\text{FAULT}}$)

The $\overline{\text{FAULT}}$ pin is used as a Fault indicator. The pin is capable of sinking a minimum of 1 mA of current while maintaining less than 50 mV of voltage across the output. An external pull-up resistor to the logic supply is required.

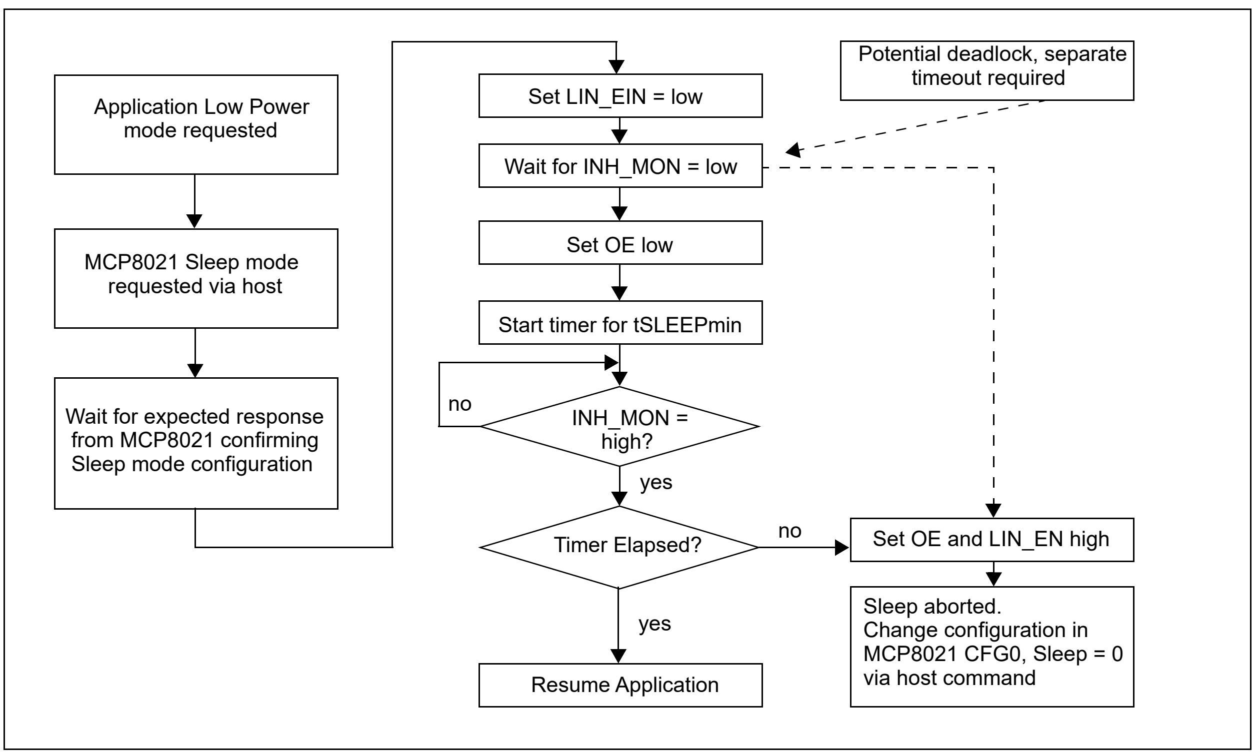
The open-drain $\overline{FAULT}$ pin transitions low when a Fault occurs. Table 12-1 lists the Faults that activate the $\overline{FAULT}$ signal. Warnings do not activate the $\overline{FAULT}$ signal; Table 12-2 lists the warnings.

12.20.2 FAULT HANDLING SEQUENCE

When a Fault occurs, the following steps will occur in sequence.

1. The gate drive outputs will be immediately turned off.
2. The $\overline{FAULT}$ pin output will go low.
3. A message will be sent via the DE2 communications link if, and only if, the Fault is not a HVDD Overvoltage Shutdown (OVSHDNACT).
4. The VREG LDO will be disabled immediately if the Fault is a HVDD Overvoltage Shutdown (OVSHDNACT) or a HVDD Undervoltage Shutdown (UVSHDNACT) Fault.
5. The VREG LDO will be disabled 5 ms after the DE2 message has been sent for an Overtemperature Shutdown (OTSHDN) Fault.

FIGURE 12-5: WAKE EXTENDER EXAMPLE SOFTWARE FLOW



12.20.3 FAULT INDICATOR

A "FAULT" indicator bit resides in the STAT0 register. The bit is the logical 'OR' of all of the Fault bits in the two Status registers. Warnings are not included in the FAULT indicator bit.

The FAULT bit will allow the user to read the STAT0 register in order to determine if a Fault is present in the system. If the bit is set, then the user may request the STAT1 message and interrogate the bits of both status messages to determine what Faults exist.

The Faults that are logically OR'd together to generate the FAULT bit are as follows:

- STAT0:OTPF
- STAT0:UVLOF
- STAT0:OVLOF
- STAT1:REGUVF
- STAT1:XUVLOF
- STAT1:XOCPF

TABLE 12-1: FAULTS

Fault	DE2 Message
Fault Active ('OR' of all Faults)	0x85 0x01
Overtemperature	0x85 0x04
HVDD Input Undervoltage	0x85 0x08
HVDD Input Overvoltage	0x85 0x10
VREG Output Undervoltage	0x86 0x01
External MOSFET Undervoltage Lockout	0x86 0x04
External MOSFET Overcurrent Detection	0x86 0x08

TABLE 12-2: WARNINGS

Fault	DE2 Message
Temperature Warning	0x85 0x02

12.20.4 POWER CONTROL STATUS (PCON)

The PCON[2:0] (STAT0[7:5]) bits are Power Control status bits that may be used to determine the cause of a shutdown. They are not Fault latches. The HVDD Overvoltage Shutdown Fault is an internally latched Fault that does not have a latched FAULT bit in the STAT0 or STAT1 register. That is because the device will be shut down immediately upon entering the Overvoltage Fault condition. When power is back within the device operating range, and the VREG supply is re-enabled, the host will be able to read the STAT0 register to determine the reason for a power cycle. The PCON power status bits will contain the cause of the power cycle. Table 12-3 lists the Power Status register bits in the STAT0 register.

TABLE 12-3: POWER STATUS

PCON[2:0] Status Bits (STAT0[7:5])	DE2 Message
Overtemperature Shutdown (OTSHDN) Occurred	0x85 0xA0
HVDD Overvoltage Shutdown (OVSHDN) Occurred	0x85 0x80
Sleep Occurred	0x85 0x60
HVDD Undervoltage Shutdown (UVSHDN) Occurred	0x85 0x40
Power-on Reset (POR) Occurred	0x85 0x20
Normal Operation	0x85 0x00

12.20.4.1 Internal Function Block Status

Table 12-4 shows the effects of the OE pin, Faults and the SLEEP bit upon the functional status of the internal blocks of the dsPIC33CDVL64MC106 family.

12.20.4.2 Start-up/$\overline{\text{FAULT}}$ Pin State

During device start-up or Power-on Reset (POR), the $\overline{\text{FAULT}}$ pin will stay active (low) to indicate to the host that the device is initializing. The $\overline{\text{FAULT}}$ pin will stay active until the state machine powers up the VREG LDO and completes the VREG state. After the VREG LDO is powered up, the $\overline{\text{FAULT}}$ pin logic checks the state of all of the latched FAULT bits. If any FAULT bit is still active, the $\overline{\text{FAULT}}$ pin will stay active and remain low.

TABLE 12-4: INTERNAL FUNCTION BLOCK STATUS

System State	Fault	Conditions	Sleep Latch	VREG LDO	VBOOT LDO	Motor Drivers	DE2	Op Amps	Internal UVLO, OVLO, OTP
Sleep		OE = 0, SLEEP = 1	W	—	—	—	—	—	—
Standby		OE = 0, SLEEP = 0	—	A	—	—	A	C	A
Operating		OE = 1, $\overline{\text{FAULT}}$ = 1	—	A	A	A	A	A	A
Faults $\overline{\text{FAULT}}$ = 0	Driver OTPF	T_J Temperature > +160°C	—	—	—	—	D	—	A
	HVDDUVLO	HVDD ≤ UVLOINACT	—	A	—	—	A	A	A
	HVDDUVSHDN	HVDD ≤ UVSHDNINACT	—	—	—	—	E	—	—
	HVDDOVLO	HVDD ≥ OVLOINACT	—	A	—	—	A	A	A
	VDDOVSHDN	HVDD ≥ OVSHDNINACT	—	—	—	—	—	—	A
	VREG LDO UVF	VREG ≤ 88% VREG	—	A	—	—	A	A	A
	MOSFET UVLO	VHS[A:C] < VDUVLO VLS[A:C] < VDUVLO	—	A	A	—	A	A	A
	MOSFET OCPF	VDRAIN SOURCE > EXTOC[1:0] setting	—	A	A	—	A	A	A
Warnings $\overline{\text{FAULT}}$ = 1	Driver Temperature	T_J Temperature > 72% TSD_MIN (+115°C for +160°C driver OTP)	—	A	A	A	A	A	A
Power Status	Configuration lost if Power-on Reset, wake from Sleep or recover from HVDD Undervoltage Shutdown occurred	Set at initial power-up when HVDD < UVSHDNACT or when waking from Sleep	—	A	A	A	A	A	A

Legend: — = Inactive (Off); A = Active (On); C = Configurable; D = Inactive (Off) 5 ms after sent Fault message; E = Inactive (Off); R = Receiver Only; W = Wake-up (from Sleep); OCPF = Overcurrent Protection; OTPF = Overtemperature Protection; UVLO = Undervoltage Lockout; OVLO = Overvoltage Lockout; UVF = Undervoltage Fault; UVSHDN = Undervoltage Shutdown; OVSHDN = Overvoltage Shutdown

12.21 Motor Control Unit

The motor control unit is comprised of the following:

- External Drive for a 3-Phase Bridge with NMOS/NMOS MOSFET Pairs
- MOSFET Driver Undervoltage Lockout
- External MOSFET Short-Circuit Current
- FAULT Pin Output
- Cross Conduction Protection
- Programmable Dead Time
- Programmable Blanking Time

12.21.1 EXTERNAL DRIVE FOR A 3-PHASE BRIDGE WITH NMOS/NMOS MOSFET PAIRS

Each motor phase is driven with external NMOS/NMOS MOSFET pairs. These are controlled by a low-side and a high-side gate driver. The gate drivers are controlled by the host dsPIC PWM interconnects found in Table 1-1. A logic high turns the associated gate driver on and a logic low turns the associated gate driver off.

The low-side gate drivers are biased by the VBOOT regulator output, referenced to ground. The high-side gate drivers are a floating drive biased by a bootstrap capacitor circuit. The bootstrap capacitor is charged by the VBOOT regulator whenever the accompanying low-side MOSFET is turned on.

The high-side and low-side driver outputs all go to a Low state whenever there is a Fault, when OE = 0 for more than 1 ms or when Sleep mode is active, regardless of the PWM[A:C]H/L inputs.

12.21.2 MOSFET GATE DRIVER UNDERVOLTAGE LOCKOUT (UVLO)

The MOSFET Gate Driver Undervoltage Lockout Fault detection monitors the available voltage used to drive the external MOSFET gates. The Fault detection is only active while the driver is actively driving the external MOSFET gate. Any time the driver bias voltage is below the gate drive Undervoltage Lockout Threshold (VDUVLO) for a time longer than specified by the tDUVLO parameter, the driver will not turn on when commanded on. A driver Fault will be indicated to the host dsPIC DSC on the FAULT open-drain output pin and also via a DE2 communications `Status_1` message. This is a latched Fault. Clearing the Fault requires either removal of device power or disabling and re-enabling the device via the device Output Enable (OE) input. The EXTUVLO bit in the CFG0 register is used to enable or disable the driver Undervoltage Lockout feature. This protection feature prevents the external MOSFETs from being controlled with a gate voltage not suitable to fully enhance the device.

12.21.3 EXTERNAL MOSFET SHORT-CIRCUIT CURRENT

Short-circuit protection monitors the voltage across the external MOSFETs during an On condition. The high-side driver voltage is measured from HVDD to PH[A:C]. The low-side driver voltage is measured from PH[A:C] to ground. If a monitored voltage rises above a user-configurable threshold after the driver HS[A:C] or LS[A:C] output voltage has been driven high, all drivers will be turned off. A driver Fault will be indicated to the host dsPIC DSC on the open-drain FAULT output pin and also via a DE2 communications `Status_1` message. This is a latched Fault. Clearing the Fault requires either removal of device power or toggling the OE input pin low-to-high. This protection feature helps detect internal motor failures, such as winding to case shorts.

Note:	The driver short-circuit protection is dependent on application parameters. A configuration message is provided for a set number of threshold levels. The MOSFET Gate Driver UVLO and short-circuit protection features have the option to be disabled.

The short-circuit voltage may be set via a DE2 `Set_Cfg_0` message. The EXTOC[1:0] bits of the CFG0 register are used to select the voltage level for the short-circuit comparison. If a monitored voltage differential between HVDD and PH[A:C], or between PH[A:C] and PGND, exceeds the selected voltage level when the MOSFET Gate Driver is active, a Fault will be triggered. The selectable voltage levels are 250 mV, 500 mV, 750 mV and 1000 mV. The EXTSC bit of the CFG0 register is used to enable or disable the MOSFET Gate Driver short-circuit detection.

12.21.4 GATE CONTROL LOGIC

The gate control logic enables level shifting of the digital inputs, polarity control and cross conduction protection.

12.21.4.1 Cross Conduction Protection

If both MOSFETs in the same half-bridge are commanded on by the digital PWM inputs, both will be turned off.

12.21.4.2 Programmable Dead Time

The gate control logic employs a break-before-make dead-time delay that is programmable. A configuration message is provided to configure the driver dead time. The programmable dead times range from 250 ns to 2000 ns (default) in 250 ns increments. The dead time allows the PWM inputs to be direct inversions of each other and still allow proper motor operation. The dead time internally modifies the PWMH/L gate drive timing to prevent cross conduction. The DRVDT[2:0] bits of the CFG2 register are used to set the dead-time value.

12.21.4.3 Programmable Blanking Time

A configuration message is provided to configure the driver current limit blanking time. The blanking time allows the driver to ignore any current spikes that may occur when switching the driver outputs. The allowable blanking times are 500 ns, 1 µs, 2 µs and 4 µs (default). The blanking time will start after the dead-time circuitry has timed out. The DRVBL[1:0] bits of the CFG2 register are used to set the blanking time value.

The blanking time also affects the driver Undervoltage Lockout. The driver Undervoltage Lockout latches the external MOSFET Undervoltage Lockout Fault if the undervoltage condition lasts longer than the time specified by the t_{DUVLO} parameter. The t_{DUVLO} parameter takes into account the blanking time if blanking is in progress.

12.22 Motor Control

The commutation loop of a BLDC motor control is a Phase-Locked Loop (PLL), which locks to the rotor's position. Note that this inner loop does not attempt to modify the position of the rotor, but modifies the commutation times to match whatever position the rotor has. An outer speed loop changes the rotor velocity and the commutation loop locks to the rotor's position to commutate the phases at the correct times.

12.22.1 SIX-STEP SENSORLESS MOTOR CONTROL

Many control algorithms can be implemented using the dsPIC33CDVL64MC106 devices with the internal MOSFET Gate Driver.

The following information provides a starting point for implementing a 3-phase sensorless motor control application. The motor is driven by energizing two windings at a time and sequencing the windings in a six-step per electrical revolution method. This method leaves one winding unenergized at all times. The voltage (Back EMF or BEMF) on that unenergized winding can be monitored to determine the rotor position.

12.22.1.1 Start-up Sequence

When the motor being driven is at rest, the BEMF voltage is equal to zero. The motor needs to be rotating for the BEMF sensor to lock onto the rotor position and commutate the motor. The recommended start-up sequence is to bring the rotor from rest, up to a speed fast enough to allow BEMF sensing. Motor operation is comprised of five modes: Disabled mode, Bootstrap mode, Lock or Align mode, Ramp mode and Run mode. Refer to the commutation state machine in Table 12-5. The order in which the host dsPIC DSC steps through the commutation state machine determines the direction that the motor rotates.

12.22.1.2 Disabled Mode (OE = 0)

When the driver output is disabled (OE = 0), all of the MOSFET driver outputs are set low.

12.22.1.3 Bootstrap Mode

The high-side driver obtains the high-side biasing voltage from the VBOOT LDO, bootstrap diode and bootstrap capacitor. The bootstrap capacitors must first be charged before the high-side drives may be used. The bootstrap capacitors are all charged by activating all three low-side drivers. The active low-side drivers pull their respective phase nodes low, charging the bootstrap capacitors to the VBOOT LDO voltage. The three low-side drivers should be active for at least 1.2 ms per 1 µF of bootstrap capacitance. This assumes a 12V voltage change and 30 mA (10 mA per phase) of current coming from the VBOOT LDO.

12.22.1.4 Lock Mode

Before the motor can be started, the rotor should be in a known position. In Lock mode, the host dsPIC DSC drives Phase B low and Phases A and C high. This aligns the rotor 30 electrical degrees before the center of the first Commutation state. Lock mode must last long enough to allow the motor and its load to settle into this position.

12.22.1.5 Ramp Mode

At the end of Lock mode, Ramp mode is entered. In Ramp mode, the host dsPIC DSC steps through the commutation state machine, increasing the step rate linearly, until a minimum speed is reached that will result in a usable BEMF voltage. Ramp mode is an open-loop commutation. No knowledge of the rotor position is used.

12.22.1.6 Run Mode

At the end of Ramp mode, Run mode is entered. In Run mode, the Back EMF sensor is enabled and commutation is now under the control of the Phase-Locked Loop. Motor speed can be regulated by an outer speed control loop.

TABLE 12-5: COMMUTATION STATE MACHINE

State	Outputs						BEMF Phase
	HSA	HSB	HSC	LSA	LSB	LSC	
OE = 0	OFF	OFF	OFF	OFF	OFF	OFF	N/A
BOOTSTRAP	OFF	OFF	OFF	ON	ON	ON	N/A
LOCK	ON	OFF	ON	OFF	ON	OFF	N/A
1	ON	OFF	OFF	OFF	OFF	ON	Phase B
2	OFF	ON	OFF	OFF	OFF	ON	Phase A
3	OFF	ON	OFF	ON	OFF	OFF	Phase C
4	OFF	OFF	ON	ON	OFF	OFF	Phase B
5	OFF	OFF	ON	OFF	ON	OFF	Phase A
6	ON	OFF	OFF	OFF	ON	OFF	Phase C

12.22.1.7 PWM Speed Control

The inner commutation loop is a Phase-Locked Loop, which locks to the rotor's position. This inner loop does not attempt to modify the position of the rotor, but modifies the commutation times to match whatever position the rotor has. The outer speed loop changes the rotor velocity and the inner commutation loop locks to the rotor's position to commutate the phase at the correct times.

The outer speed loop pulse width modulates the motor drive inverter to produce the desired wave shape and voltage at the motor. The inductance of the motor then integrates this PWM pattern to produce the desired average current, thus controlling the desired torque and speed of the motor. For a trapezoidal BLDC motor drive with six-step commutation, the PWM is used to generate the average voltage to produce the desired motor current and motor speed.

There are two basic methods to PWM the inverter switches. The first method returns the reactive energy in the motor inductance to the source by reversing the voltage on the motor winding during the current decay period. This method is referred to as fast decay or chop-chop. The second method circulates the reactive current in the motor with minimal voltage applied to the inductance. This method is referred to as slow decay or chop-coast.

The preferred control method employs a chop-chop PWM for any situations where the motor is being accelerated, either positively or negatively. For improved efficiency, chop-coast PWM is employed during steady-state conditions. The chop-chop speed loop is implemented by hysteretic control, fixed off time control or Average Current mode control of the motor current. This makes for a very robust controller, since the motor current is always in instantaneous control. The motor speed presented to the chop-chop loop is reduced by approximately 9%. A fixed frequency PWM that only modulates the high-side switches implements the chop-coast loop. The chop-coast loop is presented with the full motor speed, so if it is able to control the speed, the chop-chop loop will never be satisfied and will remain saturated. The chop-chop remains able to assume full control if the motor torque is exceeded, either through a load change or a change in speed that produces acceleration torque. The chop-coast loop will remain saturated, with the chop-chop loop in full control, during start-up and acceleration to full speed. The bandwidth of the chop-coast loop is set to be slower than the chop-chop loop so that any transients will be handled by the chop-chop loop and the chop-coast loop will only be active in steady-state operation.

12.23 DE2 Communication Port

A half-duplex 9600 baud UART interface is available to communicate with the host dsPIC DSC. The port is used to configure the MOSFET Gate Driver and also for status and Fault messages.

12.23.1 COMMUNICATIONS INTERFACE

A half-duplex, 9600 baud, 8-bit bidirectional communications interface is implemented on the DE2 interconnect. The interface consists of eight data bits, one Stop bit and one Start bit.

Dedicated UART hardware may be configured through PPS to transmit and receive messages over the DE2 communications interconnect.

The MOSFET Gate Driver side of the interface is an open-drain configuration and requires that the host dsPIC DSC uses an internal pull-up resistor to pull the DE2 interconnect high.

The auto-baud frequency is temperature-dependent, as illustrated in Figure 12-4. To establish proper DE2 communication, it is recommended to synchronize the host frequency by proceeding the auto-baud function alternatively, as described in **Section 12.23.5 "Auto-Baud Function"**. The time from receiving the last bit of a command message to sending the first bit of the response message ranges from t_{DE2_RSP} to t_{DE2_WAIT}, corresponding to 0 µs to 3.125 ms. The host should refrain from sending additional messages until the previously requested message has been received in order to prevent overwriting the driver response message.

12.23.2 PACKET FORMAT

Every internal driver status change will cause the driver to send a message to the host dsPIC DSC. The interface uses a standard UART baud rate of 9600 bits per second.

In the DE2 protocol, the transmitter and the receiver do not share a clock signal. A clock signal does not emanate from one transmitter to the other receiver. Due to this reason, the protocol is asynchronous. The protocol uses only one line to communicate, so the transmit/receive packet must be done in Half-Duplex mode. A new transmit message is allowed only when a complete packet has been transmitted and responded to.

The host must listen to the DE2 line in order to check for contentions. In case of contention, the host must release the line and wait for at least three packet length times before initiating a new transfer.

Figure 12-6 illustrates a basic DE2 data packet.

12.23.3 PACKET TIMING

While no data are being transmitted, a logic '1' must be placed on the open-drain DE2 line by the host dsPIC DSC using an internal pull-up resistor. A data packet is composed of one Start bit, which is always a logic '0', followed by eight data bits and a Stop bit. The Stop bit must always be a logic '1'. It takes ten bits to transmit a byte of data.

The DE2 interface detects the Start bit by detecting the transition from logic '1' to logic '0' (note that while the data line is Idle, the logic level is high). Once the Start bit is detected, the next data bit's "center" can be assured to be 24 ticks minus 2 (worst-case synchronizer uncertainty) later. From then on, every next data bit center is 16 clock ticks later. Figure 12-7 illustrates this point.

12.23.4 MESSAGE HANDLING

The driver will not transition to Sleep mode while a message is being received. If a message reception is in progress before the OE = 0 to Sleep Mode Transition (t_{SLEEP}) delay times out, the message will be fully received and the contents applied to the Configuration registers if applicable. The SLEEP bit will then be checked and the system enters Sleep mode if the SLEEP bit is still active.

12.23.5 AUTO-BAUD FUNCTION

The MOSFET Gate Driver provides an auto-baud feature that allows the host dsPIC DSC, communicating on the DE2 communications interconnect, to determine the actual baud rate being used by the MOSFET Gate Driver. The feature allows the host to request a 0x55 byte transmission from the MOSFET Gate Driver. The host then determines the MOSFET Gate Driver baud rate and adjusts the host internal Baud Rate Generator (BRG) to match the MOSFET Gate Driver baud rate.

The DE2 pin is used to trigger the auto-baud feature. The host sets the DE2 signal to a logic low for a period of time (auto-baud Break window) that ranges between 1.29 ms and 2.0 ms. The host then releases the DE2 pin back to the host UART control. The host UART then raises the DE2 pin to a logic high value. The MOSFET Gate Driver will respond with a standard NACK ('0b00nnnnnn', where 'nnnnnn' are the six Least Significant bits (LSbs) received) if the DE2 link was held low for less than 1.29 ms and the byte was not interpreted as a valid command. The MOSFET Gate Driver will ignore the current message if the DE2 link is held low for more than 2.0 ms.

If the driver receives a valid auto-baud request in the allotted time frame, the driver will enter an Auto-Baud state, indicating an auto-baud message has been requested. When the auto-baud function is activated, the DE2 subsystem will disable sending all unsolicited messages to the host.

If the internal Auto-Baud state is set, the driver will wait for a minimum of 0.86 ms and a maximum of 1.19 ms. After the wait time has expired, a 0x55 data byte will be immediately sent on the DE2 link by the driver.

The driver will wait 2.00 ms after sending the 0x55 baud rate data over the DE2 link before transmitting any other messages. The driver will then exit the Auto-Baud state and resume normal DE2 operations. The 2.00 ms wait is needed to allow the host to complete the auto-baud verification and update the host UART Baud Rate Generator.

The MOSFET Gate Driver will always monitor the DE2 link for a logic low before attempting to transmit.

The MOSFET Gate Driver will preempt all DE2 communications upon receiving a logic low on the DE2 link which lasts longer than ten bit times at 9600 baud (Break sequence).

The MOSFET Gate Driver will wait for a period up to 2 ms for the DE2 link to change to a Logic High state after the initial detection of a logic low on the DE2 link. If the DE2 link fails to rise to a logic high level within 2 ms of the initial logic low level, the auto-baud message will be canceled and no message will be sent. The auto-baud function will then be complete.

The driver will send any pending unsolicited messages after the auto-baud function has finished.

12.23.6 MESSAGING INTERFACE

A command byte will always have the Most Significant bit (MSb) 7 set to '1'. Bits 6 and 5 are reserved for future use and should be set to '0'. Bits[4:0] are used for commands; that allows for 32 possible commands.

12.23.6.1 Host dsPIC DSC to MOSFET Gate Driver

Messages sent from the host dsPIC DSC to the MOSFET Gate Driver consist of either one or two 8-bit bytes. The first byte transmitted is the command byte. The second byte transmitted, if required, is the data for the command.

If a multibyte command is sent to the MOSFET Gate Driver and no second byte is received by the MOSFET Gate Driver, then a "Command Not Acknowledged" message will be sent back to the host afterwards. The host must start sending the 2nd byte of a two-byte command within 1 ms of completion of the first byte to prevent a NACK message. Once the second byte Start bit is received, the MOSFET Gate Driver internal receiver logic will handle the reception of the data byte. If the data byte Stop bit is not received within the expected reception time for the last received bit, the driver will respond with a NACK message.

12.23.6.2 MOSFET Gate Driver to Host dsPIC DSC

A solicited response byte from the MOSFET Gate Driver will always echo the command byte with bit 7 set to '0' (response) and with bit 6 set to '1' for Acknowledged (ACK) or '0' for Not Acknowledged (NACK). The second byte, if required, will be the data for the host command. Any command that causes an error or is not supported will receive a NACK response.

The MOSFET Gate Driver may send unsolicited command messages to the host dsPIC DSC. All messages to the host controller do not require a response from the host controller.

12.23.7 MESSAGES

12.23.7.1 SET_CFG_0

There is a SET_CFG_0 message that is sent by the host dsPIC DSC to the MOSFET Gate Driver to configure the driver. The SET_CFG_0 message may be sent to the driver at any time. The host is responsible for making sure the system is in a state that will not be compromised by sending the SET_CFG_0 message. The SET_CFG_0 message format is indicated in Table 12-6. The response is indicated in Table 12-7.

12.23.7.2 GET_CFG_0

There is a GET_CFG_0 message that is sent by the host dsPIC DSC to the dsPIC33CDVL64MC106 device to retrieve the device Configuration register. The GET_CFG_0 message format is indicated in Table 12-6. The response is indicated in Table 12-7.

12.23.7.3 STATUS_0 and STATUS_1

There are STATUS_0 and STATUS_1 messages that are sent by the host dsPIC DSC to the MOSFET Gate Driver to retrieve the device STAT0 and STAT1 registers. Unsolicited STATUS_0 and STATUS_1 messages may also be sent to the host by the MOSFET Gate Driver to inform the host of status changes. The unsolicited STATUS_0 and STATUS_1 messages will only be sent when a status bit changes to an Active state. The STATUS_0 and STATUS_1 message format is indicated in Table 12-6. The response is indicated in Table 12-7.

When a STATUS_0 or STATUS_1 message is sent to the host dsPIC DSC in response to a new Fault becoming active, the FAULT bit will be cleared, either by the host issuing a STATUS_0 or STATUS_1 request message, or by the host toggling the OE pin low then high. The FAULT bit will stay active and not be cleared if the Fault condition still exists at the time the host attempted to clear the Fault.

The PCONx bits of the STAT0 register will be set every time the device restarts due to various events (see Table 12-3). When the driver resumes operation, a single unsolicited STATUS_0 message will be sent to the host dsPIC DSC indicating a Reset has occurred. The message will be sent five milliseconds (5 ms) after the VREG LDO has reached its Active state. The host should check the PCONx bits to determine the cause of the power cycle. In all cases, the configuration data may have been lost and should be re-sent to the driver. The PCONx flags are reset by a host STATUS_0 request message. If the host misses the unsolicited STATUS_0 message at start-up, the host may manually request the status by sending a STATUS_0 message to the driver. The PCONx bits of the STAT0 register will contain the source of the Power-on Reset until the STAT0 register is requested by the host.

12.23.7.4 SET_CFG_2

There is a SET_CFG_2 message that is sent by the host dsPIC DSC to the MOSFET Gate Driver to configure the driver current limit blanking time. The SET_CFG_2 message may be sent to the devices at any time. The host is responsible for making sure the system is in a state that will not be compromised by sending the SET_CFG_2 message. The SET_CFG_2 message format is indicated in Table 12-6. The response is indicated in Table 12-7.

12.23.7.5 GET_CFG_2

There is a GET_CFG_2 message that is sent by the host dsPIC DSC to the MOSFET Gate Driver to retrieve the device Configuration Register #2. The GET_CFG_2 message format is indicated in Table 12-6. The response is indicated in Table 12-7.

FIGURE 12-6: DE2 PACKET FORMAT

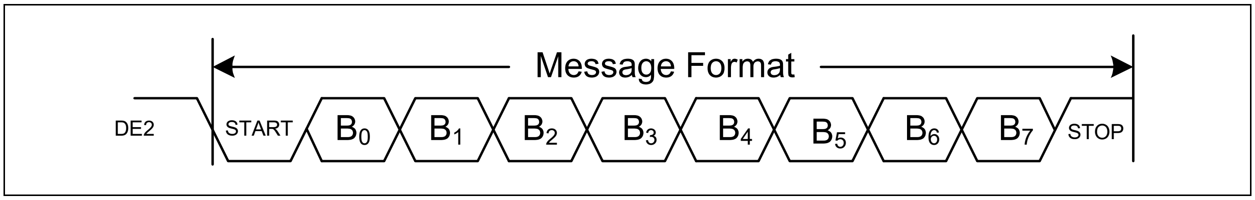


FIGURE 12-7: DE2 PACKET TIMING

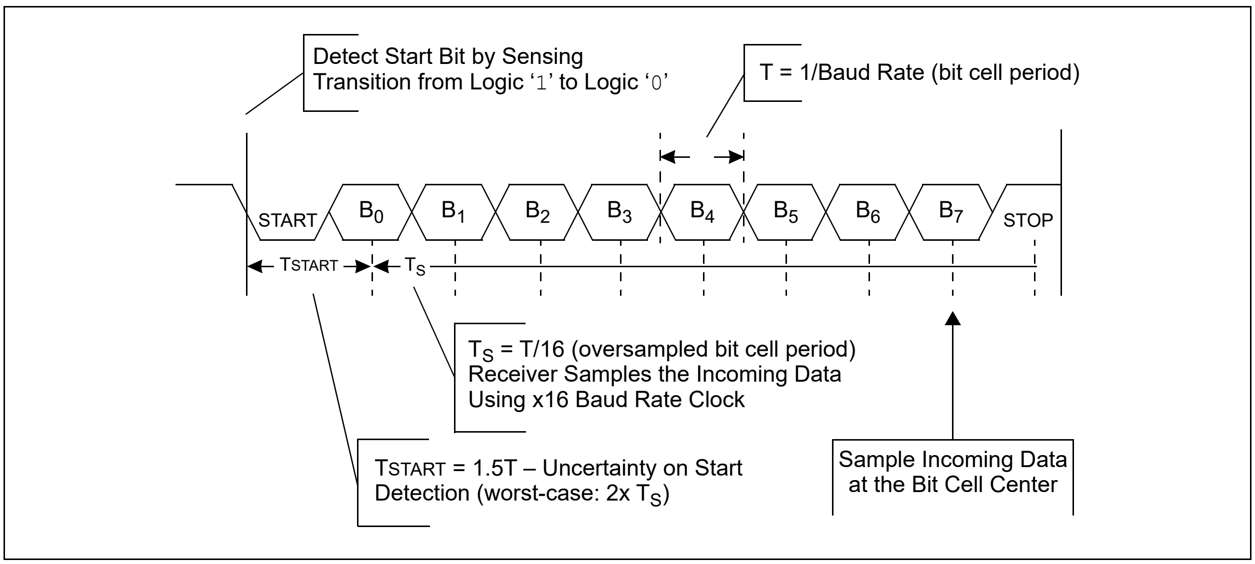


TABLE 12-6: DE2 COMMUNICATION COMMANDS FROM HOST TO dsPIC33CDVL64MC106

Command	Byte	Bit	Value	Description
SET_CFG_0	1		10000001 (81h)	Set Configuration Register 0
	2	7	0	Reserved
		6	0	Reserved
SET_CFG_0		5	0	System enters Standby mode when OE = 0, SLEEP = 0 for more than 1 ms
			1	System enters Sleep mode when OE = 0, SLEEP = 1 for more than 1 ms
		4	0	Reserved
		3	0	Enable external MOSFET Undervoltage Lockout (default)
			1	Disable external MOSFET Undervoltage Lockout
		2	0	Enable external MOSFET short-circuit detection (default)
			1	Disable external MOSFET short-circuit detection
		1:0	00	Set external MOSFET overcurrent limit to 0.250V (default)
			01	Set external MOSFET overcurrent limit to 0.500V
			10	Set external MOSFET overcurrent limit to 0.750V
			11	Set external MOSFET overcurrent limit to 1.000V
GET_CFG_0	1		10000010 (82h)	Get Configuration Register 0
STATUS_0	1		10000101 (85h)	Get Status Register 0
STATUS_1	1		10000110 (86h)	Get Status Register 1
SET_CFG_2	1		10000111 (87h)	Set Configuration Register 2
	2	7:5	00h	Reserved
		4:2	—	Driver dead time (for PWMH /PWML inputs)
			000	2000 ns (default)
			001	1750 ns
			010	1500 ns
			011	1250 ns
			100	1000 ns
			101	750 ns
			110	500 ns
			111	250 ns
		1:0	—	Driver blanking time (ignore switching current spikes)
			00	4 µs (default)
			01	2 µs
			10	1 µs
			11	500 ns
GET_CFG_2	1		10001000 (88h)	Get Configuration Register 2
GET_REV_ID	1		10010000 (90h)	Get device hardware revision

TABLE 12-7: DE2 COMMUNICATION MESSAGES FROM dsPIC33CDVL64MC106 TO HOST

Message	Byte	Bit	Value	Description
SET_CFG_0	1	7:0	00000001 (01h)	Command not Acknowledged (response)
			01000001 (41h)	Command Acknowledged (response)
	2	7	0	Reserved
		6	0	Reserved
		5	0	System enters Standby mode when OE = 0, SLEEP = 0 for more than 1 ms
			1	System enters Sleep mode when OE = 0, SLEEP = 1 for more than 1 ms
		4	0	Reserved
		3	0	External MOSFET Undervoltage Lockout enabled (default)
			1	External MOSFET Undervoltage Lockout disabled
		2	0	External MOSFET short-circuit detection enabled (default)
			1	External MOSFET short-circuit detection disabled
		1:0	00	0.250V external MOSFET overcurrent limit (default)
			01	0.500V external MOSFET overcurrent limit
			10	0.750V external MOSFET overcurrent limit
			11	1.000V external MOSFET overcurrent limit
GET_CFG_0	1	7:0	00000010 (02h)	Command not Acknowledged (response)
			01000010 (42h)	Command Acknowledged (response)
	2	7	0	Reserved
		6	0	Reserved
		5	0	System enters Standby mode when OE = 0, SLEEP = 0 for more than 1 ms
			1	System enters Sleep mode when OE = 0, SLEEP = 1 for more than 1 ms
		4	0	Reserved
		3	0	External MOSFET Undervoltage Lockout enabled
			1	External MOSFET Undervoltage Lockout disabled
		2	0	External MOSFET short-circuit detection enabled
			1	External MOSFET short-circuit detection disabled
		1:0	00	0.250V external MOSFET overcurrent limit
			01	0.500V external MOSFET overcurrent limit
			10	0.750V external MOSFET overcurrent limit
			11	1.000V external MOSFET overcurrent limit

TABLE 12-7: DE2 COMMUNICATION MESSAGES FROM dsPIC33CDVL64MC106 TO HOST (CONTINUED)

Message	Byte	Bit	Value	Description
STATUS_0	1	7:0	00000101 (05h)	Command not Acknowledged (response)
			01000101 (45h)	Command Acknowledged (response)
			10000101 (85h)	Command sent to host (unsolicited)
	2	7:5	101	Overtemperature Shutdown (OTSHDN) occurred
			100	Overvoltage Shutdown (OVSHDN) occurred
			011	Sleep Shutdown (SLEEP) occurred
			010	Undervoltage Shutdown (UVSHDN) occurred
			001	Power-on Reset (POR) occurred
			000	Normal operation
		4	1	Input Overvoltage (OVLOF), HVDD > 32V
		3	1	Input Undervoltage (UVLOF), HVDD < 5.5V
		2	1	Overtemperature (OTPF), T_J > +160°C
		1	1	Overtemperature Warning (OTPW), T_J > +115°C
		0	0	No Fault condition exists
			1	A Fault condition exists
STATUS_1	1	7:0	00000110 (06h)	Command not Acknowledged (response)
			01000110 (46h)	Command Acknowledged (response)
			10000110 (86h)	Command sent to host (unsolicited)
	2	7:4	0	Reserved
		3	1	External MOSFET Overcurrent (XOCPF) detected
		2	1	External MOSFET Undervoltage Lockout (XUVLOF)
		1	0	Reserved
		0	1	VREG LDO Undervoltage Fault (VREGUVF)
SET_CFG_2	1	7:0	00000111 (07h)	Command not Acknowledged (response)
			01000111 (47h)	Command Acknowledged (response)
	2	7:5	00h	Reserved
		4:2	—	Driver dead time (for PWMH /PWML inputs)
			000	2000 ns (default)
			001	1750 ns
			010	1500 ns
			011	1250 ns
			100	1000 ns
			101	750 ns
			110	500 ns
			111	250 ns
		1:0	—	Driver blanking time (ignore Faults)
			00	4000 ns (default)
			01	2000 ns
			10	1000 ns
			11	500 ns

TABLE 12-7: DE2 COMMUNICATION MESSAGES FROM dsPIC33CDVL64MC106 TO HOST (CONTINUED)

Message	Byte	Bit	Value	Description
GET_CFG_2	1	7:0	00001000 (08h)	Command not Acknowledged (response)
			01001000 (48h)	Command Acknowledged (response)
	2	7:5	00h	Reserved
		4:2	—	Driver dead time (for PWMH /PWML inputs)
			000	2000 ns
			001	1750 ns
			010	1500 ns
			011	1250 ns
			100	1000 ns
			101	750 ns
			110	500 ns
			111	250 ns
		1:0	—	Driver blanking time (ignore Faults)
			00	4000 ns
			01	2000 ns
			10	1000 ns
			11	500 ns
GET_REV_ID	1	7:0	00010000 (10h)	Command not Acknowledged (response)
			01010000 (50h)	Command Acknowledged (response)
	2	7:3	00h	Reserved
		2:0	00h-07h	Device hardware revision

12.24 Register Definitions

REGISTER 12-1: CFG0: CONFIGURATION REGISTER 0

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SLEEP	—	EXTUVLO	EXTSC	EXTOC1	EXTOC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **SLEEP:** Sleep Mode bit

Bit may only be changed while in Standby mode.
1 = System enters Sleep mode when OE = 0
0 = System enters Standby mode when OE = 0

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **EXTUVLO:** External MOSFET Undervoltage Lockout bit

1 = Disables
0 = Enables

bit 2 **EXTSC:** External MOSFET Short-Circuit Detection bit

1 = Disables
0 = Enables

bit 1-0 **EXTOC[1:0]:** External MOSFET Overcurrent Limit Value bits

11 = Overcurrent limit set to 1.000V
10 = Overcurrent limit set to 0.750V
01 = Overcurrent limit set to 0.500V
00 = Overcurrent limit set to 0.250V

REGISTER 12-2: CFG2: CONFIGURATION REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DRVDT2	DRVDT1	DRVDT0	DRVBL1	DRVBL0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-2 **DRVDT[2:0]:** Driver Dead-Time Selection bits
111 = 250 ns
110 = 500 ns
101 = 7500 ns
100 = 1000 ns
011 = 1250 ns
010 = 1500 ns
001 = 1750 ns
000 = 2000 ns

bit 1-0 **DRVBL[1:0]:** Driver Blanking Time Selection bits
Bits may only be changed while in Standby mode.
11 = 500 ns
10 = 1000 ns
01 = 2000 ns
00 = 4000 ns

REGISTER 12-3: STAT0: STATUS REGISTER 0

R-0	R-0	R-1	R-0	R-0	R-0	R-0	R-0
PCON2	PCON1	PCON0	OVLOF	UVLOF	OTPF	OTPW	FAULT
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **PCON[2:0]:** Power Control Status bits (configuration lost if non-zero value)
101 = Overtemperature Shutdown (OTSHDN) occurred
100 = Overvoltage Shutdown (OVSHDN) occurred
011 = Sleep (SLEEP) shutdown occurred
110 = Undervoltage Shutdown (UVSHDN) occurred
001 = Power-on Reset (POR) occurred
000 = Normal operation

bit 4 **OVLOF:** Input Overvoltage Lockout Fault bit
1 = HV_{DD} input voltage > 32V
0 = HV_{DD} input voltage < 32V

bit 3 **UVLOF:** Input Undervoltage Lockout Fault bit
1 = HV_{DD} input voltage < 5.5V
0 = HV_{DD} input voltage > 5.5V

bit 2 **OTPF:** Overtemperature Protection Fault bit
1 = Device junction temperature is > +165°C
0 = Device junction temperature is < +165°C

bit 1 **OTPW:** Overtemperature Protection Warning bit
1 = Device junction temperature is > +115°C
0 = Device junction temperature is < +115°C

bit 0 **FAULT:** Fault Status bit
1 = At least one Fault is active
0 = No active Faults

REGISTER 12-4: STAT1: STATUS REGISTER 1

U-0	U-0	U-0	U-0	R-0	R-0	U-0	R-0
—	—	—	—	XOCPF	XUVLOF	—	VREGUVF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **XOCPF:** External MOSFET Overcurrent Protection Fault bit
Only Valid when EXTSC (CFG0[2]) = 0.
1 = External MOSFET V_{DS} > EXTOC[1:0] (CFG0[1:0]) value
0 = External MOSFET V_{DS} < EXTOC[1:0] (CFG0[1:0]) value

bit 2 **XUVLOF:** External MOSFET Gate Drive Undervoltage Fault bit
Only Valid when EXTUVLO (CFG0[3]) = 0.
1 = HSx output voltage < V_{DUVLO}
0 = HSx output voltage > V_{DUVLO}

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **VREGUVF:** V_{REG} LDO Undervoltage Fault bit
1 = V_{REG} LDO output voltage < 88% of target V_{REG}
0 = V_{REG} LDO output voltage > 92% of target V_{REG}

REGISTER 12-5: REV_ID: HARDWARE REVISION ID

U-0	U-0	U-0	U-0	U-0	R-0/1	R-0/1	R-0/1
—	—	—	—	—	REVID[2:0]		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-3 **Unimplemented:** Read as ‘0’

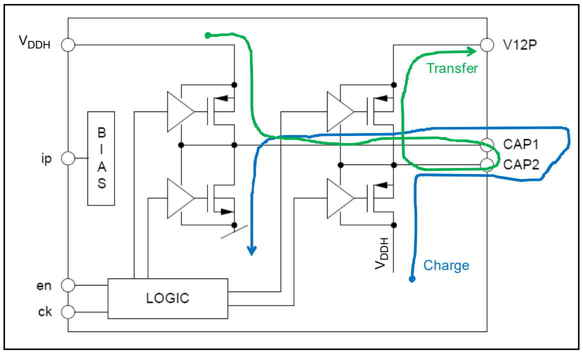
bit 2-0 **REVID[2:0]:** Device Revision bits

12.25 Application Information

12.25.1 COMPONENT CALCULATIONS

12.25.1.1 Charge Pump Capacitors

FIGURE 12-8: CHARGE PUMP



Let:

- I_{OUT} = 20 mA
- f_{CP} = 75 kHz (charge/discharge in one cycle)
- 50% duty cycle
- V_{DDH} = 5.5V (worst case)
- R_{DSON} = 7.5Ω (R_{PMOS}), 3.5Ω (R_{NMOS})
- V12P = 2 × V_{DDH} (ideal)
- C_{ESR} = 20 mΩ (ceramic capacitors)
- V_{DROP} = 100 mV (V_{OUT} ripple)
- T_{CHG} = T_{DCHG} = 0.5 × 1/75 kHz = 6.67 µs

12.25.1.2 Flying Capacitor

The flying capacitor should be chosen to charge to a minimum of 95% (3τ) of V_{DDH} within one half of a switching cycle.

- 3 × τ = T_{CHG}
- τ = T_{CHG}/3
- RC = T_{CHG}/3
- C = T_{CHG}/(R × 3)
- C = 6.67 µs/([7.5Ω + 3.5Ω + 0.02Ω] × 3)
- C = 202 nF

Choose a 180 nF capacitor.

12.25.1.3 Charge Pump Output Capacitor

Solve for the charge pump output capacitance, connected between V12P and ground, that will supply the 20 mA load for one switch cycle. The V_{BOOT} LDO pin on the dsPIC33CDVL64MC106 is the "V12P" pin referenced in the calculations.

- C = I_{OUT} × dt/dV
- C = I_{OUT} × 13.3 µs/(V_{DROP} + I_{OUT} × C_{ESR})
- C = 20 mA × 13.3 µs/(0.1V + 20 mA × 20 mΩ)
- C ≥ 2.65 µF

For stability reasons, the V_{BOOT} LDO and V_{REG} LDO capacitors must be at least 4.7 µF, so choose: C ≥ 4.7 µF.

12.25.1.4 Charging Path (Flying Capacitor Across CAP1 and CAP2)

- $V_{CAP} = V_{DDH} \times (1 - e^{-T/\tau})$
- $V_{CAP} = 5.5V \times (1 - e^{-[6.67\ \mu s/([7.5\Omega + 3.5\Omega + 20\ m\Omega] \times 180\ nF)]})$

V_{CAP} = 5.31V is available for transfer on the first cycle.

12.25.1.5 Transfer Path (Flying and Output Capacitors)

- V12P = V_{DDH} + V_{CAP} – I_{OUT} × dt/C
- V12P = 5.5V + 5.31V – (20 mA × 6.67 µs/180 nF)
- V12P = 10.066V

12.25.1.6 Calculate the Flying Capacitor Voltage Drop in One Cycle While Supplying 20 mA

- dV = I_{OUT} × dt/C
- dV = 20 mA × 6.67 µs/180 nF
- dV = 0.741V @ 20 mA

The second and subsequent transfer cycles will have a higher voltage available for transfer, since the capacitor is not completely depleted with each cycle. V_{CAP} will then be V_{CAP} – dV after the first transfer, plus V_{DDH} – (V_{CAP} – dV) times the RC constant. This repeats for each subsequent cycle, allowing a larger charge pump capacitor to be used if the system will tolerate several charge transfers before requiring full output voltage and current.

Repeating **Section 12.25.1.4 "Charging Path (Flying Capacitor Across CAP1 and CAP2)"** for the second cycle (and subsequent by recalculating for each new value of V_{CAP} after each transfer):

- $V_{CAP} = (V_{CAP} - dV) + (V_{DDH} - (V_{CAP} - dV))(1 - e^{-T/t})$
- $V_{CAP} = (5.31V - 0.741V) + (5.5V - (5.31V - 0.741V)) \times (1 - e^{-[6.67\ \mu s/([7.5W + 3.5W + 20\ mW] \times 180\ nF)]})$
- V_{CAP} = 4.567V + 0.934V × 0.96535

V_{CAP} = 5.468V is available for transfer on the second cycle.

12.25.1.7 Charge Pump Results

The maximum charge pump flying capacitor value is 202 nF to maintain a 95% voltage transfer ratio on the first charge pump cycle. Larger capacitor values may be used, but they will require more cycles to charge to maximum voltage. The minimum required output capacitor value is 2.65 µF to supply 20 mA for 13.3 µs with a 100 mV drop. A larger output capacitor may be used to cover losses due to capacitor tolerance over temperature, capacitor dielectric and PCB losses.

These are approximate calculations. The actual voltages may vary due to incomplete charging or discharging of capacitors per cycle due to load changes. The charge pump calculations assume the charge pump is able to charge up the external boot cap within a few cycles.

12.25.2 BOOTSTRAP CAPACITOR

The high-side driver bootstrap capacitor needs to power the high-side driver and gate for 1/3 of the motor electrical period for a 3-phase BLDC motor operating in Six-Step mode.

Let:

MOSFET Driver Current	=	300 mA
PWM Period	=	50 µs (20 kHz)
Minimum Duty Cycle	=	1% (500 ns)
Maximum Duty Cycle	=	99% (49.5 µs)
V_{IN}	=	12V
Minimum Gate Drive Voltage	=	8V (V_{GS})
Total Gate Charge	=	130 nC (80A MOSFET)
Allowable V_{GS} Drop (V_{DROP})	=	3V
Switch R_{DSON}	=	100 mW
Driver Internal Bias Current	=	20 µA (I_{BIAS})

Solve for the smallest capacitance that can supply:

- 130 nC of charge to the MOSFET gate
- 1 Megohm gate source resistor current
- Driver bias current and switching losses

Q_{MOSFET}	=	130 nC
$Q_{RESISTOR}$	=	$[(V_{GS}/R) \times T_{ON}]$
Q_{DRIVER}	=	$(I_{BIAS} \times T_{ON})$
T_{ON}	=	49.5 µs (99% DC) for worst case
$Q_{RESISTOR}$	=	$Q_{RESISTOR}$
Q_{DRIVER}	=	20 µA × 49.5 µs = 0.99 nC

Sum all of the energy requirements:

- $C = (Q_{MOSFET} + Q_{RESISTOR} + Q_{DRIVER})/V_{DROP}$
- C = (130 nC + 0.594 nC + 0.99 nC)/3V
- C = 43.86 nF

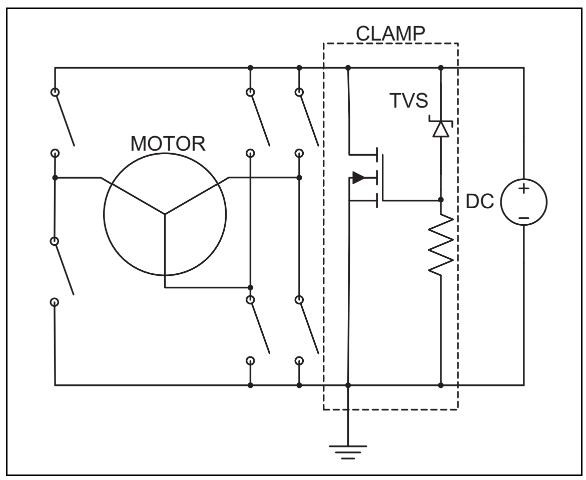
Choose a bootstrap capacitor value that is larger than 43.86 nF.

12.26 Device Protection

12.26.1 MOSFET VOLTAGE SUPPRESSION

When a motor shaft is rotating and power is removed, the magnetism of the motor components will cause the motor to act like a generator. The current that was flowing into the motor will now flow out of the motor. As the motor magnetic field decays, the generator output will also decay. The voltage across the generator terminals will be proportional to the generator current and circuit impedance of the generator circuit. If the power supply is part of the return path for the current and the power supply is disconnected, then the voltage at the generator terminals will increase until the current flows. This voltage increase must be handled externally to the driver. A voltage suppression device may be used to clamp the motor terminal voltage to a level that will not exceed the maximum system operating voltage during the high-voltage transients. A voltage suppressor circuit may be connected from power ground to the motor power supply rail to create a path for the motor current when the supply is disconnected (Figure 12-9). The PCB traces must be capable of carrying the motor current with minimum voltage and temperature rise.

FIGURE 12-9: TRANSIENT VOLTAGE CLAMP



An additional method is to inactivate the high-side drivers and to activate the low-side drivers. This allows current to flow through the low-side external MOSFETs and prevents the voltage from increasing at the power supply terminals.

12.26.2 BOOTSTRAP VOLTAGE SUPPRESSION

The pins which handle the highest voltage during motor operation are the bootstrap pins (VBx). The bootstrap pin voltage is typically V_{BOOT} (12V) higher than the associated phase voltage. When the high-side MOSFET is conducting, the phase pin voltage is typically at HV_{DD} and the bootstrap pin voltage is typically at HV_{DD} + 12V. When the phase MOSFETs switch, current induced voltage transients occur on the phase pins. These currents are caused by the MOSFET body diode reverse recovery and MOSFET turn-on/turn-off times. Those induced voltages cause the bootstrap pin voltages to also increase. Depending on the magnitude of the phase pin voltage, the bootstrap pin voltage may exceed the safe operating voltage of the device. The current induced transients may be reduced by slowing down the turn-on and turn-off times of the MOSFETs. The external MOSFETs may be slowed down by adding a 10 to 100 ohm resistor in series with the gate drive. A 3.3 nF to 10 nF ceramic capacitor may be added that connects each MOSFET gate and source terminal. The added capacitance slows down the switching times of the MOSFET, while allowing the gate resistance to remain small enough to keep the gate clamped off. The added capacitance also results in a lower slew rate of the phase node and limits the shoot-through current caused by the body diode reverse recovery.

The high-side MOSFETs may also be slowed down by inserting a 10Ω to 25Ω resistor between each bootstrap pin and the associated bootstrap diode capacitor junction. Another 25Ω to 50Ω resistor is then added between the gate drive and the MOSFET gate. This results in a high-side turn-on resistance of 25Ω plus the series gate resistor. The high-side turn-off resistance only consists of the series gate resistance and allows for a faster shut-off time. Care must be taken to make sure the voltage drop across the bootstrap pin resistor does not cause an external MOSFET Undervoltage Fault.

When a system motor power supply voltage clamp is not used, 33V or 36V transzorbs may be connected from each bootstrap pin (VBx) to the ground. This will ensure that the bootstrap voltage does not exceed the absolute maximum voltage allowed on the pins. The resistors connected between the bootstrap pins and the bootstrap diode/capacitor junctions, mentioned in the previous paragraph, may also be used in order to limit the transzorb current and reduce the transzorb package size.

12.26.3 FLOATING GATE SUPPRESSION

The gate drive pins may float when the supply voltage is lost or an overvoltage situation shuts down the driver. When an overvoltage condition exists, the driver high-side and low-side outputs are tri-state. Each external MOSFET that is connected to the gate driver should have a gate-to-source resistor to bleed off any charge that may accumulate due to the tri-state. This will help prevent inadvertent turn-on of the MOSFET.

Figure 12-10 shows the location of the overvoltage transzorbs (or equivalent circuits), gate resistors, bootstrap resistors and gate-to-source resistors.

12.26.4 MOSFET BODY DIODE REVERSE RECOVERY SNUBBER

When motor current is flowing through the external MOSFET body diodes and the complimentary MOSFET of the phase pair turns on, the body diode reverse recovery creates a momentary short circuit until the reverse recovery time is complete. When the body diode reverse recovery is complete, the current path is opened, causing the phase node voltage to slew rapidly towards ground or HV_{DD} levels. The rapid slew rate may cause an inversion of the gate-to-source voltage on the MOSFET that is turning on and result in that MOSFET turning off.

The fast slew rate may also cause ringing on the phase node and the sense resistor if the turn-off is too fast.

The first remedy for the low-side turn-off is to slow down the MOSFET gate-to-source turn-off. That causes the R_{DSON} of the low-side MOSFET to gradually increase as the gate voltage drops and the low-side MOSFET slowly turns off. The slow turn-off allows the phase voltage, generated by the motor current flowing through the low-side MOSFET R_{DSON}, to slowly rise towards the positive motor supply level.

The same scenario is also valid for turning on the low-side MOSFET when the high-side MOSFET has just been turned off and current was flowing from the high side into the motor.

The MOSFET body diode reverse recovery situation occurs when the low-side MOSFETs are turned on while the motor current is flowing to the positive source through the high-side MOSFET body diode. The diode reverse recovery time allows a short circuit to exist between the positive supply and the low-side MOSFET drain until the high-side diode is reverse biased and the reverse recovery time has elapsed. The first remedies above should be used to slow the switching speeds of the MOSFETs. Then, a snubber is added to each MOSFET to fine-tune the phase node slew rate and eliminate any further transients. Adding a drain-to-source snubber slows down the slew rate of the phase node and results in a more controlled excursion of the phase node voltage. The snubber consists of a resistor and a capacitor connected in series between the drain and source of the MOSFET. The resistor is chosen to keep the initial snubber voltage below a few volts when peak motor current is flowing through the body diode. The capacitor is then chosen to provide an RC time constant longer than the MOSFET body diode reverse recovery time. A 0.1Ω resistor is typically used, along with a 0.1 µF capacitor to provide an RC of 10 ns.

The power dissipated by the capacitor is calculated by applying Equation 12-5.

EQUATION 12-5: SNUBBER CAPACITOR POWER DISSIPATION

$$P_{DISS} = 2 \times \pi \times f \times C \times V^2 \times Dissipation\ Factor$$

Where:

f = PWM Frequency
C = Capacitance
V = Motor Voltage
$Dissipation\ Factor$ = 2 × p × f × C × ESR = ESR/XC

The capacitor and resistor form factors are chosen to handle the dissipated power.

12.26.5 MOTOR CURRENT SENSE CIRCUITRY

A sense resistor in series with the bridge ground return provides a current signal for feedback. This resistor should be non-inductive to minimize ringing from high di/dt. Any inductance in the power circuit represents potential problems in the form of additional voltage stress and ringing, as well as increasing switching times. While impractical to eliminate, careful layout and bypassing will minimize these effects. The output stage should be as compact as heat sinking will allow, with wide, short traces carrying all pulsed currents. Each half-bridge should be separately bypassed with a low-ESR/ESL capacitor, decoupling it from the rest of the circuit. Some layouts will allow the input filter capacitor to be split into three smaller values and serve double duty as the half-bridge bypass capacitors.

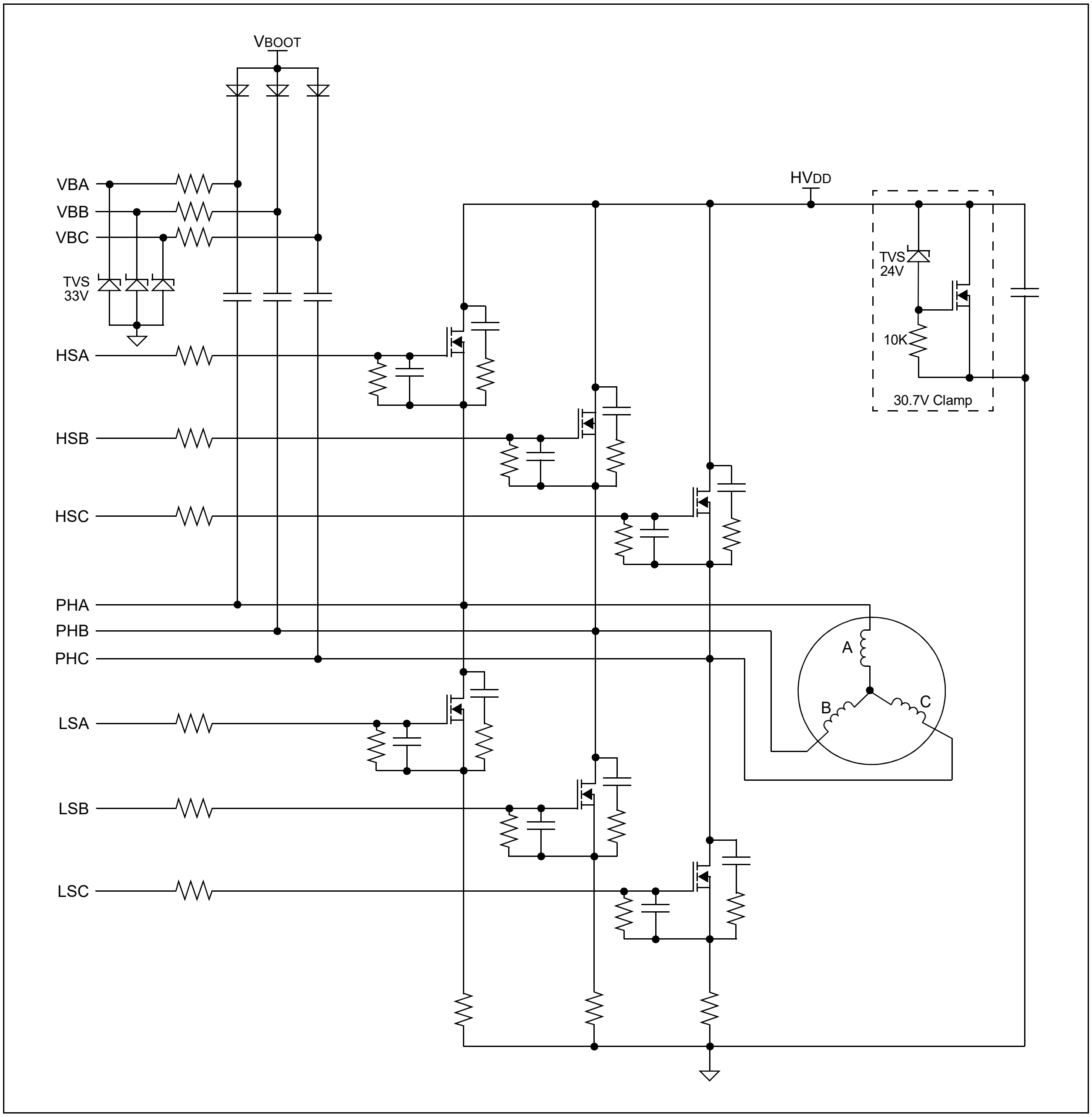
12.26.6 AUTO-BAUD CODE EXAMPLE

Example 12-1 is a dsPIC® DSC code example using the auto-baud function.

EXAMPLE 12-1: dsPIC® DSC AUTO-BAUD EXAMPLE

```
//#define FCY 70000000UL
#define TypBaudrate 9600uL
#define U1BRG_BAUDRATE (FCY/(16 * TypBaudrate)) - 1
#define U1BRG_BAUD_MIN  ((((FCY/(16 * TypBaudrate)) - 1)*1.07f))    // plus 7%
#define U1BRG_BAUD_MAX  ((((FCY/(16 * TypBaudrate)) - 1)*0.93f))    // minus 7%
#define U1BRG_BREAK  (FCY/(16 * 7880uL)) - 1 //7880 baud-rate is midpoint of required break
window for MCP8021

//set up Oscillator_Initialize code here


 void UART1_Init(void){
//UART configuration - set up PPS connections and UART module enable here
 U1MODEbits.UARTEN = 1;           //enable UART
 U1MODEbits.URXEN = 1;            //for half-duplex communication, keep RX on always and
                                   manage TX as needed in auto-baud routine
 __delay_ms(10); //10mS delay required after POR of MCP8021 and before requesting auto-baud

}

void UART1_AutoBaud(void){
   U1MODEbits.UTXEN = 1;           //Transmit enabled, UxTX pin controlled by UARTx.
   U1BRG = U1BRG_BREAK;            //7880baud representing 1.65ms dominant with 13bit BREAK
   U1MODEbits.UTXBRK = 1;          //Send BREAK command
   U1TXREG = 0x00;                 //Dummy write to start BREAK command
   while(U1STAHbits.URXBE == 0){   //wait for transmission to end then read out RXREG to
                                    avoid collision.
       unsigned short dummy = U1RXREG;
   }
   U1MODEbits.UTXEN = 0;           //disable TX while waiting on 0x55 from MCP8021
   U1MODEbits.ABAUD = 1;           //start the ABAUD counter upon receipt of next byte (0x55)
   while(U1MODEbits.ABAUD);        //application should handle timeout if auto-baud does not
                                    complete and attempt auto-baud routine again
   __delay_ms(3);                  //minimum delay of 2mS required after auto-baud complete
                                     to allow for baud rate verification by host
//verify new baud clock is within limits of MCP8021 min and max baud-rate
   if ((U1BRG > U1BRG_BAUD_MAX) && (U1BRG < U1BRG_BAUD_MIN)){
//success, use new baud-rate generator value
   }
   else{
//auto-baud out of range, reload last known good BRG value and attempt auto-baud routine again
   }
}
```

FIGURE 12-10: OVERVOLTAGE PROTECTION



NOTES:

13.0 HIGH-SPEED, 12-BIT ANALOG-TO-DIGITAL CONVERTER (ADC)

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"12-Bit High-Speed, Multiple SARs A/D Converter (ADC)"** (www.microchip.com/DS70005213).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33CDVL64MC106 devices have a high-speed, 12-bit Analog-to-Digital Converter (ADC) that features a low conversion latency, high resolution and oversampling capabilities to improve performance in AC/DC, DC/DC power converters. The devices implement the ADC with one shared SAR core.

13.1 ADC Features Overview

The High-Speed, 12-Bit Multiple SARs Analog-to-Digital Converter (ADC) includes the following features:

- One Shared (common) Core
- User-Configurable Resolution of up to 12 Bits for Each Core
- Up to 3.5 Msps Conversion Rate per Channel at 12-Bit Resolution
- Low-Latency Conversion
- Up to 16 Analog Input Channels with a Separate 16-Bit Conversion Result Register for Each Input – AN1 and AN7 Share the Same Pin
- Conversion Result can be Formatted as Unsigned or Signed Data, on a per Channel Basis, for All Channels
- Channel Scan Capability
- Multiple Conversion Trigger Options for each Core, including:
  - PWM triggers from CPU cores
  - SCCP modules triggers
  - CLC modules triggers
  - External pin trigger event (ADTRG31)
  - Software trigger
- Four Integrated Digital Comparators with Dedicated Interrupts:
  - Multiple comparison options
  - Assignable to specific analog inputs
- Four Oversampling Filters with Dedicated Interrupts:
  - Provide increased resolution
  - Assignable to a specific analog input

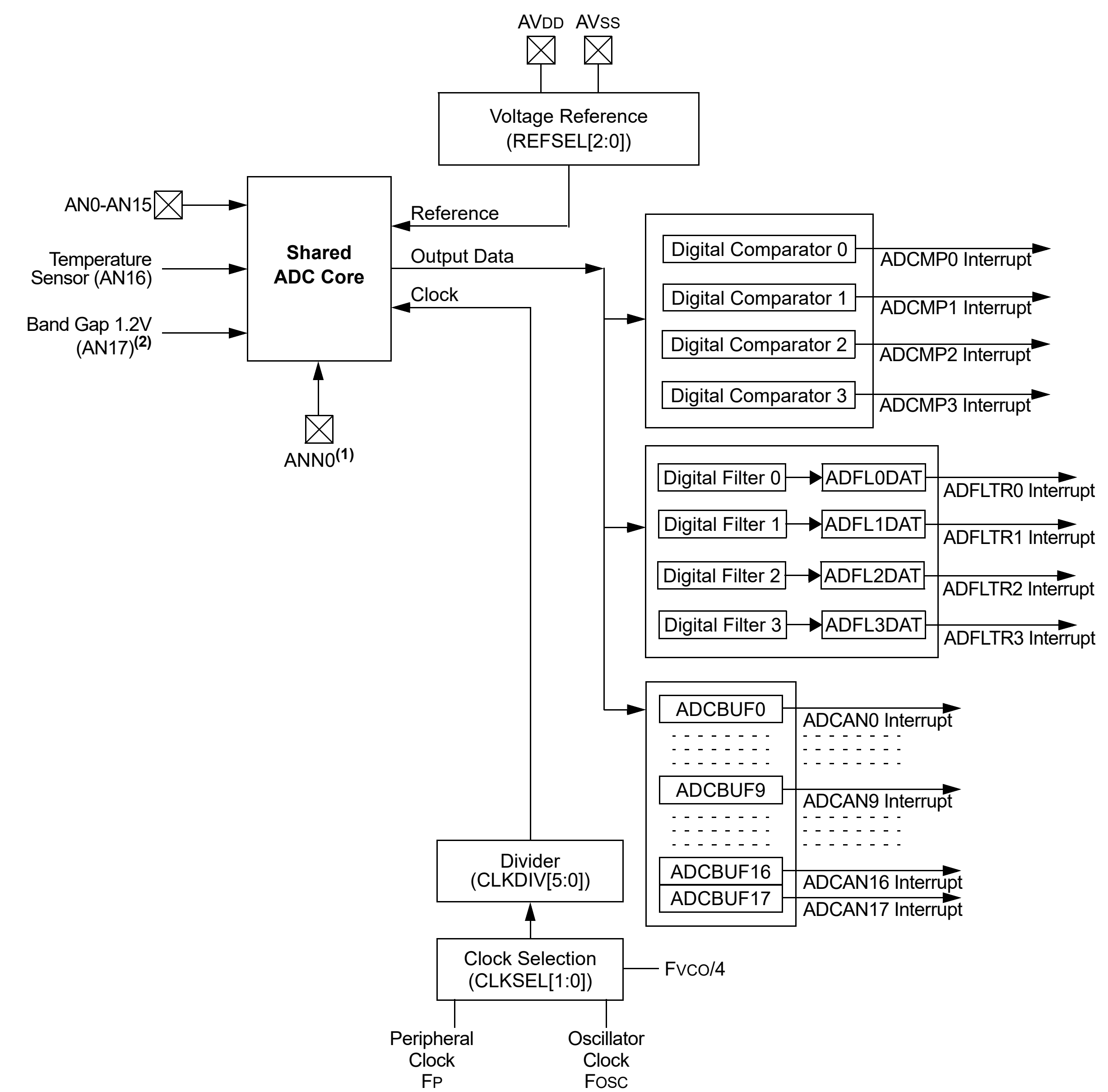
The module consists of one shared SAR ADC core. Simplified block diagrams of the Multiple SARs 12-Bit ADC are shown in Figure 13-1 and Figure 13-2.

The analog inputs (channels) are connected through multiplexers and switches to the Sample-and-Hold (S&H) circuit of each ADC core. The core uses the channel information (the output format, the Measurement mode and the input number) to process the analog sample. When conversion is complete, the result is stored in the result buffer for the specific analog input, and passed to the digital filter and digital comparator if they were configured to use data from this particular channel.

If multiple ADC inputs request conversion on the shared core, the module will convert them in a sequential manner, starting with the lowest order input.

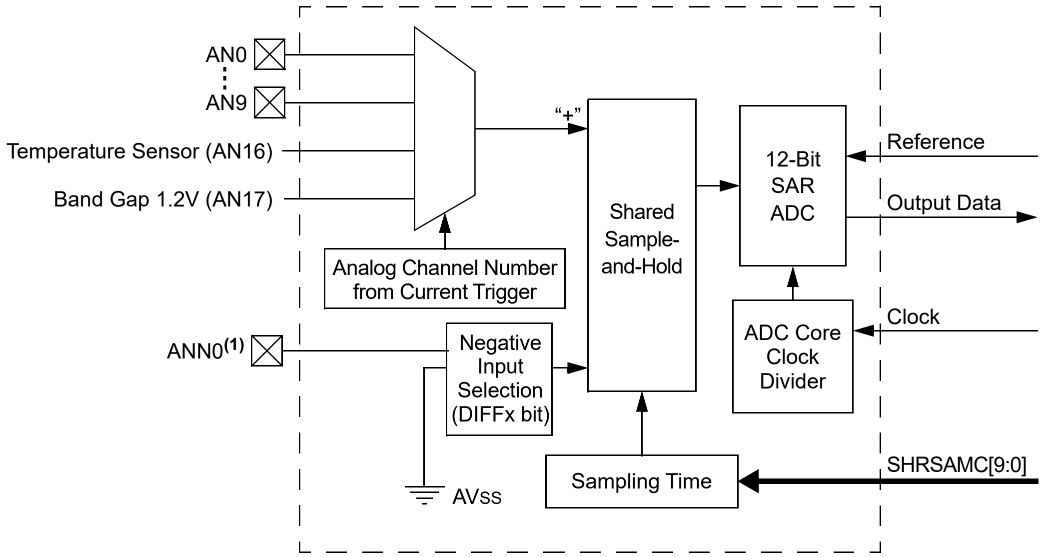
The ADC provides each analog input the ability to specify its own trigger source. This capability allows the ADC to sample and convert analog inputs that are associated with PWM Generators operating on independent time bases.

FIGURE 13-1: ADC MODULE BLOCK DIAGRAM



Note 1: Pin ANN0 is only available in the 48-pin package.
2: Band Gap Reference (VBG) is an internal analog input and is not available on device pins.

FIGURE 13-2: ADC SHARED CORE BLOCK DIAGRAM



Note 1: Pin ANN0 is only available in the 48-pin package.

13.2 Temperature Sensor

The ADC channel, AN16, is connected to a forward-biased diode. It can be used to measure a die temperature. This diode provides a voltage output that can be monitored by the ADC.

The temperature coefficient is listed in Table 33-33 in **Section 33.0 "Electrical Characteristics"**. To get the exact gain and offset numbers, the two temperature points' calibration is recommended.

13.3 Analog-to-Digital Converter Resources

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

13.3.1 KEY RESOURCES

- **"12-Bit High-Speed, Multiple SARs A/D Converter (ADC)"** (www.microchip.com/DS70005213)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

13.4 Differential-Mode

ANNx negative external inputs are used for Differential-mode, as shown in Figure 12-2. To enable Differential-mode, the DIFF bit (in the ADMODxL or ADMODxH register) is set for the corresponding channel.

13.5 ADC Control/Status Registers

REGISTER 13-1: ADCON1L: ADC CONTROL REGISTER 1 LOW

R/W-0	U-0	R/W-0	U-0	r-0	U-0	U-0	U-0
ADON[1]	—	ADSIDL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ADON:** ADC Enable bit[1]
1 = ADC module is enabled
0 = ADC module is off

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ADSIDL:** ADC Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **Unimplemented:** Read as ‘0’

bit 11 **Reserved:** Maintain as ‘0’

bit 10-0 **Unimplemented:** Read as ‘0’

Note 1: Set the ADON bit only after the ADC module has been configured. Changing ADC Configuration bits when ADON = 1 will result in unpredictable behavior.

REGISTER 13-2: ADCON1H: ADC CONTROL REGISTER 1 HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0
FORM	SHRRES1	SHRRES0	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **FORM:** Fractional Data Output Format bit
1 = Fractional
0 = Integer

bit 6-5 **SHRRES[1:0]:** Shared ADC Core Resolution Selection bits
11 = 12-bit resolution
10 = 10-bit resolution
01 = 8-bit resolution
00 = 6-bit resolution

bit 4-0 **Unimplemented:** Read as ‘0’

REGISTER 13-3: ADCON2L: ADC CONTROL REGISTER 2 LOW

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
REFCIE	REFERCIE	—	EIEN	PTGEN[3]	SHREISEL2[1]	SHREISEL1[1]	SHREISEL0[1]
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SHRADCS[6:0][2]						
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **REFCIE:** Band Gap and Reference Voltage Ready Common Interrupt Enable bit
1 = Common interrupt will be generated when the band gap will become ready
0 = Common interrupt is disabled for the band gap ready event

bit 14 **REFERCIE:** Band Gap or Reference Voltage Error Common Interrupt Enable bit
1 = Common interrupt will be generated when a band gap or reference voltage error is detected
0 = Common interrupt is disabled for the band gap and reference voltage error event

bit 13 **Unimplemented:** Read as ‘0’

bit 12 **EIEN:** Early Interrupts Enable bit
1 = The early interrupt feature is enabled for the input channel interrupts (when the EISTATx flag is set)
0 = The individual interrupts are generated when conversion is done (when the ANxRDY flag is set)

bit 11 **PTGEN:** PTG Conversion Request Interface bit[3]
1 = PTG triggers are enabled
0 = PTG triggers are disabled

bit 10-8 **SHREISEL[2:0]:** Shared Core Early Interrupt Time Selection bits[1]
111 = Early interrupt is set and interrupt is generated eight TADCORE clocks prior to when the data are ready
110 = Early interrupt is set and interrupt is generated seven TADCORE clocks prior to when the data are ready
101 = Early interrupt is set and interrupt is generated six TADCORE clocks prior to when the data are ready
100 = Early interrupt is set and interrupt is generated five TADCORE clocks prior to when the data are ready
011 = Early interrupt is set and interrupt is generated four TADCORE clocks prior to when the data are ready
010 = Early interrupt is set and interrupt is generated three TADCORE clocks prior to when the data are ready
001 = Early interrupt is set and interrupt is generated two TADCORE clocks prior to when the data are ready
000 = Early interrupt is set and interrupt is generated one TADCORE clock prior to when the data are ready

bit 7 **Unimplemented:** Read as ‘0’

bit 6-0 **SHRADCS[6:0]:** Shared ADC Core Input Clock Divider bits[2]
These bits determine the number of TCORESRC (Source Clock Periods) for one shared TADCORE (Core Clock Period).
1111111 = 254 Source Clock Periods
. . .
0000011 = 6 Source Clock Periods
0000010 = 4 Source Clock Periods
0000001 = 2 Source Clock Periods
0000000 = 2 Source Clock Periods

Note 1: For the 6-bit shared ADC core resolution (SHRRES[1:0] = 00), the SHREISEL[2:0] settings, from ‘100’ to ‘111’, are not valid and should not be used. For the 8-bit shared ADC core resolution (SHRRES[1:0] = 01), the SHREISEL[2:0] settings, ‘110’ and ‘111’, are not valid and should not be used.
2: The ADC clock frequency, selected by the SHRADCS[6:0] bits, must not exceed 70 MHz.
3: Other ADC trigger sources cannot be used if PTG triggers are enabled.

REGISTER 13-4: ADCON2H: ADC CONTROL REGISTER 2 HIGH

HSC/R-0	HSC/R-0	U-0	r-0	r-0	r-0	R/W-0	R/W-0
REFRDY	REFERR	—	—	—	—	SHRSAMC9	SHRSAMC8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SHRSAMC7	SHRSAMC6	SHRSAMC5	SHRSAMC4	SHRSAMC3	SHRSAMC2	SHRSAMC1	SHRSAMC0
bit 7							bit 0

Legend:	r = Reserved bit	U = Unimplemented bit, read as ‘0’	
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **REFRDY:** Band Gap and Reference Voltage Ready Flag bit
1 = Band gap is ready
0 = Band gap is not ready

bit 14 **REFERR:** Band Gap or Reference Voltage Error Flag bit
1 = Band gap was removed after the ADC module was enabled (ADON = 1)
0 = No band gap error was detected

bit 13 **Unimplemented:** Read as ‘0’

bit 12-10 **Reserved:** Maintain as ‘0’

bit 9-0 **SHRSAMC[9:0]:** Shared ADC Core Sample Time Selection bits
These bits specify the number of shared ADC Core Clock Periods (T_{ADCORE}) for the shared ADC core sample time.
1111111111 = 1025 T_{ADCORE}
. . .
0000000001 = 3 T_{ADCORE}
0000000000 = 2 T_{ADCORE}

REGISTER 13-5: ADCON3L: ADC CONTROL REGISTER 3 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	HSC/R-0	R/W-0	HSC/R-0
REFSEL2	REFSEL1	REFSEL0	SUSPEND	SUSPCIE	SUSPRDY	SHRSAMP	CNVRTCH
bit 15							bit 8

R/W-0	HSC/R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWLCTRG	SWCTRG	CNVCHSEL5	CNVCHSEL4	CNVCHSEL3	CNVCHSEL2	CNVCHSEL1	CNVCHSEL0
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **REFSEL[2:0]:** ADC Reference Voltage Selection bits

Value	VREFH	VREFL
000	AVDD	AVSS

001-111 = **Unimplemented:** Do not use

bit 12 **SUSPEND:** All ADC Core Triggers Disable bit
1 = All new trigger events for all ADC cores are disabled
0 = All ADC cores can be triggered

bit 11 **SUSPCIE:** Suspend All ADC Cores Common Interrupt Enable bit
1 = Common interrupt will be generated when ADC core triggers are suspended (SUSPEND bit = 1) and all previous conversions are finished (SUSPRDY bit becomes set)
0 = Common interrupt is not generated for suspend ADC cores event

bit 10 **SUSPRDY:** All ADC Cores Suspended Flag bit
1 = All ADC cores are suspended (SUSPEND bit = 1) and have no conversions in progress
0 = ADC cores have previous conversions in progress

bit 9 **SHRSAMP:** Shared ADC Core Sampling Direct Control bit
This bit should be used with the individual channel conversion trigger controlled by the CNVRTCH bit. It connects an analog input, specified by the CNVCHSEL[5:0] bits, to the shared ADC core and allows extending the sampling time. This bit is not controlled by hardware and must be cleared before the conversion starts (setting CNVRTCH to '1').
1 = Shared ADC core samples an analog input specified by the CNVCHSEL[5:0] bits
0 = Sampling is controlled by the shared ADC core hardware

bit 8 **CNVRTCH:** Software Individual Channel Conversion Trigger bit
1 = Single trigger is generated for an analog input specified by the CNVCHSEL[5:0] bits; when the bit is set, it is automatically cleared by hardware on the next instruction cycle
0 = Next individual channel conversion trigger can be generated

bit 7 **SWLCTRG:** Software Level-Sensitive Common Trigger bit
1 = Triggers are continuously generated for all channels with the software, level-sensitive common trigger selected as a source in the ADTRIGnL and ADTRIGnH registers
0 = No software, level-sensitive common triggers are generated

bit 6 **SWCTRG:** Software Common Trigger bit
1 = Single trigger is generated for all channels with the software; common trigger selected as a source in the ADTRIGnL and ADTRIGnH registers; when the bit is set, it is automatically cleared by hardware on the next instruction cycle
0 = Ready to generate the next software common trigger

bit 5-0 **CNVCHSEL [5:0]:** Channel Number Selection for Software Individual Channel Conversion Trigger bits
These bits define a channel to be converted when the CNVRTCH bit is set.

REGISTER 13-6: ADCON3H: ADC CONTROL REGISTER 3 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLKSEL1[1]	CLKSEL0[1]	CLKDIV5[2]	CLKDIV4[2]	CLKDIV3[2]	CLKDIV2[2]	CLKDIV1[2]	CLKDIV0[2]
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHREN	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **CLKSEL[1:0]:** ADC Module Clock Source Selection bits[1]
11 = FVCO/4
10 = FVCO/3
01 = FOSC
00 = FP (Peripheral Clock)

bit 13-8 **CLKDIV[5:0]:** ADC Module Clock Source Divider bits[2]
The divider forms a TCORESRC clock used by the ADC core from the TSRC ADC module clock source selected by the CLKSEL[1:0] bits. Then, each ADC core individually divides the TCORESRC clock to get a core-specific TADCORE clock using the ADCS[6:0] bits in the ADCORExH register or the SHRADCS[6:0] bits in the ADCON2L register.
111111 = 64 Source Clock Periods
...
000011 = 4 Source Clock Periods
000010 = 3 Source Clock Periods
000001 = 2 Source Clock Periods
000000 = 1 Source Clock Period

bit 7 **SHREN:** Shared ADC Core Enable bit
1 = Shared ADC core is enabled
0 = Shared ADC core is disabled

bit 6-0 **Unimplemented:** Read as ‘0’

Note 1: The ADC input clock frequency, selected by the CLKSEL[1:0] bits, must not exceed Parameters AD9, AD10 and AD11 listed in Table 33-30.

2: The ADC clock frequency, after the first divider selected by the CLKDIV[5:0] bits, must not exceed Parameters AD9, AD10 and AD11 listed in Table 33-30.

REGISTER 13-7: ADCON5L: ADC CONTROL REGISTER 5 LOW

HSC/R-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRRDY	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRPWR	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	U = Unimplemented bit, read as ‘0’		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **SHRRDY:** Shared ADC Core Ready Flag bit
1 = ADC core is powered and ready for operation
0 = ADC core is not ready for operation

bit 14-8 **Unimplemented:** Read as ‘0’

bit 7 **SHRPWR:** Shared ADC Core Power Enable bit
1 = ADC core is powered
0 = ADC core is off

bit 6-0 **Unimplemented:** Read as ‘0’

REGISTER 13-8: ADCON5H: ADC CONTROL REGISTER 5 HIGH

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WARMTIME[3:0]			
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRCIE	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-8 **WARMTIME[3:0]:** ADC Core x Power-up Delay bits

These bits determine the power-up delay in the number of the Core Source Clock Periods ($T_{CORESRC}$) for all ADC cores.

1111 = 32768 Source Clock Periods
1110 = 16384 Source Clock Periods
1101 = 8192 Source Clock Periods
1100 = 4096 Source Clock Periods
1011 = 2048 Source Clock Periods
1010 = 1024 Source Clock Periods
1001 = 512 Source Clock Periods
1000 = 256 Source Clock Periods
0111 = 128 Source Clock Periods
0110 = 64 Source Clock Periods
0101 = 32 Source Clock Periods
0100 = 16 Source Clock Periods
00xx = 16 Source Clock Periods

bit 7 **SHRCIE:** Shared ADC Core Ready Common Interrupt Enable bit

1 = Common interrupt will be generated when ADC core is powered and ready for operation
0 = Common interrupt is disabled for an ADC core ready event

bit 6-0 **Unimplemented:** Read as ‘0’

REGISTER 13-9: ADLVLTRGL: ADC LEVEL-SENSITIVE TRIGGER CONTROL REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LVLEN[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LVLEN[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **LVLEN[15:0]:** Level Trigger for Corresponding Analog Input Enable bits
1 = Input trigger is level-sensitive
0 = Input trigger is edge-sensitive

REGISTER 13-10: ADLVLTRGH: ADC LEVEL-SENSITIVE TRIGGER CONTROL REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	LVLEN[17:16]	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **LVLEN[17:16]:** Level Trigger for Corresponding Analog Input Enable bits
1 = Input trigger is level-sensitive
0 = Input trigger is edge-sensitive

REGISTER 13-11: ADEIEL: ADC EARLY INTERRUPT ENABLE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EIEN[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EIEN[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **EIEN[15:0]:** Early Interrupt Enable for Corresponding Analog Inputs bits
1 = Early interrupt is enabled for the channel
0 = Early interrupt is disabled for the channel

REGISTER 13-12: ADEIEH: ADC EARLY INTERRUPT ENABLE REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	EIEN[17:16]	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **EIEN[17:16]:** Early Interrupt Enable for Corresponding Analog Inputs bits
1 = Early interrupt is enabled for the channel
0 = Early interrupt is disabled for the channel

REGISTER 13-13: ADEISTATL: ADC EARLY INTERRUPT STATUS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EISTAT[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EISTAT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **EISTAT[15:0]:** Early Interrupt Status for Corresponding Analog Inputs bits
1 = Early interrupt was generated
0 = Early interrupt was not generated since the last ADCBUFx read

REGISTER 13-14: ADEISTATH: ADC EARLY INTERRUPT STATUS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	EISTAT[17:16]	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **EISTAT[17:16]:** Early Interrupt Status for Corresponding Analog Inputs bits
1 = Early interrupt was generated
0 = Early interrupt was not generated since the last ADCBUFx read

REGISTER 13-15: ADMOD0L: ADC INPUT MODE CONTROL REGISTER 0 LOW[(1)]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 through bit 1 (odd) **DIFF[7:0]:** Differential-Mode for Corresponding Analog Inputs bits
1 = Channel is differential
0 = Channel is single-ended

bit 14 through bit 0 (even) **SIGN[7:0]:** Output Data Sign for Corresponding Analog Inputs bits
1 = Channel output data are signed
0 = Channel output data are unsigned

Note 1: The DIFF bits are available only on devices in the 48-pin package; they are used to enable the differential input feature which is linked to the presence of the pin named ANN0. This pin is only available in 48-pin packages.

REGISTER 13-16: ADMOD0H: ADC INPUT MODE CONTROL REGISTER 0 HIGH[(1)]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIFF15	SIGN15	DIFF14	SIGN14	DIFF13	SIGN13	DIFF12	SIGN12
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 through bit 1 (odd) **DIFF[15:8]:** Differential-Mode for Corresponding Analog Inputs bits
1 = Channel is differential
0 = Channel is single-ended

bit 14 through bit 0 (even) **SIGN[15:8]:** Output Data Sign for Corresponding Analog Inputs bits
1 = Channel output data are signed
0 = Channel output data are unsigned

Note 1: The DIFF bits are available only on devices in the 48-pin package; they are used to enable the differential input feature which is linked to the presence of the pin named ANN0. This pin is only available in 48-pin packages.

REGISTER 13-17: ADMOD1L: ADC INPUT MODE CONTROL REGISTER 1 LOW

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DIFF17	SIGN17	DIFF16	SIGN16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4	**Unimplemented:** Read as ‘0’
bit 3 through bit 1 (odd)	**DIFF[17:16]:** Differential-Mode for Corresponding Analog Inputs bits 1 = Channel is differential 0 = Channel is single-ended
bit 2 through bit 0 (even)	**SIGN[17:16]:** Output Data Sign for Corresponding Analog Inputs bits 1 = Channel output data are signed 0 = Channel output data are unsigned

REGISTER 13-18: ADIEL: ADC INTERRUPT ENABLE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IE[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IE[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **IE[15:0]:** Common Interrupt Enable bits
1 = Common and individual interrupts are enabled for the corresponding channel
0 = Common and individual interrupts are disabled for the corresponding channel

REGISTER 13-19: ADIEH: ADC INTERRUPT ENABLE REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	IE[17:16]	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **IE[17:16]:** Common Interrupt Enable bits
1 = Common and individual interrupts are enabled for the corresponding channel
0 = Common and individual interrupts are disabled for the corresponding channel

REGISTER 13-20: ADSTATL: ADC DATA READY STATUS REGISTER LOW

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN[15:8]RDY							
bit 15							bit 8

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN[7:0]RDY							
bit 7							bit 0

Legend:	U = Unimplemented bit, read as ‘0’		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **AN[15:0]RDY:** Data Ready Status for Corresponding Analog Inputs bits
1 = Channel conversion result is ready in the corresponding ADCBUFx register
0 = Channel conversion result is not ready

REGISTER 13-21: ADSTATH: ADC DATA READY STATUS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	HSC/R-0	HSC/R-0
—	—	—	—	—	—	AN[17:16]RDY	
bit 7							bit 0

Legend:	U = Unimplemented bit, read as ‘0’		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **AN[17:16]RDY:** Data Ready Status for Corresponding Analog Inputs bits
1 = Channel conversion result is ready in the corresponding ADCBUFx register
0 = Channel conversion result is not ready

REGISTER 13-22: ADTRIGNL/ADTRIGNH: ADC CHANNEL TRIGGER N(X) SELECTION REGISTERS LOW AND HIGH (X = 0 TO 17; N = 0 TO 4)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TRGSRC(x+1)4	TRGSRC(x+1)3	TRGSRC(x+1)2	TRGSRC(x+1)1	TRGSRC(x+1)0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TRGSRCx4	TRGSRCx3	TRGSRCx2	TRGSRCx1	TRGSRCx0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **TRGSRC(x+1)[4:0]:** Trigger Source Selection for Corresponding Analog Inputs bits (TRGSRC1 to TRGSRC19 – Odd)

11111 = ADTRG31 (PPS input)
11110 = PTG12
11101 = CLC2
11100 = CLC1
11011 = Reserved
11010 = Reserved
11001 = Reserved
11000 = Reserved
10111 = SCCP4 input capture/output compare
10110 = SCCP3 input capture/output compare
10101 = SCCP2 input capture/output compare
10100 = SCCP1 input capture/output compare
10011 = Reserved
10010 = CLC4 output
10001 = CLC3 output
10000 = Reserved
01111 = SCCP4 trigger
01110 = SCCP3 trigger
01101 = SCCP2 trigger
01100 = SCCP1 trigger
01011 = PWM4 Trigger 2
01010 = PWM4 Trigger 1
01001 = PWM3 Trigger 2
01000 = PWM3 Trigger 1
00111 = PWM2 Trigger 2
00110 = PWM2 Trigger 1
00101 = PWM1 Trigger 2
00100 = PWM1 Trigger 1
00011 = Reserved
00010 = Level software trigger
00001 = Common software trigger
00000 = No trigger is enabled

bit 7-5 **Unimplemented:** Read as ‘0’

REGISTER 13-22: ADTRIGNL/ADTRIGNH: ADC CHANNEL TRIGGER N(X) SELECTION REGISTERS LOW AND HIGH (X = 0 TO 17; N = 0 TO 4) (CONTINUED)

bit 4-0 **TRGSRCx[4:0]:** Common Interrupt Enable for Corresponding Analog Inputs bits (TRGSRC0 to TRGSRC20 – Even)

11111 = ADTRG31 (PPS input)
11110 = PTG12
11101 = CLC2
11100 = CLC1
11011 = Reserved
11010 = Reserved
11001 = Reserved
11000 = Reserved
10111 = SCCP4 CCP interrupt
10110 = SCCP3 CCP interrupt
10101 = SCCP2 CCP interrupt
10100 = SCCP1 CCP interrupt
10011 = Reserved
10010 = CLC4 output
10001 = CLC3 output
10000 = Reserved
01111 = SCCP4 trigger
01110 = SCCP3 trigger
01101 = SCCP2 trigger
01100 = SCCP1 trigger
01011 = PWM4 Trigger 2
01010 = PWM4 Trigger 1
01001 = PWM3 Trigger 2
01000 = PWM3 Trigger 1
00111 = PWM2 Trigger 2
00110 = PWM2 Trigger 1
00101 = PWM1 Trigger 2
00100 = PWM1 Trigger 1
00011 = Reserved
00010 = Level software trigger
00001 = Common software trigger
00000 = No trigger is enabled

REGISTER 13-23: ADCMPxCON: ADC DIGITAL COMPARATOR x CONTROL REGISTER (x = 0, 1, 2, 3)

U-0	U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	—	CHNL[4:0]				
bit 15							bit 8

R/W-0	R/W-0	HC/HS/R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPEN	IE	STAT	BTWN	HIHI	HILO	LOHI	LOLO
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	U = Unimplemented bit, read as ‘0’	
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	HS = Hardware Settable bit

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **CHNL[4:0]:** Input Channel Number bits
If the comparator has detected an event for a channel, this channel number is written to these bits.
11111 = Reserved
...
10010 = Reserved
10001 = Band gap, 1.2V (AN17)
10000 = Temperature sensor (AN16)
...
01110 = Reserved
...
01010 = Reserved
01001 = AN9
...
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 7 **CMPEN:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and the STAT status bit is cleared

bit 6 **IE:** Comparator Common ADC Interrupt Enable bit
1 = Common ADC interrupt will be generated if the comparator detects a comparison event
0 = Common ADC interrupt will not be generated for the comparator

bit 5 **STAT:** Comparator Event Status bit
This bit is cleared by hardware when the channel number is read from the CHNL[4:0] bits.
1 = A comparison event has been detected since the last read of the CHNL[4:0] bits
0 = A comparison event has not been detected since the last read of the CHNL[4:0] bits

bit 4 **BTWN:** Between Low/High Comparator Event bit
1 = Generates a comparator event when ADCMPxLO ≤ ADCBUFx < ADCMPxHI
0 = Does not generate a digital comparator event when ADCMPxLO ≤ ADCBUFx < ADCMPxHI

bit 3 **HIHI:** High/High Comparator Event bit
1 = Generates a digital comparator event when ADCBUFx ≥ ADCMPxHI
0 = Does not generate a digital comparator event when ADCBUFx ≥ ADCMPxHI

bit 2 **HILO:** High/Low Comparator Event bit
1 = Generates a digital comparator event when ADCBUFx < ADCMPxHI
0 = Does not generate a digital comparator event when ADCBUFx < ADCMPxHI

bit 1 **LOHI:** Low/High Comparator Event bit
1 = Generates a digital comparator event when ADCBUFx ≥ ADCMPxLO
0 = Does not generate a digital comparator event when ADCBUFx ≥ ADCMPxLO

REGISTER 13-23: ADCMPxCON: ADC DIGITAL COMPARATOR x CONTROL REGISTER (x = 0, 1, 2, 3)

bit 0	**LOLO:** Low/Low Comparator Event bit 1 = Generates a digital comparator event when ADCBUFx < ADCMPxLO 0 = Does not generate a digital comparator event when ADCBUFx < ADCMPxLO

REGISTER 13-24: ADCMPxENL: ADC DIGITAL COMPARATOR x CHANNEL ENABLE REGISTER LOW (x = 0 or 3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPEN[15:8]							
bit 15							bit 8

R/W/0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPEN[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **CMPEN[15:0]:** Comparator Enable for Corresponding Input Channels bits
1 = Conversion result for corresponding channel is used by the comparator
0 = Conversion result for corresponding channel is not used by the comparator

REGISTER 13-25: ADCMPxENH: ADC DIGITAL COMPARATOR x CHANNEL ENABLE REGISTER HIGH (x = 0 or 3)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CMPEN[17:16]	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as ‘0’

bit 1-0 **CMPEN[17:16]:** Comparator Enable for Corresponding Input Channels bits
1 = Conversion result for corresponding channel is used by the comparator
0 = Conversion result for corresponding channel is not used by the comparator

REGISTER 13-26: ADFLxCON: ADC DIGITAL FILTER x CONTROL REGISTER (x = 0 or 3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	HSC/R-0
FLEN	MODE1	MODE0	OVRSAM2	OVRSAM1	OVRSAM0	IE	RDY
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	FLCHSEL[4:0]				
bit 7							bit 0

Legend:	U = Unimplemented bit, read as ‘0’		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **FLEN:** Filter Enable bit
1 = Filter is enabled
0 = Filter is disabled and the RDY bit is cleared

bit 14-13 **MODE[1:0]:** Filter Mode bits
11 = Averaging mode
10 = Reserved
01 = Reserved
00 = Oversampling mode

bit 12-10 **OVRSAM[2:0]:** Filter Averaging/Oversampling Ratio bits
<u>If MODE[1:0] = 00:</u>
111 = 128x (16-bit result in the ADFLxDAT register is in 12.4 format)
110 = 32x (15-bit result in the ADFLxDAT register is in 12.3 format)
101 = 8x (14-bit result in the ADFLxDAT register is in 12.2 format)
100 = 2x (13-bit result in the ADFLxDAT register is in 12.1 format)
011 = 256x (16-bit result in the ADFLxDAT register is in 12.4 format)
010 = 64x (15-bit result in the ADFLxDAT register is in 12.3 format)
001 = 16x (14-bit result in the ADFLxDAT register is in 12.2 format)
000 = 4x (13-bit result in the ADFLxDAT register is in 12.1 format)
<u>If MODE[1:0] = 11 (12-bit result in the ADFLxDAT register in all instances):</u>
111 = 256x
110 = 128x
101 = 64x
100 = 32x
011 = 16x
110 = 8x
001 = 4x
000 = 2x

bit 9 **IE:** Filter Interrupts Enable bit
1 = Individual and common interrupts will be generated when the filter result is ready
0 = Individual and common interrupts will not be generated for the filter

bit 8 **RDY:** Oversampling Filter Data Ready Flag bit
This bit is cleared by hardware when the result is read from the ADFLxDAT register.
1 = Data in the ADFLxDAT register are ready
0 = The ADFLxDAT register has been read and new data in the ADFLxDAT register are not ready

bit 7-5 **Unimplemented:** Read as ‘0’

REGISTER 13-26: ADFLxCON: ADC DIGITAL FILTER x CONTROL REGISTER (x = 0 or 3) (CONTINUED)

bit 4-0 **FLCHSEL[4:0]:** Oversampling Filter Input Channel Selection bits
11111 = Reserved
. . .
10010 = Reserved
10001 = Band gap, 1.2V (AN17)
10000 = Temperature sensor (AN16)
01111 = AN15
. . .
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

14.0 HIGH-SPEED ANALOG COMPARATOR WITH SLOPE COMPENSATION DAC

Note 1:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"High-Speed Analog Comparator Module"** (www.microchip.com/DS70005280).

The high-speed analog comparator module provides a method to monitor voltage, current and other critical signals in a power conversion application that may be too fast for the CPU and ADC to capture. There are a total of three comparator modules. The analog comparator module can be used to implement Peak Current mode control, Critical Conduction mode (variable frequency) and Hysteretic Control mode.

14.1 Overview

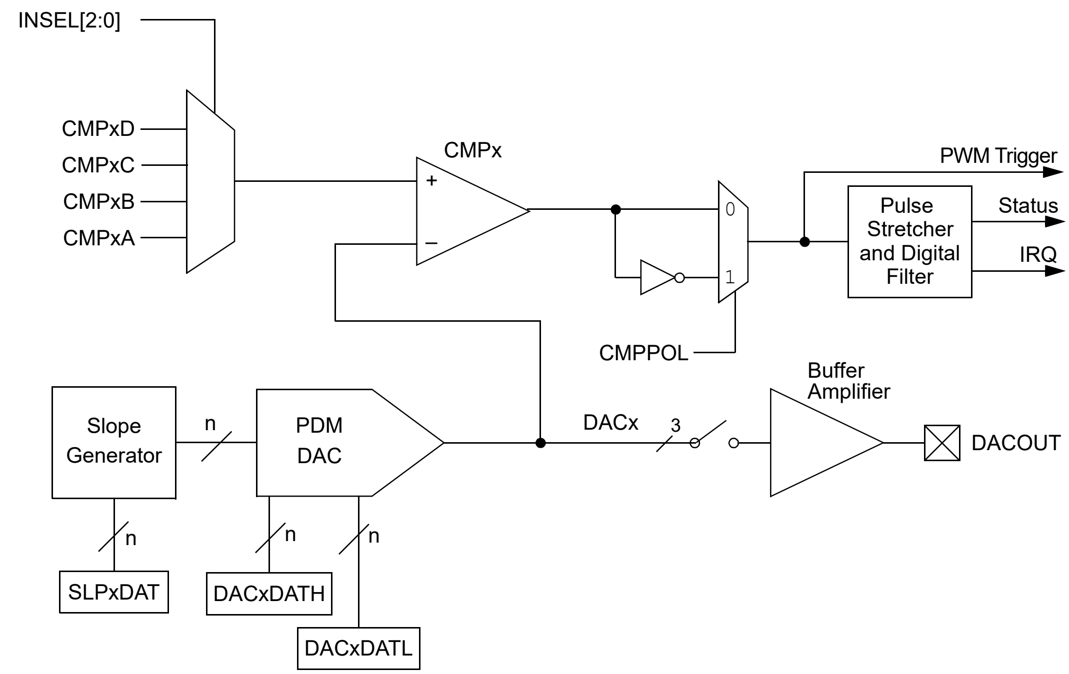
The high-speed analog comparator module is comprised of a high-speed comparator, Pulse Density Modulation (PDM) DAC and a slope compensation unit. The slope compensation unit provides a user-defined slope which can be used to alter the DAC output. This feature is useful in applications, such as Peak Current mode control, where slope compensation is required to maintain the stability of the power supply. The user simply specifies the direction and rate of change for the slope compensation and the output of the DAC is modified accordingly.

The DAC consists of a PDM unit, followed by a digitally controlled multiphase RC filter. The PDM unit uses a phase accumulator circuit to generate an output stream of pulses. The density of the pulse stream is proportional to the input data value, relative to the maximum value supported by the bit width of the accumulator. The output pulse density is representative of the desired output voltage. The pulse stream is filtered with an RC filter to yield an analog voltage. The output of the DAC is connected to the negative input of the comparator. The positive input of the comparator can be selected using a MUX from either of the input pins. The comparator provides a high-speed operation with a typical delay of 15 ns.

The output of the comparator is processed by the pulse stretcher and the digital filter blocks, which prevent comparator response to unintended fast transients in the inputs. Figure 14-1 shows a block diagram of the high-speed analog comparator module. The DAC module can be operated in one of three modes: Slope Generation mode, Hysteretic mode and Triangle Wave mode. Each of these modes can be used in a variety of power supply applications.

Note:	The DACOUT pin can only be associated with a single DAC output at any given time. If more than one DACOEN bit is set, the DACOUT pin will be a combination of the signals.

FIGURE 14-1: HIGH-SPEED ANALOG COMPARATOR MODULE BLOCK DIAGRAM



Note: n = 16

14.2 Features Overview

- Three Rail-to-Rail Analog Comparators
- Up to Four Selectable Input Sources per Comparator
- Programmable Comparator Hysteresis
- Programmable Output Polarity
- Interrupt Generation Capability
- Dedicated Pulse Density Modulation DAC for Each Analog Comparator:
  - PDM unit followed by a digitally controlled multimode multipole RC filter
- Multimode Multipole RC Output Filter:
  - Transition mode: Provides the fastest response
  - Fast mode: For tracking DAC slopes
  - Steady-State mode: Provides 12-bit resolution
- Slope Compensation along with Each DAC:
  - Slope Generation mode
  - Hysteretic Control mode
  - Triangle Wave mode
- Functional Support for the High-Speed PWM Module which Includes:
  - PWM duty cycle control
  - PWM period control
  - PWM Fault detect

14.3 Control Registers

The DACCTRL1L and DACCTRL2H/L registers are common configuration registers for DAC modules.

The DACxCON, DACxDAT, SLPxCON and SLPxDAT registers specify the operation of individual modules.

REGISTER 14-1: DACCTRL1L: DAC CONTROL 1 LOW REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
DACON	—	DACSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CLKSEL1(1,3)	CLKSEL0(1,3)	CLKDIV1(1,3)	CLKDIV0(1,3)	—	FCLKDIV2(2)	FCLKDIV1(2)	FCLKDIV0(2)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared

bit 15 **DACON:** Common DAC Module Enable bit
1 = Enables DAC modules
0 = Disables DAC modules and disables FSCM clocks to reduce power consumption; any pending Slope mode and/or underflow conditions are cleared

bit 14 **Unimplemented:** Read as '0'

bit 13 **DACSIDL:** DAC Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **Unimplemented:** Read as '0'

bit 7-6 **CLKSEL[1:0]:** DAC Clock Source Select bits(1,3)
11 = FPLLO
10 = FVCO/3
01 = FVCO/2
00 = FVCODIV

bit 5-4 **CLKDIV[1:0]:** DAC Clock Divider bits(1,3)
11 = Divide-by-4
10 = Divide-by-3 (non-uniform duty cycle)
01 = Divide-by-2
00 = 1x

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **FCLKDIV[2:0]:** Comparator Filter Clock Divider bits(2)
111 = Divide-by-8
110 = Divide-by-7
101 = Divide-by-6
100 = Divide-by-5
011 = Divide-by-4
010 = Divide-by-3
001 = Divide-by-2
000 = 1x

Note 1: These bits should only be changed when DACON = 0 to avoid unpredictable behavior.
2: The input clock to this divider is the selected clock input, CLKSEL[1:0], and then divided by 2.
3: Clock source and dividers should yield an effective DAC clock input as specified in Table 33-33.

REGISTER 14-2: DACCTRL2H: DAC CONTROL 2 HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SSTIME[9:8][1]	
bit 15							bit 8

R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
SSTIME[7:0][1]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared

bit 15-10 **Unimplemented:** Read as '0'

bit 9-0 **SSTIME[9:0]:** Time from Start of Transition Mode until Steady-State Filter is Enabled bits[1]

Note 1: The value for SSTIME[9:0] should be greater than the TMODTIME[9:0] value.

REGISTER 14-3: DACCTRL2L: DAC CONTROL 2 LOW REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	TMODTIME[9:8][1]	
bit 15							bit 8

R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1
TMODTIME[7:0][1]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared

bit 15-10 **Unimplemented:** Read as '0'

bit 9-0 **TMODTIME[9:0]:** Transition Mode Duration bits[1]

Note 1: The value for TMODTIME[9:0] should be less than the SSTIME[9:0] value.

REGISTER 14-4: DACxCONH: DACx CONTROL HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	TMCB[9:8]	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMCB[7:0]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **TMCB[9:0]:** DACx Leading-Edge Blanking bits

These register bits specify the blanking period for the comparator, following changes to the DAC output during Change-of-State (COS), for the input signal selected by the HCFSEL[3:0] bits in Register 14-9.

REGISTER 14-5: DACxCONL: DACx CONTROL LOW REGISTER

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DACEN	IRQM1[1,2]	IRQM0[1,2]	—	—	CBE	DACOEN	FLTREN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPSTAT	CMPPOL	INSEL2	INSEL1	INSEL0	HYSPOL	HYSSEL1	HYSSEL0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15 **DACEN:** Individual DACx Module Enable bit

1 = Enables DACx module

0 = Disables DACx module to reduce power consumption; any pending Slope mode and/or underflow conditions are cleared

bit 14-13 **IRQM[1:0]:** Interrupt Mode select bits[1,2]

11 = Generates an interrupt on either a rising or falling edge detect

10 = Generates an interrupt on a falling edge detect

01 = Generates an interrupt on a rising edge detect

00 = Interrupts are disabled

bit 12-11 **Unimplemented:** Read as ‘0’

Note 1: Changing these bits during operation may generate a spurious interrupt.

2: The edge selection is a post-polarity selection via the CMPPOL bit.

REGISTER 14-5: DACxCONL: DACx CONTROL LOW REGISTER (CONTINUED)

bit 10 **CBE:** Comparator Blank Enable bit
- 1 = Enables the analog comparator output to be blanked (gated off) during the recovery transition following the completion of a slope operation
- 0 = Disables the blanking signal to the analog comparator; therefore, the analog comparator output is always active

bit 9 **DACOEN:** DACx Output Buffer Enable bit
- 1 = DACx analog voltage is connected to the DACOUT pin
- 0 = DACx analog voltage is not connected to the DACOUT pin

bit 8 **FLTREN:** Comparator Digital Filter Enable bit
- 1 = Digital filter is enabled
- 0 = Digital filter is disabled

bit 7 **CMPSTAT:** Comparator Status bits

The current state of the comparator output including the CMPPOL selection.

bit 6 **CMPPOL:** Comparator Output Polarity Control bit
- 1 = Output is inverted
- 0 = Output is noninverted

bit 5-3 **INSEL[2:0]:** Comparator Input Source Select bits
- 111 = Reserved
- 110 = Reserved
- 101 = Reserved
- 100 = Reserved
- 011 = CMPxD input pin
- 010 = CMPxC input pin
- 001 = CMPxB input pin
- 000 = CMPxA input pin

bit 2 **HYSPOL:** Comparator Hysteresis Polarity Select bit
- 1 = Hysteresis is applied to the falling edge of the comparator output
- 0 = Hysteresis is applied to the rising edge of the comparator output

bit 1-0 **HYSSEL[1:0]:** Comparator Hysteresis Select bits
- 11 = 45 mv hysteresis
- 10 = 30 mv hysteresis
- 01 = 15 mv hysteresis
- 00 = No hysteresis is selected

Note 1: Changing these bits during operation may generate a spurious interrupt.

2: The edge selection is a post-polarity selection via the CMPPOL bit.

REGISTER 14-6: DACxDATH: DACx DATA HIGH REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DACDATH[11:8]			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACDATH[7:0]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-0 **DACDATH[11:0]:** DACx Data bits
This register specifies the high DACx data value. Valid values are from 205 to 3890.

REGISTER 14-7: DACxDATL: DACx DATA LOW REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DACDATL[11:8]			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACDATL[7:0]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-0 **DACDATL[11:0]:** DACx Low Data bits
In Hysteretic mode, Slope Generator mode and Triangle mode, this register specifies the low data value for the DACx module. Valid values are from 205 to 3890.

REGISTER 14-8: SLPxCONH: DACx SLOPE CONTROL HIGH REGISTER

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
SLOPEN	—	—	—	HME(1)	TWME(2)	PSE	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15 **SLOPEN:** Slope Function Enable/On bit
1 = Enables slope function
0 = Disables slope function; slope accumulator is disabled to reduce power consumption

bit 14-12 **Unimplemented:** Read as ‘0’

bit 11 **HME:** Hysteretic Mode Enable bit(1)
1 = Enables Hysteretic mode for DACx
0 = Disables Hysteretic mode for DACx

bit 10 **TWME:** Triangle Wave Mode Enable bit(2)
1 = Enables Triangle Wave mode for DACx
0 = Disables Triangle Wave mode for DACx

bit 9 **PSE:** Positive Slope Mode Enable bit
1 = Slope mode is positive (increasing)
0 = Slope mode is negative (decreasing)

bit 8-0 **Unimplemented:** Read as ‘0’

Note 1: HME mode requires the user to disable the slope function (SLOPEN = 0).
2: TWME mode requires the user to enable the slope function (SLOPEN = 1).

REGISTER 14-9: SLPxCONL: DACx SLOPE CONTROL LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HCFSEL3	HCFSEL2	HCFSEL1	HCFSEL0	SLPSTOPA3	SLPSTOPA2	SLPSTOPA1	SLPSTOPA0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPSTOPB3	SLPSTOPB2	SLPSTOPB1	SLPSTOPB0	SLPSTRT3	SLPSTRT2	SLPSTRT1	SLPSTRT0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set0	‘0’ = Bit is cleared

bit 15-12 **HCFSEL[3:0]:** Hysteretic Comparator Function Input Select bits

The selected input signal controls the switching between the DACx high limit (DACxDATH) and the DACx low limit (DACxDATL) as the data source for the PDM DAC. It modifies the polarity of the comparator, and the rising and falling edges initiate the start of the LEB counter (TMCB[9:0] bits in Register 14-4).

Input Selection	Source
0101-1111	1
0100	PWM4H
0011	PWM3H
0010	PWM2H
0001	PWM1H
0000	0

bit 11-8 **SLPSTOPA[3:0]:** Slope Stop A Signal Select bits

The selected Slope Stop A signal is logically OR’d with the selected Slope Stop B signal to terminate the slope function.

Slope Stop A Signal Selection	Source
0101-1111	1
0100	PWM4 Trigger 2
0011	PWM3 Trigger 2
0010	PWM2 Trigger 2
0001	PWM1 Trigger 2
0000	0

bit 7-4 **SLPSTOPB[3:0]:** Slope Stop B Signal Select bits

The selected Slope Stop B signal is logically OR’d with the selected Slope Stop A signal to terminate the slope function.

Slope Stop B Signal Selection	Source
0010-1111	1
0001	CMP1 Out
0000	0

REGISTER 14-9: SLPxCONL: DACx SLOPE CONTROL LOW REGISTER (CONTINUED)

bit 3-0 **SLPSTRT[3:0]:** Slope Start Signal Select bits

Slope Start Signal Selection	Source
0101-1111	1
0100	PWM4 Trigger 1
0011	PWM3 Trigger 1
0010	PWM2 Trigger 1
0001	PWM1 Trigger 1
0000	0

REGISTER 14-10: SLPxDAT: DACx SLOPE DATA REGISTER[(1)]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPDAT[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPDAT[7:0]							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared

bit 15-0 **SLPDAT[15:0]:** Slope Ramp Rate Value bits
The SLPDATx value is in 12.4 format.

Note 1: Register data are left justified.

NOTES:

15.0 QUADRATURE ENCODER INTERFACE (QEI)

Note 1:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive resource. For more information, refer to **"Quadrature Encoder Interface (QEI)"** (www.microchip.com/DS70000601).

The Quadrature Encoder Interface (QEI) module provides the interface to incremental encoders for obtaining mechanical position data. The dsPIC33CDVL64MC106 family implements one instance of the QEI. Quadrature Encoders, also known as incremental encoders or optical encoders, detect position and speed of rotating motion systems. Quadrature Encoders enable closed-loop control of motor control applications, such as Switched Reluctance (SR) and AC Induction Motors (ACIM).

A typical Quadrature Encoder includes a slotted wheel attached to the shaft of the motor and an emitter/detector module that senses the slots in the wheel. Typically, three output channels, Phase A (QEAx), Phase B (QEBx) and Index (INDXx), provide information on the movement of the motor shaft, including distance and direction.

The two channels, Phase A (QEAx) and Phase B (QEBx), are typically 90 degrees out of phase with respect to each other. The Phase A and Phase B channels have a unique relationship. If Phase A leads Phase B, the direction of the motor is deemed positive or forward. If Phase A lags Phase B, the direction of the motor is deemed negative or reverse. The Index pulse occurs once per mechanical revolution and is used as a reference to indicate an absolute position. Figure 15-1 illustrates the Quadrature Encoder Interface signals.

The Quadrature signals from the encoder can have four unique states ('01', '00', '10' and '11') that reflect the relationship between QEAx and QEBx. Figure 15-1 illustrates these states for one count cycle. The order of the states get reversed when the direction of travel changes.

The Quadrature Decoder increments or decrements the 32-bit up/down Position x Counter (POSxCNTH/L) registers for each Change-of-State (COS). The counter increments when QEAx leads QEBx and decrements when QEBx leads QEAx.

FIGURE 15-1: QUADRATURE ENCODER INTERFACE SIGNALS

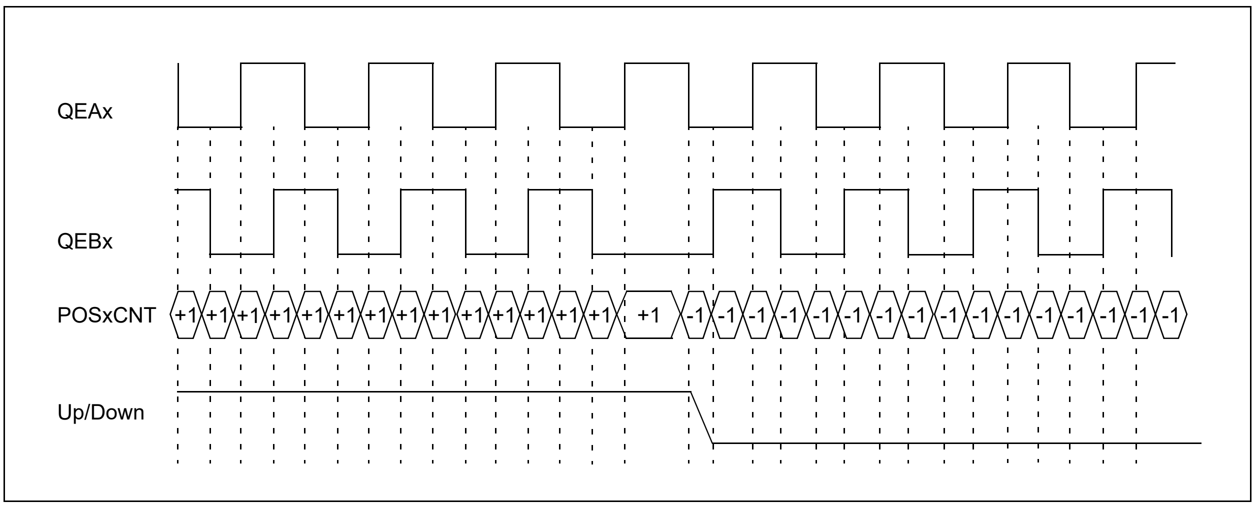


Table 15-1 shows the truth table that describes how the Quadrature signals are decoded.

TABLE 15-1: TRUTH TABLE FOR QUADRATURE ENCODER

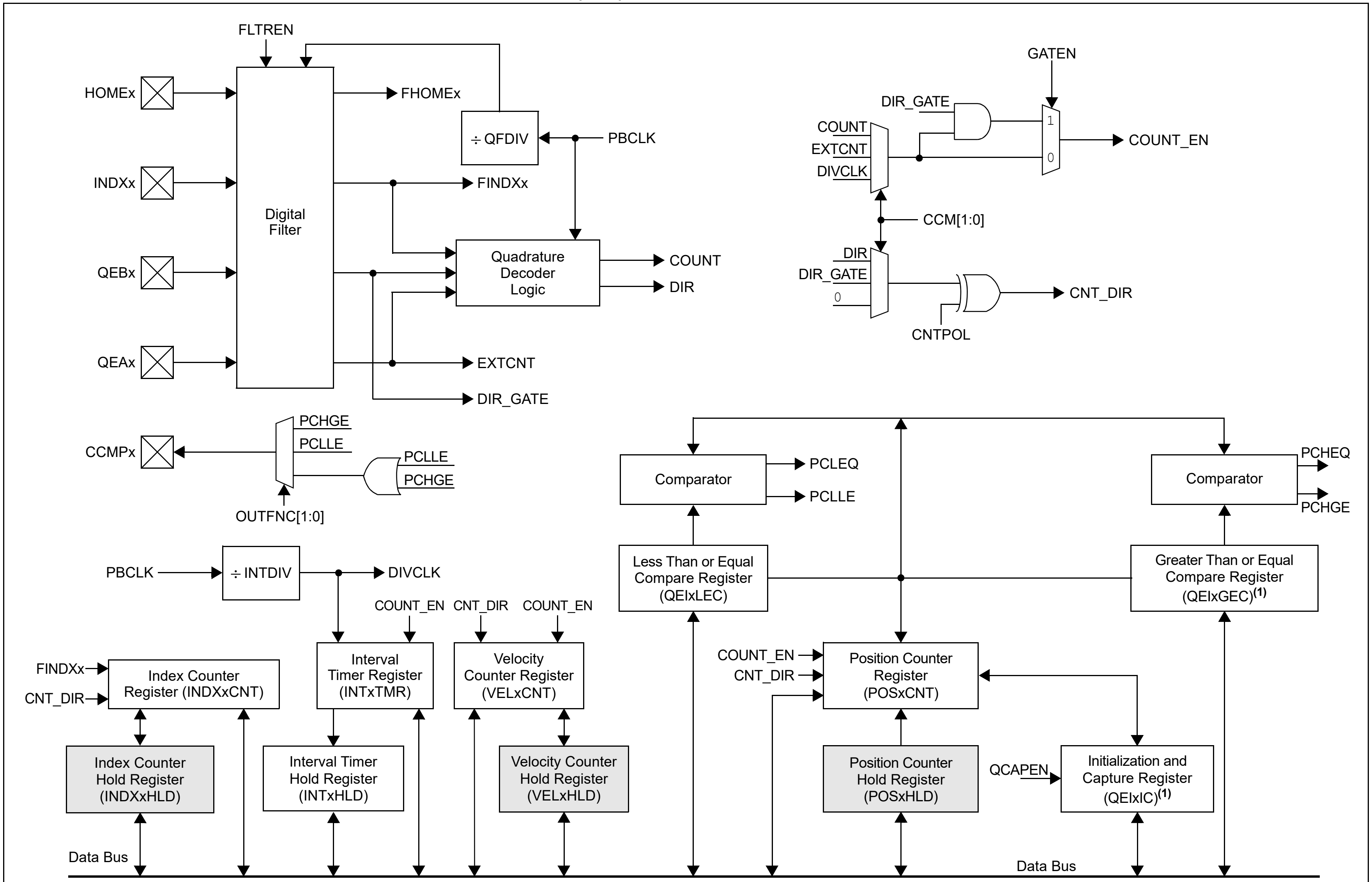
Current Quadrature State		Previous Quadrature State		Action
QEA	QEB	QEA	QEB	
1	1	1	1	No count or direction change
1	1	1	0	Count up
1	1	0	1	Count down
1	1	0	0	Invalid state change; ignore
1	0	1	1	Count down
1	0	1	0	No count or direction change
1	0	0	1	Invalid state change; ignore
1	0	0	0	Count up
0	1	1	1	Count up
0	1	1	0	Invalid state change; ignore
0	1	0	1	No count or direction change
0	1	0	0	Count down
0	0	1	1	Invalid state change; ignore
0	0	1	0	Count down
0	0	0	1	Count up
0	0	0	0	No count or direction change

Figure 15-2 illustrates the simplified block diagram of the QEI module. The QEI module consists of decoder logic to interpret the Phase A (QEAx) and Phase B (QEBx) signals, and an up/down counter to accumulate the count. The counter pulses are generated when the Quadrature state changes. The count direction information must be maintained in a register until a direction change is detected. The module also includes digital noise filters, which condition the input signal.

The QEI module consists of the following major features:

- Four Input Pins: Two Phase Signals, an Index Pulse and a Home Pulse
- Programmable Digital Noise Filters on Inputs
- Quadrature Decoder providing Counter Pulses and Count Direction
- Count Direction Status
- 4x Count Resolution
- Index (INDXx) Pulse to Reset the Position Counter
- General Purpose 32-Bit Timer/Counter Mode
- Interrupts generated by QEI or Counter Events
- 32-Bit Velocity Counter
- 32-Bit Position Counter
- 32-Bit Index Pulse Counter
- 32-Bit Interval Timer
- 32-Bit Position Initialization/Capture Register
- 32-Bit Compare Less Than and Greater Than Registers
- External Up/Down Count Mode
- External Gated Count Mode
- External Gated Timer Mode
- Interval Timer Mode

FIGURE 15-2: QUADRATURE ENCODER INTERFACE (QEI) MODULE BLOCK DIAGRAM



Note 1: These registers map to the same memory location.

15.1 QEI Control/Status Registers

REGISTER 15-1: QEIxCON: QEIx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIEN	—	QEISIDL	PIMOD2[1,5]	PIMOD1[1,5]	PIMOD0[1,5]	IMV1[2]	IMV0[2]
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	INTDIV2[3]	INTDIV1[3]	INTDIV0[3]	CNTPOL	GATEN	CCM1	CCM0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **QEIEN:** Quadrature Encoder Interface Module Enable bit
1 = Module counters are enabled
0 = Module counters are disabled, but SFRs can be read or written

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **QEISIDL:** QEI Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-10 **PIMOD[2:0]:** Position Counter Initialization Mode Select bits[1,5]
111 = Modulo Count mode for position counter and every Index event resets the position counter[4]
110 = Modulo Count mode for position counter
101 = Resets the position counter when the position counter equals the QEIxGEC register
100 = Second Index event after Home event initializes position counter with contents of QEIxIC register
011 = First Index event after Home event initializes position counter with contents of QEIxIC register
010 = Next Index input event initializes the position counter with contents of QEIxIC register
001 = Every Index input event resets the position counter
000 = Index input event does not affect the position counter

bit 9-8 **IMV[1:0]:** Index Match Value bits[2]
11 = Index match occurs when QEBx = 1 and QEAx = 1
10 = Index match occurs when QEBx = 1 and QEAx = 0
01 = Index match occurs when QEBx = 0 and QEAx = 1
00 = Index match occurs when QEBx = 0 and QEAx = 0

bit 7 **Unimplemented:** Read as ‘0’

Note 1: When CCMx = 10 or CCMx = 11, all of the QEI counters operate as timers and the PIMOD[2:0] bits are ignored.
2: When CCMx = 00, and QEAx and QEBx values match the Index Match Value (IMV), the POSxCNTH and POSxCNTL registers are reset.
3: The selected clock rate should be at least twice the expected maximum quadrature count rate.
4: Not all devices support this mode.
5: The QCAPEN and HCAPEN bits must be cleared during PIMODx Modes 2 through 7 to ensure proper functionality. Not all devices support HCAPEN.

REGISTER 15-1: QEIxCON: QEIx CONTROL REGISTER (CONTINUED)

bit 6-4 **INTDIV[2:0]:** Timer Input Clock Prescale Select bits(3)
(interval timer, main timer (position counter), velocity counter and index counter internal clock divider select)
111 = 1:128 prescale value
110 = 1:64 prescale value
101 = 1:32 prescale value
100 = 1:16 prescale value
011 = 1:8 prescale value
010 = 1:4 prescale value
001 = 1:2 prescale value
000 = 1:1 prescale value

bit 3 **CNTPOL:** Position and Index Counter/Timer Direction Select bit
1 = Counter direction is negative unless modified by external up/down signal
0 = Counter direction is positive unless modified by external up/down signal

bit 2 **GATEN:** External Count Gate Enable bit
1 = External gate signal controls position counter operation
0 = External gate signal does not affect position counter operation

bit 1-0 **CCM[1:0]:** Counter Control Mode Selection bits
11 = Internal Timer mode
10 = External Clock Count with External Gate mode
01 = External Clock Count with External Up/Down mode
00 = Quadrature Encoder mode

Note 1: When CCMx = 10 or CCMx = 11, all of the QEI counters operate as timers and the PIMOD[2:0] bits are ignored.
2: When CCMx = 00, and QEAx and QEBx values match the Index Match Value (IMV), the POSxCNTH and POSxCNTL registers are reset.
3: The selected clock rate should be at least twice the expected maximum quadrature count rate.
4: Not all devices support this mode.
5: The QCAPEN and HCAPEN bits must be cleared during PIMODx Modes 2 through 7 to ensure proper functionality. Not all devices support HCAPEN.

REGISTER 15-2: QEIxIOC: QEIx I/O CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QCAPEN	FLTREN	QFDIV2	QFDIV1	QFDIV0	OUTFNC1	OUTFNC0	SWPAB
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R-x	R-x	R-x	R-x
HOMPOL	IDXPOL	QEBPOL	QEAPOL	HOME	INDEX	QEB	QEA
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **QCAPEN:** QEIx Position Counter Input Capture Enable bit
1 = HOMEx input event (positive edge) triggers a position capture event (HCAPEN must be cleared)
0 = HOMEx input event (positive edge) does not trigger a position capture event

bit 14 **FLTREN:** QEAx/QEBx/INDXx/HOMEx Digital Filter Enable bit
1 = Input pin digital filter is enabled
0 = Input pin digital filter is disabled (bypassed)

bit 13-11 **QFDIV[2:0]:** QEAx/QEBx/INDXx/HOMEx Digital Input Filter Clock Divide Select bits
111 = 1:256 clock divide
110 = 1:64 clock divide
101 = 1:32 clock divide
100 = 1:16 clock divide
011 = 1:8 clock divide
010 = 1:4 clock divide
001 = 1:2 clock divide
000 = 1:1 clock divide

bit 10-9 **OUTFNC[1:0]:** QEIx Module Output Function Mode Select bits
11 = The QEICMPx pin goes high when POSxCNT ≤ QEIxLEC or POSxCNT ≥ QEIxGEC
10 = The QEICMPx pin goes high when POSxCNT ≤ QEIxLEC
01 = The QEICMPx pin goes high when POSxCNT ≥ QEIxGEC
00 = Output is disabled

bit 8 **SWPAB:** Swap QEAx and QEBx Inputs bit
1 = QEAx and QEBx are swapped prior to Quadrature Decoder logic
0 = QEAx and QEBx are not swapped

bit 7 **HOMPOL:** HOMEx Input Polarity Select bit
1 = Input is inverted
0 = Input is not inverted

bit 6 **IDXPOL:** INDXx Input Polarity Select bit
1 = Input is inverted
0 = Input is not inverted

bit 5 **QEBPOL:** QEBx Input Polarity Select bit
1 = Input is inverted
0 = Input is not inverted

bit 4 **QEAPOL:** QEAx Input Polarity Select bit
1 = Input is inverted
0 = Input is not inverted

bit 3 **HOME:** Status of HOMEx Input Pin After Polarity Control bit (read-only)
1 = Pin is at logic ‘1’ if the HOMPOL bit is set to ‘0’; pin is at logic ‘0’ if the HOMPOL bit is set to ‘1’
0 = Pin is at logic ‘0’ if the HOMPOL bit is set to ‘0’; pin is at logic ‘1’ if the HOMPOL bit is set to ‘1’

REGISTER 15-2: QEIxIOC: QEIx I/O CONTROL REGISTER (CONTINUED)

bit 2 **INDEX:** Status of INDXx Input Pin After Polarity Control bit (read-only)
1 = Pin is at logic '1' if the IDXPOL bit is set to '0'; pin is at logic '0' if the IDXPOL bit is set to '1'
0 = Pin is at logic '0' if the IDXPOL bit is set to '0'; pin is at logic '1' if the IDXPOL bit is set to '1'

bit 1 **QEB:** Status of QEBx Input Pin After Polarity Control and SWPAB Pin Swapping bit (read-only)
1 = Physical pin, QEBx, is at logic '1' if the QEBPOL bit is set to '0' and the SWPAB bit is set to '0'; physical pin, QEBx, is at logic '0' if the QEBPOL bit is set to '1' and the SWPAB bit is set to '0'; physical pin, QEBx, is at logic '1' if the QEBPOL bit is set to '0' and the SWPAB bit is set to '1'; physical pin, QEBx, is at logic '0' if the QEBPOL bit is set to '1' and the SWPAB bit is set to '1'
0 = Physical pin, QEBx, is at logic '0' if the QEBPOL bit is set to '0' and the SWPAB bit is set to '0'; physical pin, QEBx, is at logic '1' if the QEBPOL bit is set to '1' and the SWPAB bit is set to '0'; physical pin, QEBx, is at logic '0' if the QEBPOL bit is set to '0' and the SWPAB bit is set to '1'; physical pin, QEBx, is at logic '1' if the QEBPOL bit is set to '1' and the SWPAB bit is set to '1'

bit 0 **QEA:** Status of QEAx Input Pin After Polarity Control and SWPAB Pin Swapping bit (read-only)
1 = Physical pin, QEAx, is at logic '1' if the QEAPOL bit is set to '0' and the SWPAB bit is set to '0'; physical pin, QEAx, is at logic '0' if the QEAPOL bit is set to '1' and the SWPAB bit is set to '0'; physical pin, QEAx, is at logic '1' if the QEAPOL bit is set to '0' and the SWPAB bit is set to '1'; physical pin, QEAx, is at logic '0' if the QEAPOL bit is set to '1' and the SWPAB bit is set to '1'
0 = Physical pin, QEAx, is at logic '0' if the QEAPOL bit is set to '0' and the SWPAB bit is set to '0'; physical pin, QEAx, is at logic '1' if the QEAPOL bit is set to '1' and the SWPAB bit is set to '0'; physical pin, QEAx, is at logic '0' if the QEAPOL bit is set to '0' and the SWPAB bit is set to '1'; physical pin, QEAx, is at logic '1' if the QEAPOL bit is set to '1' and the SWPAB bit is set to '1'

REGISTER 15-3: QEIxIOCH: QEIx I/O CONTROL HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	HCAPEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-1 **Unimplemented:** Read as '0'

bit 0 **HCAPEN:** Position Counter Input Capture by Home Event Enable bit
1 = HOMEx input event (positive edge) triggers a position capture event
0 = HOMEx input event (positive edge) does not trigger a position capture event

REGISTER 15-4: QEIxSTAT: QEIx STATUS REGISTER

U-0	U-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0
—	—	PCHEQIRQ	PCHEQIEN	PCLEQIRQ	PCLEQIEN	POSOVIRQ	POSOVIEN
bit 15							bit 8

HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0
PCIIRQ[1]	PCIIEN	VELOVIRQ	VELOVIEN	HOMIRQ	HOMIEN	IDXIRQ	IDXIEN
bit 7							bit 0

Legend:	C = Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **PCHEQIRQ:** Position Counter Greater Than Compare Status bit
1 = POSxCNT ≥ QEIxGEC
0 = POSxCNT < QEIxGEC

bit 12 **PCHEQIEN:** Position Counter Greater Than Compare Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 11 **PCLEQIRQ:** Position Counter Less Than Compare Status bit
1 = POSxCNT ≤ QEIxLEC
0 = POSxCNT > QEIxLEC

bit 10 **PCLEQIEN:** Position Counter Less Than Compare Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 9 **POSOVIRQ:** Position Counter Overflow Status bit
1 = Overflow has occurred
0 = No overflow has occurred

bit 8 **POSOVIEN:** Position Counter Overflow Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 7 **PCIIRQ:** Position Counter (Homing) Initialization Process Complete Status bit[1]
1 = POSxCNT was reinitialized
0 = POSxCNT was not reinitialized

bit 6 **PCIIEN:** Position Counter (Homing) Initialization Process Complete Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 5 **VELOVIRQ:** Velocity Counter Overflow Status bit
1 = Overflow has occurred
0 = No overflow has occurred

bit 4 **VELOVIEN:** Velocity Counter Overflow Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 3 **HOMIRQ:** Status Flag for Home Event Status bit
1 = Home event has occurred
0 = No Home event has occurred

Note 1: This status bit is only applicable to PIMOD[2:0] modes, ‘011’ and ‘100’.

REGISTER 15-4: QEIxSTAT: QEIx STATUS REGISTER (CONTINUED)

bit 2 **HOMIEN:** Home Input Event Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 1 **IDXIRQ:** Status Flag for Index Event Status bit
1 = Index event has occurred
0 = No Index event has occurred

bit 0 **IDXIEN:** Index Input Event Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Note 1: This status bit is only applicable to PIMOD[2:0] modes, '011' and '100'.

REGISTER 15-5: POSxCNTL: POSITION x COUNTER REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **POSCNT[15:0]:** Low Word Used to Form 32-Bit Position Counter Register (POSxCNT) bits

REGISTER 15-6: POSxCNTH: POSITION x COUNTER REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **POSCNT[31:16]:** High Word Used to Form 32-Bit Position Counter Register (POSxCNT) bits

REGISTER 15-7: POSxHLD: POSITION x COUNTER HOLD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSHLD[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSHLD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **POSHLD[15:0]:** Hold Register for Reading/Writing Position Counter x High Word Register (POSxCNTH (POSxCNT[31:16])) bits

REGISTER 15-8: VELxCNT: VELOCITY x COUNTER REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **VELCNT[15:0]:** Velocity Counter bits

REGISTER 15-9: VELxCNTH: VELOCITY x COUNTER REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **VELCNT[31:16]:** Velocity Counter bits

REGISTER 15-10: VELxHLD: VELOCITY x COUNTER HOLD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELHLD[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELHLD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **VELHLD[15:0]:** Hold Register for Reading/Writing Velocity Counter x High Word Register (VELxCNTH (VELxCNT[31:16])) bits

REGISTER 15-11: INTxTMRL: INTERVAL x TIMER REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INTTMR[15:0]:** Low Word Used to Form 32-Bit Interval Timer Register (INTxTMR) bits

REGISTER 15-12: INTxTMRH: INTERVAL x TIMER REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INTTMR[31:16]:** High Word Used to Form 32-Bit Interval Timer Register (INTxTMR) bits

REGISTER 15-13: INTXxHLDL: INTERVAL x TIMER HOLD REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INTHLD[15:0]:** Low Word Used to Form 32-Bit Interval Timer Hold Register (INTxHLD) bits

REGISTER 15-14: INTXxHLDH: INTERVAL x TIMER HOLD REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INTHLD[31:16]:** High Word Used to Form 32-Bit Interval Timer Hold Register (INTxHLD) bits

REGISTER 15-15: INDXxCNTL: INDEX x COUNTER REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INDXCNT[15:0]:** Low Word Used to Form 32-Bit Index x Counter Register (INDXxCNT) bits

REGISTER 15-16: INDXxCNTH: INDEX x COUNTER REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INDXCNT[31:16]:** High Word Used to Form 32-Bit Index x Counter Register (INDXxCNT) bits

REGISTER 15-17: INDXxHLD: INDEX x COUNTER HOLD REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXHLD[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXHLD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **INDXHLD[15:0]:** Hold Register for Reading/Writing Index Counter x High Word Register (INDXxCNTH (INDXxCNT[31:16])) bits

REGISTER 15-18: QEIxICL: QEIx INITIALIZATION/CAPTURE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIC[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEIIC[15:0]:** Low Word Used to Form 32-Bit Initialization/Capture Register (QEIxIC) bits

REGISTER 15-19: QEIxICH: QEIx INITIALIZATION/CAPTURE REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIC[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIC[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEIIC[31:16]:** High Word Used to Form 32-Bit Initialization/Capture Register (QEIxIC) bits

REGISTER 15-20: QEIxLECL: QEIx LESS THAN OR EQUAL COMPARE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEILEC[15:0]:** Low Word Used to Form 32-Bit Less Than or Equal Compare Register (QEIxLEC) bits

REGISTER 15-21: QEIxLECH: QEIx LESS THAN OR EQUAL COMPARE REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEILEC[31:16]:** High Word Used to Form 32-Bit Less Than or Equal Compare Register (QEIxLEC) bits

REGISTER 15-22: QEIxGECL: QEIx GREATER THAN OR EQUAL COMPARE REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEIGEC[15:0]:** Low Word Used to Form 32-Bit Greater Than or Equal Compare Register (QEIxGEC) bits

REGISTER 15-23: QEIxGECH: QEIx GREATER THAN OR EQUAL COMPARE REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **QEIGEC[31:16]:** High Word Used to Form 32-Bit Greater Than or Equal Compare Register (QEIxGEC) bits

16.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note 1:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Multiprotocol Universal Asynchronous Receiver Transmitter (UART) Module"** (www.microchip.com/DS70005288).

The Universal Asynchronous Receiver Transmitter (UART) is a flexible serial communication peripheral used to interface dsPIC® microcontrollers with other equipment, including computers and peripherals. The UART is a full-duplex, asynchronous communication channel that can be used to implement protocols, such as RS-232 and RS-485. The UART also supports the following hardware extensions:

- LIN/J2602
- IrDA®
- Digital Multiplex (DMX)
- Smart Card

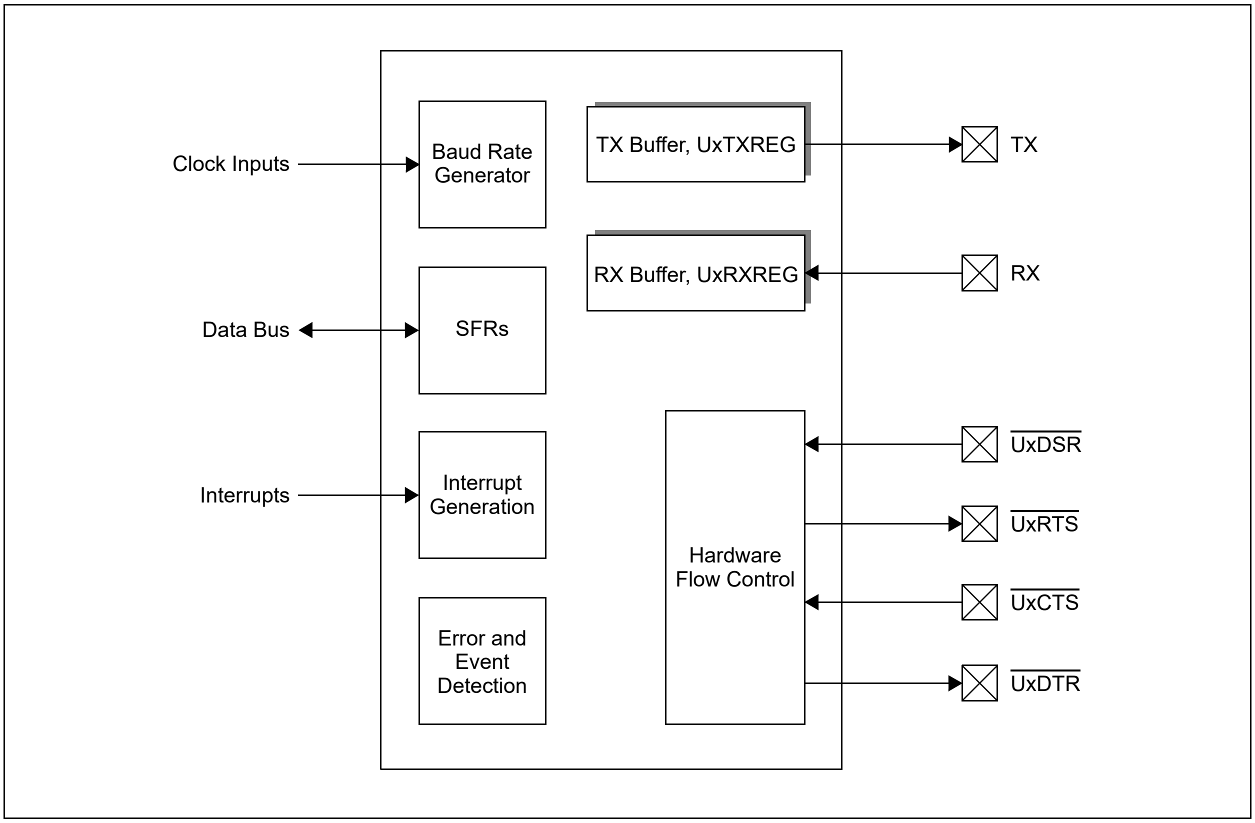
The primary features of the UART are:

- Full or Half-Duplex Operation
- Up to 8-Deep TX and RX First In, First Out (FIFO) Buffers
- 8-Bit or 9-Bit Data Width
- Configurable Stop Bit Length
- Flow Control
- Auto-Baud Calibration
- Parity, Framing and Buffer Overrun Error Detection
- Address Detect
- Break Transmission
- Transmit and Receive Polarity Control
- Manchester Encoder/Decoder
- Operation in Sleep Mx‘ode
- Wake from Sleep on Sync Break Received Interrupt

16.1 Architectural Overview

The UART transfers bytes of data, to and from device pins, using First-In First-Out (FIFO) buffers up to eight bytes deep. The status of the buffers and data is made available to user software through Special Function Registers (SFRs). The UART implements multiple interrupt channels for handling transmit, receive and error events. A simplified block diagram of the UART is shown in Figure 16-1.

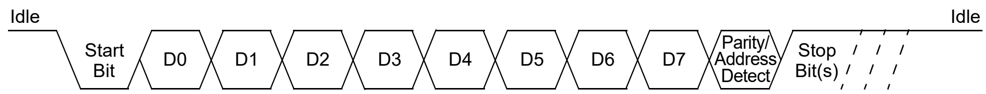
FIGURE 16-1: SIMPLIFIED UARTx BLOCK DIAGRAM



16.2 Character Frame

A typical UART character frame is shown in Figure 16-2. The Idle state is high with a 'Start' condition indicated by a falling edge. The Start bit is followed by the number of data, parity/address detect and Stop bits defined by the MOD[3:0] (UxMODE[3:0]) bits selected.

FIGURE 16-2: UART CHARACTER FRAME



16.3 Data Buffers

Both transmit and receive functions use buffers to store data shifted to/from the pins. These buffers are FIFOs and are accessed by reading the SFRs, UxTXREG and UxRXREG, respectively. Each data buffer has multiple flags associated with its operation to allow software to read the status. Interrupts can also be configured based on the space available in the buffers. The transmit and receive buffers can be cleared and their pointers reset using the associated TX/RX Buffer Empty Status bits, UTXBE (UxSTAH[5]) and URXBE (UxSTAH[1]).

16.4 Protocol Extensions

The UART provides hardware support for LIN/J2602, IrDA®, DMX and smart card protocol extensions to reduce software overhead. A protocol extension is enabled by writing a value to the MOD[3:0] (UxMODE[3:0]) selection bits and further configured using the UARTx Timing Parameter registers, UxP1 (Register 16-9), UxP2 (Register 16-10), UxP3 (Register 16-11) and UxP3H (Register 16-12). Details regarding operation and usage are discussed in their respective chapters.

16.5 UART Control/Status Registers

REGISTER 16-1: UxMODE: UARTx CONFIGURATION REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	HC/R/W-0[1]
UARTEN	—	USIDL	WAKE	RXBIMD	—	BRKOVR	UTXBRK
bit 15							bit 8

R/W-0	HC/R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRGH	ABAUD	UTXEN	URXEN	MOD3	MOD2	MOD1	MOD0
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **UARTEN:** UART Enable bit
1 = UART is ready to transmit and receive
0 = UART state machine, FIFO Buffer Pointers and counters are reset; registers are readable and writable

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** UART Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **WAKE:** Wake-up Enable bit
1 = Module will continue to sample the RX pin – interrupt generated on falling edge, bit cleared in hardware on following rising edge; if ABAUD is set, Auto-Baud Detection (ABD) will begin immediately
0 = RX pin is not monitored nor rising edge detected

bit 11 **RXBIMD:** Receive Break Interrupt Mode bit
1 = RXBKIF flag when a minimum of 23 (DMX)/11 (asynchronous or LIN/J2602) low bit periods are detected
0 = RXBKIF flag when the Break makes a low-to-high transition after being low for at least 23/11 bit periods

bit 10 **Unimplemented:** Read as '0'

bit 9 **BRKOVR:** Send Break Software Override bit
Overrides the TX Data Line:
1 = Makes the TX line active (Output 0 when UTXINV = 0, Output 1 when UTXINV = 1)
0 = TX line is driven by the shifter

bit 8 **UTXBRK:** UART Transmit Break bit[1]
1 = Sends Sync Break on next transmission; cleared by hardware upon completion
0 = Sync Break transmission is disabled or has completed

bit 7 **BRGH:** High Baud Rate Select bit
1 = High Speed: Baud rate is baudclk/4
0 = Low Speed: Baud rate is baudclk/16

bit 6 **ABAUD:** Auto-Baud Detect Enable bit (read-only when MOD[3:0] = 1xxx)
1 = Enables baud rate measurement on the next character – requires reception of a Sync field (55h); cleared in hardware upon completion
0 = Baud rate measurement is disabled or has completed

Note 1: R/HS/HC in DMX and LIN mode.

REGISTER 16-1: UxMODE: UARTx CONFIGURATION REGISTER (CONTINUED)

bit 5 **UTXEN:** UART Transmit Enable bit
1 = Transmit enabled – except during Auto-Baud Detection
0 = Transmit disabled – all transmit counters, pointers and state machines are reset; TX buffer is not flushed, status bits are not reset

bit 4 **URXEN:** UART Receive Enable bit
1 = Receive enabled – except during Auto-Baud Detection
0 = Receive disabled – all receive counters, pointers and state machines are reset; RX buffer is not flushed, status bits are not reset

bit 3-0 **MOD[3:0]:** UART Mode bits
Other = Reserved
1111 = Smart card
1110 = IrDA®
1101 = Reserved
1100 = LIN Commander/Responder
1011 = LIN Responder only
1010 = DMX
1001 = Reserved
1000 = Reserved
0111 = Reserved
0110 = Reserved
0101 = Reserved
0100 = Asynchronous 9-bit UART with address detect, ninth bit = 1 signals address
0011 = Asynchronous 8-bit UART without address detect, ninth bit is used as an even parity bit
0010 = Asynchronous 8-bit UART without address detect, ninth bit is used as an odd parity bit
0001 = Asynchronous 7-bit UART
0000 = Asynchronous 8-bit UART

Note 1: R/HS/HC in DMX and LIN mode.

REGISTER 16-2: UxMODEH: UARTx CONFIGURATION REGISTER HIGH

R/W-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPEN	ACTIVE	—	—	BCLKMOD	BCLKSEL1	BCLKSEL0	HALFDPLX
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RUNOVF	URXINV	STSEL1	STSEL0	C0EN	UTXINV	FLO1	FLO0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **SLPEN:** Run During Sleep Enable bit
1 = UART BRG clock runs during Sleep
0 = UART BRG clock is turned off during Sleep

bit 14 **ACTIVE:** UART Running Status bit
1 = UART clock request is active (user can not update the UxMODE/UxMODEH registers)
0 = UART clock request is not active (user can update the UxMODE/UxMODEH registers)

bit 13-12 **Unimplemented:** Read as ‘0’

bit 11 **BCLKMOD:** Baud Clock Generation Mode Select bit
1 = Uses fractional Baud Rate Generation
0 = Uses legacy divide-by-x counter for baud clock generation (x = 4 or 16 depending on the BRGH bit)

bit 10-9 **BCLKSEL[1:0]:** Baud Clock Source Selection bits
11 = FVCODIV
10 = FOSC
01 = PLL VCO/5
00 = FOSC/2 (FP)

bit 8 **HALFDPLX:** UART Half-Duplex Selection Mode bit
1 = Half-Duplex mode: UxTX is driven as an output when transmitting and tri-stated when TX is Idle
0 = Full-Duplex mode: UxTX is driven as an output at all times when both UARTEN and UTXEN are set

bit 7 **RUNOVF:** Run During Overflow Condition Mode bit
1 = When an Overflow Error (OERR) condition is detected, the RX shifter continues to run so as to remain synchronized with incoming RX data; data are not transferred to UxRXREG when it is full (i.e., no UxRXREG data are overwritten)
0 = When an Overflow Error (OERR) condition is detected, the RX shifter stops accepting new data (Legacy mode)

bit 6 **URXINV:** UART Receive Polarity bit
1 = Inverts RX polarity; Idle state is low
0 = Input is not inverted; Idle state is high

bit 5-4 **STSEL[1:0]:** Number of Stop Bits Selection bits
11 = 2 Stop bits sent, 1 checked at receive
10 = 2 Stop bits sent, 2 checked at receive
01 = 1.5 Stop bits sent, 1.5 checked at receive
00 = 1 Stop bit sent, 1 checked at receive

bit 3 **C0EN:** Enable Legacy Checksum (C0) Transmit and Receive bit
1 = Checksum Mode 1 (enhanced LIN checksum in LIN mode; add all TX/RX words in all other modes)
0 = Checksum Mode 0 (legacy LIN checksum in LIN mode; not used in all other modes)

REGISTER 16-2: UxMODEH: UARTx CONFIGURATION REGISTER HIGH (CONTINUED)

bit 2	**UTXINV:** UART Transmit Polarity bit 1 = Inverts TX polarity; TX is low in Idle state 0 = Output data are not inverted; TX output is high in Idle state
bit 1-0	**FLO[1:0]:** Flow Control Enable bits (only valid when MOD[3:0] = 0xxx) 11 = Reserved 10 = $\overline{\text{RTS}}$-$\overline{\text{DSR}}$ (for TX side)/$\overline{\text{CTS}}$-DTR (for RX side) hardware flow control 01 = XON/XOFF software flow control 00 = Flow control off

REGISTER 16-3: UxSTA: UARTx STATUS REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXMTIE	PERIE	ABDOVE	CERIE	FERIE	RXBKIE	OERIE	TXCIE
bit 15							bit 8

R-1	R-0	HS/R/W-0	HS/R/W-0	R-0	HS/R/W-0	HS/R/W-0	R/W-0
TRMT	PERR	ABDOVF	CERIF	FERR	RXBKIF	OERR	TXCIF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **TXMTIE:** Transmit Shifter Empty Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 14 **PERIE:** Parity Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 13 **ABDOVE:** Auto-Baud Rate Acquisition Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 12 **CERIE:** Checksum Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 11 **FERIE:** Framing Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 10 **RXBKIE:** Receive Break Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 9 **OERIE:** Receive Buffer Overflow Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 8 **TXCIE:** Transmit Collision Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 7 **TRMT:** Transmit Shifter Empty Interrupt Flag bit (read-only)
1 = Transmit Shift Register (TSR) is empty (end of last Stop bit when STPMD = 1 or middle of first Stop bit when STPMD = 0)
0 = Transmit Shift Register is not empty

bit 6 **PERR:** Parity Error/Address Received/Forward Frame Interrupt Flag bit
<u>LIN and Parity Modes:</u>
1 = Parity error detected
0 = No parity error detected
<u>Address Mode:</u>
1 = Address received
0 = No address detected
<u>All Other Modes:</u>
Not used.

REGISTER 16-3: UxSTA: UARTx STATUS REGISTER (CONTINUED)

bit 5	**ABDOVF:** Auto-Baud Rate Acquisition Interrupt Flag bit (must be cleared by software) 1 = BRG rolled over during the auto-baud rate acquisition sequence (must be cleared in software) 0 = BRG has not rolled over during the auto-baud rate acquisition sequence
bit 4	**CERIF:** Checksum Error Interrupt Flag bit (must be cleared by software) 1 = Checksum error 0 = No checksum error
bit 3	**FERR:** Framing Error Interrupt Flag bit 1 = Framing Error: Inverted level of the Stop bit corresponding to the topmost character in the buffer; propagates through the buffer with the received character 0 = No framing error
bit 2	**RXBKIF:** Receive Break Interrupt Flag bit (must be cleared by software) 1 = A Break was received 0 = No Break was detected
bit 1	**OERR:** Receive Buffer Overflow Interrupt Flag bit (must be cleared by software) 1 = Receive buffer has overflowed 0 = Receive buffer has not overflowed
bit 0	**TXCIF:** Transmit Collision Interrupt Flag bit (must be cleared by software) 1 = Transmitted word is not equal to the received word 0 = Transmitted word is equal to the received word

REGISTER 16-4: UxSTAH: UARTx STATUS REGISTER HIGH

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	UTXISEL2	UTXISEL1	UTXISEL0	—	URXISEL2[(1)]	URXISEL1[(1)]	URXISEL0[(1)]
bit 15							bit 8

HS/R/W-0	R/W-0	R/S-1	R-0	R-1	R-1	R/S-1	R-0
TXWRE	STPMD	UTXBE	UTXBF	RIDLE	XON	URXBE	URXBF
bit 7							bit 0

Legend:	HS = Hardware Settable bit	S = Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **UTXISEL[2:0]:** UART Transmit Interrupt Select bits
111 = Sets transmit interrupt when there is one empty slot left in the buffer
...
010 = Sets transmit interrupt when there are six empty slots or more in the buffer
001 = Sets transmit interrupt when there are seven empty slots or more in the buffer
000 = Sets transmit interrupt when there are eight empty slots in the buffer; TX buffer is empty

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **URXISEL[2:0]:** UART Receive Interrupt Select bits[(1)]
111 = Triggers receive interrupt when there are eight bytes in the buffer; RX buffer is full
...
001 = Triggers receive interrupt when there are two bytes or more in the buffer
000 = Triggers receive interrupt when there is one byte or more in the buffer

bit 7 **TXWRE:** TX Write Transmit Error Status bit
<u>LIN and Parity Modes:</u>
1 = A new byte was written when the buffer was full or when P2[8:0] = 0 (must be cleared by software)
0 = No error
<u>Address Detect Mode:</u>
1 = A new byte was written when the buffer was full or to P1[8:0] when P1x was full (must be cleared by software)
0 = No error
<u>Other Modes:</u>
1 = A new byte was written when the buffer was full (must be cleared by software)
0 = No error

bit 6 **STPMD:** Stop Bit Detection Mode bit
1 = Triggers RXIF at the end of the last Stop bit
0 = Triggers RXIF in the middle of the first (or second, depending on the STSEL[1:0] setting) Stop bit

bit 5 **UTXBE:** UART TX Buffer Empty Status bit
1 = Transmit buffer is empty; writing '1' when UTXEN = 0 will reset the TX FIFO Pointers and counters
0 = Transmit buffer is not empty

bit 4 **UTXBF:** UART TX Buffer Full Status bit
1 = Transmit buffer is full
0 = Transmit buffer is not full

bit 3 **RIDLE:** Receive Idle bit
1 = UART RX line is in the Idle state
0 = UART RX line is receiving something

Note 1: The receive watermark interrupt is not set if PERR or FERR is set and the corresponding IE bit is set.

REGISTER 16-4: UxSTAH: UARTx STATUS REGISTER HIGH (CONTINUED)

bit 2	**XON:** UART in XON Mode bit Only valid when FLO[1:0] control bits are set to XON/XOFF mode. 1 = UART has received XON 0 = UART has not received XON or XOFF was received
bit 1	**URXBE:** UART RX Buffer Empty Status bit 1 = Receive buffer is empty; writing '1' when URXEN = 0 will reset the RX FIFO Pointers and counters 0 = Receive buffer is not empty
bit 0	**URXBF:** UART RX Buffer Full Status bit 1 = Receive buffer is full 0 = Receive buffer is not full

Note 1: The receive watermark interrupt is not set if PERR or FERR is set and the corresponding IE bit is set.

REGISTER 16-5: UxBRG: UARTx BAUD RATE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **BRG[15:0]:** Baud Rate Divisor bits

REGISTER 16-6: UxBRGH: UARTx BAUD RATE REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BRG[19:16]			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3-0 **BRG[19:16]:** Baud Rate Divisor bits

REGISTER 16-7: UxRXREG: UARTx RECEIVE BUFFER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RXREG[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **RXREG[7:0]:** Received Character Data bits 7-0

REGISTER 16-8: UxTXREG: UARTx TRANSMIT BUFFER REGISTER

W-x	U-0	U-0	U-0	U-0	U-0	U-0	U-0
LAST	—	—	—	—	—	—	—
bit 15							bit 8

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
TXREG[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **LAST:** Last Byte Indicator for Smart Card Support bit

bit 14-8 **Unimplemented:** Read as ‘0’

bit 7-0 **TXREG[7:0]:** Transmitted Character Data bits 7-0
If the buffer is full, further writes to the buffer are ignored.

REGISTER 16-9: UxP1: UARTx TIMING PARAMETER 1 REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	P1[8]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8-0 **P1[8:0]:** Parameter 1 bits

DMX TX:
Number of Bytes to Transmit – 1 (not including Start code).

LIN Master TX:
PID to transmit (bits[5:0]).

Asynchronous TX with Address Detect:
Address to transmit. A ‘1’ is automatically inserted into bit 9 (bits[7:0]).

Smart Card Mode:
Guard Time Counter bits. This counter is operated on the bit clock whose period is always equal to one ETU (bits[8:0]).

Other Modes:
Not used.

REGISTER 16-10: UxP2: UARTx TIMING PARAMETER 2 REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	P2[8]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P2[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8-0 **P2[8:0]:** Parameter 2 bits

DMX RX:
The first byte number to receive – 1, not including Start code (bits[8:0]).

LIN Slave TX:
Number of bytes to transmit (bits[7:0]).

Asynchronous RX with Address Detect:
Address to start matching (bits[7:0]).

Smart Card Mode:
Block Time Counter bits. This counter is operated on the bit clock whose period is always equal to one ETU (bits[8:0]).

Other Modes:
Not used.

REGISTER 16-11: UxP3: UARTx TIMING PARAMETER 3 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P3[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P3[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **P3[15:0]:** Parameter 3 bits

DMX RX:
The last byte number to receive – 1, not including Start code (bits[8:0]).

LIN Slave RX:
Number of bytes to receive (bits[7:0]).

Asynchronous RX:
Used to mask the UxP2 address bits; `1` = P2 address bit is used, `0` = P2 address bit is masked off (bits[7:0]).

Smart Card Mode:
Waiting Time Counter bits (bits[15:0]).

Other Modes:
Not used.

REGISTER 16-12: UxP3H: UARTx TIMING PARAMETER 3 REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P3[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **P3[23:16]:** Parameter 3 High bits

Smart Card Mode:
Waiting Time Counter bits (bits[23:16]).

Other Modes:
Not used.

REGISTER 16-13: UxTXCHK: UARTx TRANSMIT CHECKSUM REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXCHK[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **TXCHK[7:0]:** Transmit Checksum bits (calculated from TX words)

LIN Modes:
C0EN = 1: Sum of all transmitted data + addition carries, including PID.
C0EN = 0: Sum of all transmitted data + addition carries, excluding PID.

LIN Slave:
Cleared when Break is detected.

LIN Master/Slave:
Cleared when Break is detected.

Other Modes:
C0EN = 1: Sum of every byte transmitted + addition carries.
C0EN = 0: Value remains unchanged.

REGISTER 16-14: UxRXCHK: UARTx RECEIVE CHECKSUM REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXCHK[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **RXCHK[7:0]:** Receive Checksum bits (calculated from RX words)

<u>LIN Modes:</u>
C0EN = 1: Sum of all received data + addition carries, including PID.
C0EN = 0: Sum of all received data + addition carries, excluding PID.

<u>LIN Slave:</u>
Cleared when Break is detected.

<u>LIN Master/Slave:</u>
Cleared when Break is detected.

<u>Other Modes:</u>
C0EN = 1: Sum of every byte received + addition carries.
C0EN = 0: Value remains unchanged.

REGISTER 16-15: UxSCCON: UARTx SMART CARD CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	TXRPT1	TXRPT0	CONV	T0PD	PRTCL	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5-4 **TXRPT[1:0]:** Transmit Repeat Selection bits
11 = Retransmit the error byte four times
10 = Retransmit the error byte three times
01 = Retransmit the error byte twice
00 = Retransmit the error byte once

bit 3 **CONV:** Logic Convention Selection bit
1 = Inverse logic convention
0 = Direct logic convention

bit 2 **T0PD:** Pull-Down Duration for T = 0 Error Handling bit
1 = Two ETUs
0 = One ETU

bit 1 **PRTCL:** Smart Card Protocol Selection bit
1 = T = 1
0 = T = 0

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 16-16: UxSCINT: UARTx SMART CARD INTERRUPT REGISTER

U-0	U-0	HS/R/W-0	HS/R/W-0	U-0	HS/R/W-0	HS/R/W-0	HS/R/W-0
—	—	RXRPTIF	TXRPTIF	—	BTCIF	WTCIF	GTCIF
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	RXRPTIE	TXRPTIE	—	BTCIE	WTCIE	GTCIE
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **RXRPTIF:** Receive Repeat Interrupt Flag bit
1 = Parity error has persisted after the same character has been received five times (four retransmits)
0 = Flag is cleared

bit 12 **TXRPTIF:** Transmit Repeat Interrupt Flag bit
1 = Line error has been detected after the last retransmit per TXRPT[1:0]
0 = Flag is cleared

bit 11 **Unimplemented:** Read as ‘0’

bit 10 **BTCIF:** Block Time Counter Interrupt Flag bit
1 = Block Time Counter has reached 0
0 = Block Time Counter has not reached 0

bit 9 **WTCIF:** Waiting Time Counter Interrupt Flag bit
1 = Waiting Time Counter has reached 0
0 = Waiting Time Counter has not reached 0

bit 8 **GTCIF:** Guard Time Counter Interrupt Flag bit
1 = Guard Time Counter has reached 0
0 = Guard Time Counter has not reached 0

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **RXRPTIE:** Receive Repeat Interrupt Enable bit
1 = An interrupt is invoked when a parity error has persisted after the same character has been received five times (four retransmits)
0 = Interrupt is disabled

bit 4 **TXRPTIE:** Transmit Repeat Interrupt Enable bit
1 = An interrupt is invoked when a line error is detected after the last retransmit per TXRPT[1:0] has been completed
0 = Interrupt is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **BTCIE:** Block Time Counter Interrupt Enable bit
1 = Block Time Counter interrupt is enabled
0 = Block Time Counter interrupt is disabled

bit 1 **WTCIE:** Waiting Time Counter Interrupt Enable bit
1 = Waiting Time Counter interrupt is enabled
0 = Waiting Time Counter Interrupt is disabled

bit 0 **GTCIE:** Guard Time Counter interrupt enable bit
1 = Guard Time Counter interrupt is enabled
0 = Guard Time Counter interrupt is disabled

REGISTER 16-17: UxINT: UARTx INTERRUPT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

HS/R/W-0	HS/R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0
WUIF	ABDIF	—	—	—	ABDIE	—	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **WUIF:** Wake-up Interrupt Flag bit
1 = Sets when WAKE = 1 and RX makes a ‘1’-to-‘0’ transition; triggers event interrupt (must be cleared by software)
0 = WAKE is not enabled or WAKE is enabled, but no wake-up event has occurred

bit 6 **ABDIF:** Auto-Baud Completed Interrupt Flag bit
1 = Sets when ABD sequence makes the final ‘1’-to-‘0’ transition; triggers event interrupt (must be cleared by software)
0 = ABAUD is not enabled or ABAUD is enabled but auto-baud has not completed

bit 5-3 **Unimplemented:** Read as ‘0’

bit 2 **ABDIE:** Auto-Baud Completed Interrupt Enable Flag bit
1 = Allows ABDIF to set an event interrupt
0 = ABDIF does not set an event interrupt

bit 1-0 **Unimplemented:** Read as ‘0’

NOTES:

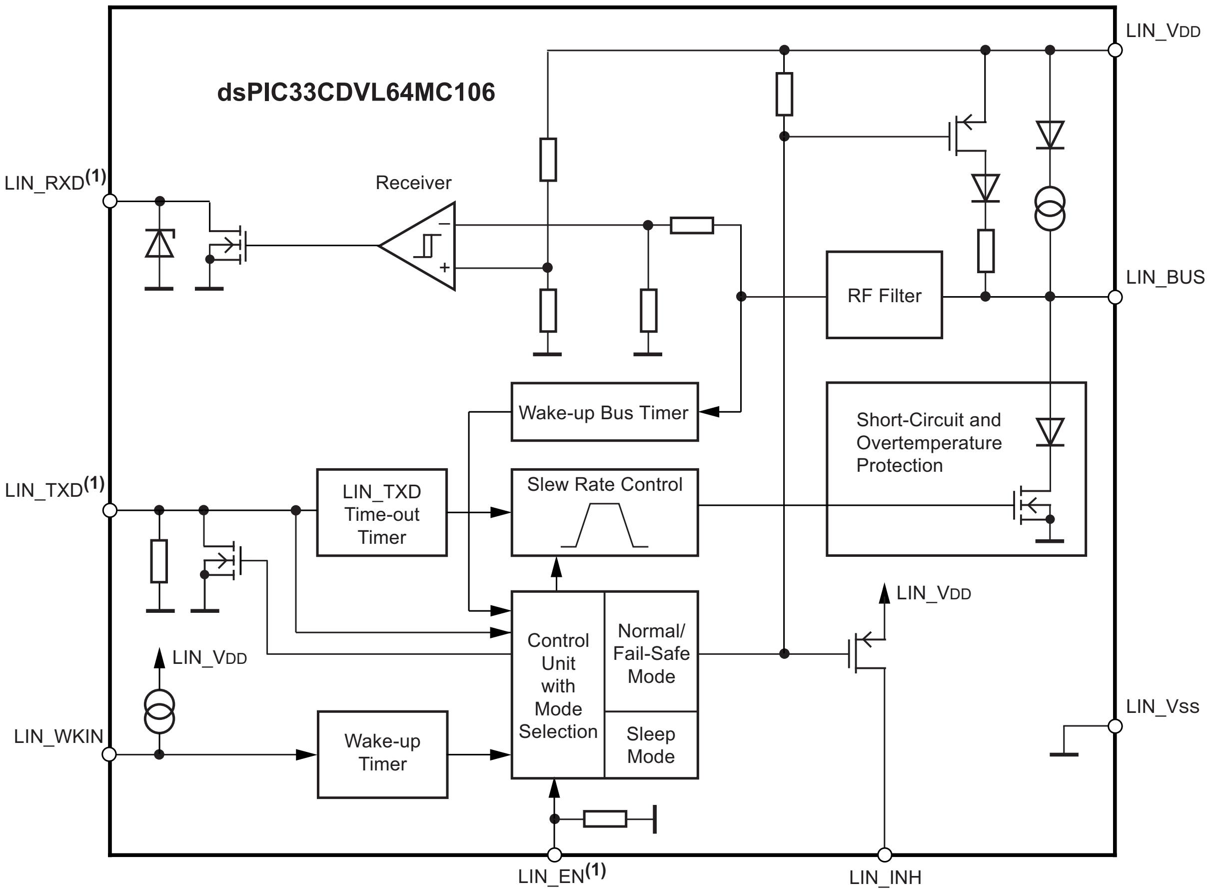
17.0 LIN TRANSCEIVER MODULE

The fully integrated LIN Transceiver is designed in compliance with the LIN Specification 2.0, 2.1, 2.2, 2.2A and SAEJ2602-2. It interfaces the LIN protocol handler and the physical layer. The device is designed to handle the low-speed data communication in vehicles, for example, in convenience electronics. Improved slope control at the LIN bus ensures data communication up to 20 kbaud. Sleep mode ensures minimal current consumption even in the case of a floating bus line or a short circuit on the LIN bus to LIN_VSS.

The primary features of the LIN Transceiver module are:

- Supply Voltage Up to 40V
- Operating Voltage VS = 5V to 28V
- Very Low Supply Current
  - Sleep mode: Typically 9 µA
  - Fail-Safe mode: Typically 80 µA
  - Normal mode: Typically 250 µA
- Fully Compatible with 3.3V and 5V Devices
- LIN Physical Layer According to LIN 2.0, 2.1, 2.2, 2.2A and SAEJ2602-2
- Wake-up Capability via LIN bus (100 µs dominant)
- External Wake-up via LIN_WKIN pin (100 µs low level)
- LIN_INH Output to Control an External Voltage Regulator or to Switch the Master Pull-up
- Wake-up Source Recognition
- LIN_TXD Time-out Timer
- Bus Pin is Overtemperature and Short-Circuit Protected vs. LIN_VSS and Battery
- Advanced EMC and ESD Performance
- Fulfills the OEM *“Hardware Requirements for LIN in Automotive Applications Rev.1.3”*
- Interference and Damage Protection According to ISO7637
- Qualified According to AEC-Q100

FIGURE 17-1: LIN TRANSCEIVER MODULE BLOCK DIAGRAM



Note 1: Pin is connected to a device interconnect; see Table 1-1 for more information.

17.1 Pin Description

17.1.1 Supply Pin (LIN_VDD)

LIN operating voltage is VS = 5V to 28V. Undervoltage detection is implemented to disable transmission if VS falls below typical, 4.5V, thereby avoiding false bus messages. After switching on VS, the IC starts in Fail-Safe mode and the LIN_INH output is switched on.

The supply current in Sleep mode is typically 9 µA.

17.1.2 GROUND PIN (LIN_VSS)

The IC does not affect the LIN bus in the event of LIN_VSS disconnection. It is able to handle a ground shift of up to 11.5% of VS.

17.1.3 BUS PIN (LIN_BUS)

A low-side driver with internal current limitation and thermal shutdown, as well as an internal pull-up resistor according to LIN Specification 2.x, is implemented. The voltage range is from -27V to +40V. This pin exhibits no reverse current from the LIN bus to VS, even in the event of a LIN_VSS shift or V_{BAT} disconnection. The LIN receiver thresholds comply with the LIN protocol specification.

The fall time (from recessive to dominant) and the rise time (from dominant to recessive) are slope controlled.

During a short circuit at LIN_BUS to VBAT, the output limits the output current to I_{BUS_LIM}. Due to the power dissipation, the chip temperature exceeds T_{off} and the LIN_BUS output is switched off. The chip cools down and after a hysteresis of T_{hys}, switches the output on again. LIN_RXD stays on high because LIN_BUS is high.

During a short circuit from LIN_BUS to LIN_VSS, the IC can be switched into Sleep mode, and even in this case, the current consumption is lower than 100 µA. If the short circuit disappears, the IC starts with a remote wake-up.

The reverse current is <2 µA at pin LIN_BUS during loss of V_{BAT}. This is optimal behavior for bus systems where some Slave nodes are supplied from battery or ignition.

17.1.4 INPUT/OUTPUT (LIN_TXD)

In Normal mode, the LIN_TXD pin is the microcontroller interface for controlling the state of the LIN_BUS output. LIN_TXD must be pulled to ground in order to drive the LIN bus low. If LIN_TXD is high, the LIN_BUS output transistor is turned off and the bus is in the Recessive state. If the LIN_TXD pin stays at LIN_VSS level while switching into Normal mode, it must be pulled to a high level longer than 10 µs before the LIN Transceiver can be activated. This feature prevents the bus line from being accidentally driven to a Dominant state after Normal mode has been activated (also in case of a short circuit at LIN_TXD to LIN_VSS). During Fail-Safe mode, this pin is used as an output and signals the Fail-Safe source.

The LIN_TXD pin provides a pull-down resistor in order to have a defined level if LIN_TXD is disconnected.

An internal timer prevents the bus line from being driven permanently in the Dominant state. If LIN_TXD is forced to low longer than t_{dom} > 20 ms, the LIN bus driver is switched to the Recessive state. Nevertheless, when switching to Sleep mode, the actual level at the LIN_TXD pin is relevant.

To reactivate the LIN bus driver, switch LIN_TXD to high (>10 µs).

17.1.5 OUTPUT PIN (LIN_RXD)

In Normal mode, this pin reports the state of the LIN bus to the microcontroller. LIN high (Recessive state) is indicated by a high level at LIN_RXD; LIN low (Dominant state) is indicated by a low level at LIN_RXD.

The output is an open-drain; therefore, it is compatible with a 3.3V or 5V power supply. The AC characteristics are defined by an external pull-up resistor of 4.7 kOhm to 5V and a load capacitor of 20 pF.

In Unpowered mode, LIN_RXD is switched off.

17.1.6 ENABLE INPUT PIN (LIN_EN)

The enable input pin controls the operating mode of the device. If LIN_EN is high, the circuit is in Normal mode, with transmission paths from LIN_TXD to LIN_BUS and from LIN_BUS to LIN_RXD both active.

If LIN_EN is switched to low while LIN_TXD is still high, the device is forced to Sleep mode. No data transmission is then possible and current consumption is reduced to $I_{VSsleep}$, typically 9 µA.

The LIN_EN pin provides a pull-down resistor to force the transceiver into Recessive mode if LIN_EN is disconnected.

17.1.7 INHIBIT OUTPUT PIN (LIN_INH)

This pin is used to control an external voltage regulator or to switch the LIN Master pull-up resistor on/off in case the device is used in a Master node. The inhibit pin provides an internal switch toward the LIN_VDD pin, which is protected by temperature monitoring. If the device is in Normal or Fail-Safe mode, the inhibit high-side switch is turned on. When the device is in Sleep mode, the inhibit switch is turned off, thus disabling the voltage regulator or other connected external devices.

A wake-up event on the LIN bus or at the LIN_WKIN pin switches the LIN_INH pin to the VS level. After a system power-up (VS rises from zero), the LIN_INH pin switches to the VS level automatically.

17.1.8 WAKE PIN (LIN_WKIN)

This pin is a high-voltage input used for waking up the device from Sleep mode. It is usually connected to an external switch in the application to generate a local wake-up. A pull-up current source with typically 10 µA is implemented. The voltage threshold for a wake-up signal is typically 2V below the VS voltage.

If a local wake-up is not needed in the application, the LIN_WKIN pin can be connected directly to the LIN_VDD pin.

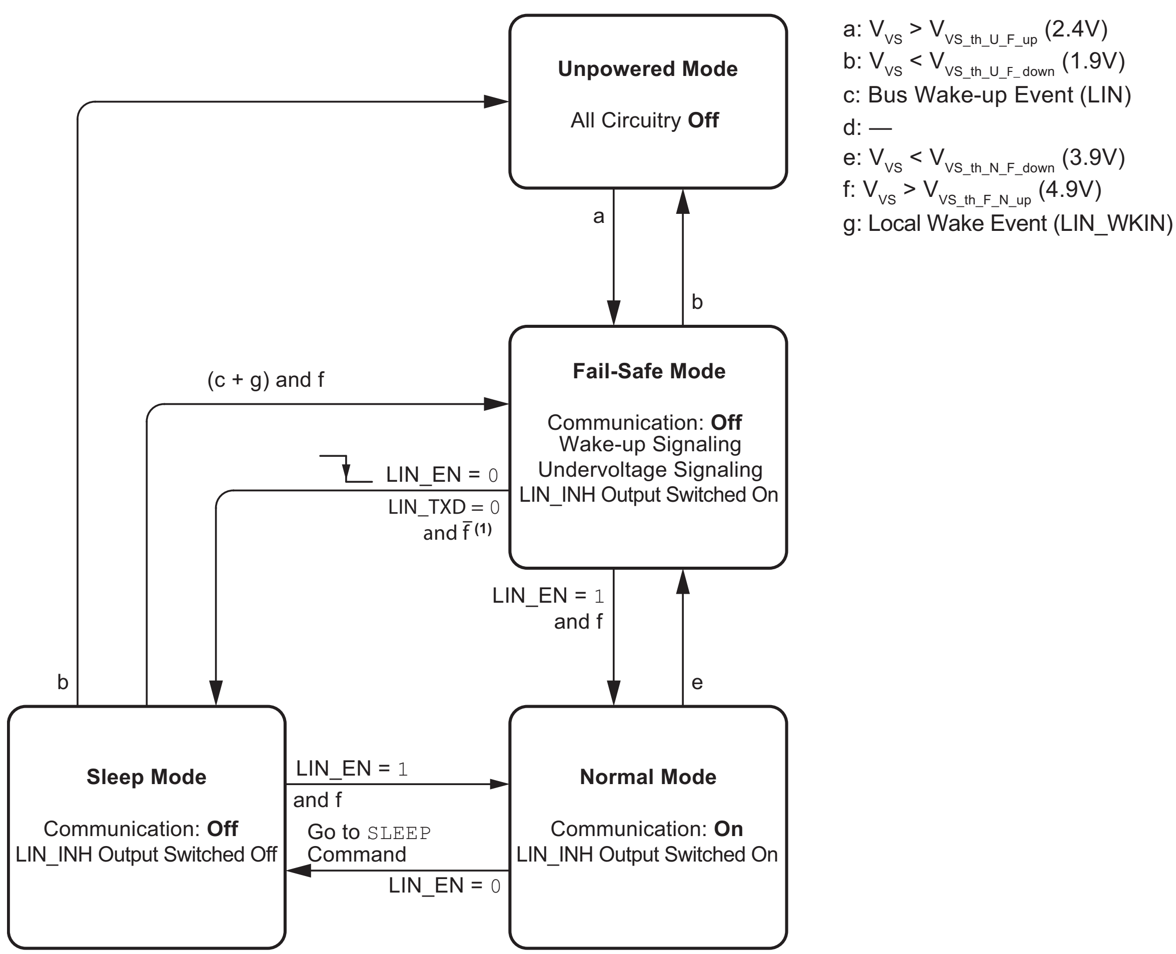
17.2 Functional Description

17.2.1 PHYSICAL LAYER COMPATIBILITY

Since the LIN physical layer is independent of higher LIN layers (for example, LIN protocol layer), all nodes with a LIN physical layer according to Revision 2.x can be mixed with LIN physical layer nodes based on earlier versions (for instance, LIN 1.0, LIN 1.1,LIN 1.2, LIN 1.3) without any restrictions.

17.2.2 OPERATING MODES

FIGURE 17-2: LIN TRANSCEIVER OPERATING MODES



Note 1: Condition f̄ is valid for Vs ramp up; at Vs ramp down condition, e is valid instead of f̄.

TABLE 17-1: OPERATING MODES

Operating Mode	Transceiver	LIN_INH	LIN_BUS	LIN_TXD	LIN_RXD
Fail-Safe	Off	On, except Vs < $V_{VS\_th\_N\_F\_down}$	Recessive	Signaling Fail-Safe Sources (see Table 17-2)	
Normal	On	On	LIN_TXD Dependent	Follows Data Transmission	
Sleep/Unpowered	Off	Off	Recessive	Low	High Ohmic

17.2.3 NORMAL MODE

This is the normal Transmitting and Receiving mode of the LIN interface in accordance with LIN Specification 2.x.

17.2.4 SLEEP MODE

A falling edge at LIN_EN switches the IC into Sleep mode. While in Sleep mode, the transmission path is disabled and the device is in Low-Power mode. Supply current from V_{BAT} is typically 9 µA. In Sleep mode, the LIN_INH pin is switched off. The internal termination between the LIN_BUS pin and LIN_VDD pin is disabled. Only a weak pull-up current (typically 10 µA) between the LIN_BUS pin and LIN_VDD pin is present. Sleep mode can be activated independently from the actual level on the LIN_BUS or LIN_WKIN pin.

If the LIN_TXD pin is short-circuited to LIN_VSS, it is possible to switch to Sleep mode via LIN_EN after $t > t_{dom}$.

17.2.5 FAIL-SAFE MODE

The device automatically switches to Fail-Safe mode at system power-up or after a wake-up event. The LIN_INH output is switched on and the LIN transceiver is switched off. The IC stays in this mode until LIN_EN is switched to high. The IC then changes to Normal mode. During Fail-Safe mode, the LIN_TXD pin is an output, and together with the LIN_RXD output pin, signals the fail-safe source.

If the device enters Fail-Safe mode coming from the Normal mode (LIN_EN = 1) due to a VS undervoltage condition ($V_{VS} < V_{VS\_th\_N\_F\_down}$), it is possible to switch into Sleep mode by a falling edge at the LIN_EN input. With this feature, the current consumption is further reduced.

A wake-up event from Sleep mode is signaled to the microcontroller using the LIN_RXD pin and the LIN_TXD pin. A VS undervoltage condition is also signaled at these two pins. The coding is shown in Table 17-2.

TABLE 17-2: SIGNALING IN FAIL-SAFE MODE

Fail-Safe Sources	LIN_TXD(1)	LIN_RXD
LIN Wake-up (LIN_BUS pin)	Low	Low
Local Wake-up (LIN_WKIN pin)	Low	High
VS_{th} (battery) Undervoltage Detection, VS < 3.9V	High	Low

Note 1: Assuming an external pull-up resistor (typically 5 kΩ) has been added on pin LIN_TXD to the power supply of the microcontroller.

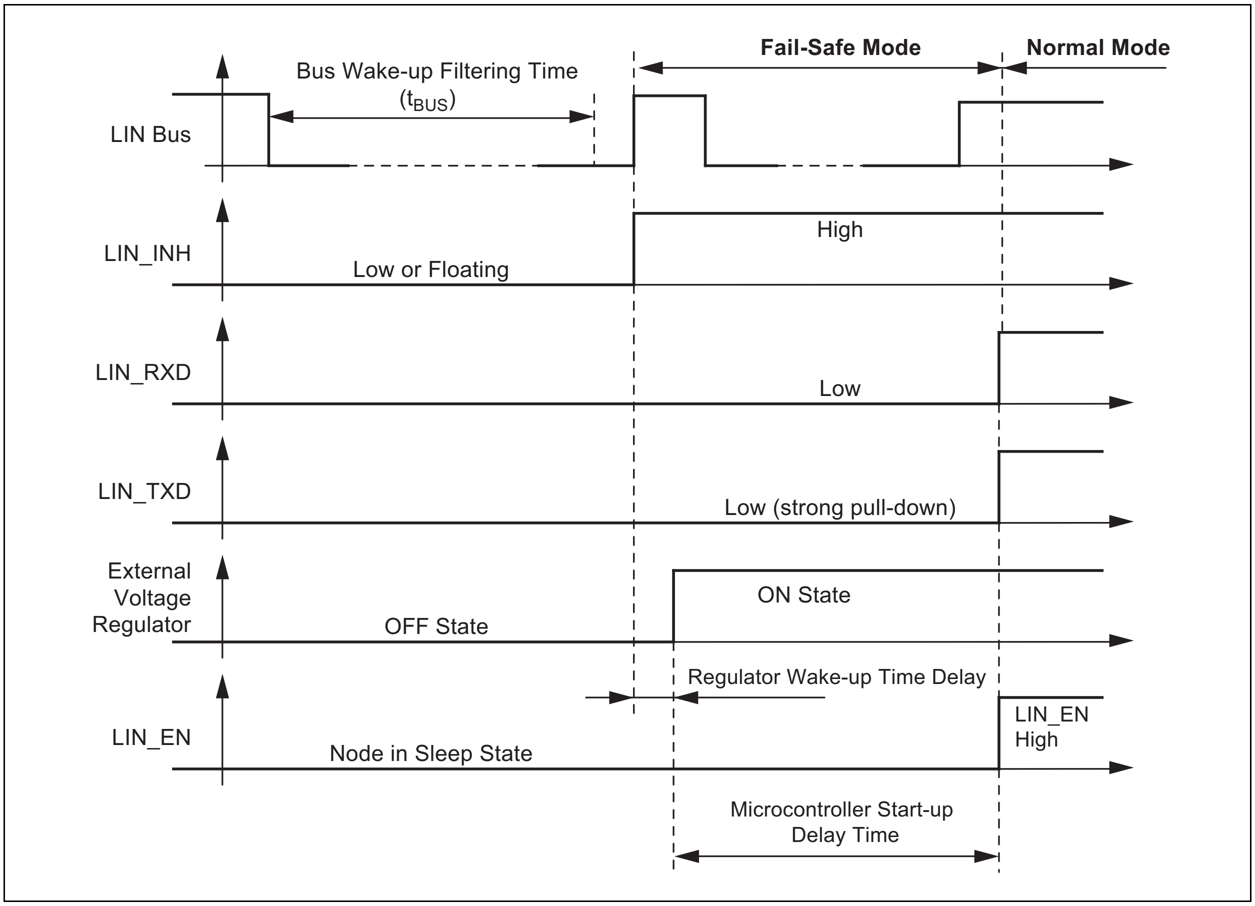
17.3 Wake-up Scenarios from Sleep Mode

17.3.1 REMOTE WAKE-UP VIA LIN_BUS

17.3.1.1 Remote Wake-up from Sleep Mode

A voltage lower than the LIN pre-wake detection, V_{LINL} at the LIN_BUS pin, activates the internal LIN receiver and starts the wake-up detection timer. A falling edge at the LIN_BUS pin, followed by a dominant bus level maintained for a certain period of time (>t_{bus}), and following a rising edge at the LIN_BUS pin, results in a remote wake-up request and the device switches to Fail-Safe mode. The LIN_INH pin is activated (switches to V_S) and the internal termination resistor is switched on. The remote wake-up request is indicated by a low level at pin LIN_RXD and interrupts the microcontroller.

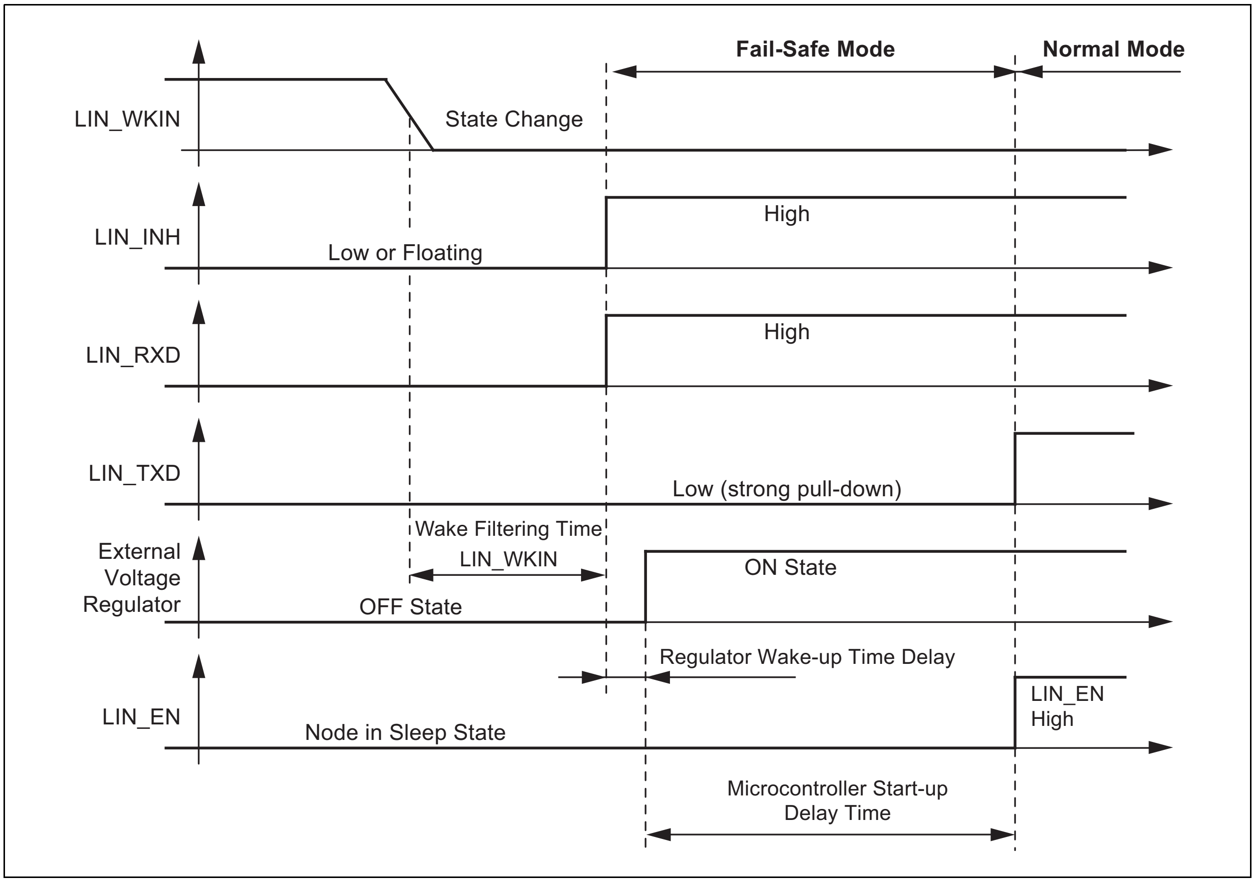
FIGURE 17-3: LIN WAKE-UP FROM SLEEP MODE



17.3.2 LOCAL WAKE-UP THROUGH LIN_WKIN PIN

A falling edge at the LIN_WKIN pin followed by a low level maintained for a certain period of time (>t_{WKin}) results in a local wake-up request and the device switches to Fail-Safe mode. The LIN_INH pin is activated (switches to VS) and the internal Slave termination resistor is switched on. The local wake-up request is indicated by a low level at the LIN_TXD pin and a high level at the LIN_RXD pin, generating an interrupt for the microcontroller. Even when the LIN_WKIN pin is low, it is possible to switch to Sleep mode via the LIN_EN pin. In this case, the wake-up signal has to be switched to high >10 µs before the negative edge at LIN_WKIN starts a new local wake-up request.

FIGURE 17-4: LOCAL WAKE-UP FROM WAKE-UP SWITCH



17.3.3 WAKE-UP SOURCE RECOGNITION

The device can distinguish between different wake-up sources. The wake-up source can be read on the LIN_TXD and LIN_RXD pin in Fail-Safe mode, according to Table 17-3, if an external pull-up resistor (typically 5 kΩ) has been added on pin LIN_TXD to the power supply of the microcontroller. These flags are reset immediately if the microcontroller sets pin LIN_EN to high and the IC is in Normal mode.

TABLE 17-3: SIGNALING IN FAIL-SAFE MODE

Fail-Safe Sources	LIN_TXD(1)	LIN_RXD
LIN Wake-up (LIN_BUS pin)	Low	Low
Local Wake-up (LIN_WKIN pin)	Low	High
$V_{S_{th}}$ (battery) Undervoltage Detection (VS < 3.9V)	High	Low

Note 1: Assuming an external pull-up resistor (typically 5 kΩ) has been added on pin LIN_TXD to the power supply of the microcontroller.

17.4 Behavior Under Low Supply Voltage Condition

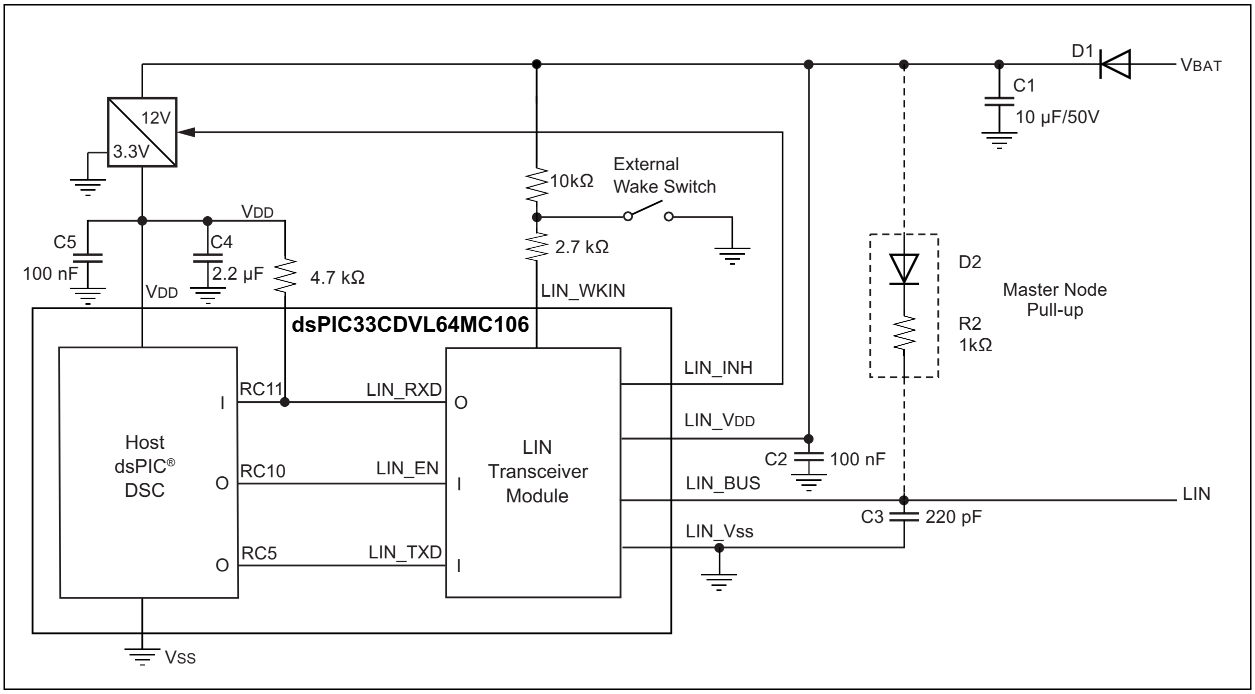
After the battery voltage has been connected to the application circuit, the voltage at the LIN_VDD pin increases according to the block capacitor used in the application (see Figure 17-1). If V_{VS} is higher than the minimum VS operation threshold, $V_{VS\_th\_U\_F\_up}$, the IC mode changes from Unpowered mode to Fail-Safe mode, the LIN_INH output is switched on and the LIN transceiver can be activated.

If during Sleep mode, the voltage level of V_{VS} drops below the undervoltage detection threshold, $V_{VS\_th\_N\_F\_down}$ (typically 4.3V), the operation mode is not changed and no wake-up is possible. Only if the supply voltage on the LIN_VDD pin drops below the VS operation threshold, $V_{VS\_th\_U\_F\_down}$ (typically 2.05V), does the IC switch to Unpowered mode.

If, during Normal mode, the voltage level on the LIN_VDD pin drops below the VS undervoltage detection threshold, $V_{VS\_th\_N\_F\_down}$ (typically 4.3V), the IC switches to Fail-Safe mode. This means that the LIN transceiver is disabled in order to avoid malfunctions or false bus messages. If the supply voltage, V_{VS}, drops further below the VS operation threshold, $V_{VS\_th\_U\_F\_down}$ (typically 2.05V), the IC switches to Unpowered mode and the LIN_INH output switches off.

17.5 Application Circuit

FIGURE 17-5: TYPICAL APPLICATION CIRCUIT



NOTES:

18.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note 1:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Serial Peripheral Interface (SPI) with Audio Codec Support"** (www.microchip.com/DS70005136).

The Serial Peripheral Interface (SPI) module is a synchronous serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with the Motorola® SPI and SIOP interfaces. The dsPIC33CDVL64MC106 device includes three SPI modules. On 48-pin devices, SPI instance of SPI2 can operate at higher speeds when selected as a non-PPS pin. The selection is done using the SPI2PIN bit (FDEVOPT[13]). If the bit for SPI2PIN is '1', the PPS pin will be used. When SPI2PIN is '0', the SPI signals are routed to dedicated pins.

The module supports operation in two Buffer modes. In Standard mode, data are shifted through a single serial buffer. In Enhanced Buffer mode, data are shifted through a FIFO buffer. The FIFO level depends on the configured mode.

Note:	FIFO depth for this device is four (in 8-Bit Data mode).

Variable length data can be transmitted and received, from 2 to 32 bits.

Note:	Do not perform Read-Modify-Write operations (such as bit-oriented instructions) on the SPIxBUF register in either Standard or Enhanced Buffer mode.

The module also supports a basic framed SPI protocol while operating in either Master or Slave mode. A total of four framed SPI configurations are supported.

The module also supports Audio modes. Four different Audio modes are available.

- I^{2}S Mode
- Left Justified Mode
- Right Justified Mode
- PCM/DSP Mode

In each of these modes, the serial clock is free-running and audio data are always transferred.

If an audio protocol data transfer takes place between two devices, then usually one device is the Master and the other is the Slave. However, audio data can be transferred between two Slaves. Because the audio protocols require free-running clocks, the Master can be a third-party controller. In either case, the Master generates two free-running clocks: SCKx and LRC (Left, Right Channel Clock/$\overline{\text{SSx}}$/FSYNC).

The SPI serial interface consists of four pins:

- SDIx: Serial Data Input
- SDOx: Serial Data Output
- SCKx: Shift Clock Input or Output
- $\overline{\text{SSx}}$: Active-Low Slave Select or Frame Synchronization I/O Pulse

The SPI module can be configured to operate using two, three or four pins. In the 3-pin mode, $\overline{\text{SSx}}$ is not used. In the 2-pin mode, both SDOx and $\overline{\text{SSx}}$ are not used.

The SPI module has the ability to generate three interrupts reflecting the events that occur during the data communication. The following types of interrupts can be generated:

1. Receive interrupts are signalled by SPIxRXIF. This event occurs when:
   - RX watermark interrupt
   - SPIROV = 1
   - SPIRBF = 1
   - SPIRBE = 1

   provided the respective mask bits are enabled in SPIxIMSKL/H.
2. Transmit interrupts are signalled by SPIxTXIF. This event occurs when:
   - TX watermark interrupt
   - SPITUR = 1
   - SPITBF = 1
   - SPITBE = 1

   provided the respective mask bits are enabled in SPIxIMSKL/H.
3. General interrupts are signalled by SPIxGIF. This event occurs when:
   - FRMERR = 1
   - SPIBUSY = 1
   - SRMT = 1

   provided the respective mask bits are enabled in SPIxIMSKL/H.

Block diagrams of the module in Standard and Enhanced modes are shown in Figure 18-1 and Figure 18-2.

Note:	In this section, the SPI modules are referred to together as SPIx, or separately as SPI1, SPI2 or SPI3. Special Function Registers will follow a similar notation. For example, SPIxCON1 and SPIxCON2 refer to the control registers for any of the three SPI modules.

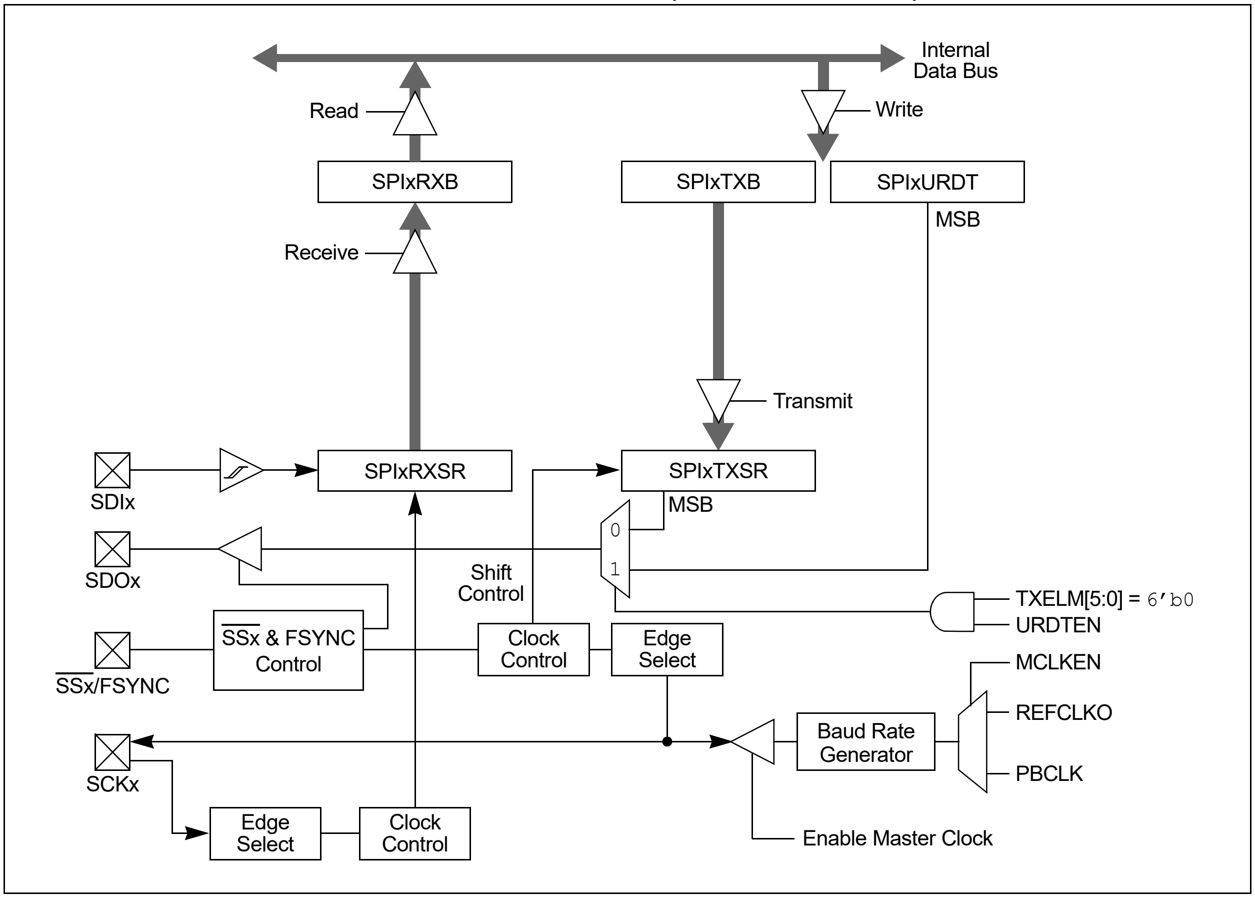
To set up the SPIx module for the Standard Master mode of operation:

1. If using interrupts:
   a) Clear the interrupt flag bits in the respective IFSx register.
   b) Set the interrupt enable bits in the respective IECx register.
   c) Write the SPIxIP bits in the respective IPCx register to set the interrupt priority.
2. Write the desired settings to the SPIxCON1L and SPIxCON1H registers with the MSTEN bit (SPIxCON1L[5]) = 1.
3. Clear the SPIROV bit (SPIxSTATL[6]).
4. Enable SPIx operation by setting the SPIEN bit (SPIxCON1L[15]).
5. Write the data to be transmitted to the SPIxBUFL and SPIxBUFH registers. Transmission (and reception) will start as soon as data are written to the SPIxBUFL and SPIxBUFH registers.

To set up the SPIx module for the Standard Slave mode of operation:

1. Clear the SPIxBUF registers.
2. If using interrupts:
   a) Clear the SPIxBUFL and SPIxBUFH registers.
   b) Set the interrupt enable bits in the respective IECx register.
   c) Write the SPIxIP bits in the respective IPCx register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1L, SPIxCON1H and SPIxCON2L registers with the MSTEN bit (SPIxCON1L[5]) = 0.
4. Clear the SMP bit.
5. If the CKE bit (SPIxCON1L[8]) is set, then the $\overline{\text{SSEN}}$ bit (SPIxCON1L[7]) must be set to enable the $\overline{\text{SSx}}$ pin.
6. Clear the SPIROV bit (SPIxSTATL[6]).
7. Enable SPIx operation by setting the SPIEN bit (SPIxCON1L[15]).

FIGURE 18-1: SPIx MODULE BLOCK DIAGRAM (STANDARD MODE)



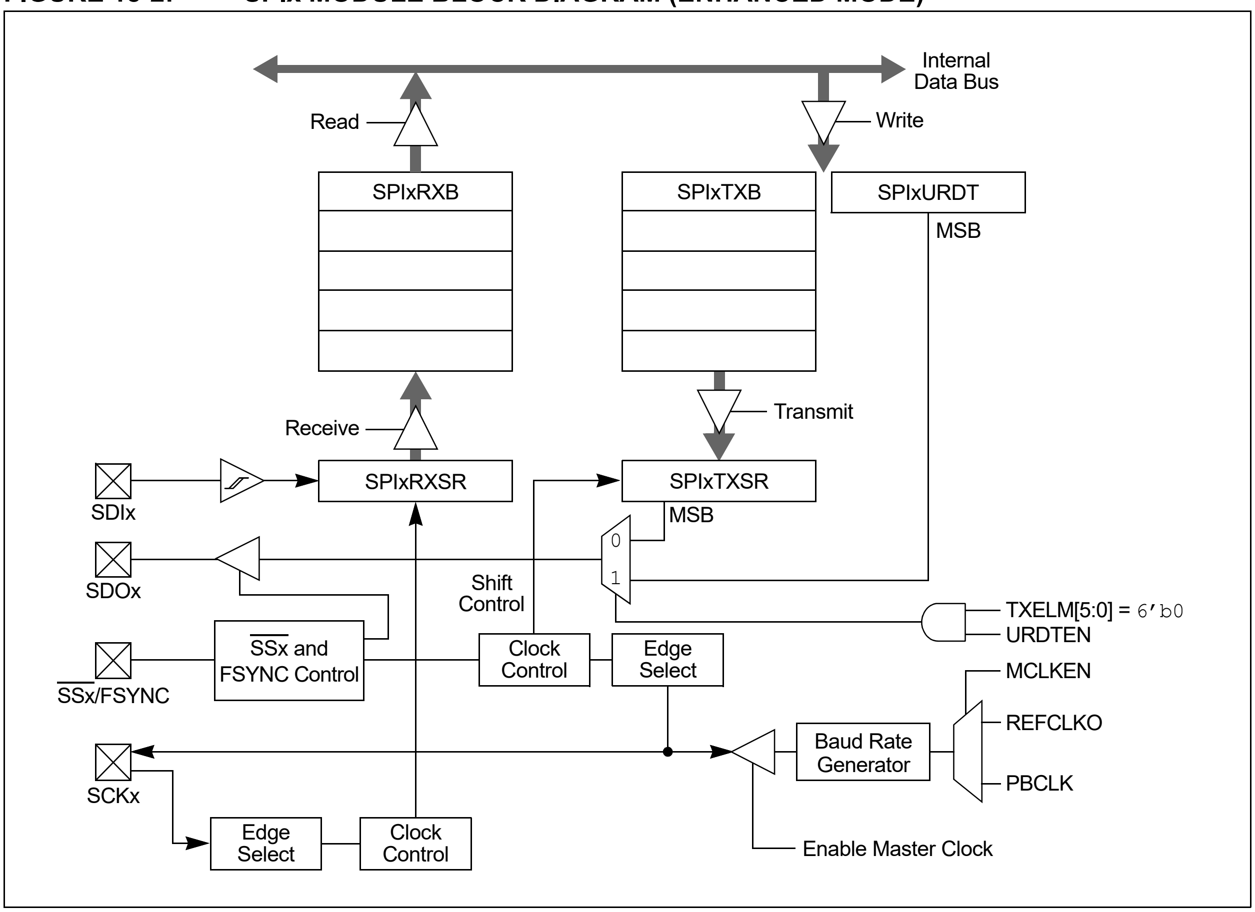
To set up the SPIx module for the Enhanced Buffer Master mode of operation:

1. If using interrupts:
   a) Clear the interrupt flag bits in the respective IFSx register.
   b) Set the interrupt enable bits in the respective IECx register.
   c) Write the SPIxIP bits in the respective IPCx register.
2. Write the desired settings to the SPIxCON1L, SPIxCON1H and SPIxCON2L registers with MSTEN (SPIxCON1L[5]) = `1`.
3. Clear the SPIROV bit (SPIxSTATL[6]).
4. Select Enhanced Buffer mode by setting the ENHBUF bit (SPIxCON1L[0]).
5. Enable SPIx operation by setting the SPIEN bit (SPIxCON1L[15]).
6. Write the data to be transmitted to the SPIxBUFL and SPIxBUFH registers. Transmission (and reception) will start as soon as data are written to the SPIxBUFL and SPIxBUFH registers.

To set up the SPIx module for the Enhanced Buffer Slave mode of operation:

1. Clear the SPIxBUFL and SPIxBUFH registers.
2. If using interrupts:
   a) Clear the interrupt flag bits in the respective IFSx register.
   b) Set the interrupt enable bits in the respective IECx register.
   c) Write the SPIxIP bits in the respective IPCx register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1L, SPIxCON1H and SPIxCON2L registers with the MSTEN bit (SPIxCON1L[5]) = `0`.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit must be set, thus enabling the $\overline{\text{SSx}}$ pin.
6. Clear the SPIROV bit (SPIxSTATL[6]).
7. Select Enhanced Buffer mode by setting the ENHBUF bit (SPIxCON1L[0]).
8. Enable SPIx operation by setting the SPIEN bit (SPIxCON1L[15]).

FIGURE 18-2: SPIx MODULE BLOCK DIAGRAM (ENHANCED MODE)



To set up the SPIx module for Audio mode:

1. Clear the SPIxBUFL and SPIxBUFH registers.
2. If using interrupts:
   a) Clear the interrupt flag bits in the respective IFSx register.
   b) Set the interrupt enable bits in the respective IECx register.
   a) Write the SPIxIP bits in the respective IPCx register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1L, SPIxCON1H and SPIxCON2L registers with AUDEN (SPIxCON1H[15]) = `1`.
4. Clear the SPIROV bit (SPIxSTATL[6]).
5. Enable SPIx operation by setting the SPIEN bit (SPIxCON1L[15]).
6. Write the data to be transmitted to the SPIxBUFL and SPIxBUFH registers. Transmission (and reception) will start as soon as data are written to the SPIxBUFL and SPIxBUFH registers.

Note:	After start-up, when configured for Slave mode, left justified for all possible configurations of MODE[32,16] and in right justified for MODE[32,16] = {`0b00`,`0b10`}, the SPI drives ones out of SDOx if the MSb bit of the first data is a one.

18.1 SPI Control/Status Registers

REGISTER 18-1: SPIxCON1L: SPIx CONTROL REGISTER 1 LOW

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	DISSDO	MODE32[1,4]	MODE16[1,4]	SMP	CKE[1]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN[2]	CKP	MSTEN	DISSDI	DISSCK	MCLKEN[3]	SPIFE	ENHBUF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **SPIEN:** SPIx On bit
1 = Enables module
0 = Turns off and resets module, disables clocks, disables interrupt event generation, allows SFR modifications

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **SPISIDL:** SPIx Stop in Idle Mode bit
1 = Halts in CPU Idle mode
0 = Continues to operate in CPU Idle mode

bit 12 **DISSDO:** Disable SDOx Output Port bit
1 = SDOx pin is not used by the module; pin is controlled by port function
0 = SDOx pin is controlled by the module

bit 11-10 **MODE32** and **MODE16:** Serial Word Length Select bits[1,4]

MODE32	MODE16	AUDEN	Communication
1	x	0	32-Bit
0	1		16-Bit
0	0		8-Bit
1	1	1	24-Bit Data, 32-Bit FIFO, 32-Bit Channel/64-Bit Frame
1	0		32-Bit Data, 32-Bit FIFO, 32-Bit Channel/64-Bit Frame
0	1		16-Bit Data, 16-Bit FIFO, 32-Bit Channel/64-Bit Frame
0	0		16-Bit FIFO, 16-Bit Channel/32-Bit Frame

bit 9 **SMP:** SPIx Data Input Sample Phase bit
Master Mode:
1 = Input data are sampled at the end of data output time
0 = Input data are sampled at the middle of data output time
Slave Mode:
Input data are always sampled at the middle of data output time, regardless of the SMP setting.

bit 8 **CKE:** SPIx Clock Edge Select bit[1]
1 = Transmit happens on transition from Active Clock state to Idle Clock state
0 = Transmit happens on transition from Idle Clock state to Active Clock state

Note 1: When AUDEN (SPIxCON1H[15]) = 1, this module functions as if CKE = 0, regardless of its actual value.
2: When FRMEN = 1, SSEN is not used.
3: MCLKEN can only be written when the SPIEN bit = 0.
4: This channel is not meaningful for DSP/PCM mode as LRC follows FRMSYPW.

REGISTER 18-1: SPIxCON1L: SPIx CONTROL REGISTER 1 LOW (CONTINUED)

bit 7 **SSEN:** Slave Select Enable bit (Slave mode)(2)
1 = $\overline{SSx}$ pin is used by the macro in Slave mode; $\overline{SSx}$ pin is used as the Slave select input
0 = $\overline{SSx}$ pin is not used by the macro ($\overline{SSx}$ pin will be controlled by the port I/O)

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; Active state is a low level
0 = Idle state for clock is a low level; Active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

bit 4 **DISSDI:** Disable SDIx Input Port bit
1 = SDIx pin is not used by the module; pin is controlled by port function
0 = SDIx pin is controlled by the module

bit 3 **DISSCK:** Disable SCKx Output Port bit
1 = SCKx pin is not used by the module; pin is controlled by port function
0 = SCKx pin is controlled by the module

bit 2 **MCLKEN:** Master Clock Enable bit(3)
1 = Reference Clock (REFCLKO) is used by the BRG
0 = Peripheral Clock (F_P = F_{OSC}/2) is used by the BRG

bit 1 **SPIFE:** Frame Sync Pulse Edge Select bit
1 = Frame Sync pulse (Idle-to-active edge) coincides with the first bit clock
0 = Frame Sync pulse (Idle-to-active edge) precedes the first bit clock

bit 0 **ENHBUF:** Enhanced Buffer Enable bit
1 = Enhanced Buffer mode is enabled
0 = Enhanced Buffer mode is disabled

Note 1: When AUDEN (SPIxCON1H[15]) = 1, this module functions as if CKE = 0, regardless of its actual value.
2: When FRMEN = 1, SSEN is not used.
3: MCLKEN can only be written when the SPIEN bit = 0.
4: This channel is not meaningful for DSP/PCM mode as LRC follows FRMSYPW.

REGISTER 18-2: SPIxCON1H: SPIx CONTROL REGISTER 1 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AUDEN(1)	SPISGNEXT	IGNROV	IGNTUR	AUDMONO(2)	URDTEN(3)	AUDMOD1(4)	AUDMOD0(4)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FRMEN	FRMSYNC	FRMPOL	MSSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **AUDEN:** Audio Codec Support Enable bit(1)
1 = Audio protocol is enabled; MSTEN controls the direction of both SCKx and frame (a.k.a. LRC), and this module functions as if FRMEN = 1, FRMSYNC = MSTEN, FRMCNT[2:0] = 001 and SMP = 0, regardless of their actual values
0 = Audio protocol is disabled

bit 14 **SPISGNEXT:** SPIx Sign-Extend RX FIFO Read Data Enable bit
1 = Data from RX FIFO are sign-extended
0 = Data from RX FIFO are not sign-extended

bit 13 **IGNROV:** Ignore Receive Overflow bit
1 = A Receive Overflow (ROV) is NOT a critical error; during ROV, data in the FIFO are not overwritten by the receive data
0 = A ROV is a critical error that stops SPI operation

bit 12 **IGNTUR:** Ignore Transmit Underrun bit
1 = A Transmit Underrun (TUR) is NOT a critical error and data indicated by URDTEN are transmitted until the SPIxTXB is not empty
0 = A TUR is a critical error that stops SPI operation

bit 11 **AUDMONO:** Audio Data Format Transmit bit(2)
1 = Audio data are mono (i.e., each data word is transmitted on both left and right channels)
0 = Audio data are stereo

bit 10 **URDTEN:** Transmit Underrun Data Enable bit(3)
1 = Transmits data out of SPIxURDT register during Transmit Underrun conditions
0 = Transmits the last received data during Transmit Underrun conditions

bit 9-8 **AUDMOD[1:0]:** Audio Protocol Mode Selection bits(4)
11 = PCM/DSP mode
10 = Right Justified mode: This module functions as if SPIFE = 1, regardless of its actual value
01 = Left Justified mode: This module functions as if SPIFE = 1, regardless of its actual value
00 = I^2S mode: This module functions as if SPIFE = 0, regardless of its actual value

bit 7 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support is enabled ($\overline{SSx}$ pin is used as the FSYNC input/output)
0 = Framed SPIx support is disabled

Note 1: AUDEN can only be written when the SPIEN bit = 0.
2: AUDMONO can only be written when the SPIEN bit = 0 and is only valid for AUDEN = 1.
3: URDTEN is only valid when IGNTUR = 1.
4: AUDMOD[1:0] can only be written when the SPIEN bit = 0 and is only valid when AUDEN = 1. When NOT in PCM/DSP mode, this module functions as if FRMSYPW = 1, regardless of its actual value.

REGISTER 18-2: SPIxCON1H: SPIx CONTROL REGISTER 1 HIGH (CONTINUED)

bit 6 **FRMSYNC:** Frame Sync Pulse Direction Control bit
1 = Frame Sync pulse input (Slave)
0 = Frame Sync pulse output (Master)

bit 5 **FRMPOL:** Frame Sync/Slave Select Polarity bit
1 = Frame Sync pulse/Slave select is active-high
0 = Frame Sync pulse/Slave select is active-low

bit 4 **MSSEN:** Master Mode Slave Select Enable bit
1 = SPIx Slave select support is enabled with polarity determined by FRMPOL ($\overline{SSx}$ pin is automatically driven during transmission in Master mode)
0 = Slave select SPIx support is disabled ($\overline{SSx}$ pin will be controlled by port I/O)

bit 3 **FRMSYPW:** Frame Sync Pulse-Width bit
1 = Frame Sync pulse is one serial word length wide (as defined by MODE[32,16]/WLENGTH[4:0])
0 = Frame Sync pulse is one clock (SCKx) wide

bit 2-0 **FRMCNT[2:0]:** Frame Sync Pulse Counter bits
Controls the number of serial words transmitted per Sync pulse.
111 = Reserved
110 = Reserved
101 = Generates a Frame Sync pulse on every 32 serial words
100 = Generates a Frame Sync pulse on every 16 serial words
011 = Generates a Frame Sync pulse on every 8 serial words
010 = Generates a Frame Sync pulse on every 4 serial words
001 = Generates a Frame Sync pulse on every 2 serial words (value used by audio protocols)
000 = Generates a Frame Sync pulse on each serial word

Note 1: AUDEN can only be written when the SPIEN bit = 0.
2: AUDMONO can only be written when the SPIEN bit = 0 and is only valid for AUDEN = 1.
3: URDTEN is only valid when IGNTUR = 1.
4: AUDMOD[1:0] can only be written when the SPIEN bit = 0 and is only valid when AUDEN = 1. When NOT in PCM/DSP mode, this module functions as if FRMSYPW = 1, regardless of its actual value.

REGISTER 18-3: SPIxCON2L: SPIx CONTROL REGISTER 2 LOW

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	WLENGTH[4:0][1,2]				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-5 **Unimplemented:** Read as '0'

bit 4-0 **WLENGTH[4:0]:** Variable Word Length bits[1,2]
11111 = 32-bit data
11110 = 31-bit data
11101 = 30-bit data
11100 = 29-bit data
11011 = 28-bit data
11010 = 27-bit data
11001 = 26-bit data
11000 = 25-bit data
10111 = 24-bit data
10110 = 23-bit data
10101 = 22-bit data
10100 = 21-bit data
10011 = 20-bit data
10010 = 19-bit data
10001 = 18-bit data
10000 = 17-bit data
01111 = 16-bit data
01110 = 15-bit data
01101 = 14-bit data
01100 = 13-bit data
01011 = 12-bit data
01010 = 11-bit data
01001 = 10-bit data
01000 = 9-bit data
00111 = 8-bit data
00110 = 7-bit data
00101 = 6-bit data
00100 = 5-bit data
00011 = 4-bit data
00010 = 3-bit data
00001 = 2-bit data
00000 = See MODE[32,16] bits in SPIxCON1L[11:10]

Note 1: These bits are effective when AUDEN = 0 only.
2: Varying the length by changing these bits does not affect the depth of the TX/RX FIFO.

REGISTER 18-4: SPIxSTATL: SPIx STATUS REGISTER LOW

U-0	U-0	U-0	HS/R/C-0	HSC/R-0	U-0	U-0	HSC/R-0
—	—	—	FRMERR	SPIBUSY	—	—	SPITUR[(1)]
bit 15							bit 8

HSC/R-0	HS/R/C-0	HSC/R-1	U-0	HSC/R-1	U-0	HSC/R-0	HSC/R-0
SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF
bit 7							bit 0

Legend:	C = Clearable bit	U = Unimplemented, read as '0'	
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	HS = Hardware Settable bit

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **FRMERR**: SPIx Frame Error Status bit
1 = Frame error is detected
0 = No frame error is detected

bit 11 **SPIBUSY:** SPIx Activity Status bit
1 = Module is currently busy with some transactions
0 = No ongoing transactions (at time of read)

bit 10-9 **Unimplemented:** Read as '0'

bit 8 **SPITUR:** SPIx Transmit Underrun Status bit[(1)]
1 = Transmit buffer has encountered a Transmit Underrun condition
0 = Transmit buffer does not have a Transmit Underrun condition

bit 7 **SRMT:** Shift Register Empty Status bit
1 = No current or pending transactions (i.e., neither SPIxTXB or SPIxTXSR contains data to transmit)
0 = Current or pending transactions

bit 6 **SPIROV:** SPIx Receive Overflow Status bit
1 = A new byte/half-word/word has been completely received when the SPIxRXB was full
0 = No overflow

bit 5 **SPIRBE:** SPIx RX Buffer Empty Status bit
1 = RX buffer is empty
0 = RX buffer is not empty
Standard Buffer Mode:
Automatically set in hardware when SPIxBUF is read from, reading SPIxRXB. Automatically cleared in hardware when SPIx transfers data from SPIxRXSR to SPIxRXB.
Enhanced Buffer Mode:
Indicates RXELM[5:0] = 000000.

bit 4 **Unimplemented:** Read as '0'

Note 1: SPITUR is cleared when SPIEN = 0. When IGNTUR = 1, SPITUR provides dynamic status of the Transmit Underrun condition, but does not stop RX/TX operation and does not need to be cleared by software.

REGISTER 18-4: SPIxSTATL: SPIx STATUS REGISTER LOW (CONTINUED)

bit 3	**SPITBE:** SPIx Transmit Buffer Empty Status bit 1 = SPIxTXB is empty 0 = SPIxTXB is not empty <u>Standard Buffer Mode:</u> Automatically set in hardware when SPIx transfers data from SPIxTXB to SPIxTXSR. Automatically cleared in hardware when SPIxBUF is written, loading SPIxTXB. <u>Enhanced Buffer Mode:</u> Indicates TXELM[5:0] = 000000.
bit 2	**Unimplemented:** Read as '0'
bit 1	**SPITBF:** SPIx Transmit Buffer Full Status bit 1 = SPIxTXB is full 0 = SPIxTXB not full <u>Standard Buffer Mode:</u> Automatically set in hardware when SPIxBUF is written, loading SPIxTXB. Automatically cleared in hardware when SPIx transfers data from SPIxTXB to SPIxTXSR. <u>Enhanced Buffer Mode:</u> Indicates TXELM[5:0] = 111111.
bit 0	**SPIRBF:** SPIx Receive Buffer Full Status bit 1 = SPIxRXB is full 0 = SPIxRXB is not full <u>Standard Buffer Mode:</u> Automatically set in hardware when SPIx transfers data from SPIxRXSR to SPIxRXB. Automatically cleared in hardware when SPIxBUF is read from, reading SPIxRXB. <u>Enhanced Buffer Mode:</u> Indicates RXELM[5:0] = 111111.

Note 1: SPITUR is cleared when SPIEN = 0. When IGNTUR = 1, SPITUR provides dynamic status of the Transmit Underrun condition, but does not stop RX/TX operation and does not need to be cleared by software.

REGISTER 18-5: SPIxSTATH: SPIx STATUS REGISTER HIGH

U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	RXELM5[3]	RXELM4[2]	RXELM3[1]	RXELM2	RXELM1	RXELM0
bit 15							bit 8

U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	TXELM5[3]	TXELM4[2]	TXELM3[1]	TXELM2	TXELM1	TXELM0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RXELM[5:0]:** Receive Buffer Element Count bits (valid in Enhanced Buffer mode)[1,2,3]

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **TXELM[5:0]:** Transmit Buffer Element Count bits (valid in Enhanced Buffer mode)[1,2,3]

Note 1: RXELM3 and TXELM3 bits are only present when FIFODEPTH = 8 or higher.

2: RXELM4 and TXELM4 bits are only present when FIFODEPTH = 16 or higher.

3: RXELM5 and TXELM5 bits are only present when FIFODEPTH = 32.

REGISTER 18-6: SPIxIMSKL: SPIx INTERRUPT MASK REGISTER LOW

U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **FRMERREN:** Enable Interrupt Events via FRMERR bit
1 = Frame error generates an interrupt event
0 = Frame error does not generate an interrupt event

bit 11 **BUSYEN:** Enable Interrupt Events via SPIBUSY bit
1 = SPIBUSY generates an interrupt event
0 = SPIBUSY does not generate an interrupt event

bit 10-9 **Unimplemented:** Read as '0'

bit 8 **SPITUREN:** Enable Interrupt Events via SPITUR bit
1 = Transmit Underrun (TUR) generates an interrupt event
0 = Transmit Underrun does not generate an interrupt event

bit 7 **SRMTEN:** Enable Interrupt Events via SRMT bit
1 = Shift Register Empty (SRMT) generates interrupt events
0 = Shift Register Empty does not generate interrupt events

bit 6 **SPIROVEN:** Enable Interrupt Events via SPIROV bit
1 = SPIx Receive Overflow (ROV) generates an interrupt event
0 = SPIx Receive Overflow does not generate an interrupt event

bit 5 **SPIRBEN:** Enable Interrupt Events via SPIRBE bit
1 = SPIx RX buffer empty generates an interrupt event
0 = SPIx RX buffer empty does not generate an interrupt event

bit 4 **Unimplemented:** Read as '0'

bit 3 **SPITBEN:** Enable Interrupt Events via SPITBE bit
1 = SPIx transmit buffer empty generates an interrupt event
0 = SPIx transmit buffer empty does not generate an interrupt event

bit 2 **Unimplemented:** Read as '0'

bit 1 **SPITBFEN:** Enable Interrupt Events via SPITBF bit
1 = SPIx transmit buffer full generates an interrupt event
0 = SPIx transmit buffer full does not generate an interrupt event

bit 0 **SPIRBFEN:** Enable Interrupt Events via SPIRBF bit
1 = SPIx receive buffer full generates an interrupt event
0 = SPIx receive buffer full does not generate an interrupt event

REGISTER 18-7: SPIxIMSKH: SPIx INTERRUPT MASK REGISTER HIGH

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXWIEN	—	RXMSK5[1]	RXMSK4[1,4]	RXMSK3[1,3]	RXMSK2[1,2]	RXMSK1[1]	RXMSK0[1]
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXWIEN	—	TXMSK5[1]	TXMSK4[1,4]	TXMSK3[1,3]	TXMSK2[1,2]	TXMSK1[1]	TXMSK0[1]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **RXWIEN:** Receive Watermark Interrupt Enable bit
1 = Triggers receive buffer element watermark interrupt when RXMSK[5:0] ≤ RXELM[5:0]
0 = Disables receive buffer element watermark interrupt

bit 14 **Unimplemented:** Read as ‘0’

bit 13-8 **RXMSK[5:0]:** RX Buffer Mask bits[1,2,3,4]
RX mask bits; used in conjunction with the RXWIEN bit.

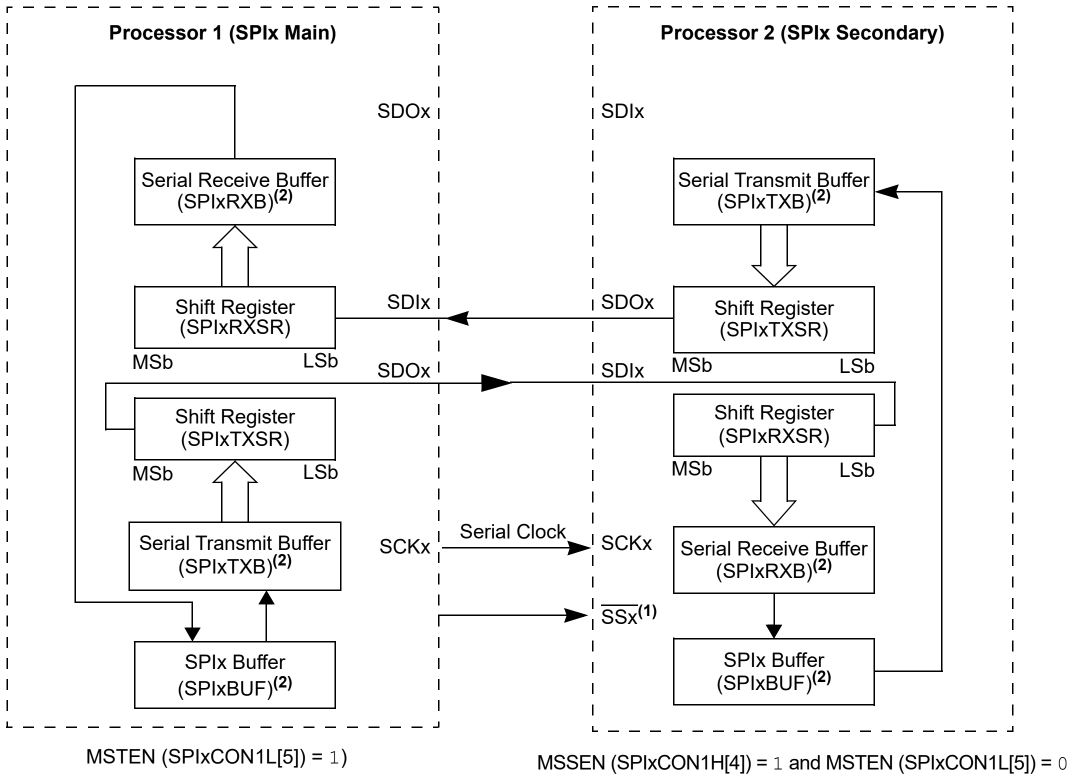
bit 7 **TXWIEN:** Transmit Watermark Interrupt Enable bit
1 = Triggers transmit buffer element watermark interrupt when TXMSK[5:0] = TXELM[5:0]
0 = Disables transmit buffer element watermark interrupt

bit 6 **Unimplemented:** Read as ‘0’

bit 5-0 **TXMSK[5:0]:** TX Buffer Mask bits[1,2,3,4]
TX mask bits; used in conjunction with the TXWIEN bit.

Note 1: Mask values higher than FIFODEPTH are not valid. The module will not trigger a match for any value in this case.
2: RXMSK2 and TXMSK2 bits are only present when FIFODEPTH = 8 or higher.
3: RXMSK3 and TXMSK3 bits are only present when FIFODEPTH = 16 or higher.
4: RXMSK4 and TXMSK4 bits are only present when FIFODEPTH = 32.

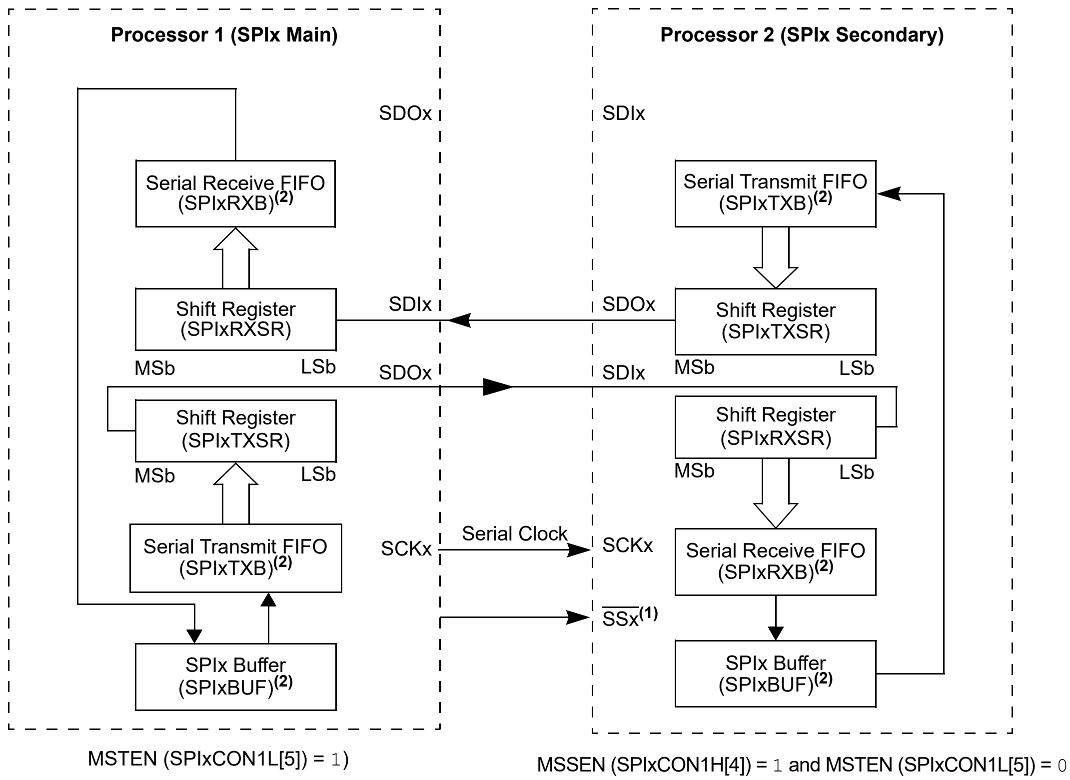
FIGURE 18-3: SPIx MAIN/SECONDARY CONNECTION (STANDARD MODE)



Note 1: Using the $\overline{SSx}$ pin in Secondary mode of operation is optional.

2: User must write transmit data to read the received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory-mapped to SPIxBUF.

FIGURE 18-4: SPIx MAIN/SECONDARY CONNECTION (ENHANCED BUFFER MODES)



Note 1: Using the SSx pin in Secondary mode of operation is optional.

2: User must write transmit data to read the received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory-mapped to SPIxBUF.

FIGURE 18-5: SPIx MAIN, FRAME MASTER CONNECTION DIAGRAM

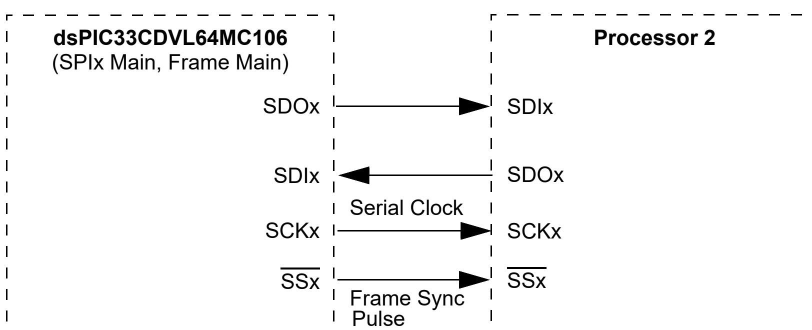


FIGURE 18-6: SPIx MAIN, FRAME SECONDARY CONNECTION DIAGRAM

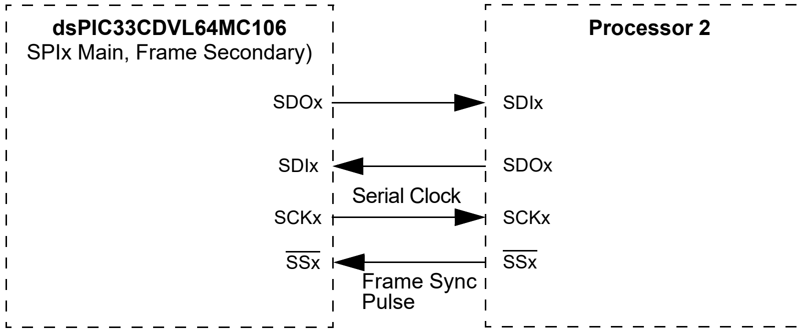


FIGURE 18-7: SPIx SECONDARY, FRAME MAIN CONNECTION DIAGRAM

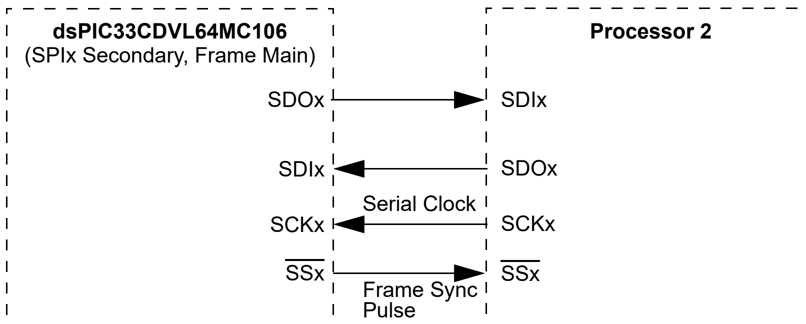
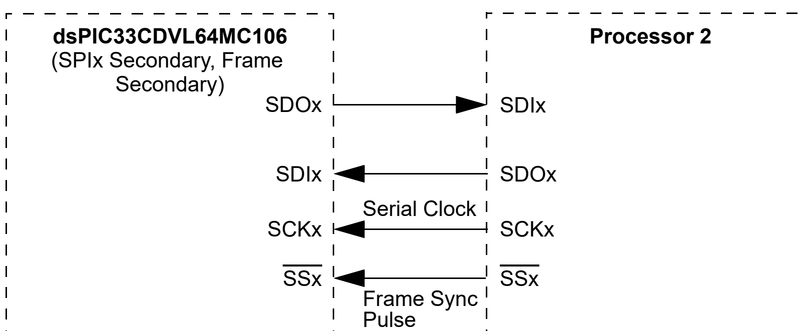


FIGURE 18-8: SPIx SECONDARY, FRAME SECONDARY CONNECTION DIAGRAM



EQUATION 18-1: RELATIONSHIP BETWEEN DEVICE AND SPIx CLOCK SPEED

$$Baud\ Rate = \frac{F_P}{(2 * (SPIxBRG + 1))}$$

Where:

F_P is the Peripheral Bus Clock Frequency.

19.0 INTER-INTEGRATED CIRCUIT (I^2C)

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. For more information, refer to **"Inter-Integrated Circuit (I^2C)"** (www.microchip.com/DS70000195).

The Inter-Integrated Circuit (I^2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent Master and Slave Logic
- 7-Bit and 10-Bit Device Addresses
- General Call Address as Defined in the I^2C Protocol
- Clock Stretching to Provide Delays for the Processor to Respond to a Slave Data Request
- Both 100 kHz and 400 kHz Bus Specifications
- Configurable Address Masking
- Multi-Master modes to Prevent Loss of Messages in Arbitration
- Bus Repeater mode, Allowing the Acceptance of All Messages as a Slave, regardless of the Address
- Automatic SCL

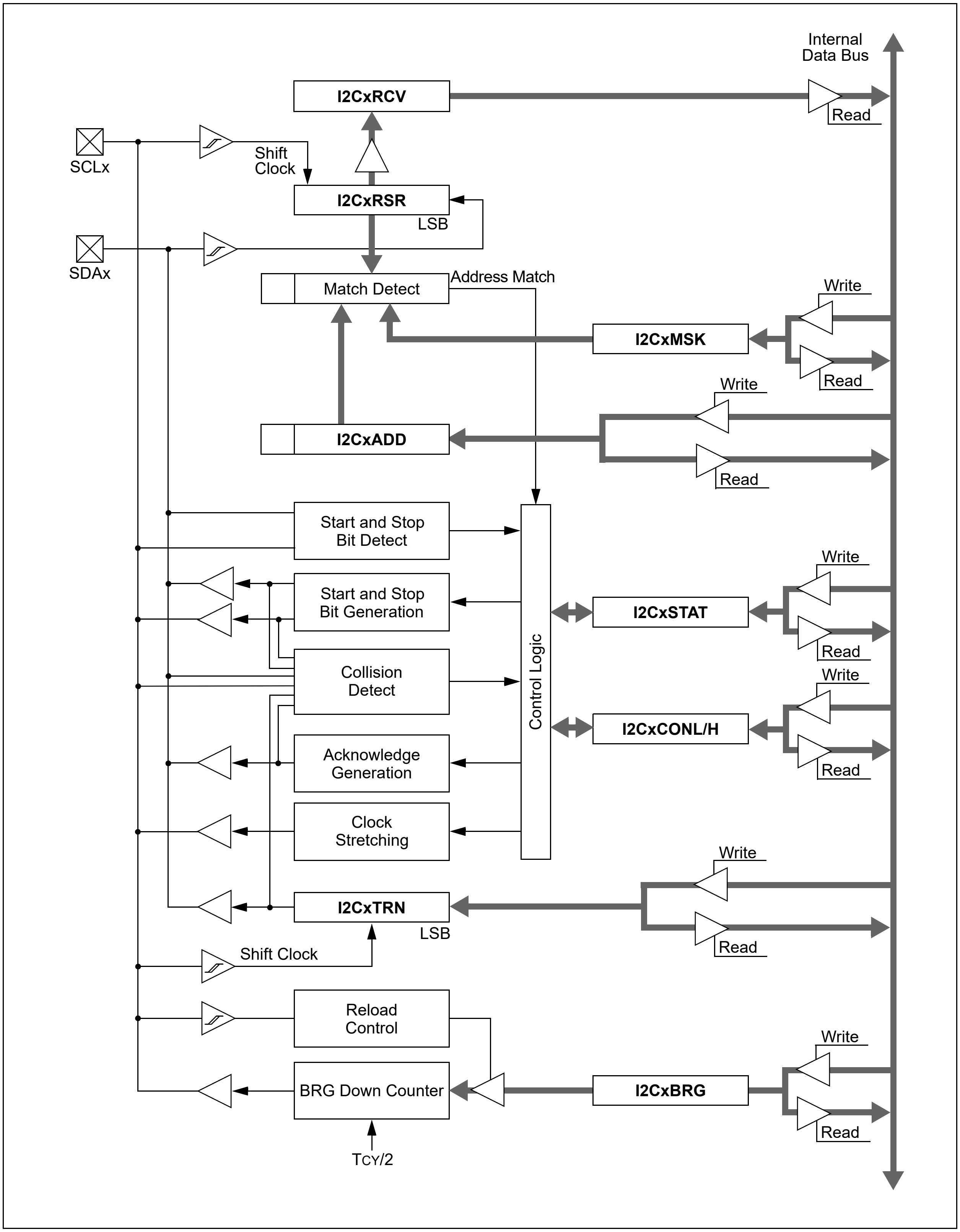
A block diagram of the module is shown in Figure 19-1.

19.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communication protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the Slave with a write indication.
3. Wait for and verify an Acknowledge from the Slave.
4. Send the first data byte (sometimes known as the command) to the Slave.
5. Wait for and verify an Acknowledge from the Slave.
6. Send the serial memory address low byte to the Slave.
7. Repeat Steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the Slave with a read indication.
10. Wait for and verify an Acknowledge from the Slave.
11. Enable Master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

FIGURE 19-1: I2Cx BLOCK DIAGRAM



19.2 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator reload value, use Equation 19-1.

EQUATION 19-1: COMPUTING BAUD RATE RELOAD VALUE[1,2,3,4]

$$I2CxBRG = ((1/F_{SCL} – Delay) \cdot F_P/2) – 2$$

Note 1: Based on FP = FOSC/2.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system-level parameters. The actual clock rate should be measured in its intended application.

3: Typical value of delay varies from 110 ns to 150 ns.

4: I2CxBRG values of 0 to 3 are expressly forbidden. The user should never program the I2CxBRG with a value of 0x0, 0x1, 0x2 or 0x3 as indeterminate results may occur.

19.3 Slave Address Masking

The I2CxMSK register (Register 19-4) designates address bit positions as “don’t care” for both 7-Bit and 10-Bit Addressing modes. Setting a particular bit location (= `1`) in the I2CxMSK register causes the Slave module to respond, whether the corresponding address bit value is a ‘`0`’ or a ‘`1`’. For example, when I2CxMSK is set to ‘`0010000000`’, the Slave module will detect both addresses, ‘`0000000000`’ and ‘`0010000000`’.

To enable address masking, the Intelligent Peripheral Management Interface (IPMI) must be disabled by clearing the STRICT bit (I2CxCONL[11]).

Note: As a result of changes in the I^{2}C protocol, the addresses in Table 19-2 are reserved and will not be Acknowledged in Slave mode. This includes any address mask settings that include any of these addresses.

TABLE 19-1: I2Cx CLOCK RATES[1,2]

FCY	FSCL	I2CxBRG Value	
		Decimal	Hexadecimal
100 MHz	1 MHz	41	29
100 MHz	400 kHz	116	74
100 MHz	100 kHz	491	1EB
80 MHz	1 MHz	32	20
80 MHz	400 kHz	92	5C
80 MHz	100 kHz	392	188
60 MHz	1 MHz	24	18
60 MHz	400 kHz	69	45
60 MHz	100 kHz	294	126
40 MHz	1 MHz	15	0F
40 MHz	400 kHz	45	2D
40 MHz	100 kHz	195	C3
20 MHz	1 MHz	7	7
20 MHz	400 kHz	22	16
20 MHz	100 kHz	97	61

Note 1: Based on FP = FOSC/2.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system-level parameters. The actual clock rate should be measured in its intended application.

TABLE 19-2: I2Cx RESERVED ADDRESSES[1]

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General Call Address[2]
0000 000	1	Start Byte
0000 001	x	Cbus Address
0000 01x	x	Reserved
0000 1xx	x	HS Mode Master Code
1111 0xx	x	10-Bit Slave Upper Byte[3]
1111 1xx	x	Reserved

Note 1: The address bits listed here will never cause an address match independent of address mask settings.
2: This address will be Acknowledged only if GCEN = 1.
3: A match on this address can only occur on the upper byte in 10-Bit Addressing mode.

19.4 I²C Control/Status Registers

REGISTER 19-1: I2CxCONL: I2Cx CONTROL REGISTER LOW

R/W-0	U-0	HC/R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL(1)	STRICT	A10M	DISSLW	SMEN(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **I2CEN:** I2Cx Enable bit (writable from software only)
1 = Enables the I2Cx module, and configures the SDAx and SCLx pins as serial port pins
0 = Disables the I2Cx module; all I²C pins are controlled by port functions

bit 14 **Unimplemented:** Read as '0'

bit 13 **I2CSIDL:** I2Cx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **SCLREL:** SCLx Release Control bit (I²C Slave mode only)(1)
1 = Releases the SCLx clock
0 = Holds the SCLx clock low (clock stretch)

If STREN = 1:(2)
User software may write '0' to initiate a clock stretch and write '1' to release the clock. Hardware clears at the beginning of every Slave data byte transmission. Hardware clears at the end of every Slave address byte reception. Hardware clears at the end of every Slave data byte reception.

If STREN = 0:
User software may only write '1' to release the clock. Hardware clears at the beginning of every Slave data byte transmission. Hardware clears at the end of every Slave address byte reception.

bit 11 **STRICT:** I2Cx Strict Reserved Address Rule Enable bit
1 = Strict reserved addressing is enforced; for reserved addresses, refer to Table 19-2.
(In Slave Mode) – The device doesn't respond to reserved address space and addresses falling in that category are NACKed.
(In Master Mode) – The device is allowed to generate addresses with reserved address space.
0 = Reserved addressing would be Acknowledged.
(In Slave Mode) – The device will respond to an address falling in the reserved address space. When there is a match with any of the reserved addresses, the device will generate an ACK.
(In Master Mode) – Reserved.

bit 10 **A10M:** 10-Bit Slave Address Flag bit
1 = I2CxADD is a 10-bit Slave address
0 = I2CxADD is a 7-bit Slave address

Note 1: Automatically cleared to '0' at the beginning of Slave transmission; automatically cleared to '0' at the end of Slave reception.
2: Automatically cleared to '0' at the beginning of Slave transmission.
3: The SMB3EN Configuration bit (FDEVOPT[10]) selects between normal and SMBus 3.0 levels.

REGISTER 19-1: I2CxCONL: I2Cx CONTROL REGISTER LOW (CONTINUED)

bit 9 **DISSLW:** Slew Rate Control Disable bit
1 = Slew rate control is disabled for Standard Speed mode (100 kHz, also disabled for 1 MHz mode)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

bit 8 **SMEN:** SMBus Input Levels Enable bit(3)
1 = Enables input logic so thresholds are compliant with the SMBus specification
0 = Disables SMBus-specific inputs

bit 7 **GCEN:** General Call Enable bit (I^2C Slave mode only)
1 = Enables interrupt when a general call address is received in I2CxRSR; module is enabled for reception
0 = General call address is disabled.

bit 6 **STREN:** SCLx Clock Stretch Enable bit
In I^2C Slave mode only; used in conjunction with the SCLREL bit.
1 = Enables clock stretching
0 = Disables clock stretching

bit 5 **ACKDT:** Acknowledge Data bit
In I^2C Master mode during Master Receive mode. The value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
In I^2C Slave mode when AHEN = 1 or DHEN = 1. The value that the Slave will transmit when it initiates an Acknowledge sequence at the end of an address or data reception.
1 = NACK is sent
0 = ACK is sent

bit 4 **ACKEN:** Acknowledge Sequence Enable bit
In I^2C Master mode only; applicable during Master Receive mode.
1 = Initiates Acknowledge sequence on SDAx and SCLx pins, and transmits ACKDT data bit
0 = Acknowledge sequence is Idle

bit 3 **RCEN:** Receive Enable bit (I^2C Master mode only)
1 = Enables Receive mode for I^2C; automatically cleared by hardware at end of 8-bit receive data byte
0 = Receive sequence is not in progress

bit 2 **PEN:** Stop Condition Enable bit (I^2C Master mode only)
1 = Initiates Stop condition on SDAx and SCLx pins
0 = Stop condition is Idle

bit 1 **RSEN:** Restart Condition Enable bit (I^2C Master mode only)
1 = Initiates Restart condition on SDAx and SCLx pins
0 = Restart condition is Idle

bit 0 **SEN:** Start Condition Enable bit (I^2C Master mode only)
1 = Initiates Start condition on SDAx and SCLx pins
0 = Start condition is Idle

Note 1: Automatically cleared to ‘0’ at the beginning of Slave transmission; automatically cleared to ‘0’ at the end of Slave reception.
2: Automatically cleared to ‘0’ at the beginning of Slave transmission.
3: The SMB3EN Configuration bit (FDEVOPT[10]) selects between normal and SMBus 3.0 levels.

REGISTER 19-2: I2CxCONH: I2Cx CONTROL REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6 **PCIE:** Stop Condition Interrupt Enable bit (I^2C Slave mode only).
1 = Enables interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled

bit 5 **SCIE:** Start Condition Interrupt Enable bit (I^2C Slave mode only)
1 = Enables interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled

bit 4 **BOEN:** Buffer Overwrite Enable bit (I^2C Slave mode only)
1 = I2CxRCV is updated and an ACK is generated for a received address/data byte, ignoring the state of the I2COV bit only if RBF bit = 0
0 = I2CxRCV is only updated when I2COV is clear

bit 3 **SDAHT:** SDAx Hold Time Selection bit
1 = Minimum of 300 ns hold time on SDAx after the falling edge of SCLx
0 = Minimum of 100 ns hold time on SDAx after the falling edge of SCLx

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)
If, on the rising edge of SCLx, SDAx is sampled low when the module is outputting a High state, the BCL bit is set and the bus goes Idle. This Detection mode is only valid during data and ACK transmit sequences.
1 = Enables Slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (I^2C Slave mode only)
1 = Following the 8th falling edge of SCLx for a matching received address byte; SCLREL bit (I2CxCONL[12]) will be cleared and the SCLx will be held low
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^2C Slave mode only)
1 = Following the 8th falling edge of SCLx for a received data byte; Slave hardware clears the SCLREL bit (I2CxCONL[12]) and SCLx is held low
0 = Data holding is disabled

REGISTER 19-3: I2CxSTAT: I2Cx STATUS REGISTER

HSC/R-0	HSC/R-0	HSC/R-0	U-0	U-0	HSC/R/C-0	HSC/R-0	HSC/R-0
ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

HS/R/C-0	HS/R/C-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
IWCOL	I2COV	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	RBF	TBF
bit 7							bit 0

Legend:	C = Clearable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	HS = Hardware Settable bit

bit 15 **ACKSTAT:** Acknowledge Status bit (updated in all Master and Slave modes)
1 = Acknowledge was not received from Slave
0 = Acknowledge was received from Slave

bit 14 **TRSTAT:** Transmit Status bit (when operating as I^{2}C Master; applicable to Master transmit operation)
1 = Master transmit is in progress (eight bits + ACK)
0 = Master transmit is not in progress

bit 13 **ACKTIM:** Acknowledge Time Status bit (valid in I^{2}C Slave mode only)
1 = Indicates I^{2}C bus is in an Acknowledge sequence, set on 8th falling edge of SCLx clock
0 = Not an Acknowledge sequence, cleared on 9th rising edge of SCLx clock

bit 12-11 **Unimplemented:** Read as ‘0’

bit 10 **BCL:** Bus Collision Detect bit (cleared when I^{2}C module is disabled, I2CEN = 0)
1 = A bus collision has been detected during a transmit operation
0 = No bus collision has been detected

bit 9 **GCSTAT:** General Call Status bit (cleared after Stop detection)
1 = General call address was received
0 = General call address was not received

bit 8 **ADD10:** 10-Bit Address Status bit (cleared after Stop detection)
1 = 10-bit address was matched
0 = 10-bit address was not matched

bit 7 **IWCOL:** I2Cx Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I^{2}C module is busy; must be cleared in software
0 = No collision

bit 6 **I2COV:** I2Cx Receive Overflow Flag bit
1 = A byte was received while the I2CxRCV register is still holding the previous byte; I2COV is a “don’t care” in Transmit mode, must be cleared in software
0 = No overflow

bit 5 **D/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ bit (when operating as I^{2}C Slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received or transmitted was an address

bit 4 **P:** I2Cx Stop bit
Updated when Start, Reset or Stop is detected; cleared when the I^{2}C module is disabled, I2CEN = 0.
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

REGISTER 19-3: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

bit 3	**S:** I2Cx Start bit Updated when Start, Reset or Stop is detected; cleared when the I^2C module is disabled, I2CEN = 0. 1 = Indicates that a Start (or Repeated Start) bit has been detected last 0 = Start bit was not detected last
bit 2	**R/$\overline{W}$:** Read/$\overline{Write}$ Information bit (when operating as I^2C Slave) 1 = Read: Indicates the data transfer is output from the Slave 0 = Write: Indicates the data transfer is input to the Slave
bit 1	**RBF:** Receive Buffer Full Status bit 1 = Receive is complete, I2CxRCV is full 0 = Receive is not complete, I2CxRCV is empty
bit 0	**TBF:** Transmit Buffer Full Status bit 1 = Transmit is in progress, I2CxTRN is full (eight bits of data) 0 = Transmit is complete, I2CxTRN is empty

REGISTER 19-4: I2CxMSK: I2Cx SLAVE MODE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MSK[9:8]	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MSK[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10	**Unimplemented:** Read as ‘0’
bit 9-0	**MSK[9:0]:** I2Cx Mask for Address Bit x Select bits 1 = Enables masking for bit x of the incoming message address; bit match is not required in this position 0 = Disables masking for bit x; bit match is required in this position

NOTES:

20.0 SINGLE-EDGE NIBBLE TRANSMISSION (SENT)

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Single-Edge Nibble Transmission (SENT) Module"** (www.microchip.com/DS70005145).

The Single-Edge Nibble Transmission (SENT) module is based on the SAE J2716, *"SENT – Single-Edge Nibble Transmission for Automotive Applications"*. The SENT protocol is a one-way, single wire, time-modulated serial communication, based on successive falling edges. It is intended for use in applications where high-resolution sensor data need to be communicated from a sensor to an Engine Control Unit (ECU).

The SENTx module has the following major features:

- Selectable Transmit or Receive Mode
- Synchronous or Asynchronous Transmit Modes
- Automatic Data Rate Synchronization
- Optional Automatic Detection of CRC Errors in Receive Mode
- Optional Hardware Calculation of CRC in Transmit Mode
- Support for Optional Pause Pulse Period
- Data Buffering for One Message Frame
- Selectable Data Length for Transmit/Receive, Up to Six Nibbles
- Automatic Detection of Framing Errors

SENT protocol timing is based on a predetermined time unit, T_{TICK}. Both the transmitter and receiver must be preconfigured for T_{TICK}, which can vary from 3 to 90 µs. A SENT message frame starts with a Sync pulse. The purpose of the Sync pulse is to allow the receiver to calculate the data rate of the message encoded by the transmitter. The SENT specification allows messages to be validated with up to a 20% variation in T_{TICK}. This allows for the transmitter and receiver to run from different clocks that may be inaccurate, and drift with time and temperature. The data nibbles are 4 bits in length and are encoded as the data value + 12 ticks. This yields a 0 value of 12 ticks and the maximum value, 0xF, of 27 ticks.

A SENT message consists of the following:

- A synchronization/calibration period of 56 tick times
- A status nibble of 12-27 tick times
- Up to six data nibbles of 12-27 tick times
- A CRC nibble of 12-27 tick times
- An optional pause pulse period of 12-768 tick times

Figure 20-1 shows a block diagram of the SENTx module.

Figure 20-2 shows the construction of a typical 6-nibble data frame, with the numbers representing the minimum or maximum number of tick times for each section.

FIGURE 20-1: SENTx MODULE BLOCK DIAGRAM

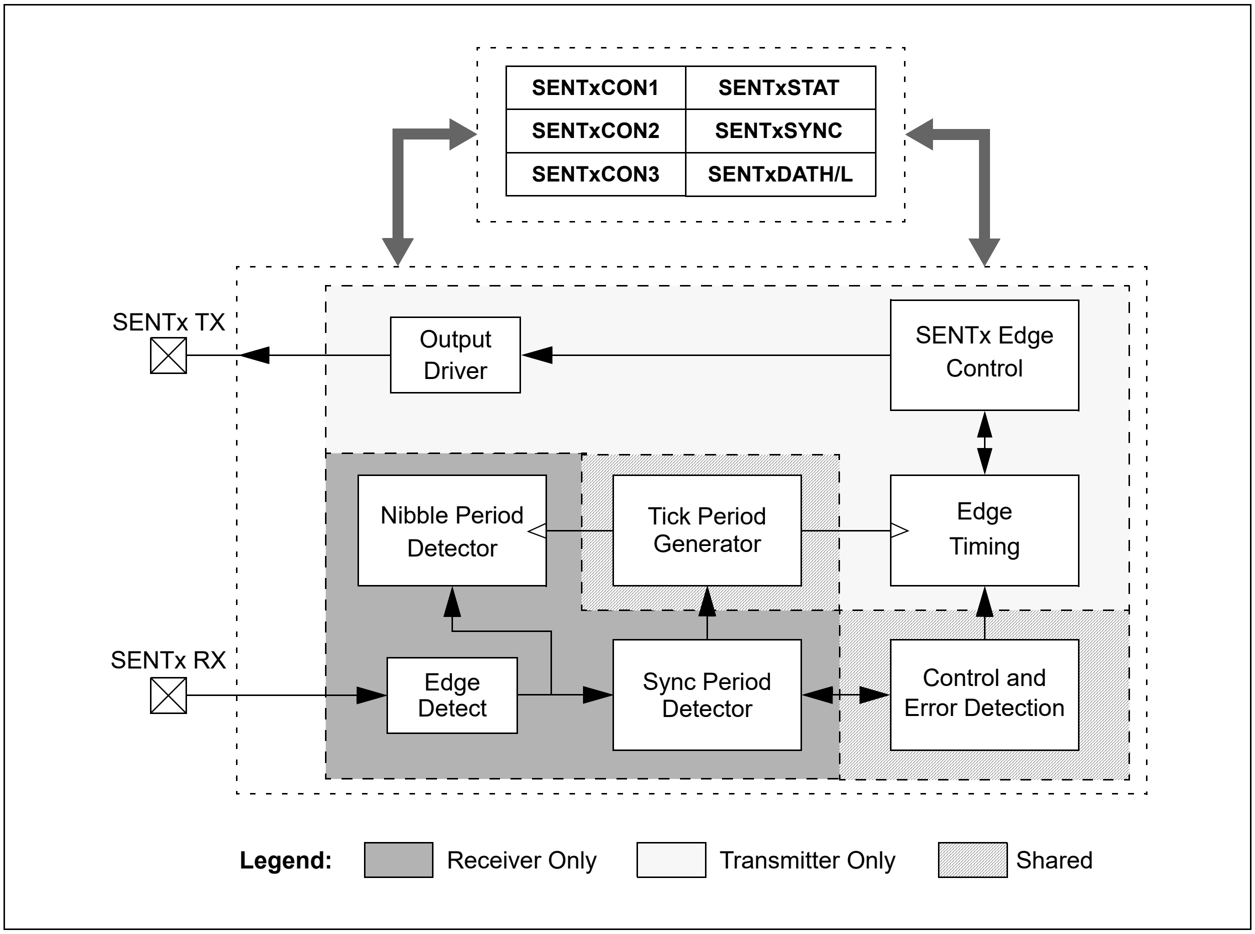
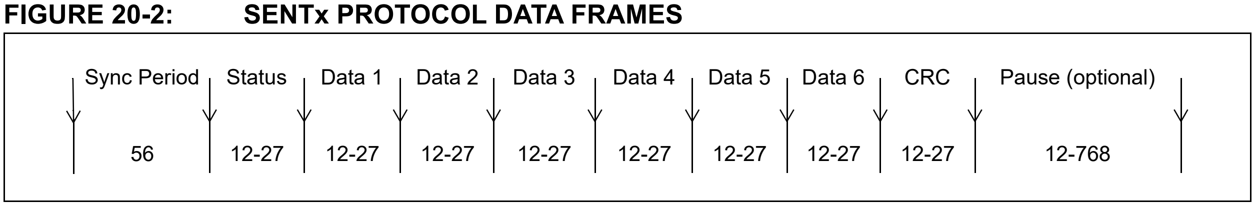


FIGURE 20-2: SENTx PROTOCOL DATA FRAMES



20.1 Transmit Mode

By default, the SENTx module is configured for transmit operation. The module can be configured for continuous asynchronous message frame transmission, or alternatively, for Synchronous mode triggered by software. When enabled, the transmitter will send a Sync, followed by the appropriate number of data nibbles, an optional CRC and optional pause pulse. The tick period used by the SENTx transmitter is set by writing a value to the TICKTIME[15:0] (SENTxCON2[15:0]) bits. The tick period calculations are shown in Equation 20-1.

EQUATION 20-1: TICK PERIOD CALCULATION

$$TICKTIME[15:0] = \frac{T_{TICK}}{T_{CLK}} - 1$$

An optional pause pulse can be used in Asynchronous mode to provide a fixed message frame time period. The frame period used by the SENTx transmitter is set by writing a value to the FRAMETIME[15:0] (SENTxCON3[15:0]) bits. The formulas used to calculate the value of frame time are shown in Equation 20-2.

EQUATION 20-2: FRAME TIME CALCULATIONS

$$FRAMETIME[15:0] = T_{TICK}/T_{FRAME}$$

$$FRAMETIME[15:0] \geq 122 + 27N$$

$$FRAMETIME[15:0] \geq 848 + 12N$$

Where:

T_{FRAME} = Total time of the message in ms

N = The number of data nibbles in message, 1-6

Note:	The module will not produce a pause period with less than 12 ticks, regardless of the FRAMETIME[15:0] value. FRAMETIME[15:0] values beyond 2047 will have no effect on the length of a data frame.

20.1.1 TRANSMIT MODE CONFIGURATION

20.1.1.1 Initializing the SENTx Module

Perform the following steps to initialize the module:

1. Write RCVEN (SENTxCON1[11]) = `0` for Transmit mode.
2. Write TXM (SENTxCON1[10]) = `0` for Asynchronous Transmit mode or TXM = `1` for Synchronous mode.
3. Write NIBCNT[2:0] (SENTxCON1[2:0]) for the desired data frame length.
4. Write CRCEN (SENTxCON1[8]) for hardware or software CRC calculation.
5. Write PPP (SENTxCON1[7]) for optional pause pulse.
6. If PPP = `1`, write T_{FRAME} to SENTxCON3.
7. Write SENTxCON2 with the appropriate value for the desired tick period.
8. Enable interrupts and set interrupt priority.
9. Write initial status and data values to SENTxDATH/L.
10. If CRCEN = `0`, calculate CRC and write the value to CRC[3:0] (SENTxDATL[3:0]).
11. Set the SNTEN (SENTxCON1[15]) bit to enable the module.

User software updates to SENTxDATH/L must be performed after the completion of the CRC and before the next message frame's status nibble. The recommended method is to use the message frame completion interrupt to trigger data writes.

20.2 Receive Mode

The module can be configured for receive operation by setting the RCVEN (SENTxCON1[11]) bit. The time between each falling edge is compared to SYNCMIN[15:0] (SENTxCON3[15:0]) and SYNCMAX[15:0] (SENTxCON2[15:0]), and if the measured time lies between the minimum and maximum limits, the module begins to receive data. The validated Sync time is captured in the SENTxSYNC register and the tick time is calculated. Subsequent falling edges are verified to be within the valid data width and the data are stored in the SENTxDATL/H registers. An interrupt event is generated at the completion of the message and the user software should read the SENTx Data registers before the reception of the next nibble. The equation for SYNCMIN[15:0] and SYNCMAX[15:0] is shown in Equation 20-3.

EQUATION 20-3: SYNCMIN[15:0] AND SYNCMAX[15:0] CALCULATIONS

$$T_{TICK} = T_{CLK} \bullet (TICKTIME[15:0] + 1)$$

$$FRAMETIME[15:0] = T_{TICK}/T_{FRAME}$$

$$SyncCount = 8 \times F_{RCV} \times T_{TICK}$$

$$SYNCMIN[15:0] = 0.8 \times SyncCount$$

$$SYNCMAX[15:0] = 1.2 \times SyncCount$$

$$FRAMETIME[15:0] \geq 122 + 27N$$

$$FRAMETIME[15:0] \geq 848 + 12N$$

Where:

T_{FRAME} = Total time of the message in ms

N = The number of data nibbles in message, 1-6

F_{RCV} = F_{CY} x Prescaler

T_{CLK} = F_{CY}/Prescaler

For T_{TICK} = 3.0 µs and F_{CLK} = 4 MHz, SYNCMIN[15:0] = 76.

Note:	To ensure a Sync period can be identified, the value written to SYNCMIN[15:0] must be less than the value written to SYNCMAX[15:0].

20.2.1 RECEIVE MODE CONFIGURATION

20.2.1.1 Initializing the SENTx Module

Perform the following steps to initialize the module:

1. Write RCVEN (SENTxCON1[11]) = `1` for Receive mode.
2. Write NIBCNT[2:0] (SENTxCON1[2:0]) for the desired data frame length.
3. Write CRCEN (SENTxCON1[8]) for hardware or software CRC validation.
4. Write PPP (SENTxCON1[7]) = `1` if pause pulse is present.
5. Write SENTxCON2 with the value of SYNCMAXx (Nominal Sync Period + 20%).
6. Write SENTxCON3 with the value of SYNCMINx (Nominal Sync Period – 20%).
7. Enable interrupts and set interrupt priority.
8. Set the SNTEN (SENTxCON1[15]) bit to enable the module.

The data should be read from the SENTxDATL/H registers after the completion of the CRC and before the next message frame's status nibble. The recommended method is to use the message frame completion interrupt trigger.

20.3 SENT Control/Status Registers

REGISTER 20-1: SENTxCON1: SENTx CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SNTEN	—	SNTSIDL	—	RCVEN	TXM(1)	TXPOL(1)	CRCEN
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PPP	SPCEN(2)	—	PS	—	NIBCNT2	NIBCNT1	NIBCNT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **SNTEN:** SENTx Enable bit
1 = SENTx is enabled
0 = SENTx is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **SNTSIDL:** SENTx Stop in Idle Mode bit
1 = Discontinues module operation when the device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **Unimplemented:** Read as '0'

bit 11 **RCVEN:** SENTx Receive Enable bit
1 = SENTx operates as a receiver
0 = SENTx operates as a transmitter (sensor)

bit 10 **TXM:** SENTx Transmit Mode bit(1)
1 = SENTx transmits data frame only when triggered using the SYNCTXEN status bit
0 = SENTx transmits data frames continuously while SNTEN = 1

bit 9 **TXPOL:** SENTx Transmit Polarity bit(1)
1 = SENTx data output pin is low in the Idle state
0 = SENTx data output pin is high in the Idle state

bit 8 **CRCEN:** CRC Enable bit
Module in Receive Mode (RCVEN = 1):
1 = SENTx performs CRC verification on received data using the preferred J2716 method
0 = SENTx does not perform CRC verification on received data
Module in Transmit Mode (RCVEN = 0):
1 = SENTx automatically calculates CRC using the preferred J2716 method
0 = SENTx does not calculate CRC

bit 7 **PPP:** Pause Pulse Present bit
1 = SENTx is configured to transmit/receive SENT messages with pause pulse
0 = SENTx is configured to transmit/receive SENT messages without pause pulse

bit 6 **SPCEN:** Short PWM Code Enable bit(2)
1 = SPC control from external source is enabled
0 = SPC control from external source is disabled

bit 5 **Unimplemented:** Read as '0'

Note 1: This bit has no function in Receive mode (RCVEN = 1).
2: This bit has no function in Transmit mode (RCVEN = 0).

REGISTER 20-1: SENTxCON1: SENTx CONTROL REGISTER 1 (CONTINUED)

bit 4 **PS:** SENTx Module Clock Prescaler (divider) bits
1 = Divide-by-4
0 = Divide-by-1

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **NIBCNT[2:0]:** Nibble Count Control bits
111 = Reserved; do not use
110 = Module transmits/receives six data nibbles in a SENT data packet
101 = Module transmits/receives five data nibbles in a SENT data packet
100 = Module transmits/receives four data nibbles in a SENT data packet
011 = Module transmits/receives three data nibbles in a SENT data packet
010 = Module transmits/receives two data nibbles in a SENT data packet
001 = Module transmits/receives one data nibble in a SENT data packet
000 = Reserved; do not use

Note 1: This bit has no function in Receive mode (RCVEN = 1).
2: This bit has no function in Transmit mode (RCVEN = 0).

REGISTER 20-2: SENTxSTAT: SENTx STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0	R-0	R-0	R-0	R/C-0	R/C-0	R-0	HC/R/W-0
PAUSE	NIB2	NIB1	NIB0	CRCERR	FRMERR	RXIDLE	SYNCTXEN[1]
bit 7							bit 0

Legend:	C = Clearable bit	HC = Hardware Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **PAUSE:** Pause Period Status bit
1 = The module is transmitting/receiving a pause period
0 = The module is not transmitting/receiving a pause period

bit 6-4 **NIB[2:0]:** Nibble Status bits
Module in Transmit Mode (RCVEN = 0):
111 = Module is transmitting a CRC nibble
110 = Module is transmitting Data Nibble 6
101 = Module is transmitting Data Nibble 5
100 = Module is transmitting Data Nibble 4
011 = Module is transmitting Data Nibble 3
010 = Module is transmitting Data Nibble 2
001 = Module is transmitting Data Nibble 1
000 = Module is transmitting a status nibble or pause period, or is not transmitting
Module in Receive Mode (RCVEN = 1):
111 = Module is receiving a CRC nibble or was receiving this nibble when an error occurred
110 = Module is receiving Data Nibble 6 or was receiving this nibble when an error occurred
101 = Module is receiving Data Nibble 5 or was receiving this nibble when an error occurred
100 = Module is receiving Data Nibble 4 or was receiving this nibble when an error occurred
011 = Module is receiving Data Nibble 3 or was receiving this nibble when an error occurred
010 = Module is receiving Data Nibble 2 or was receiving this nibble when an error occurred
001 = Module is receiving Data Nibble 1 or was receiving this nibble when an error occurred
000 = Module is receiving a status nibble or waiting for Sync

bit 3 **CRCERR:** CRC Status bit (Receive mode only)
1 = A CRC error has occurred for the 1-6 data nibbles in SENTxDATL/H
0 = A CRC error has not occurred

bit 2 **FRMERR:** Framing Error Status bit (Receive mode only)
1 = A data nibble was received with less than 12 tick periods or greater than 27 tick periods
0 = Framing error has not occurred

bit 1 **RXIDLE:** SENTx Receiver Idle Status bit (Receive mode only)
1 = The SENTx data bus has been Idle (high) for a period of SYNCMAX[15:0] or greater
0 = The SENTx data bus is not Idle

Note 1: In Receive mode (RCVEN = 1), the SYNCTXEN bit is read-only.

REGISTER 20-2: SENTxSTAT: SENTx STATUS REGISTER (CONTINUED)

bit 0 **SYNCTXEN:** SENTx Synchronization Period Status/Transmit Enable bit(1)

Module in Receive Mode (RCVEN = 1):
1 = A valid synchronization period was detected; the module is receiving nibble data
0 = No synchronization period has been detected; the module is not receiving nibble data

Module in Asynchronous Transmit Mode (RCVEN = 0, TXM = 0):
The bit always reads as '1' when the module is enabled, indicating the module transmits SENTx data frames continuously. The bit reads '0' when the module is disabled.

Module in Synchronous Transmit Mode (RCVEN = 0, TXM = 1):
1 = The module is transmitting a SENTx data frame
0 = The module is not transmitting a data frame, user software may set SYNCTXEN to start another data frame transmission

Note 1: In Receive mode (RCVEN = 1), the SYNCTXEN bit is read-only.

REGISTER 20-3: SENTxDATL: SENTx RECEIVE DATA REGISTER LOW[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA4[3:0]				DATA5[3:0]			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA6[3:0]				CRC[3:0]			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **DATA4[3:0]:** Data Nibble 4 Data bits

bit 11-8 **DATA5[3:0]:** Data Nibble 5 Data bits

bit 7-4 **DATA6[3:0]:** Data Nibble 6 Data bits

bit 3-0 **CRC[3:0]:** CRC Nibble Data bits

Note 1: Register bits are read-only in Receive mode (RCVEN = 1). In Transmit mode, the CRC[3:0] bits are read-only when automatic CRC calculation is enabled (RCVEN = 0, CRCEN = 1).

REGISTER 20-4: SENTxDATH: SENTx RECEIVE DATA REGISTER HIGH[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STAT[3:0]				DATA1[3:0]			
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA2[3:0]				DATA3[3:0]			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **STAT[3:0]:** Status Nibble Data bits

bit 11-8 **DATA1[3:0]:** Data Nibble 1 Data bits

bit 7-4 **DATA2[3:0]:** Data Nibble 2 Data bits

bit 3-0 **DATA3[3:0]:** Data Nibble 3 Data bits

Note 1: Register bits are read-only in Receive mode (RCVEN = 1). In Transmit mode, the CRC[3:0] bits are read-only when automatic CRC calculation is enabled (RCVEN = 0, CRCEN = 1).

NOTES:

21.0 TIMER1

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Timer1 Module"** (www.microchip.com/DS70005279).

The Timer1 module is a 16-bit timer that can operate as a free-running interval timer/counter.

The Timer1 module has the following unique features over other timers:

- Can be Operated in Asynchronous Counter Mode
- Asynchronous Timer
- Operational during CPU Sleep Mode
- Software Selectable Prescalers 1:1, 1:8, 1:64 and 1:256
- External Clock Selection Control
- The Timer1 External Clock Input (T1CK) can Optionally be Synchronized to the Internal Device Clock and the Clock Synchronization is Performed after the Prescaler

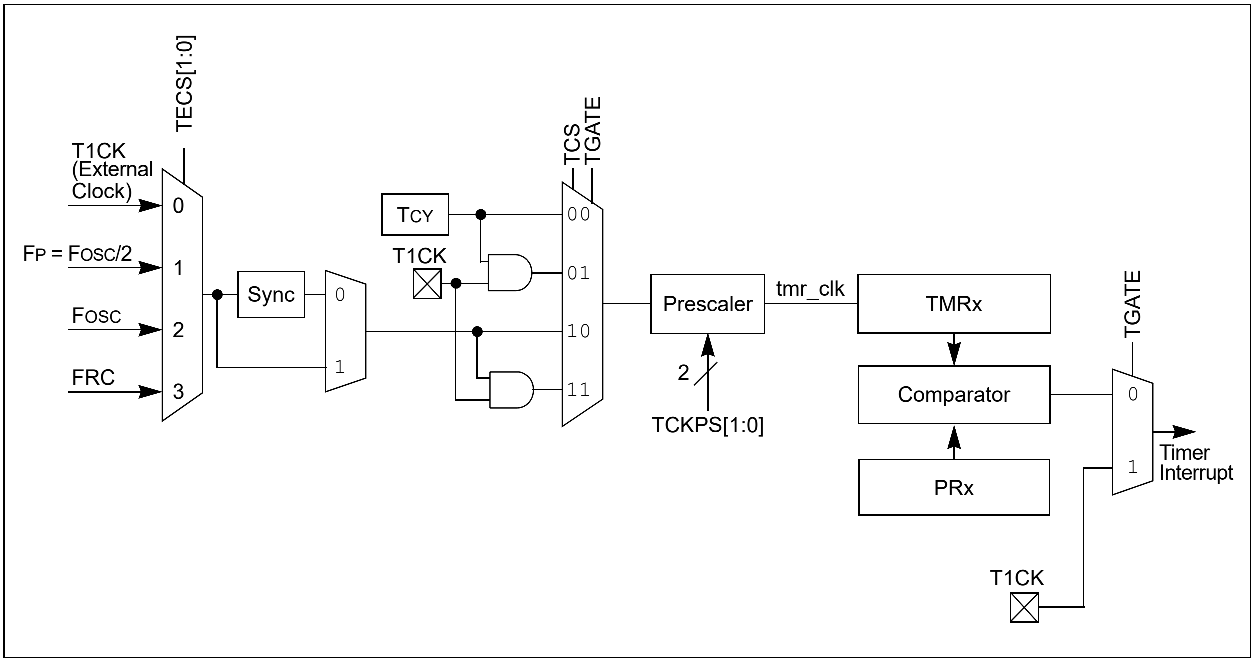
If Timer1 is used for SCCP, the timer should be running in Synchronous mode.

The Timer1 module can operate in one of the following modes:

- Timer Mode
- Gated Timer Mode
- Synchronous Counter Mode
- Asynchronous Counter Mode

A block diagram of Timer1 is shown in Figure 21-1.

FIGURE 21-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



21.1 Timer1 Control Register

REGISTER 21-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
TON[(1)]	—	SIDL	TMWDIS	TMWIP	PRWIP	TECS1	TECS0
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
TGATE	—	TCKPS1	TCKPS0	—	TSYNC[(1)]	TCS[(1)]	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **TON:** Timer1 On bit[(1)]
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **SIDL:** Timer1 Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **TMWDIS:** Asynchronous Timer1 Write Disable bit
1 = Timer writes are ignored while a posted write to TMR1 or PR1 is synchronized to the asynchronous clock domain
0 = Back-to-back writes are enabled in Asynchronous mode

bit 11 **TMWIP:** Asynchronous Timer1 Write in Progress bit
1 = Write to the timer in Asynchronous mode is pending
0 = Write to the timer in Asynchronous mode is complete

bit 10 **PRWIP:** Asynchronous Period Write in Progress bit
1 = Write to the Period register in Asynchronous mode is pending
0 = Write to the Period register in Asynchronous mode is complete

bit 9-8 **TECS[1:0]:** Timer1 Extended Clock Select bits
11 = FRC Clock
10 = FOSC Oscillator Clock
01 = FP = FOSC/2 Peripheral Clock
00 = External Clock comes from the T1CK pin

bit 7 **TGATE:** Timer1 Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 6 **Unimplemented:** Read as ‘0’

Note 1: When Timer1 is enabled in External Synchronous Counter mode (TCS = 1, TSYNC = 1, TON = 1), any attempts by user software to write to the TMR1 register are ignored.

REGISTER 21-1: T1CON: TIMER1 CONTROL REGISTER (CONTINUED)

bit 5-4	**TCKPS[1:0]:** Timer1 Input Clock Prescale Select bits 11 = 1:256 10 = 1:64 01 = 1:8 00 = 1:1
bit 3	**Unimplemented:** Read as '0'
bit 2	**TSYNC:** Timer1 External Clock Input Synchronization Select bit(1) When TCS = 1: 1 = Synchronizes the External Clock input 0 = Does not synchronize the External Clock input When TCS = 0: This bit is ignored.
bit 1	**TCS:** Timer1 Clock Source Select bit(1) 1 = External Clock source selected by TECS[1:0] 0 = Internal Peripheral Clock (FP)
bit 0	**Unimplemented:** Read as '0'

Note 1: When Timer1 is enabled in External Synchronous Counter mode (TCS = 1, TSYNC = 1, TON = 1), any attempts by user software to write to the TMR1 register are ignored.

NOTES:

22.0 CAPTURE/COMPARE/PWM/TIMER MODULES (SCCP)

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. For more information on the SCCP modules, refer to **"Capture/Compare/PWM/Timer (MCCP and SCCP)"** (www.microchip.com/DS30003035)

The dsPIC33CDVL64MC106 devices include four SCCP Capture/Compare/PWM/Timer base modules, which provide the functionality of three different peripherals from earlier PIC24F devices. The module can operate in one of three major modes:

- General Purpose Timer
- Input Capture
- Output Compare/PWM

Single Capture/Compare/PWM (SCCP) output modules provide only one PWM output.

The SCCPx modules can be operated in only one of the three major modes at any time. The other modes are not available unless the module is reconfigured for the new mode.

A conceptual block diagram for the module is shown in Figure 22-1. All three modes share a time base generator and a common Timer register pair (CCPxTMRH/L); other shared hardware components are added as a particular mode requires.

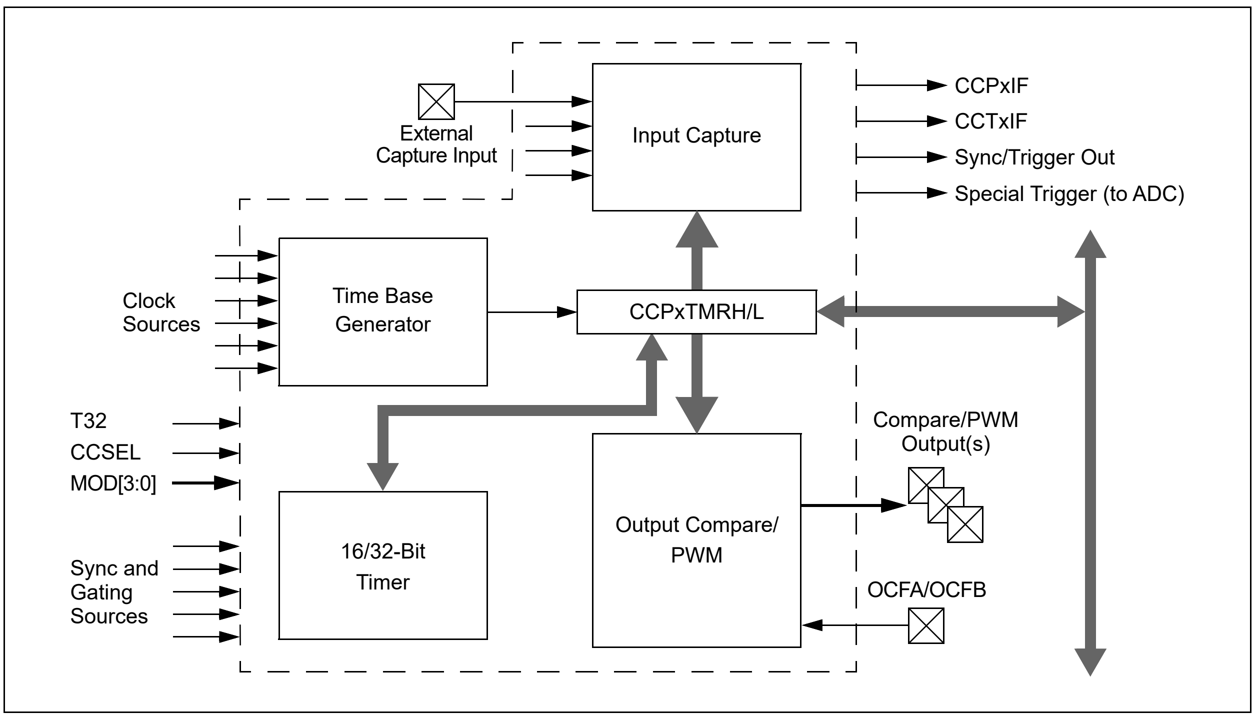
Each module has a total of six control and status registers:

- CCPxCON1L (Register 22-1)
- CCPxCON1H (Register 22-2)
- CCPxCON2L (Register 22-3)
- CCPxCON2H (Register 22-4)
- CCPxCON3H (Register 22-5)
- CCPxSTATL (Register 22-6)

Each module also includes eight buffer/counter registers that serve as Timer Value registers or data holding buffers:

- CCPxTMRH/CCPxTMRL (CCPx Timer High/Low Counters)
- CCPxPRH/CCPxPRL (CCPx Timer Period High/Low)
- CCPxRA (CCPx Primary Output Compare Data Buffer)
- CCPxRB (CCPx Secondary Output Compare Data Buffer)
- CCPxBUFH/CCPxBUFL (CCPx Input Capture High/Low Buffers)

FIGURE 22-1: SCCPx CONCEPTUAL BLOCK DIAGRAM

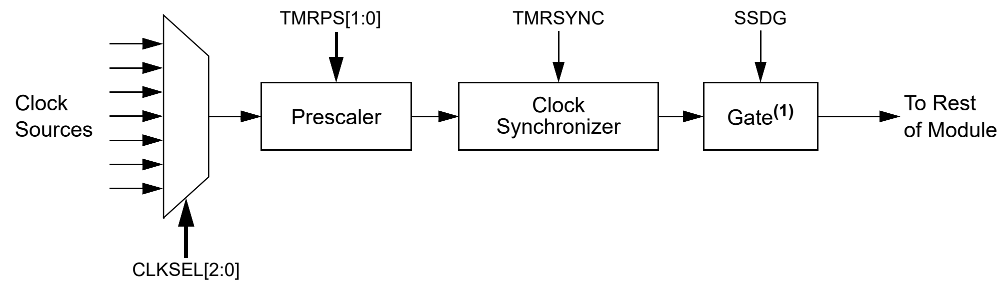


22.1 Time Base Generator

The Timer Clock Generator (TCG) generates a clock for the module's internal time base, using one of the clock signals already available on the microcontroller. This is used as the time reference for the module in its three major modes. The internal time base is shown in Figure 22-2.

There are eight inputs available to the clock generator, which are selected using the CLKSEL[2:0] bits (CCPxCON1L[10:8]). Available sources include the FRC and LPRC, the Secondary Oscillator and the TCLKI External Clock inputs. The system clock is the default source (CLKSEL[2:0] = 000).

FIGURE 22-2: TIMER CLOCK GENERATOR



Note 1: Gating is available in Timer modes only.

22.2 General Purpose Timer

Timer mode is selected when CCSEL = 0 and MOD[3:0] = 0000. The timer can function as a 32-bit timer or a dual 16-bit timer, depending on the setting of the T32 bit (Table 22-1).

TABLE 22-1: TIMER OPERATION MODE

T32 (CCPxCON1L[5])	Operating Mode
0	Dual Timer mode (16-bit)
1	Timer mode (32-bit)

Dual 16-Bit Timer mode provides a simple timer function with two independent 16-bit timer/counters. The primary timer uses CCPxTMRL and CCPxPRL. Only the primary timer can interact with other modules on the device. It generates the SCCPx sync out signals for use by other SCCP modules. It can also use the SYNC[4:0] bits signal generated by other modules.

The secondary timer uses CCPxTMRH and CCPxPRH. It is intended to be used only as a periodic interrupt source for scheduling CPU events. It does not generate an output sync/trigger signal like the primary time base. In Dual Timer mode, the CCPx Secondary Timer Period register, CCPxPRH, generates the SCCP compare event (CCPxIF) used by many other modules on the device.

The 32-Bit Timer mode uses the CCPxTMRL and CCPxTMRH registers, together, as a single 32-bit timer. When CCPxTMRL overflows, CCPxTMRH increments by one. This mode provides a simple timer function when it is important to track long time periods. Note that the T32 bit (CCPxCON1L[5]) should be set before the CCPxTMRL or CCPxPRH registers are written to initialize the 32-bit timer.

22.2.1 SYNC AND TRIGGER OPERATION

In both 16-bit and 32-bit modes, the timer can also function in either synchronization ("sync") or trigger operation. Both use the SYNC[4:0] bits (CCPxCON1H[4:0]) to determine the input signal source. The difference is how that signal affects the timer.

In sync operation, the timer Reset or clear occurs when the input selected by SYNC[4:0] is asserted. The timer immediately begins to count again from zero unless it is held for some other reason. Sync operation is used whenever the TRIGEN bit (CCPxCON1H[7]) is cleared. SYNC[4:0] can have any value, except '11111'.

In trigger operation, the timer is held in Reset until the input selected by SYNC[4:0] is asserted; when it occurs, the timer starts counting. Trigger operation is used whenever the TRIGEN bit is set. In Trigger mode, the timer will continue running after a trigger event as long as the CCPTRIG bit (CCPxSTATL[7]) is set. To clear CCPTRIG, the TRCLR bit (CCPxSTATL[5]) must be set to clear the trigger event, reset the timer and hold it at zero until another trigger event occurs. On the dsPIC33CDVL64MC106 device, trigger operation can only be used when the system clock is the time base source (CLKSEL[2:0] = 000).

FIGURE 22-3: DUAL 16-BIT TIMER MODE

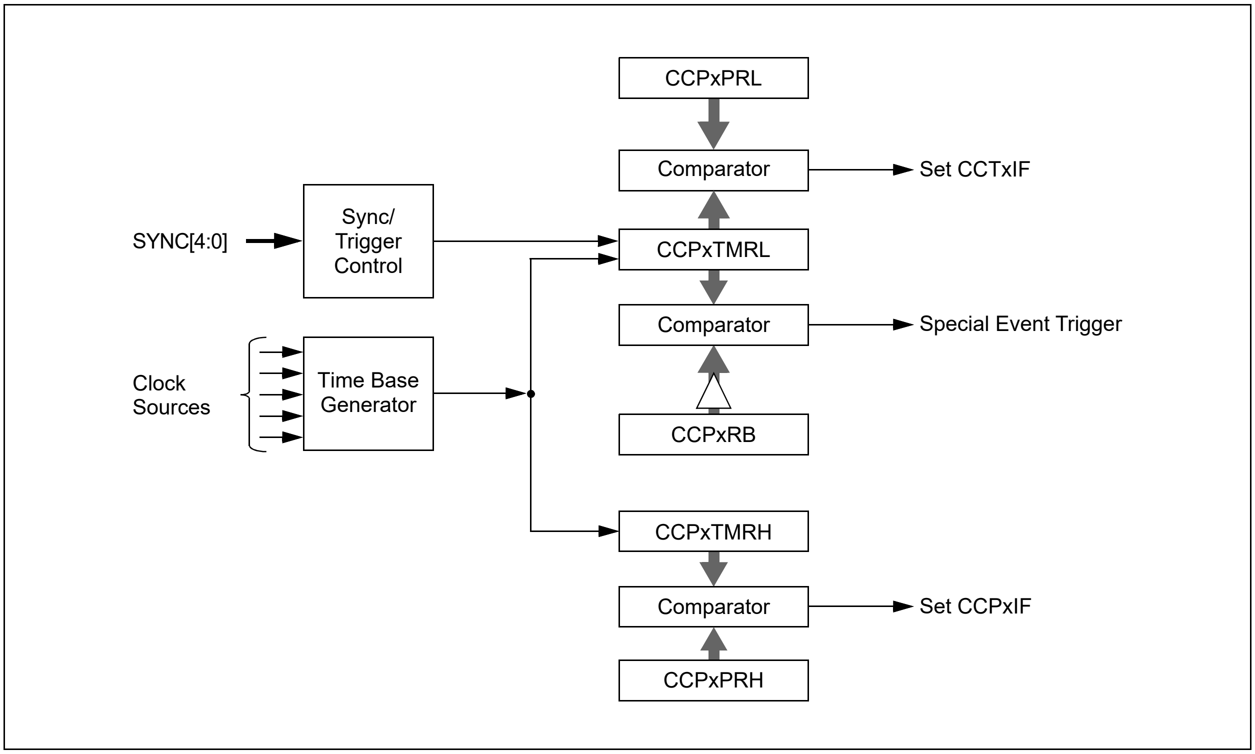
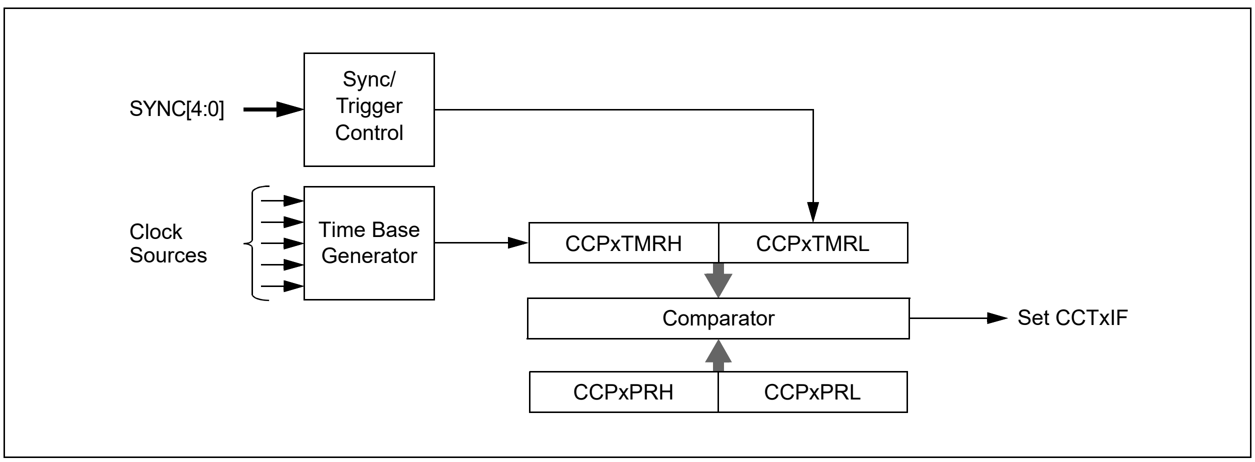


FIGURE 22-4: 32-BIT TIMER MODE



22.3 Output Compare Mode

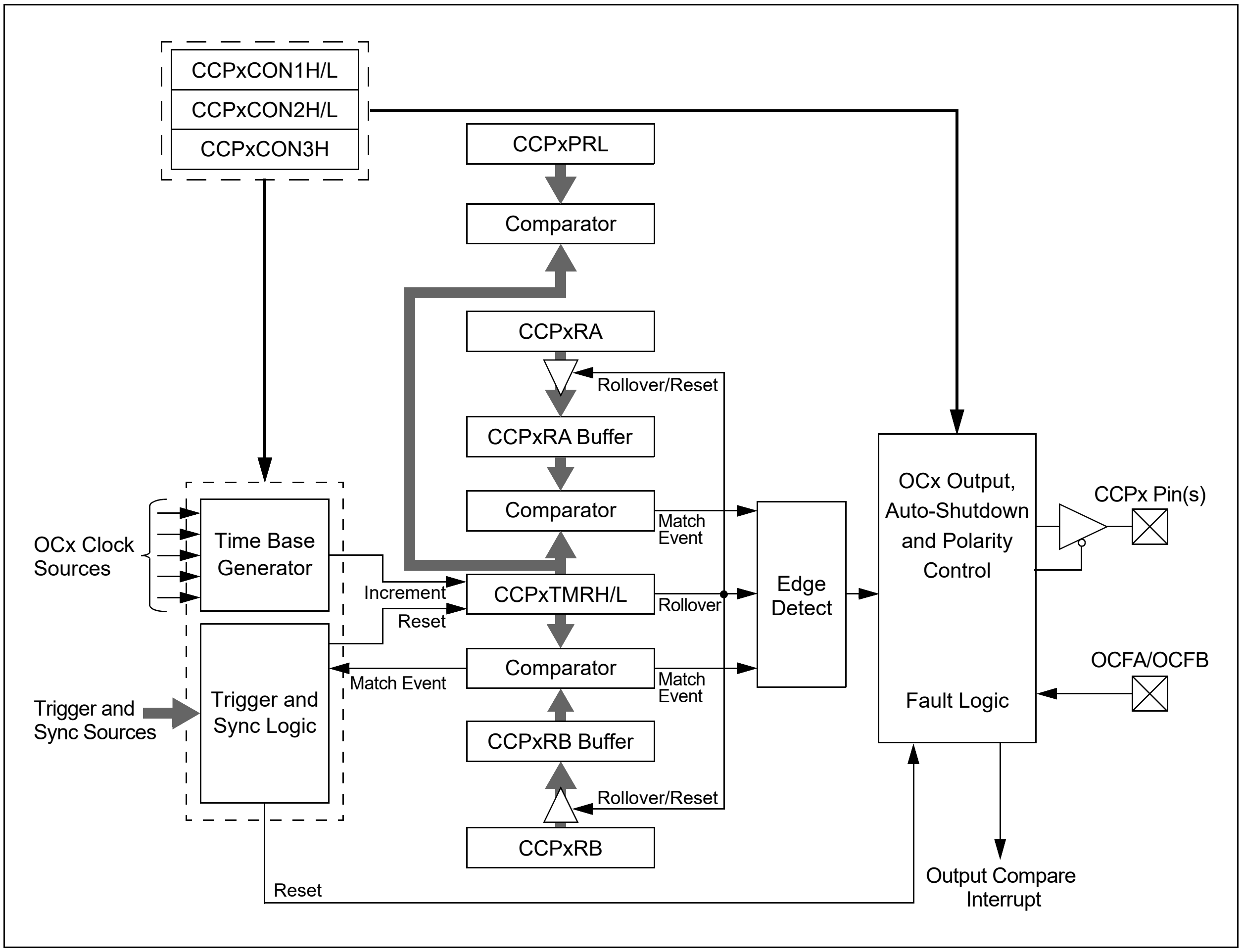
Output Compare mode compares the Timer register value with the value of one or two Compare registers, depending on its mode of operation. The Output Compare x module, on compare match events, has the ability to generate a single output transition or a train of output pulses. Like most PIC® MCU peripherals, the Output Compare x module can also generate interrupts on a compare match event.

Table 22-2 shows the various modes available in Output Compare modes.

TABLE 22-2: OUTPUT COMPARE x/PWMx MODES

MOD[3:0] (CCPxCON1L[3:0])	T32 (CCPxCON1L[5])	Operating Mode	
0001	0	Output High on Compare (16-bit)	Single-Edge mode
0001	1	Output High on Compare (32-bit)	
0010	0	Output Low on Compare (16-bit)	
0010	1	Output Low on Compare (32-bit)	
0011	0	Output Toggle on Compare (16-bit)	
0011	1	Output Toggle on Compare (32-bit)	
0100	0	Dual Edge Compare (16-bit)	Dual Edge mode
0101	0	Dual Edge Compare (16-bit buffered)	PWM mode

FIGURE 22-5: OUTPUT COMPARE x BLOCK DIAGRAM



22.4 Input Capture Mode

Input Capture mode is used to capture a timer value from an independent timer base, upon an event, on an input pin or other internal trigger source. The input capture features are useful in applications requiring frequency (time period) and pulse measurement. Figure 22-6 depicts a simplified block diagram of Input Capture mode.

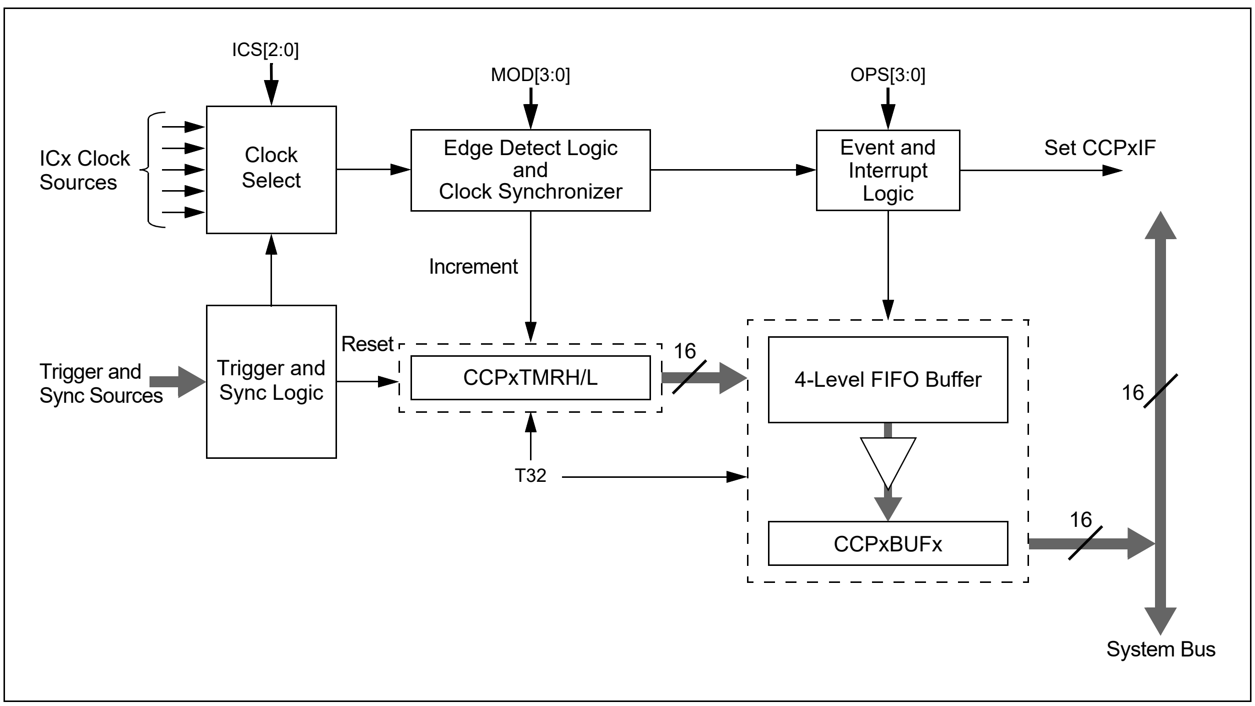
Input Capture mode uses a dedicated 16/32-bit, synchronous, up counting timer for the capture function. The timer value is written to the FIFO when a capture event occurs. The internal value may be read (with a synchronization delay) using the CCPxTMRH/L register.

To use Input Capture mode, the CCSEL bit (CCPxCON1L[4]) must be set. The T32 and the MOD[3:0] bits are used to select the proper Capture mode, as shown in Table 22-3.

TABLE 22-3: INPUT CAPTURE x MODES

MOD[3:0] (CCPxCON1L[3:0])	T32 (CCPxCON1L[5])	Operating Mode
0000	0	Edge Detect (16-bit capture)
0000	1	Edge Detect (32-bit capture)
0001	0	Every Rising (16-bit capture)
0001	1	Every Rising (32-bit capture)
0010	0	Every Falling (16-bit capture)
0010	1	Every Falling (32-bit capture)
0011	0	Every Rising/Falling (16-bit capture)
0011	1	Every Rising/Falling (32-bit capture)
0100	0	Every 4th Rising (16-bit capture)
0100	1	Every 4th Rising (32-bit capture)
0101	0	Every 16th Rising (16-bit capture)
0101	1	Every 16th Rising (32-bit capture)

FIGURE 22-6: INPUT CAPTURE x BLOCK DIAGRAM



22.5 Auxiliary Output

The SCCPx modules have an auxiliary (secondary) output that provides other peripherals access to internal module signals. The auxiliary output is intended to connect to other SCCP modules, or other digital peripherals, to provide these types of functions:

- Time Base Synchronization
- Peripheral Trigger and Clock Inputs
- Signal Gating

The type of output signal is selected using the AUXOUT[1:0] control bits (CCPxCON2H[4:3]). The type of output signal is also dependent on the module operating mode.

TABLE 22-4: AUXILIARY OUTPUT

AUXOUT[1:0]	CCSEL	MOD[3:0]	Comments	Signal Description
00	x	xxxx	Auxiliary Output Disabled	No Output
01	0	0000	Time Base Modes	Time Base Period Reset or Rollover
10				Special Event Trigger Output
11				No Output
01	0	0001 through 1111	Output Compare Modes	Time Base Period Reset or Rollover
10				Output Compare Event Signal
11				Output Compare Signal
01	1	xxxx	Input Capture Modes	Time Base Period Reset or Rollover
10				Reflects the Value of the ICDIS bit
11				Input Capture Event Signal

22.6 SCCP Control/Status Registers

REGISTER 22-1: CCPxCON1L: CCPx CONTROL 1 LOW REGISTERS

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCPON	—	CCPSIDL	CCPSLP	TMRSYNC	CLKSEL2	CLKSEL1	CLKSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMRPS1	TMRPS0	T32	CCSEL	MOD3	MOD2	MOD1	MOD0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **CCPON:** CCPx Module Enable bit
1 = Module is enabled with an operating mode specified by the MOD[3:0] control bits
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CCPSIDL:** CCPx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **CCPSLP:** CCPx Sleep Mode Enable bit
1 = Module continues to operate in Sleep modes
0 = Module does not operate in Sleep modes

bit 11 **TMRSYNC:** Time Base Clock Synchronization bit
1 = Asynchronous module time base clock is selected and synchronized to the internal system clocks (CLKSEL[2:0] ≠ 000)
0 = Synchronous module time base clock is selected and does not require synchronization (CLKSEL[2:0] = 000)

bit 10-8 **CLKSEL[2:0]:** CCPx Time Base Clock Select bits
111 = PPS TxCK input
110 = CLC4
101 = CLC3
100 = CLC2
011 = CLC1
010 = Reserved
001 = Reference Clock (REFCLKO)
000 = Peripheral Clock ($F_P = F_{OSC}/2$)

bit 7-6 **TMRPS[1:0]:** Time Base Prescale Select bits
11 = 1:64 Prescaler
10 = 1:16 Prescaler
01 = 1:4 Prescaler
00 = 1:1 Prescaler

bit 5 **T32:** 32-Bit Time Base Select bit
1 = Uses 32-bit time base for timer, single-edge output compare or input capture function
0 = Uses 16-bit time base for timer, single-edge output compare or input capture function

bit 4 **CCSEL:** Capture/Compare Mode Select bit
1 = Input Capture peripheral
0 = Output Compare/PWM/Timer peripheral (exact function is selected by the MOD[3:0] bits)

REGISTER 22-1: CCPxCON1L: CCPx CONTROL 1 LOW REGISTERS (CONTINUED)

bit 3-0 **MOD[3:0]:** CCPx Mode Select bits

For CCSEL = 1 (Input Capture modes):
1xxx = Reserved
011x = Reserved
0101 = Capture every 16th rising edge
0100 = Capture every 4th rising edge
0011 = Capture every rising and falling edge
0010 = Capture every falling edge
0001 = Capture every rising edge
0000 = Capture every rising and falling edge (Edge Detect mode)

For CCSEL = 0 (Output Compare/Timer modes):
1111 = External Input mode: Pulse generator is disabled, source is selected by ICS[2:0]
1110 = Reserved
110x = Reserved
10xx = Reserved
0111 = Reserved
0110 = Reserved
0101 = Dual Edge Compare mode, buffered
0100 = Dual Edge Compare mode
0011 = 16-Bit/32-Bit Single-Edge mode, toggles output on compare match
0010 = 16-Bit/32-Bit Single-Edge mode, drives output low on compare match
0001 = 16-Bit/32-Bit Single-Edge mode, drives output high on compare match
0000 = 16-Bit/32-Bit Timer mode, output functions are disabled

REGISTER 22-2: CCPxCON1H: CCPx CONTROL 1 HIGH REGISTERS

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OPSSRC[(1)]	RTRGEN[(2)]	—	—	OPS3[(3)]	OPS2[(3)]	OPS1[(3)]	OPS0[(3)]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGEN	ONESHOT	ALTSYNC	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **OPSSRC:** Output Postscaler Source Select bit[(1)]
1 = Output postscaler scales module trigger output events
0 = Output postscaler scales time base interrupt events

bit 14 **RTRGEN:** Retrigger Enable bit[(2)]
1 = Time base can be retriggered when TRIGEN bit = 1
0 = Time base may not be retriggered when TRIGEN bit = 1

bit 13-12 **Unimplemented:** Read as '0'

bit 11-8 **OPS3[3:0]:** CCPx Interrupt Output Postscale Select bits[(3)]
1111 = Interrupt every 16th time base period match
1110 = Interrupt every 15th time base period match
. . .
0100 = Interrupt every 5th time base period match
0011 = Interrupt every 4th time base period match or 4th input capture event
0010 = Interrupt every 3rd time base period match or 3rd input capture event
0001 = Interrupt every 2nd time base period match or 2nd input capture event
0000 = Interrupt after each time base period match or input capture event

bit 7 **TRIGEN:** CCPx Trigger Enable bit
1 = Trigger operation of time base is enabled
0 = Trigger operation of time base is disabled

bit 6 **ONESHOT:** One-Shot Trigger Mode Enable bit
1 = One-Shot Trigger mode is enabled; trigger duration is set by OSCNT[2:0]
0 = One-Shot Trigger mode is disabled

bit 5 **ALTSYNC:** CCPx Alternate Synchronization Output Signal Select bit
1 = An alternate signal is used as the module synchronization output signal
0 = The module synchronization output signal is the Time Base Reset/rollover event

bit 4-0 **SYNC[4:0]:** CCPx Synchronization Source Select bits
See Table 22-5 for the definition of inputs.

Note 1: This control bit has no function in Input Capture modes.
2: This control bit has no function when TRIGEN = 0.
3: Output postscale settings, from 1:5 to 1:16 (0100-1111), will result in a FIFO buffer overflow for Input Capture modes.

TABLE 22-5: SYNCHRONIZATION SOURCES

SYNC[4:0]	Synchronization Source
00000	None; Timer with Rollover on CCPxPR Match or FFFFh
00001	Sync Output SCCP1
00010	Sync Output SCCP2
00011	Sync Output SCCP3
00100	Sync Output SCCP4
00101-01000	Reserved
01001	INT0
01010	INT1
01011	INT2
01100	UART1 RX Edge Detect
01101	UART1 TX Edge Detect
01110	UART2 RX Edge Detect
01111	UART2 TX Edge Detect
10000	CLC1 Output
10001	CLC2 Output
10010	CLC3 Output
10011	CLC4 Output
10100	UART3 RX Edge Detect
10101	UART3 TX Edge Detect
10111	Comparator 1 Output
11000-11110	Reserved
11111	None; Timer with Auto-Rollover (FFFFh → 0000h)

REGISTER 22-3: CCPxCON2L: CCPx CONTROL 2 LOW REGISTERS

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
PWMRSEN	ASDGM	—	SSDG	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ASDG[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **PWMRSEN:** CCPx PWM Restart Enable bit
1 = ASEVT bit clears automatically at the beginning of the next PWM period, after the shutdown input has ended
0 = ASEVT bit must be cleared in software to resume PWM activity on output pins

bit 14 **ASDGM:** CCPx Auto-Shutdown Gate Mode Enable bit
1 = Waits until the next Time Base Reset or rollover for shutdown to occur
0 = Shutdown event occurs immediately

bit 13 **Unimplemented:** Read as ‘0’

bit 12 **SSDG:** CCPx Software Shutdown/Gate Control bit
1 = Manually forces auto-shutdown, timer clock gate or input capture signal gate event (setting of ASDGM bit still applies)
0 = Normal module operation

bit 11-8 **Unimplemented:** Read as ‘0’

bit 7-0 **ASDG[7:0]:** CCPx Auto-Shutdown/Gating Source Enable bits
1 = ASDGx Source n is enabled (see Table 22-6 for auto-shutdown/gating sources)
0 = ASDGx Source n is disabled

TABLE 22-6: AUTO-SHUTDOWN AND GATING SOURCES

ASDG[x] Bit	Auto-Shutdown/Gating Source			
	SCCP1	SCCP2	SCCP3	SCCP4
0	Comparator 1 Output			
2	OCFC			
3	OCFD			
4	ICM1[1]	ICM2[1]	ICM3[1]	ICM4[1]
5	CLC1[1]			
6	OCFA[1]			
7	OCFB[1]			

Note 1: Selected by Peripheral Pin Select (PPS).

REGISTER 22-4: CCPxCON2H: CCPx CONTROL 2 HIGH REGISTERS

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
OENSYNC	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICGSM1	ICGSM0	—	AUXOUT1	AUXOUT0	ICS2	ICS1	ICS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **OENSYNC:** Output Enable Synchronization bit
1 = Update by output enable bits occurs on the next Time Base Reset or rollover
0 = Update by output enable bits occurs immediately

bit 14-8 **Unimplemented:** Read as ‘0’

bit 7-6 **ICGSM[1:0]:** Input Capture Gating Source Mode Control bits
11 = Reserved
10 = One-Shot mode: Falling edge from gating source disables future capture events (ICDIS = 1)
01 = One-Shot mode: Rising edge from gating source enables future capture events (ICDIS = 0)
00 = Level-Sensitive mode: A high level from gating source will enable future capture events; a low level will disable future capture events

bit 5 **Unimplemented:** Read as ‘0’

bit 4-3 **AUXOUT[1:0]:** Auxiliary Output Signal on Event Selection bits
11 = Input capture or output compare event; no signal in Timer mode
10 = Signal output is defined by module operating mode (see Table 22-4)
01 = Time base rollover event (all modes)
00 = Disabled

bit 2-0 **ICS[2:0]:** Input Capture Source Select bits
111 = CLC4 output
110 = CLC3 output
101 = CLC2 output
100 = CLC1 output
011 = Reserved
010 = Reserved
001 = Comparator 1 output
000 = Input Capture ICMx pin (PPS)

REGISTER 22-5: CCPxCON3H: CCPx CONTROL 3 HIGH REGISTERS

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
OETRIG	OSCNT2	OSCNT1	OSCNT0	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0
—	—	POLACE	—	PSSACE1	PSSACE0	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **OETRIG:** CCPx Dead-Time Select bit
1 = For Triggered mode (TRIGEN = 1): Module does not drive enabled output pins until triggered
0 = Normal output pin operation

bit 14-12 **OSCNT[2:0]:** One-Shot Event Count bits
111 = Extends one-shot event by seven time base periods (eight time base periods total)
110 = Extends one-shot event by six time base periods (seven time base periods total)
101 = Extends one-shot event by five time base periods (six time base periods total)
100 = Extends one-shot event by four time base periods (five time base periods total)
011 = Extends one-shot event by three time base periods (four time base periods total)
010 = Extends one-shot event by two time base periods (three time base periods total)
001 = Extends one-shot event by one time base period (two time base periods total)
000 = Does not extend one-shot Trigger event

bit 11-6 **Unimplemented:** Read as ‘0’

bit 5 **POLACE:** CCPx Output Pins, OCMxA, OCMxC and OCMxE, Polarity Control bit
1 = Output pin polarity is active-low
0 = Output pin polarity is active-high

bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **PSSACE[1:0]:** PWMx Output Pins, OCMxA, OCMxC and OCMxE, Shutdown State Control bits
11 = Pins are driven active when a shutdown event occurs
10 = Pins are driven inactive when a shutdown event occurs
0x = Pins are tri-stated when a shutdown event occurs

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 22-6: CCPxSTATL: CCPx STATUS REGISTER LOW

U-0	U-0	U-0	U-0	U-0	W1-0	U-0	U-0
—	—	—	—	—	ICGARM	—	—
bit 15							bit 8

R-0	W1-0	W1-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W1 = Write ‘1’ Only bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10 **ICGARM:** Input Capture Gate Arm bit
A write of ‘1’ to this location will arm the input capture gating logic for a one-shot gate event when ICGSM[1:0] = 01 or 10. Bit always reads as ‘0’.

bit 9-8 **Unimplemented:** Read as ‘0’

bit 7 **CCPTRIG:** CCPx Trigger Status bit
1 = Timer has been triggered and is running
0 = Timer has not been triggered and is held in Reset

bit 6 **TRSET:** CCPx Trigger Set Request bit
Writes ‘1’ to this location to trigger the timer when TRIGEN = 1 (location always reads as ‘0’).

bit 5 **TRCLR:** CCPx Trigger Clear Request bit
Writes ‘1’ to this location to cancel the timer trigger when TRIGEN = 1 (location always reads as ‘0’).

bit 4 **ASEVT:** CCPx Auto-Shutdown Event Status/Control bit
1 = A shutdown event is in progress; CCPx outputs are in the Shutdown state
0 = CCPx outputs operate normally

bit 3 **SCEVT:** Single-Edge Compare Event Status bit
1 = A single-edge compare event has occurred
0 = A single-edge compare event has not occurred

bit 2 **ICDIS:** Input Capture x Disable bit
1 = Event on Input Capture x pin (ICx) does not generate a capture event
0 = Event on Input Capture x pin will generate a capture event

bit 1 **ICOV:** Input Capture x Buffer Overflow Status bit
1 = The Input Capture x FIFO buffer has overflowed
0 = The Input Capture x FIFO buffer has not overflowed

bit 0 **ICBNE:** Input Capture x Buffer Status bit
1 = Input Capture x buffer has data available
0 = Input Capture x buffer is empty

REGISTER 22-7: CCPxSTATH: CCPx STATUS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
—	—	—	PRLWIP	TMRHWIP	TMRLWIP	RBWIP	RAWIP
bit 7							bit 0

Legend:	C = Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-5 **Unimplemented:** Read as ‘0’

bit 4 **PRLWIP:** CCPxPRL Write In Progress Status bit
1 = An update to the CCPxPRL register with the buffered contents is in progress
0 = An update to the CCPxPRL register is not in progress

bit 3 **TMRHWIP:** CCPxTMRH Write In Progress Status bit
1 = An update to the CCPxTMRH register with the buffered contents is in progress
0 = An update to the CCPxTMRH register is not in progress

bit 2 **TMRLWIP:** CCPxTMRL Write In Progress Status bit
1 = An update to the CCPxTMRL register with the buffered contents is in progress
0 = An update to the CCPxTMRL register is not in progress

bit 1 **RBWIP:** CCPxRB Write In Progress Status bit
1 = An update to the CCPxRB register with the buffered contents is in progress
0 = An update to the CCPxRB register is not in progress

bit 0 **RAWIP:** CCPxRA Write In Progress Status bit
1 = An update to the CCPxRA register with the buffered contents is in progress
0 = An update to the CCPxRA register is not in progress

23.0 CONFIGURABLE LOGIC CELL (CLC)

Note 1: This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. For more information, refer to **"Configurable Logic Cell (CLC)"** (www.microchip.com/DS70005298). The information in this data sheet supersedes the information in the FRM.

The Configurable Logic Cell (CLC) module allows the user to specify combinations of signals as inputs to a logic function and to use the logic output to control other peripherals or I/O pins. This provides greater flexibility and potential in embedded designs, since the CLC module can operate outside the limitations of software execution, and supports a vast amount of output designs.

There are four input gates to the selected logic function. These four input gates select from a pool of up to 32 signals that are selected using four data source selection multiplexers. Figure 23-1 shows an overview of the module.

Figure 23-3 shows the details of the data source multiplexers and Figure 23-2 shows the logic input gate connections.

FIGURE 23-1: CLCx MODULE

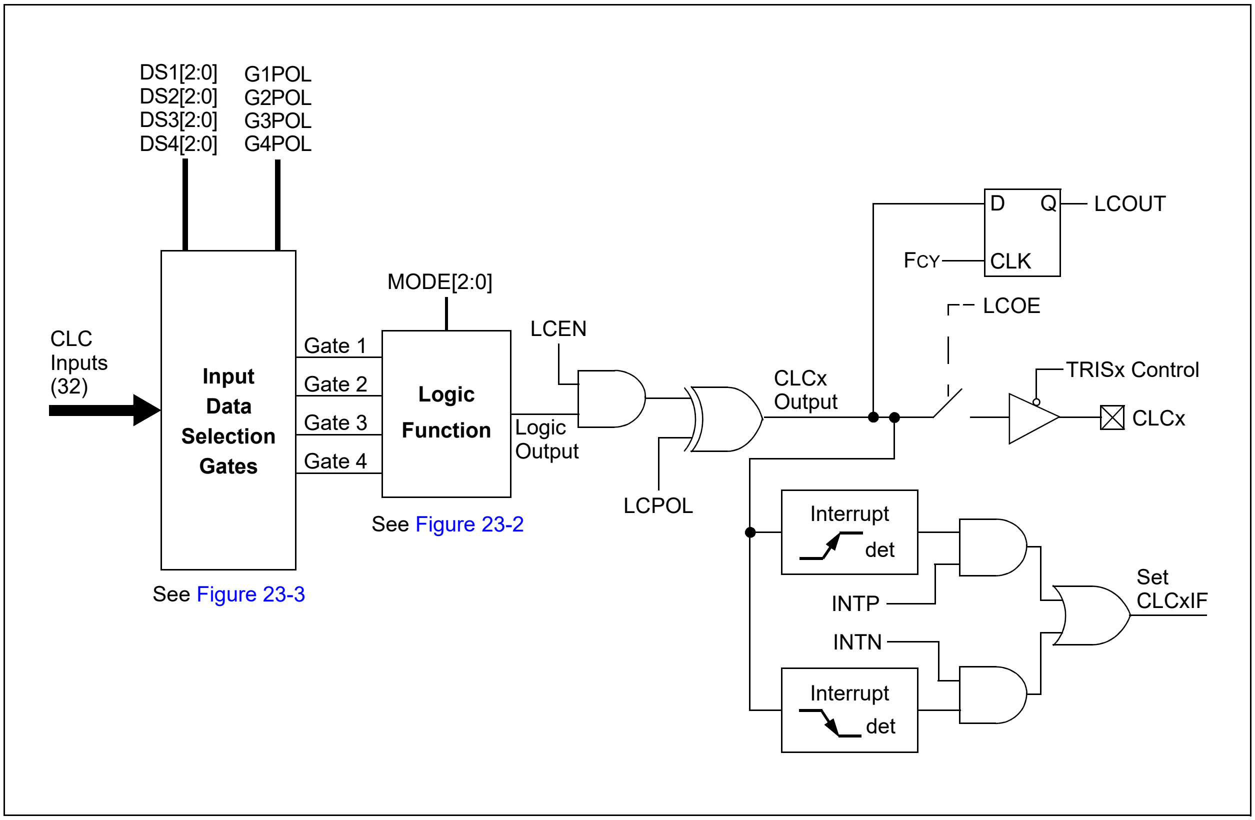


FIGURE 23-2: CLCx LOGIC FUNCTION COMBINATORIAL OPTIONS

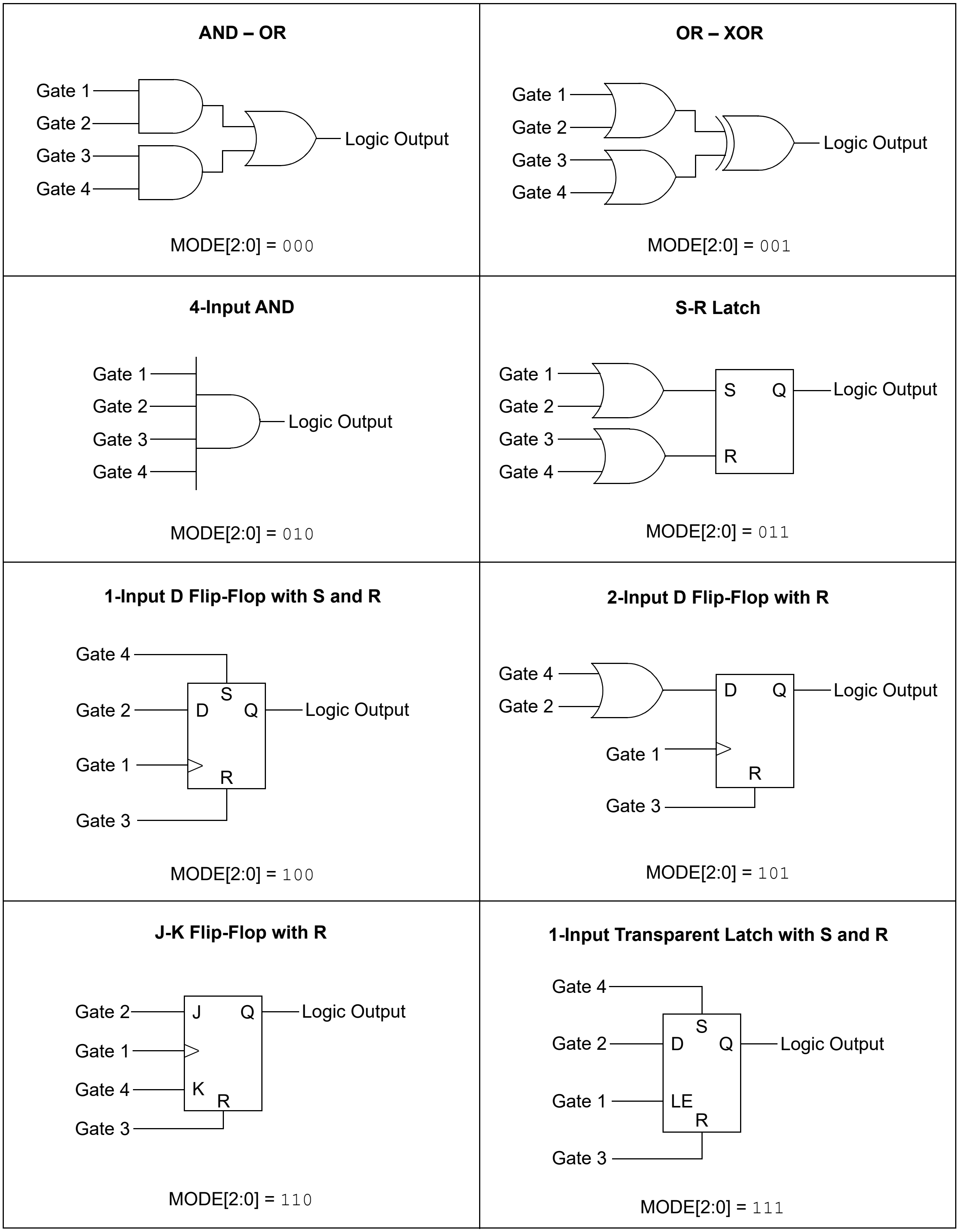
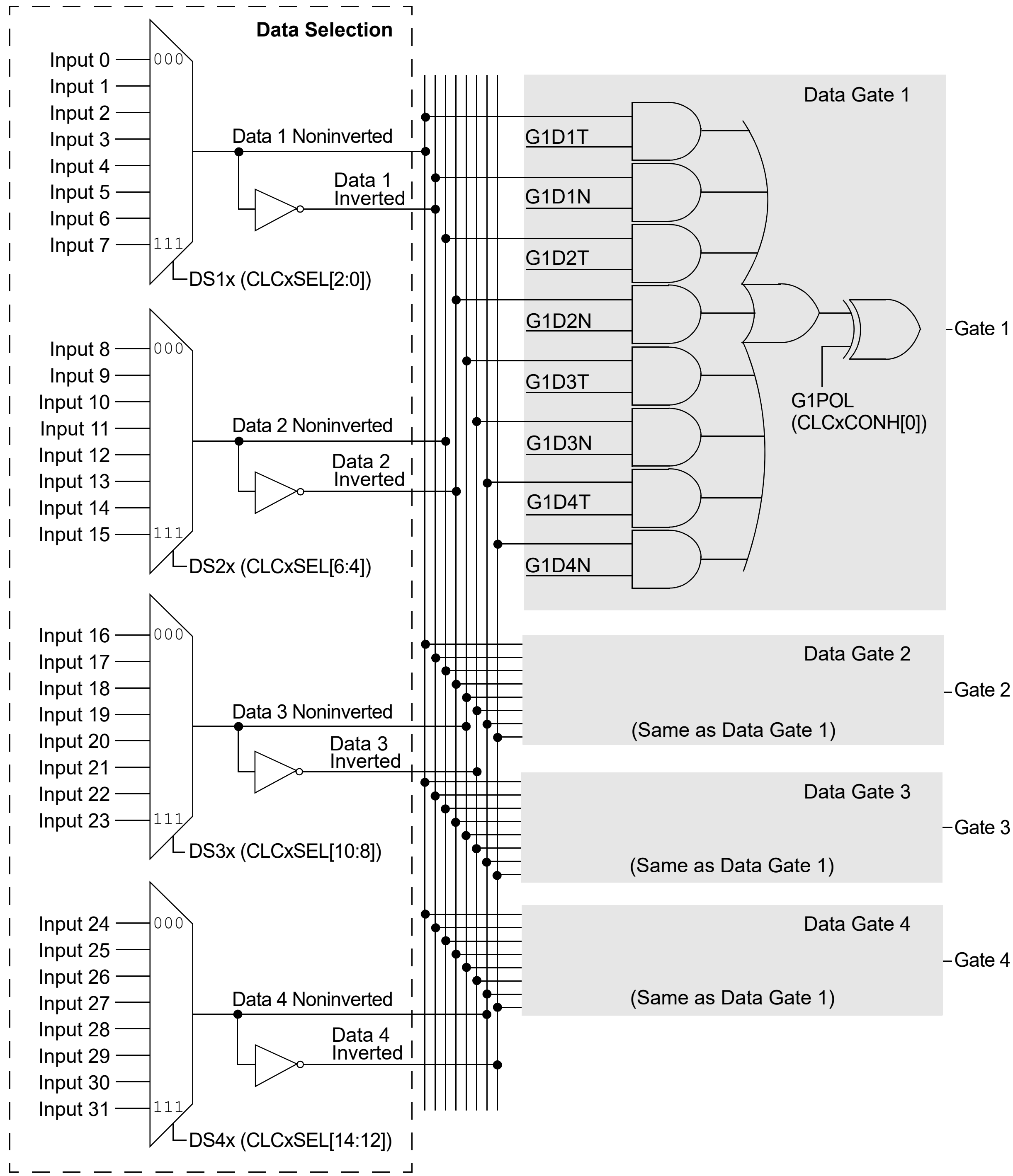


FIGURE 23-3: CLCx INPUT SOURCE SELECTION DIAGRAM



Note: All controls are undefined at power-up.

23.1 Control Registers

The CLCx module is controlled by the following registers:

- CLCxCONL
- CLCxCONH
- CLCxSEL
- CLCxGLSL
- CLCxGLSH

The CLCx Control registers (CLCxCONL and CLCxCONH) are used to enable the module and interrupts, control the output enable bit, select output polarity and select the logic function. The CLCx Control registers also allow the user to control the logic polarity of not only the cell output, but also some intermediate variables.

The CLCx Input MUX Select register (CLCxSEL) allows the user to select up to four data input sources using the four data input selection multiplexers. Each multiplexer has a list of eight data sources available.

The CLCx Gate Logic Input Select registers (CLCxGLSL and CLCxGLSH) allow the user to select which outputs from each of the selection MUXes are used as inputs to the input gates of the logic cell. Each data source MUX outputs both a true and a negated version of its output. All of these eight signals are enabled, ORed together by the logic cell input gates. If no inputs are selected (CLCxGLS = 0x00), the output will be zero or one, depending on the GxPOL bits.

REGISTER 23-1: CLCxCONL: CLCx CONTROL REGISTER (LOW)

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
LCEN	—	—	—	INTP	INTN	—	—
bit 15							bit 8

R-0	R-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
LCOE	LCOUT	LCPOL	—	—	MODE2	MODE1	MODE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **LCEN:** CLCx Enable bit
1 = CLCx is enabled and mixing input signals
0 = CLCx is disabled and has logic zero outputs

bit 14-12 **Unimplemented:** Read as ‘0’

bit 11 **INTP:** CLCx Positive Edge Interrupt Enable bit
1 = Interrupt will be generated when a rising edge occurs on LCOUT
0 = Interrupt will not be generated

bit 10 **INTN:** CLCx Negative Edge Interrupt Enable bit
1 = Interrupt will be generated when a falling edge occurs on LCOUT
0 = Interrupt will not be generated

bit 9-8 **Unimplemented:** Read as ‘0’

bit 7 **LCOE:** CLCx Port Enable bit
1 = CLCx port pin output is enabled
0 = CLCx port pin output is disabled

bit 6 **LCOUT:** CLCx Data Output Status bit
1 = CLCx output high
0 = CLCx output low

bit 5 **LCPOL:** CLCx Output Polarity Control bit
1 = The output of the module is inverted
0 = The output of the module is not inverted

bit 4-3 **Unimplemented:** Read as ‘0’

REGISTER 23-1: CLCxCONL: CLCx CONTROL REGISTER (LOW) (CONTINUED)

bit 2-0 **MODE[2:0]:** CLCx Mode bits
111 = Single input transparent latch with S and R
110 = JK flip-flop with R
101 = Two-input D flip-flop with R
100 = Single input D flip-flop with S and R
011 = SR latch
010 = Four-input AND
001 = Four-input OR-XOR
000 = Four-input AND-OR

REGISTER 23-2: CLCxCONH: CLCx CONTROL REGISTER (HIGH)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	G4POL	G3POL	G2POL	G1POL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **G4POL:** Gate 4 Polarity Control bit
1 = Channel 4 logic output is inverted when applied to the logic cell
0 = Channel 4 logic output is not inverted

bit 2 **G3POL:** Gate 3 Polarity Control bit
1 = Channel 3 logic output is inverted when applied to the logic cell
0 = Channel 3 logic output is not inverted

bit 1 **G2POL:** Gate 2 Polarity Control bit
1 = Channel 2 logic output is inverted when applied to the logic cell
0 = Channel 2 logic output is not inverted

bit 0 **G1POL:** Gate 1 Polarity Control bit
1 = Channel 1 logic output is inverted when applied to the logic cell
0 = Channel 1 logic output is not inverted

REGISTER 23-3: CLCxSEL: CLCx INPUT MUX SELECT REGISTER

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS4[2:0]			—	DS3[2:0]		
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS2[2:0]			—	DS1[2:0]		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **DS4[2:0]:** Data Selection MUX 4 Signal Selection bits
111 = SCCP3 auxiliary out
110 = SCCP1 auxiliary out
101 = CLCIND pin
100 = Reserved
011 = SPI1 Input (SDIx)(1)
010 = Reserved
001 = CLC2 output
000 = PWM Event A

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **DS3[2:0]:** Data Selection MUX 3 Signal Selection bits
111 = SCCP4 output compare
110 = SCCP3 output compare
101 = CLC4 out
100 = UART1 RX
011 = SPI1 Output (SDOx)(1)
010 = Reserved
001 = CLC1 output
000 = CLCINC I/O pin

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **DS2[2:0]:** Data Selection MUX 2 Signal Selection bits
111 = SCCP2 output compare
110 = SCCP1 output compare
101 = Reserved
100 = Reserved
011 = UART1 TX input corresponding to CLCx module
010 = Comparator 1 output
001 = Reserved
000 = CLCINB I/O pin

bit 3 **Unimplemented:** Read as ‘0’

Note 1: Valid only when SPI is used on PPS.

REGISTER 23-3: CLCxSEL: CLCx INPUT MUX SELECT REGISTER (CONTINUED)

bit 2-0 **DS1[2:0]:** Data Selection MUX 1 Signal Selection bits

`111` = SCCP4 auxiliary out
`110` = SCCP2 auxiliary out
`101` = Reserved
`100` = REFCLKO output
`011` = INTRC/LPRC clock source
`010` = CLC3 out
`001` = System clock (F_{CY})
`000` = CLCINA I/O pin

Note 1: Valid only when SPI is used on PPS.

REGISTER 23-4: CLCxGLSL: CLCx GATE LOGIC INPUT SELECT LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **G2D4T:** Gate 2 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 2
0 = Data Source 4 signal is disabled for Gate 2

bit 14 **G2D4N:** Gate 2 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 2
0 = Data Source 4 inverted signal is disabled for Gate 2

bit 13 **G2D3T:** Gate 2 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 2
0 = Data Source 3 signal is disabled for Gate 2

bit 12 **G2D3N:** Gate 2 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 2
0 = Data Source 3 inverted signal is disabled for Gate 2

bit 11 **G2D2T:** Gate 2 Data Source 2 True Enable bit
1 = Data Source 2 signal is enabled for Gate 2
0 = Data Source 2 signal is disabled for Gate 2

bit 10 **G2D2N:** Gate 2 Data Source 2 Negated Enable bit
1 = Data Source 2 inverted signal is enabled for Gate 2
0 = Data Source 2 inverted signal is disabled for Gate 2

bit 9 **G2D1T:** Gate 2 Data Source 1 True Enable bit
1 = Data Source 1 signal is enabled for Gate 2
0 = Data Source 1 signal is disabled for Gate 2

bit 8 **G2D1N:** Gate 2 Data Source 1 Negated Enable bit
1 = Data Source 1 inverted signal is enabled for Gate 2
0 = Data Source 1 inverted signal is disabled for Gate 2

bit 7 **G1D4T:** Gate 1 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 1
0 = Data Source 4 signal is disabled for Gate 1

bit 6 **G1D4N:** Gate 1 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 1
0 = Data Source 4 inverted signal is disabled for Gate 1

bit 5 **G1D3T:** Gate 1 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 1
0 = Data Source 3 signal is disabled for Gate 1

bit 4 **G1D3N:** Gate 1 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 1
0 = Data Source 3 inverted signal is disabled for Gate 1

REGISTER 23-4: CLCxGLSL: CLCx GATE LOGIC INPUT SELECT LOW REGISTER (CONTINUED)

bit 3	**G1D2T:** Gate 1 Data Source 2 True Enable bit 1 = Data Source 2 signal is enabled for Gate 1 0 = Data Source 2 signal is disabled for Gate 1
bit 2	**G1D2N:** Gate 1 Data Source 2 Negated Enable bit 1 = Data Source 2 inverted signal is enabled for Gate 1 0 = Data Source 2 inverted signal is disabled for Gate 1
bit 1	**G1D1T:** Gate 1 Data Source 1 True Enable bit 1 = Data Source 1 signal is enabled for Gate 1 0 = Data Source 1 signal is disabled for Gate 1
bit 0	**G1D1N:** Gate 1 Data Source 1 Negated Enable bit 1 = Data Source 1 inverted signal is enabled for Gate 1 0 = Data Source 1 inverted signal is disabled for Gate 1

REGISTER 23-5: CLCxGLSH: CLCx GATE LOGIC INPUT SELECT HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **G4D4T:** Gate 4 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 4
0 = Data Source 4 signal is disabled for Gate 4

bit 14 **G4D4N:** Gate 4 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 4
0 = Data Source 4 inverted signal is disabled for Gate 4

bit 13 **G4D3T:** Gate 4 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 4
0 = Data Source 3 signal is disabled for Gate 4

bit 12 **G4D3N:** Gate 4 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 4
0 = Data Source 3 inverted signal is disabled for Gate 4

bit 11 **G4D2T:** Gate 4 Data Source 2 True Enable bit
1 = Data Source 2 signal is enabled for Gate 4
0 = Data Source 2 signal is disabled for Gate 4

bit 10 **G4D2N:** Gate 4 Data Source 2 Negated Enable bit
1 = Data Source 2 inverted signal is enabled for Gate 4
0 = Data Source 2 inverted signal is disabled for Gate 4

bit 9 **G4D1T:** Gate 4 Data Source 1 True Enable bit
1 = Data Source 1 signal is enabled for Gate 4
0 = Data Source 1 signal is disabled for Gate 4

bit 8 **G4D1N:** Gate 4 Data Source 1 Negated Enable bit
1 = Data Source 1 inverted signal is enabled for Gate 4
0 = Data Source 1 inverted signal is disabled for Gate 4

bit 7 **G3D4T:** Gate 3 Data Source 4 True Enable bit
1 = Data Source 4 signal is enabled for Gate 3
0 = Data Source 4 signal is disabled for Gate 3

bit 6 **G3D4N:** Gate 3 Data Source 4 Negated Enable bit
1 = Data Source 4 inverted signal is enabled for Gate 3
0 = Data Source 4 inverted signal is disabled for Gate 3

bit 5 **G3D3T:** Gate 3 Data Source 3 True Enable bit
1 = Data Source 3 signal is enabled for Gate 3
0 = Data Source 3 signal is disabled for Gate 3

bit 4 **G3D3N:** Gate 3 Data Source 3 Negated Enable bit
1 = Data Source 3 inverted signal is enabled for Gate 3
0 = Data Source 3 inverted signal is disabled for Gate 3

REGISTER 23-5: CLCxGLSH: CLCx GATE LOGIC INPUT SELECT HIGH REGISTER (CONTINUED)

bit 3	**G3D2T:** Gate 3 Data Source 2 True Enable bit 1 = Data Source 2 signal is enabled for Gate 3 0 = Data Source 2 signal is disabled for Gate 3
bit 2	**G3D2N:** Gate 3 Data Source 2 Negated Enable bit 1 = Data Source 2 inverted signal is enabled for Gate 3 0 = Data Source 2 inverted signal is disabled for Gate 3
bit 1	**G3D1T:** Gate 3 Data Source 1 True Enable bit 1 = Data Source 1 signal is enabled for Gate 3 0 = Data Source 1 signal is disabled for Gate 3
bit 0	**G3D1N:** Gate 3 Data Source 1 Negated Enable bit 1 = Data Source 1 inverted signal is enabled for Gate 3 0 = Data Source 1 inverted signal is disabled for Gate 3

NOTES:

24.0 PERIPHERAL TRIGGER GENERATOR (PTG)

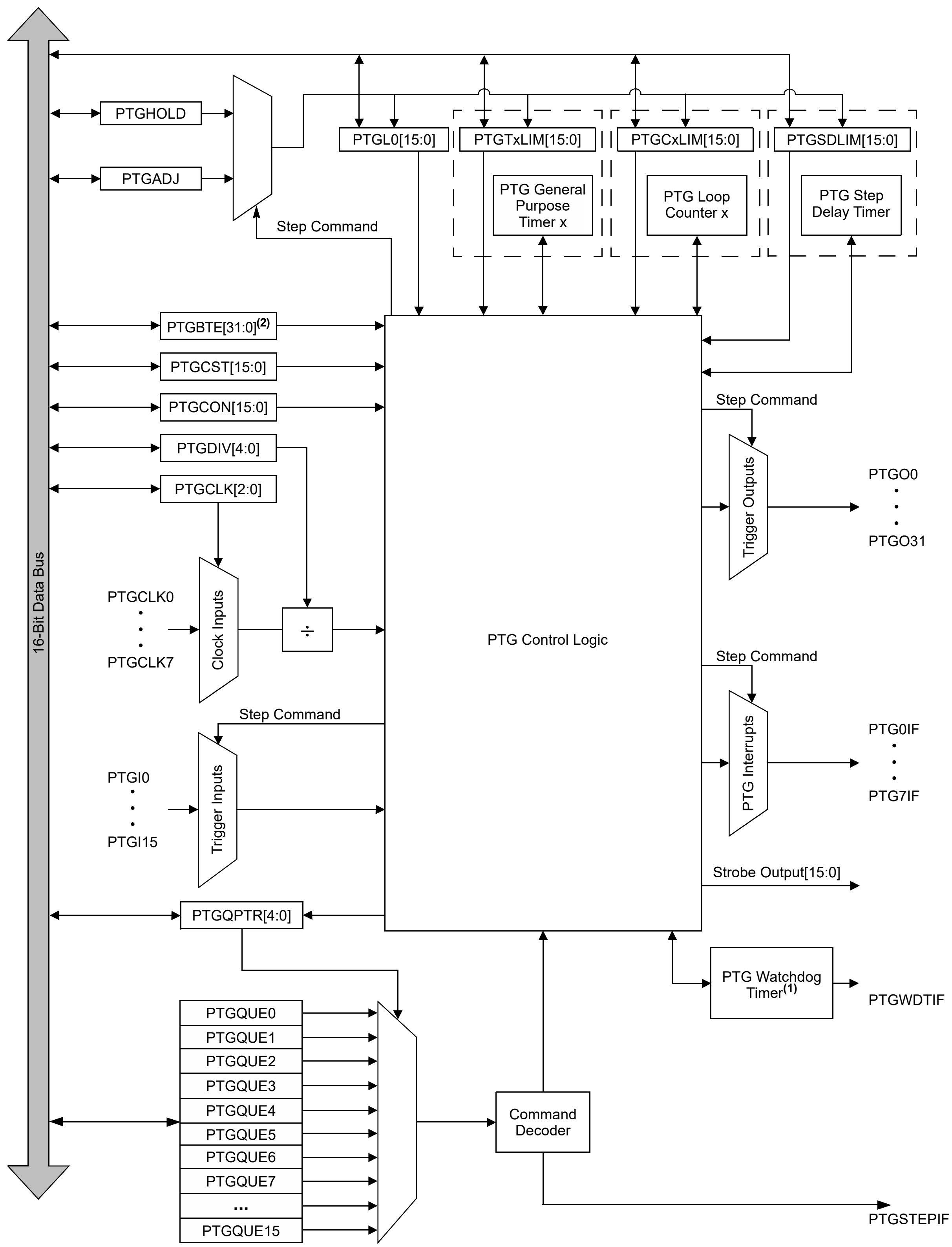
> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Peripheral Trigger Generator (PTG)"** (www.microchip.com/DS70000669).

The dsPIC33CDVL64MC106 Peripheral Trigger Generator (PTG) module is a user-programmable sequencer that is capable of generating complex trigger signal sequences to coordinate the operation of other peripherals. The PTG module is designed to interface with the modules, such as an Analog-to-Digital Converter (ADC), output compare and PWM modules, timers and interrupt controllers.

24.1 Features

- Behavior is Step Command Driven:
  - Step commands are eight bits wide.
- Commands are Stored in a Step Queue:
  - Queue depth is up to 32 entries.
  - Programmable Step execution time (Step delay)
- Supports the Command Sequence Loop:
  - Can be nested one-level deep.
  - Conditional or unconditional loop
  - Two 16-bit loop counters
- 15 Hardware Input Triggers:
  - Sensitive to either positive or negative edges, or a high or low level
- One Software Input Trigger
- Generates up to 32 Unique Output Trigger Signals
- Generates Two Types of Trigger Outputs:
  - Individual
  - Broadcast
- Generates up to Ten Unique Interrupt Signals
- Two 16-Bit General Purpose Timers
- Flexible Self-Contained Watchdog Timer (WDT) to Set an Upper Limit to Trigger Wait Time
- Single-Step Command Capability in Debug Mode
- Selectable Clock (System, Pulse-Width Modulator (PWM) or ADC)
- Programmable Clock Divider

FIGURE 24-1: PTG BLOCK DIAGRAM



Note 1: This is a dedicated Watchdog Timer for the PTG module and is independent of the device Watchdog Timer.

2: Some devices support only PTGBTE[15:0] (16 outputs).

24.2 PTG Control/Status Registers

REGISTER 24-1: PTGCST: PTG CONTROL/STATUS LOW REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	HC/R/W-0	R/W-0	R/W-0
PTGEN	—	PTGSIDL	PTGTOGL	—	PTGSWT[2]	PTGSSEN[3]	PTGIVIS
bit 15							bit 8

HC/R/W-0	HS/R/W-0	HS/HC/R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
PTGSTRT	PTGWDTO	PTGBUSY	—	—	—	PTGITM1[1]	PTGITM0[1]
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **PTGEN:** PTG Enable bit
1 = PTG is enabled
0 = PTG is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **PTGSIDL:** PTG Freeze in Debug Mode bit
1 = Halts PTG operation when device is Idle
0 = PTG operation continues when device is Idle

bit 12 **PTGTOGL:** PTG Toggle Trigger Output bit
1 = Toggles state of TRIG output for each execution of PTGTRIG
0 = Generates a single TRIG pulse for each execution of PTGTRIG

bit 11 **Unimplemented:** Read as ‘0’

bit 10 **PTGSWT:** PTG Software Trigger bit[2]
1 = If the PTG state machine is executing the “Wait for software trigger” Step command (OPTION[3:0] = 1010 or 1011), the command will complete and execution will continue
0 = No action other than to clear the bit

bit 9 **PTGSSEN:** PTG Single-Step Command bit[3]
1 = Enables single step when in Debug mode
0 = Disables single step

bit 8 **PTGIVIS:** PTG Counter/Timer Visibility bit
1 = Reading the PTGSDLIM, PTGCxLIM or PTGTxLIM registers returns the current values of their corresponding Counter/Timer registers (PTGSDLIM, PTGCxLIM and PTGTxLIM)
0 = Reading the PTGSDLIM, PTGCxLIM or PTGTxLIM registers returns the value of these Limit registers

bit 7 **PTGSTRT:** PTG Start Sequencer bit
1 = Starts to sequentially execute the commands (Continuous mode)
0 = Stops executing the commands

bit 6 **PTGWDTO:** PTG Watchdog Timer Time-out Status bit
1 = PTG Watchdog Timer has timed out
0 = PTG Watchdog Timer has not timed out

Note 1: These bits apply to the PTGWHI and PTGWLO commands only.
2: This bit is only used with the PTGCTRL Step command software trigger option.
3: The PTGSSEN bit may only be written when in Debug mode.

REGISTER 24-1: PTGCST: PTG CONTROL/STATUS LOW REGISTER (CONTINUED)

bit 5 **PTGBUSY:** PTG State Machine Busy bit
1 = PTG is running on the selected clock source; no SFR writes are allowed to PTGCLK[2:0] or PTGDIV[4:0]
0 = PTG state machine is not running

bit 4-2 **Unimplemented:** Read as '0'

bit 1-0 **PTGITM[1:0]:** PTG Input Trigger Operation Selection bit[1]
11 = Single-level detect with Step delay not executed on exit of command (regardless of the PTGCTRL command) (Mode 3)
10 = Single-level detect with Step delay executed on exit of command (Mode 2)
01 = Continuous edge detect with Step delay not executed on exit of command (regardless of the PTGCTRL command) (Mode 1)
00 = Continuous edge detect with Step delay executed on exit of command (Mode 0)

Note 1: These bits apply to the PTGWHI and PTGWLO commands only.
2: This bit is only used with the PTGCTRL Step command software trigger option.
3: The PTGSSEN bit may only be written when in Debug mode.

REGISTER 24-2: PTGCON: PTG CONTROL/STATUS HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGCLK2	PTGCLK1	PTGCLK0	PTGDIV4	PTGDIV3	PTGDIV2	PTGDIV1	PTGDIV0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTGPWD3	PTGPWD2	PTGPWD1	PTGPWD0	—	PTGWDT2	PTGWDT1	PTGWDT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **PTGCLK[2:0]:** PTG Module Clock Source Selection bits
111 = CLC1
110 = PLL VCO DIV 4 output
101 = Reserved
100 = Reserved
011 = Input from Timer1 Clock pin, T1CK
010 = PTG module clock source will be ADC clock
001 = PTG module clock source will be FOSC
000 = PTG module clock source will be FOSC/2 (FP)

bit 12-8 **PTGDIV[4:0]:** PTG Module Clock Prescaler (Divider) bits
11111 = Divide-by-32
11110 = Divide-by-31
. . .
00001 = Divide-by-2
00000 = Divide-by-1

bit 7-4 **PTGPWD[3:0]:** PTG Trigger Output Pulse-Width (in PTG clock cycles) bits
1111 = All trigger outputs are 16 PTG clock cycles wide
1110 = All trigger outputs are 15 PTG clock cycles wide
. . .
0001 = All trigger outputs are 2 PTG clock cycles wide
0000 = All trigger outputs are 1 PTG clock cycle wide

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **PTGWDT[2:0]:** PTG Watchdog Timer Time-out Selection bits
111 = Watchdog Timer will time out after 512 PTG clocks
110 = Watchdog Timer will time out after 256 PTG clocks
101 = Watchdog Timer will time out after 128 PTG clocks
100 = Watchdog Timer will time out after 64 PTG clocks
011 = Watchdog Timer will time out after 32 PTG clocks
010 = Watchdog Timer will time out after 16 PTG clocks
001 = Watchdog Timer will time out after 8 PTG clocks
000 = Watchdog Timer is disabled

REGISTER 24-3: PTGBTE: PTG BROADCAST TRIGGER ENABLE LOW REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGBTE[15:0]:** PTG Broadcast Trigger Enable bits
1 = Generates trigger when the broadcast command is executed
0 = Does not generate trigger when the broadcast command is executed

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-4: PTGBTEH: PTG BROADCAST TRIGGER ENABLE HIGH REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGBTE[31:16]:** PTG Broadcast Trigger Enable bits
1 = Generates trigger when the broadcast command is executed
0 = Does not generate trigger when the broadcast command is executed

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-5: PTGHOLD: PTG HOLD REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGHOLD[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGHOLD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGHOLD[15:0]:** PTG General Purpose Hold Register bits
This register holds the user-supplied data to be copied to the PTGTxLIM, PTGCxLIM, PTGSDLIM or PTGL0 register using the `PTGCOPY` command.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-6: PTGT0LIM: PTG TIMER0 LIMIT REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT0LIM[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT0LIM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGT0LIM[15:0]:** PTG Timer0 Limit Register bits
General Purpose Timer0 Limit register.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-7: PTGT1LIM: PTG TIMER1 LIMIT REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT1LIM[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT1LIM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGT1LIM[15:0]:** PTG Timer1 Limit Register bits
General Purpose Timer1 Limit register.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-8: PTGSDLIM: PTG STEP DELAY LIMIT REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGSDLIM[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGSDLIM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGSDLIM[15:0]:** PTG Step Delay Limit Register bits
This register holds a PTG Step delay value representing the number of additional PTG clocks between the start of a Step command and the completion of a Step command.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-9: PTGC0LIM: PTG COUNTER 0 LIMIT REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC0LIM[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC0LIM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGC0LIM[15:0]:** PTG Counter 0 Limit Register bits

This register is used to specify the loop count for the `PTGJMPC0` Step command or as a Limit register for the General Purpose Counter 0.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-10: PTGC1LIM: PTG COUNTER 1 LIMIT REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC1LIM[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC1LIM[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGC1LIM[15:0]:** PTG Counter 1 Limit Register bits

This register is used to specify the loop count for the `PTGJMPC1` Step command or as a Limit register for the General Purpose Counter 1.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-11: PTGADJ: PTG ADJUST REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGADJ[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGADJ[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGADJ[15:0]:** PTG Adjust Register bits
This register holds the user-supplied data to be added to the PTGTxLIM, PTGCxLIM, PTGSDLIM or PTGL0 register using the `PTGADD` command.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-12: PTGL0: PTG LITERAL 0 REGISTER[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGL0[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGL0[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PTGL0[15:0]:** PTG Literal 0 Register bits

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-13: PTGQPTR: PTG STEP QUEUE POINTER REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PTGQPTR[4:0]				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PTGQPTR[4:0]:** PTG Step Queue Pointer Register bits
This register points to the currently active Step command in the Step queue.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 24-14: PTGQUEn: PTG STEP QUEUE n POINTER REGISTER (n = 0-15)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP2n+1[7:0](2)							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP2n[7:0](2)							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **STEP2n+1[7:0]:** PTG Command 4n+1 bits(2)
A queue location for storage of the STEP2n+1 command byte, where ‘n’ is from PTGQUEn.

bit **STEP2n[7:0]:** PTG Command 4n+2 bits(2)
A queue location for storage of the STEP2n command byte, where ‘n’ are the odd numbered Step Queue Pointers.

Note 1: These bits are read-only when the module is executing Step commands.
2: Refer to Table 24-1 for the Step command encoding.

TABLE 24-1: PTG STEP COMMAND FORMAT AND DESCRIPTION

Step Command Byte			
STEPx[7:0]			
CMD[3:0]		OPTION[3:0]	
bit 7	bit 4	bit 3	bit 0

bit 7-4	Step Command	CMD[3:0]	Command Description
	PTGCTRL	0000	Execute the control command as described by the OPTION[3:0] bits.
	PTGADD	0001	Add contents of the PTGADJ register to the target register as described by the OPTION[3:0] bits.
	PTGCOPY		Copy contents of the PTGHOLD register to the target register as described by the OPTION[3:0] bits.
	PTGSTRB	001x	This command starts an ADC conversion of the channels specified in CMD[0] and OPTION[3:0] bits.
	PTGWHI	0100	Wait for a low-to-high edge input from a selected PTG trigger input as described by the OPTION[3:0] bits.
	PTGWLO	0101	Wait for a high-to-low edge input from a selected PTG trigger input as described by the OPTION[3:0] bits.
	—	0110	Reserved; do not use.(1)
	PTGIRQ	0111	Generate individual interrupt request as described by the OPTION[3:0] bits.
	PTGTRIG	100x	Generate individual trigger output as described by the bits, CMD[0]:OPTION[3:0].
	PTGJMP	101x	Copy the values contained in the bits, CMD[0]:OPTION[3:0], to the PTGQPTR register and jump to that Step queue.
	PTGJMPC0	110x	PTGC0 = PTGC0LIM: Increment the PTGQPTR register.
			PTGC0 ≠ PTGC0LIM: Increment Counter 0 (PTGC0) and copy the values contained in the bits, CMD[0]:OPTION[3:0], to the PTGQPTR register, and jump to that Step queue.
	PTGJMPC1	111x	PTGC1 = PTGC1LIM: Increment the PTGQPTR register.
			PTGC1 ≠ PTGC1LIM: Increment Counter 1 (PTGC1) and copy the values contained in the bits, CMD[0]:OPTION[3:0], to the PTGQPTR register, and jump to that Step queue.

Note 1: All reserved commands or options will execute, but they do not have any affect (i.e., execute as a `NOP` instruction).

TABLE 24-2: PTG COMMAND OPTIONS

bit 3-0	Step Command	OPTION[3:0]	Command Description
	PTGCTRL[(1)]	0000	NOP.
		0001	Reserved; do not use.
		0010	Disable Step delay timer (PTGSD).
		0011	Reserved; do not use.
		0100	Reserved; do not use.
		0101	Reserved; do not use.
		0110	Enable Step delay timer (PTGSD).
		0111	Reserved; do not use.
		1000	Start and wait for the PTG Timer0 to match the PTGT0LIM register.
		1001	Start and wait for the PTG Timer1 to match the PTGT1LIM register.
		1010	Wait for the software trigger (level, PTGSWT = 1).
		1011	Wait for the software trigger (positive edge, PTGSWT = 0 to 1).
		1100	Copy the PTGC0LIM register contents to the strobe output.
		1101	Copy the PTGC1LIM register contents to the strobe output.
		1110	Reserved; do not use.
		1111	Generate the triggers indicated in the PTGBTE register.
	PTGADD[(1)]	0000	Add the PTGADJ register contents to the PTGC0LIM register.
		0001	Add the PTGADJ register contents to the PTGC1LIM register.
		0010	Add the PTGADJ register contents to the PTGT0LIM register.
		0011	Add the PTGADJ register contents to the PTGT1LIM register.
		0100	Add the PTGADJ register contents to the PTGSDLIM register.
		0101	Add the PTGADJ register contents to the PTGL0 register.
		0110	Reserved; do not use.
		0111	Reserved; do not use.
	PTGCOPY[(1)]	1000	Copy the PTGHOLD register contents to the PTGC0LIM register.
		1001	Copy the PTGHOLD register contents to the PTGC1LIM register.
		1010	Copy the PTGHOLD register contents to the PTGT0LIM register.
		1011	Copy the PTGHOLD register contents to the PTGT1LIM register.
		1100	Copy the PTGHOLD register contents to the PTGSDLIM register.
		1101	Copy the PTGHOLD register contents to the PTGL0 register.
		1110	Reserved; do not use.
		1111	Reserved; do not use.

Note 1: All reserved commands or options will execute, but they do not have any affect (i.e., execute as a NOP instruction).

TABLE 24-2: PTG COMMAND OPTIONS (CONTINUED)

bit 3-0	Step Command	OPTION[3:0]	Option Description
	PTGWHI[(1)] or PTGWLO[(1)]	0000	PTGI0 (see Table 24-3 for input assignments).
		• • •	• • •
		1111	PTGI15 (see Table 24-3 for input assignments).
	PTGIRQ[(1)]	0000	Generate PTG Interrupt 0.
		• • •	• • •
		0111	Generate PTG Interrupt 7.
		1000	Reserved; do not use.
		• • •	• • •
		1111	Reserved; do not use.
	PTGTRIG	00000	PTGO0 (see Table 24-4 for output assignments).
		00001	PTGO1 (see Table 24-4 for output assignments).
		• • •	• • •
		11110	PTGO30 (see Table 24-4 for output assignments).
		11111	PTGO31 (see Table 24-4 for output assignments).
	PTGWHI[(1)] or PTGWLO[(1)]	0000	PTGI0 (see Table 24-3 for input assignments).
		• • •	• • •
		1111	PTGI15 (see Table 24-3 for input assignments).
	PTGIRQ[(1)]	0000	Generate PTG Interrupt 0.
		• • •	• • •
		0111	Generate PTG Interrupt 7.
		1000	Reserved; do not use.
		• • •	• • •
		1111	Reserved; do not use.
	PTGTRIG	00000	PTGO0 (see Table 24-4 for output assignments).
		00001	PTGO1 (see Table 24-4 for output assignments).

Note 1: All reserved commands or options will execute, but they do not have any affect (i.e., execute as a NOP instruction).

TABLE 24-3: PTG INPUT DESCRIPTIONS

PTG Input Number	PTG Input Description
PTG Trigger Input 0	Trigger Input from PWM1 ADC Trigger 2
PTG Trigger Input 1	Trigger Input from PWM2 ADC Trigger 2
PTG Trigger Input 2	Trigger Input from PWM3 ADC Trigger 2
PTG Trigger Input 3	Trigger Input from PWM4 ADC Trigger 2
PTG Trigger Input 4	Reserved
PTG Trigger Input 5	Reserved
PTG Trigger Input 6	Reserved
PTG Trigger Input 7	Trigger Input from SCCP4 Input Capture/Output Compare
PTG Trigger Input 8	Reserved
PTG Trigger Input 9	Trigger Input from Comparator 1
PTG Trigger Input 10	Reserved
PTG Trigger Input 11	Reserved
PTG Trigger Input 12	Trigger Input from CLC1
PTG Trigger Input 13	Trigger Input from ADC Common Interrupt
PTG Trigger Input 14	Reserved
PTG Trigger Input 15	Trigger Input from INT2 PPS

TABLE 24-4: PTG OUTPUT DESCRIPTIONS

PTG Output Number	PTG Output Description
PTGO0 to PTGO11	Reserved
PTGO12	ADC TRGSRC[30]
PTGO13 to PTGO23	Reserved
PTGO24	PPS Output RP46
PTGO25	PPS Output RP47
PTGO26	PPS Input RP6
PTGO27	PPS Input to P7
PTGO28	PPS Input to PTGO31
PTGO29 to PTGO31	Reserved

NOTES:

25.0 CURRENT BIAS GENERATOR (CBG)

Note 1: This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Current Bias Generator (CBG)"** (www.microchip.com/DS70005253).

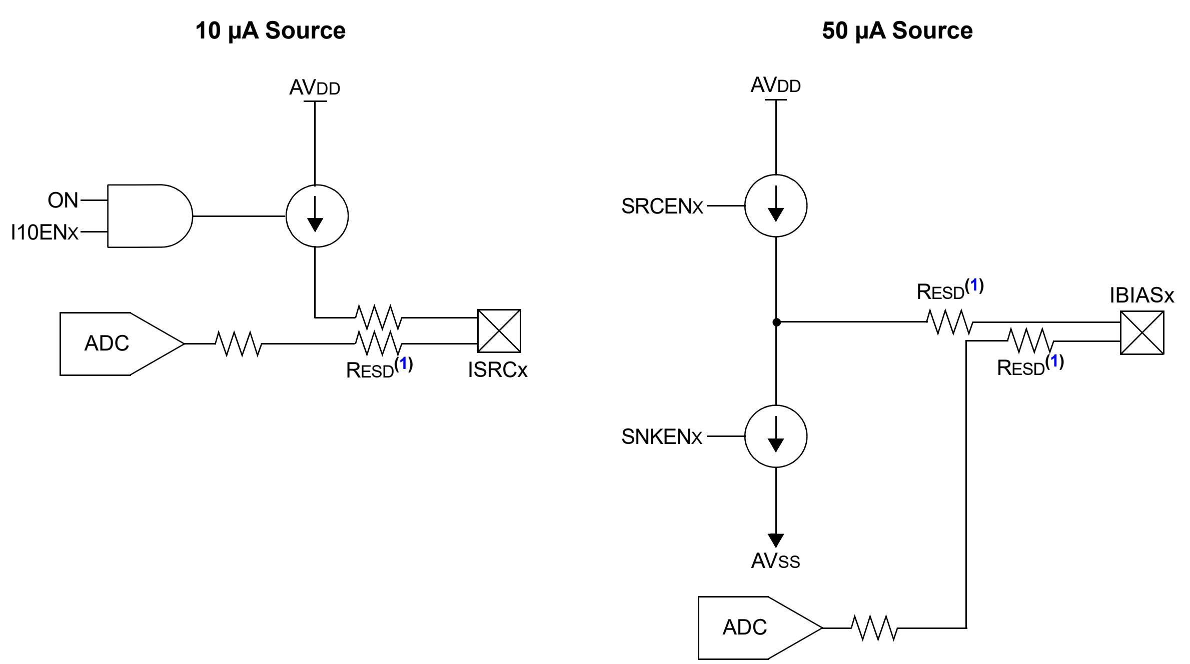
2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The Current Bias Generator (CBG) consists of two classes of current sources: 10 µA and 50 µA sources. The major features of each current source are:

- 10 µA Current Sources:
  - Current sourcing only
  - Up to four independent sources
- 50 µA Current Sources:
  - Selectable current sourcing or sinking
  - Selectable current mirroring for sourcing and sinking

A simplified block diagram of the CBG module is shown in Figure 25-1.

FIGURE 25-1: CONSTANT-CURRENT SOURCE MODULE BLOCK DIAGRAM(2)



Note 1: RESD is typically 350 Ohms.

2: In Figure 25-1, the ADC analog input is shown for clarity. Each analog peripheral connected to the pin has a separate Electrostatic Discharge (ESD) resistor.

25.1 Current Bias Generator Control Registers

REGISTER 25-1: BIASCON: CURRENT BIAS GENERATOR CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	I10EN3	I10EN2(1)	I10EN1(2)	I10EN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ON:** Current Bias Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14-4 **Unimplemented:** Read as ‘0’

bit 3 **I10EN3:** 10 µA Enable for Output 3 bit
1 = 10 µA output is enabled
0 = 10 µA output is disabled

bit 2 **I10EN2:** 10 µA Enable for Output 2 bit(1)
1 = 10 µA output is enabled
0 = 10 µA output is disabled

bit 1 **I10EN1:** 10 µA Enable for Output 1 bit(2)
1 = 10 µA output is enabled
0 = 10 µA output is disabled

bit 0 **I10EN0:** 10 µA Enable for Output 0 bit
1 = 10 µA output is enabled
0 = 10 µA output is disabled

Note 1: This bit is only available for the 36 and 48-pin package devices.
2: This bit is only available for the 48-pin package devices.

REGISTER 25-2: IBIASCONH: CURRENT BIAS GENERATOR 50 μA CURRENT SOURCE CONTROL HIGH REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN3[(1)]	SHRSNKEN3[(1)]	GENSRCEN3[(1)]	GENSNKEN3[(1)]	SRCEN3[(1)]	SNKEN3[(1)]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN2[(1)]	SHRSNKEN2[(1)]	GENSRCEN2[(1)]	GENSNKEN2[(1)]	SRCEN2[(1)]	SNKEN2[(1)]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **SHRSRCEN3:** Share Source Enable for Output #3 bit[(1)]
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 12 **SHRSNKEN3:** Share Sink Enable for Output #3 bit[(1)]
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 11 **GENSRCEN3:** Generated Source Enable for Output #3 bit[(1)]
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 10 **GENSNKEN3:** Generated Sink Enable for Output #3 bit[(1)]
1 = Source generates the current sink mirror reference
0 = Source does not generate the current sink mirror reference

bit 9 **SRCEN3:** Source Enable for Output #3 bit[(1)]
1 = Current source is enabled
0 = Current source is disabled

bit 8 **SNKEN3:** Sink Enable for Output #3 bit[(1)]
1 = Current sink is enabled
0 = Current sink is disabled

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **SHRSRCEN2:** Share Source Enable for Output #2 bit[(1)]
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 4 **SHRSNKEN2:** Share Sink Enable for Output #2 bit[(1)]
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 3 **GENSRCEN2:** Generated Source Enable for Output #2 bit[(1)]
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 2 **GENSNKEN2:** Generated Sink Enable for Output #2 bit[(1)]
1 = Source generates the current sink mirror reference
0 = Source does not generate the current sink mirror reference

Note 1: This bit is only available in 48-pin package devices.

REGISTER 25-2: IBIASCONH: CURRENT BIAS GENERATOR 50 μA CURRENT SOURCE CONTROL HIGH REGISTER (CONTINUED)

bit 1 **SRCEN2:** Source Enable for Output #2 bit[1]
1 = Current source is enabled
0 = Current source is disabled

bit 0 **SNKEN2:** Sink Enable for Output #2 bit[1]
1 = Current sink is enabled
0 = Current sink is disabled

Note 1: This bit is only available in 48-pin package devices.

REGISTER 25-3: IBIASCONL: CURRENT BIAS GENERATOR 50 μA CURRENT SOURCE CONTROL LOW REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN1[1]	SHRSNKEN1[1]	GENSRCEN1[1]	GENSNKEN1[1]	SRCEN1[1]	SNKEN1[1]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN0[1]	SHRSNKEN0[1]	GENSRCEN0[1]	GENSNKEN0[1]	SRCEN0[1]	SNKEN0[1]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **SHRSRCEN1:** Share Source Enable for Output #1 bit[1]
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 12 **SHRSNKEN1:** Share Sink Enable for Output #1 bit[1]
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 11 **GENSRCEN1:** Generated Source Enable for Output #1 bit[1]
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 10 **GENSNKEN1:** Generated Sink Enable for Output #1 bit[1]
1 = Source generates the current sink mirror reference
0 = Source does not generate the current sink mirror reference

bit 9 **SRCEN1:** Source Enable for Output #1 bit[1]
1 = Current source is enabled
0 = Current source is disabled

bit 8 **SNKEN1:** Sink Enable for Output #1 bit[1]
1 = Current sink is enabled
0 = Current sink is disabled

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **SHRSRCEN0:** Share Source Enable for Output #0 bit[1]
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 4 **SHRSNKEN0:** Share Sink Enable for Output #0 bit[1]
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 3 **GENSRCEN0:** Generated Source Enable for Output #0 bit[1]
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 2 **GENSNKEN0:** Generated Sink Enable for Output #0 bit[1]
1 = Source generates the current sink mirror reference
0 = Source does not generate the current sink mirror reference

Note 1: This bit is only available in 36 and 48-pin package devices.

REGISTER 25-3: IBIASCONL: CURRENT BIAS GENERATOR 50 µA CURRENT SOURCE CONTROL LOW REGISTER (CONTINUED)

bit 1 **SRCEN0:** Source Enable for Output #0 bit[(1)]
1 = Current source is enabled
0 = Current source is disabled

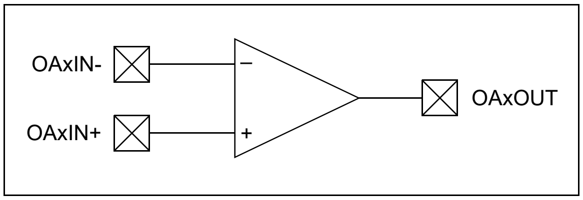
bit 0 **SNKEN0:** Sink Enable for Output #0 bit[(1)]
1 = Current sink is enabled
0 = Current sink is disabled

Note 1: This bit is only available in 36 and 48-pin package devices.

26.0 OPERATIONAL AMPLIFIER

The dsPIC33CDVL64MC106 devices implement three instances of operational amplifiers (op amps). The op amps can be used for a wide variety of purposes, including signal conditioning and filtering. The three op amps are functionally identical. The block diagram for a single amplifier is shown in Figure 26-1.

FIGURE 26-1: SINGLE OPERATIONAL AMPLIFIER BLOCK DIAGRAM



The op amps are controlled by two SFR registers: AMPCON1L and AMPCON1H. They remain in a Low-Power state until the AMPON bit is set. Each op amp can then be enabled independently by setting the corresponding AMPENx bit (x = 1, 2, 3).

The NCHDISx bit provides some flexibility regarding input range versus Integral Nonlinearity (INL). When NCHDISx = 0 (default), the op amps have a wider input voltage range (see Table 33-37 in **Section 33.0 "Electrical Characteristics"**). When NCHDISx = 1, the wider input range is traded for improved INL performance (lower INL).

26.1 Operational Amplifier Control Registers

REGISTER 26-1: AMPCON1L: OP AMP CONTROL REGISTER LOW

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
AMPON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	AMPEN3	AMPEN2	AMPEN1
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **AMPON:** Op Amp Enable/On bit
1 = Enables op amp modules if their respective AMPENx bits are also asserted
0 = Disables all op amp modules

bit 14-3 **Unimplemented:** Read as ‘0’

bit 2 **AMPEN3:** Op Amp #3 Enable bit
1 = Enables Op Amp #3 if the AMPON bit is also asserted
0 = Disables Op Amp #3

bit 1 **AMPEN2:** Op Amp #2 Enable bit
1 = Enables Op Amp #2 if the AMPON bit is also asserted
0 = Disables Op Amp #2

bit 0 **AMPEN1:** Op Amp #1 Enable bit
1 = Enables Op Amp #1 if the AMPON bit is also asserted
0 = Disables Op Amp #1

REGISTER 26-2: AMPCON1H: OP AMP CONTROL REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NCHDIS3[(1)]	NCHDIS2	NCHDIS1
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2 **NCHDIS3:** Op Amp #3 N Channel Disable bit[(1)]
1 = Disables Op Amp #3 N channel input stage; reduced INL, but lowered input voltage range
0 = Wide input range for Op Amp #3

bit 1 **NCHDIS2:** Op Amp #2 N Channel Disable bit
1 = Disables Op Amp #2 N channel input stage; reduced INL, but lowered input voltage range
0 = Wide input range for Op Amp #2

bit 0 **NCHDIS1:** Op Amp #1 N Channel Disable bit
1 = Disables Op Amp #1 N channel input stage; reduced INL, but lowered input voltage range
0 = Wide input range for Op Amp #1

Note 1: This bit is not available on 28-pin devices.

NOTES:

27.0 DEADMAN TIMER (DMT)

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Deadman Timer (DMT)"** (www.microchip.com/DS70005155).

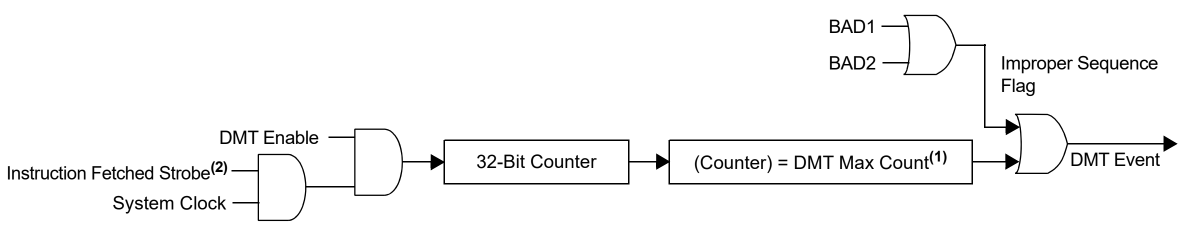
The primary function of the Deadman Timer (DMT) is to interrupt the processor in the event of a software malfunction. The DMT, which works on the system clock, is a free-running instruction fetch timer, which is clocked whenever an instruction fetch occurs, until a count match occurs. Instructions are not fetched when the processor is in Sleep mode.

DMT can be enabled in the Configuration fuse or by software in the DMTCON register by setting the ON bit. A DMT event results in a DMT interrupt on this device. The DMT consists of a 32-bit counter with a time-out count match value, as specified by the two 16-bit Configuration Fuse registers: FDMTCNTL and FDMTCNTH.

A DMT is typically used in mission-critical and safety-critical applications, where any single failure of the software functionality and sequencing must be detected.

Figure 27-1 shows a block diagram of the Deadman Timer module.

FIGURE 27-1: DEADMAN TIMER BLOCK DIAGRAM



Note 1: DMT Max Count is controlled by the initial value of the FDMTCNTL and FDMTCNTH Configuration registers.
2: DMT window interval is controlled by the value of the FDMTIVTL and FDMTIVTH Configuration registers.

27.1 Deadman Timer Control/Status Registers

REGISTER 27-1: DMTCON: DEADMAN TIMER CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ON[1,2]	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ON:** DMT Module Enable bit[1]
1 = Deadman Timer module is enabled
0 = Deadman Timer module is not enabled

bit 14-0 **Unimplemented:** Read as ‘0’

Note 1: This bit has control only when DMTDIS = 0 in the FDMT register.
2: DMT cannot be disabled in software. Writing ‘0’ to this bit has no effect.

REGISTER 27-2: DMTPRECLR: DEADMAN TIMER PRECLEAR REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP1[7:0][1]							
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **STEP1[7:0]:** DMT Preclear Enable bits[1]
01000000 = Enables the Deadman Timer preclear (Step 1)
All Other Write Patterns = Sets the BAD1 flag

bit 7-0 **Unimplemented:** Read as ‘0’

Note 1: Bits[15:8] are cleared when the DMT counter is reset by writing a correct sequence of STEP1 and STEP2. STEP1 is also cleared if DMTCLR[STEP2] is loaded with the correct value in the correct sequence.

REGISTER 27-3: DMTCLR: DEADMAN TIMER CLEAR REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP2[7:0][(1)]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **STEP2[7:0]:** DMT Clear Timer bits[(1)]

`00001000` = Clears STEP1[7:0], STEP2[7:0] and the Deadman Timer if preceded by the correct loading of the STEP1[7:0] bits in the correct sequence. The write to these bits may be verified by reading the DMTCNTL/H register and observing the counter being reset.

All Other Write Patterns = Sets the BAD2 bit; the value of STEP1[7:0] will remain unchanged, and the new value being written to STEP2[7:0] will be captured.

Note 1: Bits[7:0] are cleared when the DMT counter is reset by writing a correct sequence of STEP1 and STEP2.

REGISTER 27-4: DMTSTAT: DEADMAN TIMER STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

HC/R-0	HC/R-0	HC/R-0	U-0	U-0	U-0	U-0	R-0
BAD1[1]	BAD2[1]	DMTEVENT[1]	—	—	—	—	WINOPN
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **BAD1:** Deadman Timer Bad STEP1[7:0] Value Detect bit[1]
1 = Incorrect STEP1[7:0] value was detected
0 = Incorrect STEP1[7:0] value was not detected

bit 6 **BAD2:** Deadman Timer Bad STEP2[7:0] Value Detect bit[1]
1 = Incorrect STEP2[7:0] value was detected
0 = Incorrect STEP2[7:0] value was not detected

bit 5 **DMTEVENT:** Deadman Timer Event bit[1]
1 = Deadman Timer event was detected (counter expired, or bad STEP1[7:0] or STEP2[7:0] value was entered prior to counter increment)
0 = Deadman Timer event was not detected

bit 4-1 **Unimplemented:** Read as '0'

bit 0 **WINOPN:** Deadman Timer Clear Window bit
1 = Deadman Timer clear window is open
0 = Deadman Timer clear window is not open

Note 1: The BAD1, BAD2 and DMTEVENT bits are cleared only on a Reset.

REGISTER 27-5: DMTCNTL: DEADMAN TIMER COUNT REGISTER LOW

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **COUNTER[15:0]:** Read Current Contents of Lower DMT Counter bits

REGISTER 27-6: DMTCNTH: DEADMAN TIMER COUNT REGISTER HIGH

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[31:24]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **COUNTER[31:16]:** Read Current Contents of Higher DMT Counter bits

REGISTER 27-7: DMTPSCNTL: DMT POST-CONFIGURE COUNT STATUS REGISTER LOW

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PSCNT[15:0]:** Lower DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTL Configuration register.

REGISTER 27-8: DMTPSCNTH: DMT POST-CONFIGURE COUNT STATUS REGISTER HIGH

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[31:24]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PSCNT[31:16]:** Higher DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTH Configuration register.

REGISTER 27-9: DMTPSINTVL: DMT POST-CONFIGURE INTERVAL STATUS REGISTER LOW

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PSINTV[15:0]:** Lower DMT Window Interval Configuration Status bits
This is always the value of the FDMTIVTL Configuration register.

REGISTER 27-10: DMTPSINTVH: DMT POST-CONFIGURE INTERVAL STATUS REGISTER HIGH

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[31:24]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **PSINTV[31:16]:** Higher DMT Window Interval Configuration Status bits
This is always the value of the FDMTIVTH Configuration register.

REGISTER 27-11: DMTHOLDREG: DMT HOLD REGISTER[(1)]

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
UPRCNT[15:8]							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
UPRCNT[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **UPRCNT[15:0]:** DMTCNTH Register Value when DMTCNTL and DMTCNTH were Last Read bits

Note 1: The DMTHOLDREG register is initialized to ‘0’ on Reset and is only loaded when the DMTCNTL and DMTCNTH registers are read.

28.0 32-BIT PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

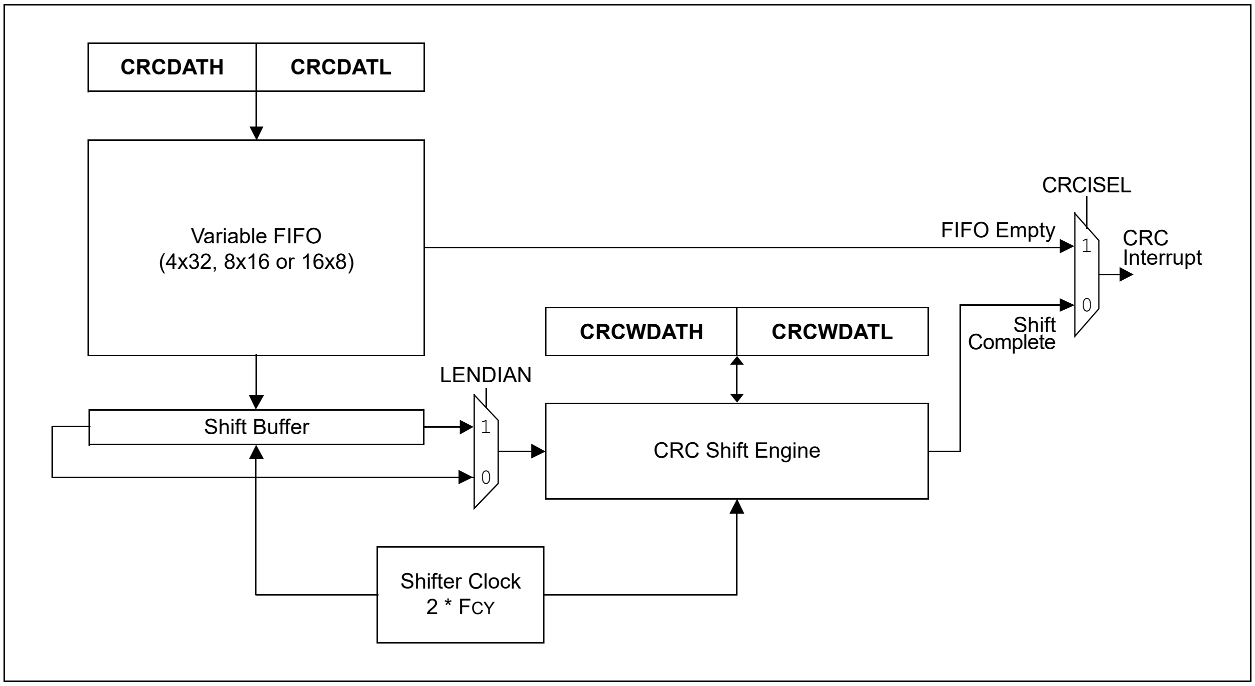
Note 1: This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. For more information, refer to **"32-Bit Programmable Cyclic Redundancy Check (CRC)"** (www.microchip.com/DS30009729).

The 32-bit programmable CRC generator provides a hardware implemented method of quickly generating checksums for various networking and security applications. It offers the following features:

- User-Programmable CRC Polynomial Equation, up to 32 Bits
- Programmable Shift Direction (little or big-endian)
- Independent Data and Polynomial Lengths
- Configurable Interrupt Output
- Data FIFO

A simple version of the CRC shift engine is displayed in Figure 28-1.

FIGURE 28-1: CRC MODULE BLOCK DIAGRAM



28.1 CRC Control Registers

REGISTER 28-1: CRCCONL: CRC CONTROL REGISTER LOW

R/W-0	U-0	R/W-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

HSC/R-0	HSC/R-1	R/W-0	HC/R/W-0	R/W-0	R/W-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	MOD	—	—
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **CRCEN:** CRC Enable bit
1 = Enables module
0 = Disables module

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CSIDL:** CRC Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **VWORD[4:0]:** Pointer Value bits
Indicates the number of valid words in the FIFO. Has a maximum value of 8 when PLEN[4:0] ≥ 7 or 16 when PLEN[4:0] ≤ 7.

bit 7 **CRCFUL:** CRC FIFO Full bit
1 = FIFO is full
0 = FIFO is not full

bit 6 **CRCMPT:** CRC FIFO Empty bit
1 = FIFO is empty
0 = FIFO is not empty

bit 5 **CRCISEL:** CRC Interrupt Selection bit
1 = Interrupt on FIFO is empty; the final word of data is still shifting through the CRC
0 = Interrupt on shift is complete and results are ready

bit 4 **CRCGO:** CRC Start bit
1 = Starts CRC serial shifter
0 = CRC serial shifter is turned off

bit 3 **LENDIAN:** Data Shift Direction Select bit
1 = Data word is shifted into the FIFO, starting with the LSb (little-endian)
0 = Data word is shifted into the FIFO, starting with the MSb (big-endian)

bit 2 **MOD:** CRC Calculation Mode bit
1 = Alternate mode
0 = Legacy mode bit

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 28-2: CRCCONH: CRC CONTROL REGISTER HIGH

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH[4:0]				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN[4:0]				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **DWIDTH[4:0]:** Data Word Width Configuration bits
Configures the width of the data word (Data Word Width – 1).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PLEN[4:0]:** Polynomial Length Configuration bits
Configures the length of the polynomial (Polynomial Length – 1).

REGISTER 28-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X[15:8]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X[7:1]							—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **X[15:1]:** XOR of Polynomial Term x^n Enable bits

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 28-4: CRCXORH: CRC XOR POLYNOMIAL REGISTER, HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X[31:24]							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X[23:16]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **X[31:16]:** XOR of Polynomial Term x^n Enable bits

29.0 POWER-SAVING FEATURES

> **Note 1:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Watchdog Timer and Power-Saving Modes"** (www.microchip.com/DS70615).

The dsPIC33CDVL64MC106 devices provide the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of peripherals being clocked constitutes lower consumed power.

The dsPIC33CDVL64MC106 devices can manage power consumption in four ways:

- Clock Frequency
- Instruction-Based Sleep and Idle Modes
- Software-Controlled Doze Mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption while still maintaining critical application features, such as timing-sensitive communications.

29.1 Clock Frequency and Clock Switching

The dsPIC33CDVL64MC106 family allows a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSCx bits (OSCCON[10:8]). The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 9.0 "Oscillator with High-Frequency PLL"**.

29.2 Instruction-Based Power-Saving Modes

The dsPIC33CDVL64MC106 family has two special power-saving modes that are entered through the execution of a special `PWRSAV` instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembler syntax of the `PWRSAV` instruction is shown in Example 29-1.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

EXAMPLE 29-1: `PWRSAV` INSTRUCTION SYNTAX

```
PWRSAV  #0            ; Put the device into Sleep mode
PWRSAV  #1            ; Put the device into Idle mode
```

29.2.1 SLEEP MODE

The following occurs in Sleep mode:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption is reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor does not operate, since the system clock source is disabled.
- The LPRC clock continues to run in Sleep mode if the WDT is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals can continue to operate. This includes items such as the Input Change Notification on the I/O ports or peripherals that use an External Clock input.
- Any peripheral that requires the system clock source for its operation is disabled.

The device wakes up from Sleep mode on any of the these events:

- Any interrupt source that is individually enabled
- Any form of device Reset
- A WDT time-out

On wake-up from Sleep mode, the processor restarts with the same clock source that was active when Sleep mode was entered.

For optimal power savings, the regulators can be configured to go into standby when Sleep mode is entered by clearing the VREGS (RCON[8]) bit (default configuration).

If the application requires a faster wake-up time, and can accept higher current requirements, the VREGS (RCON[8]) bit can be set to keep the regulators active during Sleep mode. The available Low-Power Sleep modes are shown in Table 29-1. Additional regulator information is available in **Section 30.4 "On-Chip Voltage Regulator"**.

TABLE 29-1: LOW-POWER SLEEP MODES

Relative Power	LPWREN	VREGS	MODE
Highest	0	1	Full power, active
—	0	0	Full power, standby
—	1(1)	1	Low power, active
Lowest	1(1)	0	Low power, standby

Note 1: Low-Power modes; when LPWREN = 1, can only be used in the industrial temperature range.

29.2.2 IDLE MODE

The following occurs in Idle mode:

- The CPU stops executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see **Section 29.4 "Peripheral Module Disable"**).
- If the WDT or FSCM is enabled, the LPRC also remains active.

The device wakes from Idle mode on any of these events:

- Any interrupt that is individually enabled
- Any device Reset
- A WDT time-out

On wake-up from Idle mode, the clock is reapplied to the CPU and instruction execution will begin (2-4 clock cycles later), starting with the instruction following the `PWRSAV` instruction or the first instruction in the ISR.

All peripherals also have the option to discontinue operation when Idle mode is entered to allow for increased power savings. This option is selectable in the control register of each peripheral; for example, the SIDL bit in the Timer1 Control register (T1CON[13]).

29.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a `PWRSAV` instruction is held off until entry into Sleep or Idle mode has completed. The device then wakes up from Sleep or Idle mode.

29.3 Doze Mode

The preferred strategies for reducing power consumption are changing clock speed and invoking one of the power-saving modes. In some circumstances, this cannot be practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed can introduce communication errors, while using a power-saving mode can stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed, while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV[11]). The ratio between peripheral and core clock speed is determined by the DOZE[2:0] bits (CLKDIV[14:12]). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default setting.

Programs can use Doze mode to selectively reduce power consumption in event-driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU Idles, waiting for something to invoke an interrupt routine. An automatic return to full-speed CPU operation on interrupts can be enabled by setting the ROI bit (CLKDIV[15]). By default, interrupt events have no effect on Doze mode operation.

29.4 Peripheral Module Disable

The Peripheral Module Disable (PMD) registers provide a method to disable a peripheral module by stopping all clock sources supplied to that module. When a peripheral is disabled using the appropriate PMD control bit, the peripheral is in a Minimum Power Consumption state. The control and status registers associated with the peripheral are also disabled, so writes to those registers do not have any effect and read values are invalid.

A peripheral module is enabled only if both the associated bit in the PMD register is cleared and the peripheral is supported by the specific dsPIC® DSC variant. If the peripheral is present in the device, it is enabled in the PMD register by default.

Note 1:	If a PMD bit is set, the corresponding module is disabled after a delay of one instruction cycle. Similarly, if a PMD bit is cleared, the corresponding module is enabled after a delay of one instruction cycle (assuming the module control registers are already configured to enable module operation).

29.5 Power-Saving Resources

Many useful resources are provided on the main product page of the Microchip website for the devices listed in this data sheet. This product page contains the latest updates and additional information.

29.5.1 KEY RESOURCES

- **"Watchdog Timer and Power-Saving Modes"** (www.microchip.com/DS70615)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- Development Tools

29.6 PMD Control Registers

REGISTER 29-1: PMD1: PERIPHERAL MODULE DISABLE 1 CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	T1MD	QEI1MD	PWMMD	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11 **T1MD:** Timer1 Module Disable bit
1 = Timer1 module is disabled
0 = Timer1 module is enabled

bit 10 **QEI1MD:** QEI1 Module Disable bit
1 = QEI1 module is disabled
0 = QEI1 module is enabled

bit 9 **PWMMD:** PWM Module Disable bit
1 = PWM module is disabled
0 = PWM module is enabled

bit 8 **Unimplemented:** Read as ‘0’

bit 7 **I2C1MD:** I2C1 Module Disable bit
1 = I2C1 module is disabled
0 = I2C1 module is enabled

bit 6 **U2MD:** UART2 Module Disable bit
1 = UART2 module is disabled
0 = UART2 module is enabled

bit 5 **U1MD:** UART1 Module Disable bit
1 = UART1 module is disabled
0 = UART1 module is enabled

bit 4 **SPI2MD:** SPI2 Module Disable bit
1 = SPI2 module is disabled
0 = SPI2 module is enabled

bit 3 **SPI1MD:** SPI1 Module Disable bit
1 = SPI1 module is disabled
0 = SPI1 module is enabled

bit 2-1 **Unimplemented:** Read as ‘0’

bit 0 **ADC1MD:** ADC Module Disable bit
1 = ADC module is disabled
0 = ADC module is enabled

REGISTER 29-2: PMD2: PERIPHERAL MODULE DISABLE 2 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CCP4MD	CCP3MD	CCP2MD	CCP1MD
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **CCP4MD:** SCCP4 Module Disable bit
1 = SCCP4 module is disabled
0 = SCCP4 module is enabled

bit 2 **CCP3MD:** SCCP3 Module Disable bit
1 = SCCP3 module is disabled
0 = SCCP3 module is enabled

bit 1 **CCP2MD:** SCCP2 Module Disable bit
1 = SCCP2 module is disabled
0 = SCCP2 module is enabled

bit 0 **CCP1MD:** SCCP1 Module Disable bit
1 = SCCP1 module is disabled
0 = SCCP1 module is enabled

REGISTER 29-3: PMD3: PERIPHERAL MODULE DISABLE 3 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
CRCMD	—	—	—	U3MD	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **CRCMD:** CRC Module Disable bit
1 = CRC module is disabled
0 = CRC module is enabled

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **U3MD:** UART3 Module Disable bit
1 = UART3 module is disabled
0 = UART3 module is enabled

bit 2-0 **Unimplemented:** Read as ‘0’

REGISTER 29-4: PMD4: PERIPHERAL MODULE DISABLE 4 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	REFOMD	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **REFOMD:** Reference Clock Module Disable bit
1 = Reference clock module is disabled
0 = Reference clock module is enabled

bit 2-0 **Unimplemented:** Read as ‘0’

REGISTER 29-5: PMD6: PERIPHERAL MODULE DISABLE 6 CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3MD	DMA2MD	DMA1MD	DMA0MD
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11 **DMA3MD:** DMA3 Module Disable bit
1 = DMA3 module is disabled
0 = DMA3 module is enabled

bit 10 **DMA2MD:** DMA2 Module Disable bit
1 = DMA2 module is disabled
0 = DMA2 module is enabled

bit 9 **DMA1MD:** DMA1 Module Disable bit
1 = DMA1 module is disabled
0 = DMA1 module is enabled

bit 8 **DMA0MD:** DMA0 Module Disable bit
1 = DMA0 module is disabled
0 = DMA0 module is enabled

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 29-6: PMD7: PERIPHERAL MODULE DISABLE 7 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CMP1MD
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	PTGMD	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as ‘0’

bit 8 **CMP1MD:** Comparator 1 Module Disable bit
1 = Comparator 1 module is disabled
0 = Comparator 1 module is enabled

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3 **PTGMD**: PTG Module Disable bit
1 = PTG module is disabled
0 = PTG module is enabled

bit 2-0 **Unimplemented:** Read as ‘0’

REGISTER 29-7: PMD8: PERIPHERAL MODULE DISABLE 8 CONTROL REGISTER

U-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0
—	—	OPAMPMD	—	SENT1MD	—	—	DMTMD
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	BIASMD	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13 **OPAMPMD:** Op Amp Module Disable bit
1 = Op amp modules are disabled
0 = Op amp modules are enabled

bit 12 **Unimplemented:** Read as ‘0’

bit 11 **SENT1MD:** SENT1 Module Disable bit
1 = SENT1 module is disabled
0 = SENT1 module is enabled

bit 10-9 **Unimplemented:** Read as ‘0’

bit 8 **DMTMD:** Deadman Timer Module Disable bit
1 = DMT module is disabled
0 = DMT module is enabled

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **CLC4MD:** CLC4 Module Disable bit
1 = CLC4 module is disabled
0 = CLC4 module is enabled

bit 4 **CLC3MD**: CLC3 Module Disable bit
1 = CLC3 module is disabled
0 = CLC3 module is enabled

bit 3 **CLC2MD:** CLC2 Module Disable bit
1 = CLC2 module is disabled
0 = CLC2 module is enabled

bit 2 **CLC1MD:** CLC1 Module Disable bit
1 = CLC1 module is disabled
0 = CLC1 module is enabled

bit 1 **BIASMD:** Constant-Current Source Module Disable bit
1 = Constant-current source module is disabled
0 = Constant-current source module is enabled

bit 0 **Unimplemented:** Read as ‘0’

TABLE 29-2: PMD REGISTERS

Register	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMD1	—	—	—	—	T1MD	QEIMD	PWMMD	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD
PMD2	—	—	—	—	—	—	—	—	—	—	—	—	CCP4MD	CCP3MD	CCP2MD	CCP1MD
PMD3	—	—	—	—	—	—	—	—	CRCMD	—	—	—	U3MD	—	—	—
PMD4	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—
PMD6	—	—	—	—	DMA3MD	DMA2MD	DMA1MD	DMA0MD	—	—	—	—	—	—	—	—
PMD7	—	—	—	—	—	—	—	CMP1MD	—	—	—	—	PTGMD	—	—	—
PMD8	—	—	OPAMPMD	SENT2MD	SENT1MD	—	—	DMTMD	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	BIASMD	—

NOTES:

30.0 SPECIAL FEATURES

Note:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section of the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip website (www.microchip.com).

The dsPIC33CDVL64MC106 devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection and CodeGuard™ Security
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation
- Brown-out Reset (BOR)

30.1 Configuration Bits

In the dsPIC33CDVL64MC106 devices, the Configuration Words are implemented as volatile memory. This means that configuration data will get loaded to volatile memory (from the Flash Configuration Words) each time the device is powered up. Configuration data are stored at the end of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 30-1. The configuration data are automatically loaded from the Flash Configuration Words to the proper Configuration Shadow registers during device Resets.

Note:	Configuration data are reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Words for configuration data in their code for the compiler. This is to make certain that program code is not stored in this address when the code is compiled. Program code executing out of configuration space will cause a device Reset.

Note:	Performing a page erase operation on the last page of program memory clears the Flash Configuration Words.

TABLE 30-1: dsPIC33CDVL64MC106 FAMILY CONFIGURATION ADDRESSES

Register Name	64 KB	32 KB
FSEC	0x00AF00	0x005F00
FBSLIM	0x00AF10	0x005F10
FSIGN	0x00AF14	0x005F14
FOSCSEL	0x00AF18	0x005F18
FOSC	0x00AF1C	0x005F1C
FWDT	0x00AF20	0x005F20
FPOR	0x00AF24	0x005F24
FICD	0x00AF28	0x005F28
FDMTIVTL	0x00AF2C	0x005F2C
FDMTIVTH	0x00AF30	0x005F30
FDMTCNTL	0x00AF34	0x005F34
FDMTCNTH	0x00AF38	0x005F38
FDMT	0x00AF3C	0x005F3C
FDEVOPT	0x00AF40	0x005F40
FALTREG	0x00AF44	0x005F44

TABLE 30-2: CONFIGURATION REGISTERS MAP

Register Name	Bits 23-16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSEC	—	AIVTDIS	—	—	—	CSS[2:0]			CWRP	GSS[1:0]		GWRP	—	BSEN	BSS[1:0]		BWRP
FBSLIM	—	—	—	—	BSLIM[12:0]												
FSIGN	—	r[(2)]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FOSCSEL	—	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC[2:0]		
FOSC	—	—	—	—	XTBST	XTCFG[1:0]		—	PLLKEN	FCKSM[1:0]		—	—	—	OSCIOFNC	POSCMD[1:0]	
FWDT	—	FWDTEN	SWDTPS[4:0]					WDTWIN[1:0]		WINDIS	RCLKSEL[1:0]		RWDTPS[4:0]				
FPOR	—	—	—	—	—	—	r[(1)]	—	—	—	BISTDIS	r[(1)]	r[(1)]	—	—	—	—
FICD	—	—	—	—	—	—	—	—	—	r[(1)]	—	JTAGEN	—	—	—	ICS[1:0]	
FDMTIVTL	—	DMTIVT[15:0]															
FDMTIVTH	—	DMTIVT[31:16]															
FDMTCNTL	—	DMTCNT[15:0]															
FDMTCNTH	—	DMTCNT[31:16]															
FDMT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMTDIS
FDEVOPT	—	—	—	SPI2PIN	—	—	SMB3EN	r[(2)]	r[(2)]	r[(1)]	—	—	—	ALTI2C1	r[(1)]	—	—
FALTREG	—	—	CTXT4[2:0]			—	CTXT3[2:0]			—	CTXT2[2:0]			—	CTXT1[2:0]		

Legend: — = unimplemented bit, read as ‘1’; r = reserved bit.

Note 1: Bit reserved, maintain as ‘1’.

2: Bit reserved, maintain as ‘0’.

REGISTER 30-1: FSEC CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
AIVTDIS	—	—	—	CSS2	CSS1	CSS0	CWRP
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
GSS1	GSS0	GWRP	—	BSEN	BSS1	BSS0	BWRP
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15 **AIVTDIS:** Alternate Interrupt Vector Table Disable bit
1 = Disables AIVT
0 = Enables AIVT

bit 14-12 **Unimplemented:** Read as ‘1’

bit 11-9 **CSS[2:0]:** Configuration Segment Code Flash Protection Level bits
111 = No protection (other than CWRP write protection)
110 = Standard security
10x = Enhanced security
0xx = High security

bit 8 **CWRP:** Configuration Segment Write-Protect bit
1 = Configuration Segment is not write-protected
0 = Configuration Segment is write-protected

bit 7-6 **GSS[1:0]:** General Segment Code Flash Protection Level bits
11 = No protection (other than GWRP write protection)
10 = Standard security
0x = High security

bit 5 **GWRP:** General Segment Write-Protect bit
1 = User program memory is not write-protected
0 = User program memory is write-protected

bit 4 **Unimplemented:** Read as ‘1’

bit 3 **BSEN:** Boot Segment Control bit
1 = No Boot Segment
0 = Boot Segment size is determined by BSLIM[12:0]

bit 2-1 **BSS[1:0]:** Boot Segment Code Flash Protection Level bits
11 = No protection (other than BWRP write protection)
10 = Standard security
0x = High security

bit 0 **BWRP:** Boot Segment Write-Protect bit
1 = User program memory is not write-protected
0 = User program memory is write-protected

REGISTER 30-2: FBSLIM CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	—	BSLIM[12:8][(1)]				
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
BSLIM[7:0][(1)]							
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-13 **Unimplemented:** Read as ‘1’

bit 12-0 **BSLIM[12:0]:** Boot Segment Code Flash Page Address Limit bits[(1)]
Contains the page address of the first active General Segment page. The value to be programmed is the inverted page address, such that programming additional ‘0’s can only increase the Boot Segment size.

Note 1: The BSLIMx bits are a ‘write-once’ element. If, after the Reset sequence, they are not erased (all ‘1’s), then programming of the FBSLIM bits is prohibited. An attempt to do so will fail to set the WR bit (NVMCON[15]), and consequently, have no effect.

REGISTER 30-3: FSIGN CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-0	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	r = Reserved bit	PO = Program Once bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15 **Reserved:** Maintain as ‘0’

bit 14-0 **Unimplemented:** Read as ‘1’

REGISTER 30-4: FOSCSEL CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-8 **Unimplemented:** Read as ‘1’

bit 7 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)

bit 6-3 **Unimplemented:** Read as ‘1’

bit 2-0 **FNOSC[2:0]:** Initial Oscillator Source Selection bits
111 = Internal Fast RC (FRC) Oscillator with Postscaler
110 = Backup Fast RC (BFRC)
101 = LPRC Oscillator
100 = Reserved
011 = Primary Oscillator with PLL (XTPLL, HSPLL, ECPLL)
010 = Primary (XT, HS, EC) Oscillator
001 = Internal Fast RC Oscillator with PLL (FRCPLL)
000 = Fast RC (FRC) Oscillator

REGISTER 30-5: FOSC CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1
—	—	—	XTBST	XTCFG1	XTCFG0	—	PLLKEN[(1)]
bit 15							bit 8

R/PO-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	—	—	—	OSCIOFNC	POSCMD1	POSCMD0
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-13 **Unimplemented:** Read as ‘1’

bit 12 **XTBST:** Oscillator Kick-Start Programmability bit
1 = Boosts the kick-start
0 = Default kick-start

bit 11-10 **XTCFG[1:0]:** Crystal Oscillator Drive Select bits
Current gain programmability for oscillator (output drive).
11 = Gain3 (use for 24-32 MHz crystals)
10 = Gain2 (use for 16-24 MHz crystals)
01 = Gain1 (use for 8-16 MHz crystals)
00 = Gain0 (use for 4-8 MHz crystals)

bit 9 **Unimplemented:** Read as ‘1’

bit 8 **PLLKEN:** PLL Lock Enable bit[(1)]
1 = PLL clock output will be disabled if lock is lost
0 = PLL clock output will not be disabled if lock is lost

bit 7-6 **FCKSM[1:0]:** Clock Switching Mode bits
1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled
01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled

bit 5-3 **Unimplemented:** Read as ‘1’

bit 2 **OSCIOFNC:** OSCO Pin Function bit (except in XT and HS modes)
1 = OSCO is the clock output
0 = OSCO is the general purpose digital I/O pin

bit 1-0 **POSCMD[1:0]:** Primary Oscillator Mode Select bits
11 = Primary Oscillator is disabled
10 = HS Crystal Oscillator mode (10 MHz-32 MHz)
01 = XT Crystal Oscillator mode (3.5 MHz-10 MHz)
00 = EC (External Clock) mode

Note 1: A time-out period will occur when the system clock switching logic requests the PLL clock source and the PLL is not already enabled.

REGISTER 30-6: FWDT CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	SWDTPS4	SWDTPS3	SWDTPS2	SWDTPS1	SWDTPS0	WDTWIN1	WDTWIN0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WINDIS	RCLKSEL1	RCLKSEL0	RWDTPS4	RWDTPS3	RWDTPS2	RWDTPS1	RWDTPS0
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15 **FWDTEN:** Watchdog Timer Enable bit
1 = WDT is enabled in hardware
0 = WDT controller via the ON bit (WDTCONL[15])

bit 14-10 **SWDTPS[4:0]:** Sleep Mode Watchdog Timer Period Select bits
11111 = Divide by 2 ^ 31 = 2,147,483,648
11110 = Divide by 2 ^ 30 = 1,073,741,824
. . .
00001 = Divide by 2^1 = 2
00000 = Divide by 2^0 = 1

bit 9-8 **WDTWIN[1:0]:** Watchdog Timer Window Select bits
11 = WDT window is 25% of the WDT period
10 = WDT window is 37.5% of the WDT period
01 = WDT window is 50% of the WDT period
00 = WDT Window is 75% of the WDT period

bit 7 **WINDIS:** Watchdog Timer Window Enable bit
1 = Watchdog Timer is in Non-Window mode
0 = Watchdog Timer is in Window mode

bit 6-5 **RCLKSEL[1:0]:** Watchdog Timer Clock Select bits
11 = LPRC clock
10 = Uses FRC when WINDIS = 0, system clock is not INTOSC/LPRC and device is not in Sleep; otherwise, uses INTOSC/LPRC
01 = Uses Peripheral Clock when system clock is not INTOSC/LPRC and device is not in Sleep; otherwise, uses INTOSC/LPRC
00 = Reserved

bit 4-0 **RWDTPS[4:0]:** Run Mode Watchdog Timer Period Select bits
11111 = Divide by 2 ^ 31 = 2,147,483,648
11110 = Divide by 2 ^ 30 = 1,073,741,824
. . .
00001 = Divide by 2^1 = 2
00000 = Divide by 2^0 = 1

REGISTER 30-7: FPOR CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	r-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	R/PO-1[1]	r-1	r-1	U-1	U-1	U-1	U-1
—	BISTDIS	—	—	—	—	—	—
bit 7							bit 0

Legend:	PO = Program Once bit	r = Reserved bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Erased value	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-11 **Unimplemented:** Read as '1'

bit 10 **Reserved:** Maintain as '1'

bit 9-7 **Unimplemented:** Read as '1'

bit 6 **BISTDIS:** Memory BIST Feature Disable bit[1]
1 = MBIST on Reset feature is disabled
0 = MBIST on Reset feature is enabled

bit 5-4 **Reserved:** Maintain as '0b11'

bit 3-0 **Unimplemented:** Read as '1'

Note 1: Applies to a Power-on Reset (POR) only.

REGISTER 30-8: FICD CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

r-1	U-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1
—	—	JTAGEN	—	—	—	ICS1	ICS0
bit 7							bit 0

Legend:	PO = Program Once bit	r = Reserved bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-8 **Unimplemented:** Read as ‘1’

bit 7 **Reserved:** Maintain as ‘1’

bit 6 **Unimplemented:** Read as ‘1’

bit 5 **JTAGEN:** JTAG Enable bit
1 = JTAG port is enabled
0 = JTAG port is disabled

bit 4-2 **Unimplemented:** Read as ‘1’

bit 1-0 **ICS[1:0]:** ICD Communication Channel Select bits
11 = Communicates on PGC1 and PGD1
10 = Communicates on PGC2 and PGD2
01 = Communicates on PGC3 and PGD3
00 = Reserved, do not use

REGISTER 30-9: FDMTIVTL CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[15:8]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[7:0]							
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15-0 **DMTIVT[15:0]:** DMT Window Interval Lower 16 bits

REGISTER 30-10: FDMTIVTH CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[31:24]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[23:16]							
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15-0 **DMTIVT[31:16]:** DMT Window Interval Higher 16 bits

REGISTER 30-11: FDMTCNTL CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[15:8]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[7:0]							
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15-0 **DMTCNT[15:0]:** DMT Instruction Count Time-out Value Lower 16 bits

REGISTER 30-12: FDMTCNTH CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[31:24]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[23:16]							
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘1’

bit 15-0 **DMTCNT[31:16]:** DMT Instruction Count Time-out Value Upper 16 bits

REGISTER 30-13: FDMT CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/PO-1
—	—	—	—	—	—	—	DMTDIS
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-1 **Unimplemented:** Read as ‘1’

bit 0 **DMTDIS:** DMT Disable bit
1 = Deadman Timer is disabled and can be enabled by software using the ON bit (DMTCON[15])
0 = Deadman Timer is enabled and cannot be disabled by software

REGISTER 30-14: FDEVOPT CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	R/PO-1	U-1	U-1	R/PO-1	r-0	r-0
—	—	SPI2PIN[(1)]	—	—	SMB3EN[(2)]	—	—
bit 15							bit 8

r-1	U-1	U-1	U-1	R/PO-1	r-1	U-1	U-1
—	—	—	—	ALTI2C1	—	—	—
bit 7							bit 0

Legend:	PO = Program Once bit	r = Reserved bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Erased value	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-14 **Unimplemented:** Read as ‘1’

bit 13 **SPI2PIN:** Master SPI #2 Fast I/O Pad Disable bit[(1)]
1 = Master SPI2 uses PPS (I/O remap) to make connections with device pins
0 = Master SPI2 uses direct connections with specified device pins

bit 12-11 **Unimplemented:** Read as ‘1’

bit 10 **SMB3EN:** SMBus 3.0 Levels Enable bit[(2)]
1 = SMBus 3.0 input levels
0 = Normal SMBus input levels

bit 9-8 **Reserved:** Maintain as ‘0’

bit 7 **Reserved:** Maintain as ‘1’

bit 6-4 **Unimplemented:** Read as ‘1’

bit 3 **ALTI2C1:** Alternate I2C1 Pin Mapping bit
1 = Default location for SCL1/SDA1 pins
0 = Alternate location for SCL1/SDA1 pins (ASCL1/ASDA1)

bit 2 **Reserved:** Maintain as ‘1’

bit 1-0 **Unimplemented:** Read as ‘1’

Note 1: Fixed pin option is only available for 48-pin packages.
2: SMBus mode is enabled by the SMEN bit (I2CxCONL[8]).

REGISTER 30-15: FALTREG CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
—	CTXT4[2:0]			—	CTXT3[2:0]		
bit 15							bit 8

U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
—	CTXT2[2:0]			—	CTXT1[2:0]		
bit 7							bit 0

Legend:	PO = Program Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Erased value	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-15 **Unimplemented:** Read as '1'

bit 14-12 **CTXT4[2:0]:** Specifies the Alternate Working Register Set #4 with Interrupt Priority Levels (IPL) bits
111 = Not assigned
110 = Alternate Register Set #4 is assigned to IPL Level 7
101 = Alternate Register Set #4 is assigned to IPL Level 6
100 = Alternate Register Set #4 is assigned to IPL Level 5
011 = Alternate Register Set #4 is assigned to IPL Level 4
010 = Alternate Register Set #4 is assigned to IPL Level 3
001 = Alternate Register Set #4 is assigned to IPL Level 2
000 = Alternate Register Set #4 is assigned to IPL Level 1

bit 11 **Unimplemented:** Read as '1'

bit 10-8 **CTXT3[2:0]:** Specifies the Alternate Working Register Set #3 with Interrupt Priority Levels (IPL) bits
111 = Not assigned
110 = Alternate Register Set #3 is assigned to IPL Level 7
101 = Alternate Register Set #3 is assigned to IPL Level 6
100 = Alternate Register Set #3 is assigned to IPL Level 5
011 = Alternate Register Set #3 is assigned to IPL Level 4
010 = Alternate Register Set #3 is assigned to IPL Level 3
001 = Alternate Register Set #3 is assigned to IPL Level 2
000 = Alternate Register Set #3 is assigned to IPL Level 1

bit 7 **Unimplemented:** Read as '1'

bit 6-4 **CTXT2[2:0]:** Specifies the Alternate Working Register Set #2 with Interrupt Priority Levels (IPL) bits
111 = Not assigned
110 = Alternate Register Set #2 is assigned to IPL Level 7
101 = Alternate Register Set #2 is assigned to IPL Level 6
100 = Alternate Register Set #2 is assigned to IPL Level 5
011 = Alternate Register Set #2 is assigned to IPL Level 4
010 = Alternate Register Set #2 is assigned to IPL Level 3
001 = Alternate Register Set #2 is assigned to IPL Level 2
000 = Alternate Register Set #2 is assigned to IPL Level 1

bit 3 **Unimplemented:** Read as '1'

REGISTER 30-15: FALTREG CONFIGURATION REGISTER (CONTINUED)

bit 2-0 **CTXT1[2:0]:** Specifies the Alternate Working Register Set #1 with Interrupt Priority Levels (IPL) bits

111 = Not assigned
110 = Alternate Register Set #1 is assigned to IPL Level 7
101 = Alternate Register Set #1 is assigned to IPL Level 6
100 = Alternate Register Set #1 is assigned to IPL Level 5
011 = Alternate Register Set #1 is assigned to IPL Level 4
010 = Alternate Register Set #1 is assigned to IPL Level 3
001 = Alternate Register Set #1 is assigned to IPL Level 2
000 = Alternate Register Set #1 is assigned to IPL Level 1

30.2 Device Identification

The dsPIC33CDVL64MC106 family has two Identification registers, near the end of configuration memory space, that store the Device ID (DEVID) and Device Revision (DEVREV). These registers are used to determine the mask, variant and manufacturing information about the device. These registers are read-only and are shown in Register 30-16 and Register 30-17.

REGISTER 30-16: DEVREV: DEVICE REVISION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R	R	R	R
—	—	—	—	DEVREV[3:0]			
bit 7							bit 0

Legend:			
R = Read-Only bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-4 **Unimplemented:** Read as ‘0’

bit 3-0 **DEVREV[3:0]:** Device Revision bits

REGISTER 30-17: DEVID: DEVICE ID REGISTERS

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R-1	R-0	R-0	R-0	R-1	R-1	R-1	R-0
FAMID[7:0]							
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV[7:0][(1)]							
bit 7							bit 0

Legend:			
R = Read-Only bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as ‘0’

bit 15-8 **FAMID[7:0]:** Device Family Identifier bits

bit 7-0 **DEV[7:0]:** Individual Device Identifier bits[(1)]

Note 1: See Table 30-3 for the list of Device Identifier bits.

TABLE 30-3: DEVICE IDs FOR THE dsPIC33CDVL64MC106 FAMILY

Device	DEVID
dsPIC33CDVL64MC106	0x991A
dsPIC33CDV64MC106	0x991B

30.3 User OTP Memory

The dsPIC33CDVL64MC106 family contains 64 One-Time-Programmable (OTP) double words, located at addresses, 801700h through 8017FEh. Each 48-bit OTP double word can only be written one time. The OTP Words can be used for storing checksums, code revisions, manufacturing dates, manufacturing lot numbers or any other application-specific information.

The OTP area is not cleared by any erase command. This memory can be written only once.

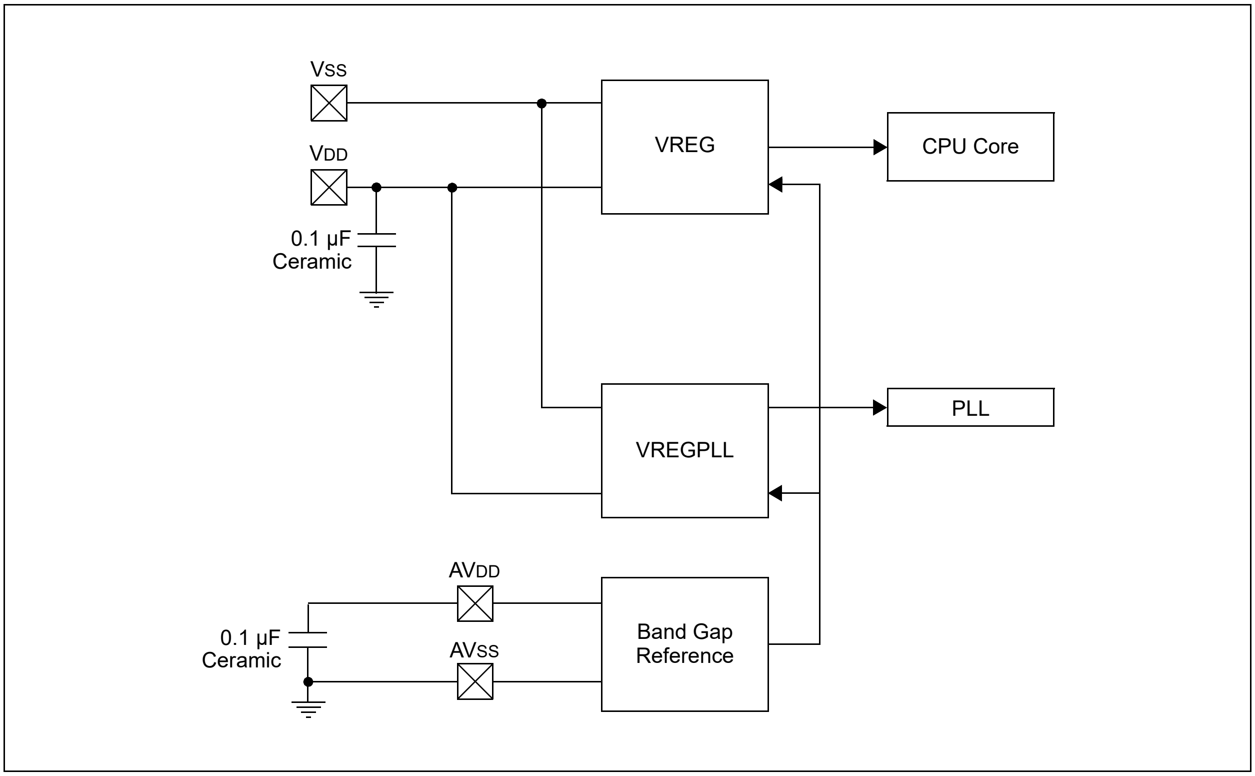
30.4 On-Chip Voltage Regulator

The dsPIC33CDVL64MC106 family has a capacitorless internal voltage regulator to supply power to the core at 1.2V (typical). The voltage regulator, VREG, provides power for the core. The PLL is powered using a separate regulator, VREGPLL, as shown in Figure 30-1. The regulators have Low-Power and Standby modes for use in Sleep modes. For additional information about Sleep, see **Section 29.2.1 "Sleep Mode"**.

When the regulators are in Low-Power mode (LPWREN = `1`), the power available to the core is limited.

Before the LPWREN bit is set, the device should be placed into a Lower Power state by disabling peripherals and lowering CPU frequency (e.g., 8 MHz FRC without PLL).

FIGURE 30-1: INTERNAL REGULATOR



REGISTER 30-18: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER[2]

R/W-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
LPWREN[1]	—	—	—	—	—	—	—
bit 15							bit 8

r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **LPWREN:** Low-Power Mode Enable bit[1]
1 = Voltage regulators are in Low-Power mode
0 = Voltage regulators are in Full Power mode

bit 14-0 **Reserved:** Maintained as ‘0’

Note 1: Low-Power mode can only be used within the industrial temperature range. The CPU should be run at slow speed (8 MHz or less) before setting this bit.
2: HW resets this register only on a POR Reset.

30.5 Brown-out Reset (BOR)

The Brown-out Reset (BOR) module is based on an internal voltage reference circuit that monitors the regulated supply voltage. The main purpose of the BOR module is to generate a device Reset when a brown-out condition occurs. Brown-out conditions are generally caused by glitches on the AC mains (for example, missing portions of the AC cycle waveform due to bad power transmission lines or voltage sags due to excessive current draw when a large inductive load is turned on).

A BOR generates a Reset pulse which resets the device. The BOR selects the clock source based on the device Configuration bit selections.

If an oscillator mode is selected, the BOR activates the Oscillator Start-up Timer (OST). The system clock is held until OST expires. If the PLL is used, the clock is held until the LOCK bit (OSCCON[5]) is '1'.

The BOR status bit (RCON[1]) is set to indicate that a BOR has occurred. The BOR circuit continues to operate while in Sleep or Idle mode and resets the device should V_{DD} fall below the BOR threshold voltage.

30.6 Dual Watchdog Timer (WDT)

Note 1:	This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Dual Watchdog Timer"**, (www.microchip.com/DS70005250).

The dsPIC33 dual Watchdog Timer (WDT) is described in this section. Refer to Figure 30-2 for a block diagram of the WDT.

The WDT, when enabled, operates from the internal Low-Power RC (BFRC/244) Oscillator clock source or a selectable clock source in Run mode. The WDT can be used to detect system software malfunctions by resetting the device if the WDT is not cleared periodically in software. The WDT can be configured in Windowed mode or Non-Windowed mode. Various WDT time-out periods can be selected using the WDT postscaler. The WDT can also be used to wake the device from Sleep or Idle mode (Power Save mode). If the WDT expires and issues a device Reset, the WDTO bit in RCON (Register 6-1) will be set.

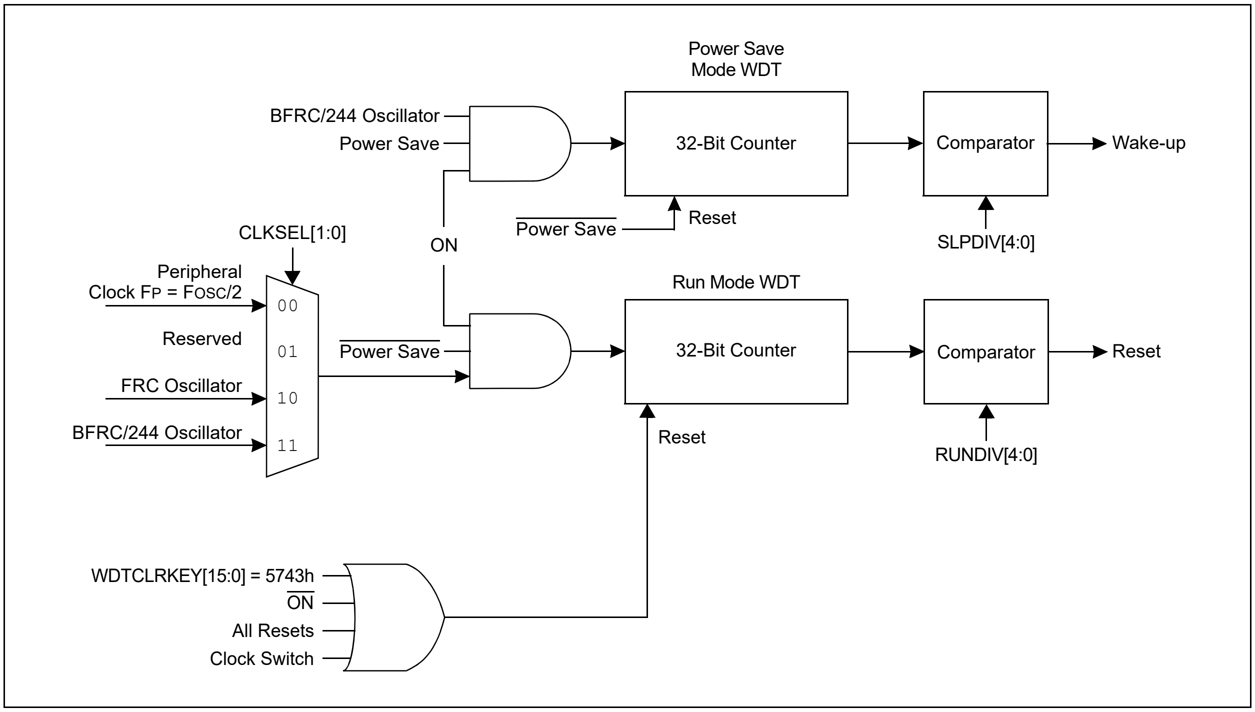
Note:	It is recommended to have at least 1-2 WDT clock cycles of delay after a `CLRWDT` instruction, in case one needs to perform a `PWRSAV`/NVM operation soon after the `CLRWDT` instruction.

The following are some of the key features of the WDT modules:

- Configuration or Software Controlled
- Separate User-Configurable Time-out Periods for Run and Sleep/Idle
- Can Wake the Device from Sleep or Idle
- User-Selectable Clock Source in Run Mode
- Operates from BFRC/244 in Sleep/Idle Mode

Note:	The WDT is not automatically reset when a Fail-Safe Clock Monitor event occurs. The user should issue a `CLRWDT` instruction after a clock fail event is detected.

FIGURE 30-2: WATCHDOG TIMER BLOCK DIAGRAM



REGISTER 30-19: WDTCONL: WATCHDOG TIMER CONTROL REGISTER LOW

R/W-0	U-0	U-0	R-y	R-y	R-y	R-y	R-y
ON(1,2)	—	—	RUNDIV4(3)	RUNDIV3(3)	RUNDIV2(3)	RUNDIV1(3)	RUNDIV0(3)
bit 15							bit 8

R	R	R-y	R-y	R-y	R-y	R-y	HS/R/W-0
CLKSEL1(3,5)	CLKSEL0(3,5)	SLPDIV4(3)	SLPDIV3(3)	SLPDIV2(3)	SLPDIV1(3)	SLPDIV0(3)	WDTWINEN(4)
bit 7							bit 0

Legend:	HS = Hardware Settable bit	y = Value from Configuration bit on POR	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **ON:** Watchdog Timer Enable bit(1,2)
1 = Enables the Watchdog Timer if it is not enabled by the device configuration
0 = Disables the Watchdog Timer if it was enabled in software

bit 14-13 **Unimplemented:** Read as ‘0’

bit 12-8 **RUNDIV[4:0]:** Sleep and Idle Mode WDT Postscaler Status bits(3)
11111 = Divide by 2^{31} = 2,147,483,648
11110 = Divide by 2^{30} = 1,073,741,824
...
00001 = Divide by 2^1 = 2
00000 = Divide by 2^0 = 1

bit 7-6 **CLKSEL[1:0]:** WDT Run Mode Clock Select Status bits(3,5)
11 = BFRC/244 Oscillator
10 = FRC Oscillator
01 = Reserved
00 = SYSCLK

bit 5-1 **SLPDIV[4:0]:** Sleep and Idle Mode WDT Postscaler Status bits(3)
11111 = Divide by 2^{31} = 2,147,483,648
11110 = Divide by 2^{30} = 1,073,741,824
...
00001 = Divide by 2^1 = 2
00000 = Divide by 2^0 = 1

bit 0 **WDTWINEN:** Watchdog Timer Window Enable bit(4)
1 = Enables Window mode
0 = Disables Window mode

Note 1: A read of this bit will result in a ‘1’ if the WDT is enabled by the device configuration or by software.
2: The user’s software should not read or write the peripheral’s SFRs immediately following the instruction that clears the module’s ON bit.
3: These bits reflect the value of the Configuration bits.
4: The WDTWINEN bit reflects the status of the Configuration bit if the bit is set. If the bit is cleared, the value is controlled by software.
5: The available clock sources are device-dependent.

REGISTER 30-20: WDTCONH: WATCHDOG TIMER CONTROL REGISTER HIGH

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
WDTCLRKEY[15:8]							
bit 15							bit 8

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
WDTCLRKEY[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-0 **WDTCLRKEY[15:0]:** Watchdog Timer Clear Key bits

To clear the Watchdog Timer to prevent a time-out, software must write the value, 0x5743, to this location using a single 16-bit write.

30.7 JTAG Interface

The dsPIC33CDVL64MC106 devices implement a JTAG interface, which supports boundary scan device testing. Programming is not supported through the JTAG interface; only boundary scan is supported.

Note:	Refer to **"Programming and Diagnostics"** (www.microchip.com/DS70608) for further information on usage, configuration and operation of the JTAG interface.

30.8 In-Circuit Serial Programming™ (ICSP™)

The dsPIC33CDVL64MC106 devices can be serially programmed while in the end application circuit. This is done with two lines for clock and data, and three other lines for power, ground and the programming sequence. Serial programming allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. Serial programming also allows the most recent firmware or a custom firmware to be programmed.

Any of the three pairs of programming clock/data pins can be used:

- PGC1 and PGD1
- PGC2 and PGD2
- PGC3 and PGD3

30.9 In-Circuit Debugger

When the MPLAB® tool is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGCx (Emulation/Debug Clock) and PGDx (Emulation/Debug Data) pin functions.

Any of the three pairs of debugging clock/data pins can be used:

- PGC1 and PGD1
- PGC2 and PGD2
- PGC3 and PGD3

To use the in-circuit debugger function of the device, the design must implement ICSP connections to $\overline{\text{MCLR}}$, V_{DD}, V_{SS} and the PGCx/PGDx pin pair. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins (PGCx and PGDx).

30.10 Code Protection and CodeGuard™ Security

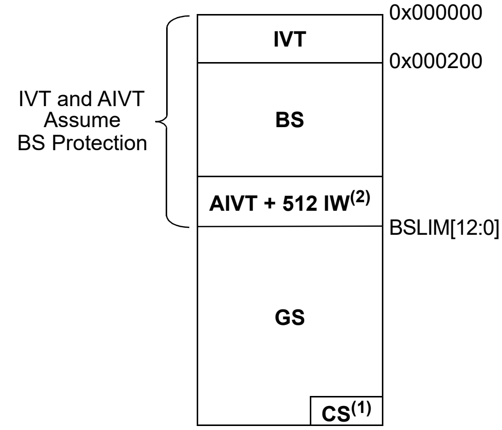
The dsPIC33CDVL64MC106 devices offer multiple levels of security for protecting individual intellectual property. The program Flash protection can be broken up into three segments: Boot Segment (BS), General Segment (GS) and Configuration Segment (CS). Boot Segment has the highest security privilege and can be thought to have limited restrictions when accessing other segments. General Segment has the least security and is intended for the end user system code. Configuration Segment contains only the device user configuration data, which is located at the end of the program memory space.

The code protection features are controlled by the Configuration registers, FSEC and FBSLIM. The FSEC register controls the code-protect level for each segment and if that segment is write-protected. The size of BS and GS will depend on the BSLIM[12:0] bits setting and if the Alternate Interrupt Vector Table (AIVT) is enabled. The BSLIM[12:0] bits define the number of pages for BS with each page containing 1024 IW. The smallest BS size is one page, which will consist of the Interrupt Vector Table (IVT) and 512 IW of code protection.

If the AIVT is enabled, the last page of BS will contain the AIVT and will not contain any BS code. With AIVT enabled, the smallest BS size is now two pages (2048 IW), with one page for the IVT and BS code, and the other page for the AIVT. Write protection of the BS does not cover the AIVT. The last page of BS can always be programmed or erased by BS code. The General Segment will start at the next page and will consume the rest of program Flash, except for the Flash Configuration Words. The IVT will assume GS security only if BS is not enabled. The IVT is protected from being programmed or page erased when either security segment has enabled write protection.

The different device security segments are shown in Figure 30-3. Here, all three segments are shown, but are not required. If only basic code protection is required, then GS can be enabled independently or combined with CS, if desired.

FIGURE 30-3: SECURITY SEGMENTS EXAMPLE



Note 1: If CS is write-protected, the last page (GS + CS) of program memory will be protected from an erase condition.

2: The last half (256 IW) of the last page of BS is unusable program memory.

NOTES:

31.0 INSTRUCTION SET SUMMARY

> **Note:** This data sheet summarizes the features of the dsPIC33CDVL64MC106 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *"16-Bit MCU and DSC Programmer's Reference Manual"* (www.microchip.com/DS70000157).

The dsPIC33CDVL64MC106 family instruction set is almost identical to that of the dsPIC30F and dsPIC33F.

Most instructions are a single program memory word (24 bits). Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word, divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into five basic categories:

- Word or Byte-oriented Operations
- Bit-oriented Operations
- Literal Operations
- DSP Operations
- Control Operations

Table 31-1 lists the general symbols used in describing the instructions.

The dsPIC33 instruction set summary in Table 31-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register 'Wb' without any address modifier
- The second source operand, which is typically a register 'Ws' with or without an address modifier
- The destination of the result, which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value 'f'
- The destination, which could be either the file register 'f' or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register 'Wb')

The literal instructions that involve data movement can use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register 'Wb' without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register 'Wd' with or without an address modifier

The `MAC` class of DSP instructions can use some of the following operands:

- The accumulator (A or B) to be used (required operand)
- The W registers to be used as the two operands
- The X and Y address space prefetch operations
- The X and Y address space prefetch destinations
- The accumulator write-back destination

The other DSP instructions do not involve any multiplication and can include:

- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift specified by a W register 'Wn' or a literal value

The control instructions can use some of the following operands:

- A program memory address
- The mode of the Table Read and Table Write instructions

Most instructions are a single word. Certain double-word instructions are designed to provide all the required information in these 48 bits. In the second word, the 8 MSbs are ‘0’s. If this second word is executed as an instruction (by itself), it executes as a NOP.

The double-word instructions execute in two instruction cycles.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter is changed as a result of the instruction, or a PSV or Table Read is performed. In these cases, the execution takes multiple instruction cycles, with the additional instruction cycle(s) executed as a NOP. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles.

Note: In the dsPIC33CDVL64MC106 devices, read and Read-Modify-Write operations on non-CPU Special Function Registers require an additional cycle when compared to dsPIC30F, dsPIC33F, PIC24F and PIC24H devices.

Note: For more details on the instruction set, refer to the *“16-Bit MCU and DSC Programmer’s Reference Manual”* (www.microchip.com/DS70000157).

TABLE 31-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by “text”
(text)	Means “content of text”
[text]	Means “the location addressed by text”
{ }	Optional field or operation
a ∈ {b, c, d}	a is selected from the set of values b, c, d
[n:m]	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator Write-Back Destination Address register ∈ {W13, [W13]+ = 2}
bit4	4-bit bit selection field (used in word-addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x0000...0x1FFF}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSb must be ‘0’
None	Field does not require an entry, can be blank
OA, OB, SA, SB	DSP Status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0...W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor Working register pair (direct addressing)

TABLE 31-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

Field	Description
Wm*Wm	Multiplicand and Multiplier Working register pair for Square instructions ∈ {W4 * W4,W5 * W5,W6 * W6,W7 * W7}
Wm*Wn	Multiplicand and Multiplier Working register pair for DSP instructions ∈ {W4 * W5,W4 * W6,W4 * W7,W5 * W6,W5 * W7,W6 * W7}
Wn	One of 16 Working registers ∈ {W0...W15}
Wnd	One of 16 Destination Working registers ∈ {W0...W15}
Wns	One of 16 Source Working registers ∈ {W0...W15}
WREG	W0 (Working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }
Wx	X Data Space Prefetch Address register for DSP instructions ∈ {[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] - = 6, [W8] - = 4, [W8] - = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] - = 6, [W9] - = 4, [W9] - = 2, [W9 + W12], none}
Wxd	X Data Space Prefetch Destination register for DSP instructions ∈ {W4...W7}
Wy	Y Data Space Prefetch Address register for DSP instructions ∈ {[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] - = 6, [W11] - = 4, [W11] - = 2, [W11 + W12], none}
Wyd	Y Data Space Prefetch Destination register for DSP instructions ∈ {W4...W7}

TABLE 31-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles[(1)]	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BFEXT	BFEXT	bit4,wid5,Ws,Wb	Bit Field Extract from Ws to Wb	2	2	None
		BFEXT	bit4,wid5,f,Wb	Bit Field Extract from f to Wb	2	2	None
7	BFINS	BFINS	bit4,wid5,Wb,Ws	Bit Field Insert from Wb into Ws	2	2	None
		BFINS	bit4,wid5,Wb,f	Bit Field Insert from Wb into f	2	2	None
		BFINS	bit4,wid5,lit8,Ws	Bit Field Insert from #lit8 to Ws	2	2	None

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a "REPEAT #5" instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles[1]	Status Flags Affected
9	BRA	BRA	C,Expr	Branch if Carry	1	1 (4)	None
		BRA	GE,Expr	Branch if Greater Than or Equal	1	1 (4)	None
		BRA	GEU,Expr	Branch if unsigned Greater Than or Equal	1	1 (4)	None
		BRA	GT,Expr	Branch if Greater Than	1	1 (4)	None
		BRA	GTU,Expr	Branch if Unsigned Greater Than	1	1 (4)	None
		BRA	LE,Expr	Branch if Less Than or Equal	1	1 (4)	None
		BRA	LEU,Expr	Branch if Unsigned Less Than or Equal	1	1 (4)	None
		BRA	LT,Expr	Branch if Less Than	1	1 (4)	None
		BRA	LTU,Expr	Branch if Unsigned Less Than	1	1 (4)	None
		BRA	N,Expr	Branch if Negative	1	1 (4)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (4)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (4)	None
		BRA	NOV,Expr	Branch if Not Overflow	1	1 (4)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (4)	None
		BRA	OA,Expr	Branch if Accumulator A Overflow	1	1 (4)	None
		BRA	OB,Expr	Branch if Accumulator B Overflow	1	1 (4)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (4)	None
		BRA	SA,Expr	Branch if Accumulator A Saturated	1	1 (4)	None
		BRA	SB,Expr	Branch if Accumulator B Saturated	1	1 (4)	None
		BRA	Expr	Branch Unconditionally	1	4	None
		BRA	Z,Expr	Branch if Zero	1	1 (4)	None
		BRA	Wn	Computed Branch	1	4	None
10	BREAK	BREAK		Stop User Code Execution	1	1	None
11	BSET	BSET	f,#bit4	Bit Set f	1	1	None
			Ws,#bit4	Bit Set Ws	1	1	None
12	BSW	BSW.C	Ws,Wb	Write C bit to Ws[Wb]	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws[Wb]	1	1	None
13	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
14	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
15	BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
16	BTST	BTST	f,#bit4	Bit Test f	1	1	Z
		BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	C
		BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C	Ws,Wb	Bit Test Ws[Wb] to C	1	1	C
		BTST.Z	Ws,Wb	Bit Test Ws[Wb] to Z	1	1	Z
17	BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
18	CALL	CALL	lit23	Call Subroutine	2	4	SFA
		CALL	Wn	Call Indirect Subroutine	1	4	SFA
		CALL.L	Wn	Call Indirect Subroutine (long address)	1	4	SFA
19	CLR	CLR	f	f = 0x0000	1	1	None
		CLR	WREG	WREG = 0x0000	1	1	None
		CLR	Ws	Ws = 0x0000	1	1	None
		CLR	Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a "`REPEAT #5`" instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles[(1)]	Status Flags Affected
20	CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO,Sleep
21	COM	COM	f	f = $\overline{f}$	1	1	N,Z
		COM	f,WREG	WREG = $\overline{f}$	1	1	N,Z
		COM	Ws,Wd	Wd = $\overline{Ws}$	1	1	N,Z
22	CP	CP	f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP	Wb,#lit8	Compare Wb with lit8	1	1	C,DC,N,OV,Z
		CP	Wb,Ws	Compare Wb with Ws (Wb – Ws)	1	1	C,DC,N,OV,Z
23	CP0	CP0	f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0	Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
24	CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,#lit8	Compare Wb with lit8, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,Ws	Compare Wb with Ws, with Borrow (Wb – Ws – $\overline{C}$)	1	1	C,DC,N,OV,Z
25	CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
	CPBEQ	CPBEQ	Wb,Wn,Expr	Compare Wb with Wn, Branch if =	1	1 (5)	None
26	CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
	CPBGT	CPBGT	Wb,Wn,Expr	Compare Wb with Wn, Branch if >	1	1 (5)	None
27	CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
	CPBLT	CPBLT	Wb,Wn,Expr	Compare Wb with Wn, Branch if <	1	1 (5)	None
28	CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
	CPBNE	CPBNE	Wb,Wn,Expr	Compare Wb with Wn, Branch if ≠	1	1 (5)	None
29	CTXTSWP	CTXTSWP	#lit3	Switch CPU Register Context to Context Defined by lit3	1	2	None
30	CTXTSWP	CTXTSWP	Wn	Switch CPU Register Context to Context Defined by Wn	1	2	None
31	DAW.B	DAW.B	Wn	Wn = Decimal Adjust Wn	1	1	C
32	DEC	DEC	f	f = f – 1	1	1	C,DC,N,OV,Z
		DEC	f,WREG	WREG = f – 1	1	1	C,DC,N,OV,Z
		DEC	Ws,Wd	Wd = Ws – 1	1	1	C,DC,N,OV,Z
33	DEC2	DEC2	f	f = f – 2	1	1	C,DC,N,OV,Z
		DEC2	f,WREG	WREG = f – 2	1	1	C,DC,N,OV,Z
		DEC2	Ws,Wd	Wd = Ws – 2	1	1	C,DC,N,OV,Z
34	DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
35	DIVF	DIVF	Wm,Wn	Signed 16/16-bit Fractional Divide	1	18	N,Z,C,OV
36	DIV.S[(2)]	DIV.S	Wm,Wn	Signed 16/16-bit Integer Divide	1	6	N,Z,C,OV
		DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	6	N,Z,C,OV
37	DIV.U[(2)]	DIV.U	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	6	N,Z,C,OV
		DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	6	N,Z,C,OV
38	DIVF2[(2)]	DIVF2	Wm,Wn	Signed 16/16-bit Fractional Divide (W1:W0 preserved)	1	6	N,Z,C,OV
39	DIV2.S[(2)]	DIV2.S	Wm,Wn	Signed 16/16-bit Integer Divide (W1:W0 preserved)	1	6	N,Z,C,OV
		DIV2.SD	Wm,Wn	Signed 32/16-bit Integer Divide (W1:W0 preserved)	1	6	N,Z,C,OV
40	DIV2.U[(2)]	DIV2.U	Wm,Wn	Unsigned 16/16-bit Integer Divide (W1:W0 preserved)	1	6	N,Z,C,OV
		DIV2.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide (W1:W0 preserved)	1	6	N,Z,C,OV
41	DO	DO	#lit15,Expr	Do Code to PC + Expr, lit15 + 1 Times	2	2	None
		DO	Wn,Expr	Do code to PC + Expr, (Wn) + 1 Times	2	2	None

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a “`REPEAT #5`” instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles(1)	Status Flags Affected
42	ED	ED	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance (no accumulate)	1	1	OA,OB,OAB, SA,SB,SAB
43	EDAC	EDAC	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance	1	1	OA,OB,OAB, SA,SB,SAB
44	EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
46	FBCL	FBCL	Ws,Wnd	Find Bit Change from Left (MSb) Side	1	1	C
47	FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
48	FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	C
49	FLIM	FLIM	Wb, Ws	Force Data (Upper and Lower) Range Limit without Limit Excess Result	1	1	N,Z,OV
		FLIM.V	Wb, Ws, Wd	Force Data (Upper and Lower) Range Limit with Limit Excess Result	1	1	N,Z,OV
50	GOTO	GOTO	Expr	Go to Address	2	4	None
		GOTO	Wn	Go to Indirect	1	4	None
		GOTO.L	Wn	Go to Indirect (long address)	1	4	None
51	INC	INC	f	f = f + 1	1	1	C,DC,N,OV,Z
		INC	f,WREG	WREG = f + 1	1	1	C,DC,N,OV,Z
		INC	Ws,Wd	Wd = Ws + 1	1	1	C,DC,N,OV,Z
52	INC2	INC2	f	f = f + 2	1	1	C,DC,N,OV,Z
		INC2	f,WREG	WREG = f + 2	1	1	C,DC,N,OV,Z
		INC2	Ws,Wd	Wd = Ws + 2	1	1	C,DC,N,OV,Z
53	IOR	IOR	f	f = f .IOR. WREG	1	1	N,Z
		IOR	f,WREG	WREG = f .IOR. WREG	1	1	N,Z
		IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N,Z
		IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N,Z
		IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N,Z
54	LAC	LAC	Wso,#Slit4,Acc	Load Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		LAC.D	Wso, #Slit4, Acc	Load Accumulator Double	1	2	OA,SA,OB,SB
56	LNK	LNK	#lit14	Link Frame Pointer	1	1	SFA
57	LSR	LSR	f	f = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	f,WREG	WREG = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C,N,OV,Z
		LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N,Z
		LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N,Z
58	MAC	MAC	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
		MAC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
59	MAX	MAX	Acc	Force Data Maximum Range Limit	1	1	N,OV,Z
		MAX.V	Acc, Wnd	Force Data Maximum Range Limit with Result	1	1	N,OV,Z
60	MIN	MIN	Acc	If Accumulator A Less than B, Load Accumulator with B or vice versa	1	1	N,OV,Z
		MIN.V	Acc, Wd	If Accumulator A Less than B Accumulator, Force Minimum Data Range Limit with Limit Excess Result	1	1	N,OV,Z
		MINZ	Acc	Accumulator Force Minimum Data Range Limit	1	1	N,OV,Z
		MINZ.V	Acc, Wd	Accumulator Force Minimum Data Range Limit with Limit Excess Result	1	1	N,OV,Z

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a "`REPEAT #5`" instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles[(1)]	Status Flags Affected
61	MOV	MOV f,Wn	Move f to Wn	1	1	None
		MOV f	Move f to f	1	1	None
		MOV f,WREG	Move f to WREG	1	1	None
		MOV #lit16,Wn	Move 16-bit Literal to Wn	1	1	None
		MOV.b #lit8,Wn	Move 8-bit Literal to Wn	1	1	None
		MOV Wn,f	Move Wn to f	1	1	None
		MOV Wso,Wdo	Move Ws to Wd	1	1	None
		MOV WREG,f	Move WREG to f	1	1	None
		MOV.D Wns,Wd	Move Double from W(ns):W(ns + 1) to Wd	1	2	None
		MOV.D Ws,Wnd	Move Double from Ws to W(nd + 1):W(nd)	1	2	None
62	MOVPAG	MOVPAG #lit10,DSRPAG	Move 10-bit Literal to DSRPAG	1	1	None
		MOVPAG #lit8,TBLPAG	Move 8-bit Literal to TBLPAG	1	1	None
		MOVPAG Ws, DSRPAG	Move Ws[9:0] to DSRPAG	1	1	None
		MOVPAG Ws, TBLPAG	Move Ws[7:0] to TBLPAG	1	1	None
64	MOVSAC	MOVSAC Acc,Wx,Wxd,Wy,Wyd,AWB	Prefetch and Store Accumulator	1	1	None
65	MPY	MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
66	MPY.N	MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
67	MSC	MSC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
68	MUL	MUL.SS Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
		MUL.SS Wb,Ws,Acc	Accumulator = Signed(Wb) * Signed(Ws)	1	1	None
		MUL.SU Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,Ws,Acc	Accumulator = Signed(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Acc	Accumulator = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.US Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
		MUL.US Wb,Ws,Acc	Accumulator = Unsigned(Wb) * Signed(Ws)	1	1	None
		MUL.UU Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MUL.UU Wb,#lit5,Acc	Accumulator = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,Ws,Acc	Accumulator = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MULW.SS Wb,Ws,Wnd	Wnd = Signed(Wb) * Signed(Ws)	1	1	None
		MULW.SU Wb,Ws,Wnd	Wnd = Signed(Wb) * Unsigned(Ws)	1	1	None
		MULW.US Wb,Ws,Wnd	Wnd = Unsigned(Wb) * Signed(Ws)	1	1	None
		MULW.UU Wb,Ws,Wnd	Wnd = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.SU Wb,#lit5,Wnd	Wnd = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	Wnd = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL f	W3:W2 = f * WREG	1	1	None

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a "REPEAT #5" instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles[1]	Status Flags Affected
69	NEG	NEG Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG f	f = $\overline{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG f,WREG	WREG = $\overline{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C,DC,N,OV,Z
70	NOP	NOP	No Operation	1	1	None
		NOPR	No Operation	1	1	None
71	NORM	NORM Acc, Wd	Normalize Accumulator	1	1	N,OV,Z
72	POP	POP f	Pop f from Top-of-Stack (TOS)	1	1	None
		POP Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
		POP.D Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd + 1)	1	2	None
		POP.S	Pop Shadow Registers	1	1	All
73	PUSH	PUSH f	Push f to Top-of-Stack (TOS)	1	1	None
		PUSH Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
		PUSH.D Wns	Push W(ns):W(ns + 1) to Top-of-Stack (TOS)	1	2	None
		PUSH.S	Push Shadow Registers	1	1	None
74	PWRSAV	PWRSAV #lit1	Go into Sleep or Idle mode	1	1	WDTO,Sleep
75	RCALL	RCALL Expr	Relative Call	1	4	SFA
		RCALL Wn	Computed Call	1	4	SFA
76	REPEAT	REPEAT #lit15	Repeat Next Instruction lit15 + 1 Times	1	1	None
		REPEAT Wn	Repeat Next Instruction (Wn) + 1 Times	1	1	None
77	RESET	RESET	Software Device Reset	1	1	None
78	RETFIE	RETFIE	Return from Interrupt	1	6 (5)	SFA
79	RETLW	RETLW #lit10,Wn	Return with Literal in Wn	1	6 (5)	SFA
80	RETURN	RETURN	Return from Subroutine	1	6 (5)	SFA
81	RLC	RLC f	f = Rotate Left through Carry f	1	1	C,N,Z
		RLC f,WREG	WREG = Rotate Left through Carry f	1	1	C,N,Z
		RLC Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C,N,Z
82	RLNC	RLNC f	f = Rotate Left (No Carry) f	1	1	N,Z
		RLNC f,WREG	WREG = Rotate Left (No Carry) f	1	1	N,Z
		RLNC Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N,Z
83	RRC	RRC f	f = Rotate Right through Carry f	1	1	C,N,Z
		RRC f,WREG	WREG = Rotate Right through Carry f	1	1	C,N,Z
		RRC Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C,N,Z
84	RRNC	RRNC f	f = Rotate Right (No Carry) f	1	1	N,Z
		RRNC f,WREG	WREG = Rotate Right (No Carry) f	1	1	N,Z
		RRNC Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N,Z
85	SAC	SAC Acc,#Slit4,Wdo	Store Accumulator	1	1	None
		SAC.R Acc,#Slit4,Wdo	Store Rounded Accumulator	1	1	None
86	SE	SE Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C,N,Z
87	SETM	SETM f	f = 0xFFFF	1	1	None
		SETM WREG	WREG = 0xFFFF	1	1	None
		SETM Ws	Ws = 0xFFFF	1	1	None
88	SFTAC	SFTAC Acc,Wn	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC Acc,#Slit6	Arithmetic Shift Accumulator by Slit6	1	1	OA,OB,OAB, SA,SB,SAB

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a “`REPEAT #5`” instruction, such that they are executed six consecutive times.

TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles[1]	Status Flags Affected
89	SL	SL f	f = Left Shift f	1	1	C,N,OV,Z
		SL f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
		SL Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
91	SUB	SUB Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB f	f = f – WREG	1	1	C,DC,N,OV,Z
		SUB f,WREG	WREG = f – WREG	1	1	C,DC,N,OV,Z
		SUB #lit10,Wn	Wn = Wn – lit10	1	1	C,DC,N,OV,Z
		SUB Wb,Ws,Wd	Wd = Wb – Ws	1	1	C,DC,N,OV,Z
		SUB Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C,DC,N,OV,Z
92	SUBB	SUBB f	f = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB #lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
93	SUBR	SUBR f	f = WREG – f	1	1	C,DC,N,OV,Z
		SUBR f,WREG	WREG = WREG – f	1	1	C,DC,N,OV,Z
		SUBR Wb,Ws,Wd	Wd = Ws – Wb	1	1	C,DC,N,OV,Z
		SUBR Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C,DC,N,OV,Z
94	SUBBR	SUBBR f	f = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
95	SWAP	SWAP.b Wn	Wn = Nibble Swap Wn	1	1	None
		SWAP Wn	Wn = Byte Swap Wn	1	1	None
96	TBLRDH	TBLRDH Ws,Wd	Read Prog[23:16] to Wd[7:0]	1	5	None
97	TBLRDL	TBLRDL Ws,Wd	Read Prog[15:0] to Wd	1	5	None
98	TBLWTH	TBLWTH Ws,Wd	Write Ws[7:0] to Prog[23:16]	1	2	None
99	TBLWTL	TBLWTL Ws,Wd	Write Ws to Prog[15:0]	1	2	None
101	ULNK	ULNK	Unlink Frame Pointer	1	1	SFA
104	XOR	XOR f	f = f .XOR. WREG	1	1	N,Z
		XOR f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR #lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
105	ZE	ZE Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.
2: The divide instructions must be preceded with a "REPEAT #5" instruction, such that they are executed six consecutive times.

32.0 DEVELOPMENT SUPPORT

Move a design from concept to production in record time with Microchip's award-winning development tools. Microchip tools work together to provide state of the art debugging for any project with easy-to-use Graphical User Interfaces (GUIs) in our free MPLAB® X and Atmel Studio Integrated Development Environments (IDEs), and our code generation tools. Providing the ultimate ease-of-use experience, Microchip's line of programmers, debuggers and emulators work seamlessly with our software tools. Microchip development boards help evaluate the best silicon device for an application, while our line of third party tools round out our comprehensive development tool solutions.

Microchip's MPLAB X and Atmel Studio ecosystems provide a variety of embedded design tools to consider, which support multiple devices, such as PIC® MCUs, AVR® MCUs, SAM MCUs and dsPIC® DSCs. MPLAB X tools are compatible with Windows®, Linux® and Mac® operating systems while Atmel Studio tools are compatible with Windows.

Go to the following website for more information and details:

https://www.microchip.com/development-tools/

NOTES:

33.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the dsPIC33CDVL64MC106 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the dsPIC33CDVL64MC106 devices are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings(1)

Ambient temperature under bias .... -40°C to +125°C
Storage temperature .... -65°C to +150°C
Voltage on VDD with respect to VSS .... -0.3V to +4.0V
Voltage on any pin that is not 5V tolerant with respect to VSS(3) .... -0.3V to (VDD + 0.3V)
Voltage on any 5V tolerant pin with respect to VSS(3) .... -0.3V to +5.5V
Maximum current out of VSS pins .... 300 mA
Maximum current into VDD pins(2) .... 300 mA
Maximum current sunk/sourced by any regular I/O pin .... 15 mA
Maximum current sunk/sourced by an I/O pin with increased current drive strength (RB1, RC8, RC9 and RD8) .... 25 mA
Maximum current sunk by a group of I/Os between two VSS pins(4) .... 75 mA
Maximum current sourced by a group of I/Os between two VDD pins(4) .... 75 mA
Maximum current sunk by all I/Os(2,5) .... 200 mA
Maximum current sourced by all I/Os(2,5) .... 200 mA

Note 1: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those, or any other conditions above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

2: Maximum allowable current is a function of device maximum power dissipation (see Table 33-2).

3: See the **“Pin Diagrams”** section for the 5V tolerant pins.

4: Not applicable to AVDD and AVSS pins.

5: The maximum current sunk/sourced by all I/Os is limited by 150 mA.

33.1 DC Characteristics

TABLE 33-1: dsPIC33CDVL64MC106 FAMILY OPERATING CONDITIONS

VDD Range	Temperature Range	Maximum CPU Clock Frequency
3.0V to 3.6V	-40°C to +125°C	100 MHz

TABLE 33-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min.	Max.	Unit
Industrial Temperature Devices				
Operating Junction Temperature Range	T_J	-40	+125	°C
Operating Ambient Temperature Range	T_A	-40	+85	°C
Extended Temperature Devices				
Operating Junction Temperature Range	T_J	-40	+150	°C
Operating Ambient Temperature Range	T_A	-40	+125	°C
High-Temperature Devices				
Operating Junction Temperature Range	T_J	-40	+165	°C
Operating Ambient Temperature Range	T_A	-40	+150	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $I/O = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$		W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$		W

TABLE 33-3: PACKAGE THERMAL RESISTANCE(1)

Package	Symbol	Typ.	Unit
64-Pin VGQFN 9x9 mm	θ_{JA}	20.5	°C/W

Note 1: Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

TABLE 33-4: OPERATING VOLTAGE SPECIFICATIONS

Operating Conditions (unless otherwise stated):
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
DC10	VDD	**Supply Voltage**	3.0	3.6	V	
DC11	AVDD	**Supply Voltage**	Greater of: VDD – 0.3 or 3.0	Lesser of: VDD + 0.3 or 3.6	V	The difference between the AVDD supply and VDD supply must not exceed ±300 mV at all times, including during device power-up
DC16	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	—	VSS	V	
DC17	SVDD(2)	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.03	—	V/ms	0V-3V in 100 ms
BO10	VBOR(1)	**BOR Event** on VDD Transition High-to-Low	2.68	2.99	V	

TABLE 33-4: OPERATING VOLTAGE SPECIFICATIONS

Note 1: Device is functional at $V_{BORMIN} < V_{DD} < V_{DDMIN}$. Analog modules (ADC and comparators) may have degraded performance. The V_{BOR} parameter is for design guidance only and is not tested in manufacturing.

2: Failure to observe SV_{DD} can result in the device remaining in Reset, even after V_{DD} is raised past V_{BORMAX}.

TABLE 33-5: OPERATING CURRENT (IDD)[2]

Parameter No.	Typ.[1]	Max.	Units	Conditions		
DC20	5.5	6.6	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	5.2	6.4	mA	+25°C		
	5.7	11.9	mA	+85°C		
	7.2	16	mA	+125°C		
DC21	7.1	8.4	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 280 MHz)
	7.1	8.1	mA	+25°C		
	7.5	13.4	mA	+85°C		
	8.8	17.5	mA	+125°C		
DC22	10.6	12.6	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	10.3	12.3	mA	+25°C		
	11.6	17.1	mA	+85°C		
	12.3	21.3	mA	+125°C		
DC23	15.4	18	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	15.2	17.4	mA	+25°C		
	16	22	mA	+85°C		
	17.5	26.3	mA	+125°C		
DC24	19	22.5	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	18.9	21.6	mA	+25°C		
	19.7	26	mA	+85°C		
	21.2	29.6	mA	+125°C		
DC25	20.7	22.4	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz)
	20.7	21.5	mA	+25°C		
	21.4	25.4	mA	+85°C		
	23	29.5	mA	+125°C		

Note 1: Data in the “Typ.” column are for design guidance only and are not tested.

2: Base run current (IDD) is measured as follows:

- Oscillator is switched to EC+PLL mode in software
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to VDD – 0.3V
- OSC2 pin is configured as an I/O in the Configuration Words (OSCIOFNC (FOSC[2]) = `0`)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = `01`)
- Watchdog Timer is disabled (FWDTEN (FWDT[15]) = `0`)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all ‘`1`’s)
- JTAG is disabled (JTAGEN (FICD[5]) = `0`)
- `NOP` instructions are executed

TABLE 33-6: IDLE CURRENT (IIDLE)[2]

Parameter No.	Typ.[1]	Max.	Units	Conditions		
DC30	4.5	5.4	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	4.1	5.2	mA	+25°C		
	4.7	9.1	mA	+85°C		
	6.0	14.2	mA	+125°C		
DC31	4.8	5.9	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 80 MHz)
	4.7	5.7	mA	+25°C		
	5.0	9.6	mA	+85°C		
	6.6	14.8	mA	+125°C		
DC32	6.2	7.3	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	5.8	7.1	mA	+25°C		
	6.5	11	mA	+85°C		
	8.0	16.1	mA	+125°C		
DC33	7.8	9.3	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	7.6	9.0	mA	+25°C		
	8.1	12.8	mA	+85°C		
	9.8	18	mA	+125°C		
DC34	9.3	11.4	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	9.2	11.2	mA	+25°C		
	10.0	14.5	mA	+85°C		
	11.6	19.8	mA	+125°C		
DC35	10.0	12	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz)
	10.0	12	mA	+25°C		
	10.7	13.4	mA	+85°C		
	12.5	18.6	mA	+125°C		

Note 1: Data in the “Typ.” column are for design guidance only and are not tested.

2: Base Idle current (IIDLE) is measured as follows:

- Oscillator is switched to EC+PLL mode in software
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to VDD – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFNC (FOSC[2]) = `0`)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = `01`)
- Watchdog Timer is disabled (FWDTEN (FWDT[15]) = `0`)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all ‘`1`’s)
- JTAG is disabled (JTAGEN (FICD[5]) = `0`)
- `NOP` instructions are executed

TABLE 33-7: POWER-DOWN CURRENT (IPD)[2]

Parameter No.	Typ.[1]	Max.	Units	Conditions		
DC40[3,4]	0.15	0.48	mA	-40°C	3.3V	VREGS bit (RCON[8]) = 0 LPWREN bit (VREGCON[15]) = 1
	0.23	1.1	mA	+25°C		
	0.86	4.2	mA	+85°C		
DC41	0.9	—	mA	-40°C	3.3V	VREGS bit (RCON[8]) = 1 LPWREN bit (VREGCON[15]) = 1
	0.9	—	mA	+25°C		
	1.5	—	mA	+85°C		
	2.9	11	mA	+125°C		

Note 1: Data in the "Typ." column are for design guidance only and are not tested.

2: Base Sleep current (IPD) is measured with:
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to VDD – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFNC (FOSC[2]) = 0)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = 01)
- Watchdog Timer is disabled (FWDTEN (FWDT[15]) = 0)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all '1's)
- JTAG is disabled (JTAGEN (FICD[5]) = 0)

3: The Regulator Standby mode, when the VREGS bit = 0, is operational only in industrial temperature range: -40°C ≤ TA ≤ +85°C.

4: The Regulator Low-Power mode, when LPWREN = 1, is operational only in industrial temperature range: -40°C ≤ TA ≤ +85°C.

TABLE 33-8: DOZE CURRENT (IDOZE)

Parameter No.	Typ.[1]	Doze Ratio	Units	Conditions		
DC70	12.1	1:2	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	8.0	1:128	mA			
	12.0	1:2	mA	+25°C		
	7.8	1:128	mA			
	12.4	1:2	mA	+85°C		
	8.3	1:128	mA			
	13.8	1:2	mA	+125°C		
	8.8	1:128	mA			
DC71	15.8	1:2	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz)
	10.4	1:128	mA			
	15.7	1:2	mA	+25°C		
	10.3	1:128	mA			
	16.6	1:2	mA	+85°C		
	11.2	1:128	mA			
	18.1	1:2	mA	+125°C		
	12.7	1:128	mA			

Note 1: Data in the "Typ." column are for design guidance only and are not tested.

TABLE 33-9: WATCHDOG TIMER DELTA CURRENT (ΔIWDT)[1]

Parameter No.	Typ.	Units	Conditions	
DC61	1	μA	-40°C	3.3V
	2	μA	+25°C	
	4	μA	+85°C	
	11	μA	+125°C	

Note 1: The ΔIWDT current is the additional current consumed when the module is enabled. This current should be added to the base IPD current. All parameters are for design guidance only and are not tested.

TABLE 33-10: PWM DELTA CURRENT[1]

Parameter No.	Typ.	Max.	Units	Conditions		
DC100	5.59	6.8	mA	-40°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (FPLLO = 500 MHz), (VCO = 1000 MHz, PLLFBD = 125)
	5.87	6.9	mA	+25°C		
	5.99	7.4	mA	+85°C		
	6.05	7.4	mA	+125°C		
DC101	4.52	6.2	mA	-40°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (FPLLO = 400 MHz), (VCO = 400 MHz, PLLFBD = 100)
	4.68	6.5	mA	+25°C		
	4.77	6.5	mA	+85°C		
	4.81	6.7	mA	+125°C		
DC102	2.39	3.96	mA	-40°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (FPLLO = 200 MHz), (VCO = 200 MHz, PLLFBD = 50)
	2.42	3.44	mA	+25°C		
	2.47	3.4	mA	+85°C		
	2.49	4.2	mA	+125°C		
DC103	1.24	2	mA	-40°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (FPLLO = 100 MHz), (VCO = 100 MHz, PLLFBD = 25)
	1.26	2.1	mA	+25°C		
	1.28	2.2	mA	+85°C		
	1.31	2.2	mA	+125°C		

Note 1: PLL current is not included. The PLL current will be the same if more than one PWM is running. All parameters are characterized but not tested during manufacturing.

TABLE 33-11: ADC DELTA CURRENT[1]

Parameter No.	Typ.	Max.	Units	Conditions		
DC120	5.35	5.9	mA	-40°C	3.3V	TAD = 14.3 ns (3.5 Msps conversion rate)
	5.42	5.9	mA	+25°C		
	5.44	5.7	mA	+85°C		
	5.46	5.7	mA	+125°C		

Note 1: Shared core continuous conversion. TAD = 14.3 ns (3.5 Msps conversion rate). Listed delta currents are for only one ADC core. All parameters are characterized but not tested during manufacturing.

TABLE 33-12: COMPARATOR + DAC DELTA CURRENT

Parameter No.	Typ.	Max.	Units	Conditions		
DC130	1.38	—	mA	-40°C	3.3V	FPLLO @ 500 MHz[1]
	1.28	—	mA	+25°C		
	1.30	—	mA	+85°C		
	1.37	—	mA	+125°C		

Note 1: Listed delta currents are for only one comparator + DAC instance. All parameters are characterized but not tested during manufacturing.

TABLE 33-13: OP AMP DELTA CURRENT[1]

Parameter No.	Typ.	Max.	Units	Conditions	
DC140	0.21	0.42	mA	-40°C	3.3V
	0.22	0.44	mA	+25°C	
	0.23	0.52	mA	+85°C	
	0.47	0.89	mA	+125°C	

Note 1: Listed delta currents are for only one op amp instance. All parameters are characterized but not tested during manufacturing.

TABLE 33-14: I/O PIN INPUT SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended						
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
DI10	VIL	**Input Low-Level Voltage**				
		Any I/O Pin and $\overline{\text{MCLR}}$	VSS	0.2 VDD	V	
		I/O Pins with SDAx, SCLx	VSS	0.3 VDD	V	SMBus disabled
		I/O Pins with SDAx, SCLx	VSS	0.8	V	SMBus enabled
		I/O Pins with SDAx, SCLx	VSS	0.8	V	SMBus 3.0 enabled
DI20	VIH	**Input High-Level Voltage**[(1)]				
		I/O Pins Not 5V Tolerant	0.8 VDD	VDD	V	
		I/O Pins 5V Tolerant and $\overline{\text{MCLR}}$	0.8 VDD	5.5	V	
		I/O Pins 5V Tolerant with SDAx, SCLx	0.8 VDD	5.5	V	SMBus disabled
		I/O Pins 5V Tolerant with SDAx, SCLx	2.1	5.5	V	SMBus enabled
		I/O Pins 5V Tolerant with SDAx, SCLx	1.35	VDD	V	SMBus 3.0 enabled
		I/O Pins Not 5V Tolerant with SDAx, SCLx	0.8 VDD	VDD	V	SMBus disabled
		I/O Pins Not 5V Tolerant with SDAx, SCLx	2.1	VDD	V	SMBus enabled
		I/O Pins Not 5V Tolerant with SDAx, SCLx	1.35	VDD	V	SMBus 3.0 enabled
DI30	ICNPU	**Input Current with Pull-up Resistor Enabled**[(2)]	175	545	µA	VDD = 3.3V, VPIN = VSS
DI31	ICNPD	**Input Current with Pull-Down Resistor Enabled**[(2)]	65	360	µA	VDD = 3.3V, VPIN = VDD
DI50	IIL	**Input Leakage Current** I/O Pins and $\overline{\text{MCLR}}$ Pin	-700	—	nA	VPIN = VSS
			—	700	nA	VPIN = VDD

Note 1: See the **"Pin Diagrams"** section for the 5V tolerant I/O pins.

2: Characterized but not tested.

TABLE 33-15: I/O PIN INPUT INJECTION CURRENT SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended						
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
DI60a	IICL	**Input Low Injection Current**	0	-5[1,4]	mA	This parameter applies to all pins
DI60b	IICH	**Input High Injection Current**	0	+5[2,3,4]	mA	This parameter applies to all pins, except all 5V tolerant pins
DI60c	∑IICT	**Total Input Injection Current (sum of all I/O and control pins)**	-20[5]	+20[5]	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins Σ (| IICL | + | IICH |) ≤ ∑IICT

Note 1: VIL Source < (VSS – 0.3).
2: VIH Source > (VDD + 0.3) for non-5V tolerant pins only.
3: 5V tolerant pins do not have an internal high-side diode to VDD, and therefore, cannot tolerate any “positive” input injection current.
4: Injection currents can affect the ADC results.
5: Any number and/or combination of I/O pins, not excluded under IICL or IICH conditions, are permitted in the sum.

TABLE 33-16: I/O PIN OUTPUT SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param.	**Symbol**	**Characteristic**	**Typ.[1]**	**Units**	**Conditions**
DO10	VOL	**Sink Driver Voltage**	0.2	V	ISINK = 3.0 mA, VDD = 3.3V
			0.4	V	ISINK = 6.0 mA, VDD = 3.3V
			0.6	V	ISINK = 9.0 mA, VDD = 3.3V
		Sink Driver Voltage for RB1, RC8, RC9 and RD8 Pins	0.25	V	ISINK = 6.0 mA, VDD = 3.3V
			0.5	V	ISINK = 12.0 mA, VDD = 3.3V
			0.75	V	ISINK = 18.0 mA, VDD = 3.3V
DO20	VOH	**Source Driver Voltage**	3.1	V	ISOURCE = 3.0 mA, VDD = 3.3V
			2.9	V	ISOURCE = 6.0 mA, VDD = 3.3V
			2.7	V	ISOURCE = 9.0 mA, VDD = 3.3V
		Source Driver Voltage for RB1, RC8, RC9 and RD8 Pins	3.1	V	ISOURCE = 6.0 mA, VDD = 3.3V
			2.8	V	ISOURCE = 12.0 mA, VDD = 3.3V
			2.6	V	ISOURCE = 18.0 mA, VDD = 3.3V

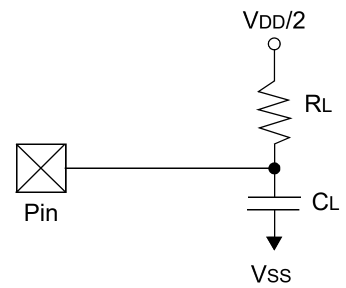
Note 1: Data in the “Typ.” column are at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 33-17: PROGRAM FLASH MEMORY SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended						
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
		Program Flash Memory				
D130	EP	Cell Endurance	10,000	—	E/W	
D134	TRETD	Characteristic Retention	20	—	Year	
D137a	TPE	Self-Timed Page Erase Time	—	20	ms	
D137b	TCE	Self-Timed Chip Erase Time	—	20	ms	
D138a	TWW	Self-Timed Double-Word Write Cycle Time	—	20	µs	6 bytes, data are not all '1's
D138b	TRW	Self-Timed Row Write Cycle Time	—	1.28	ms	384 bytes, data are not all '1's

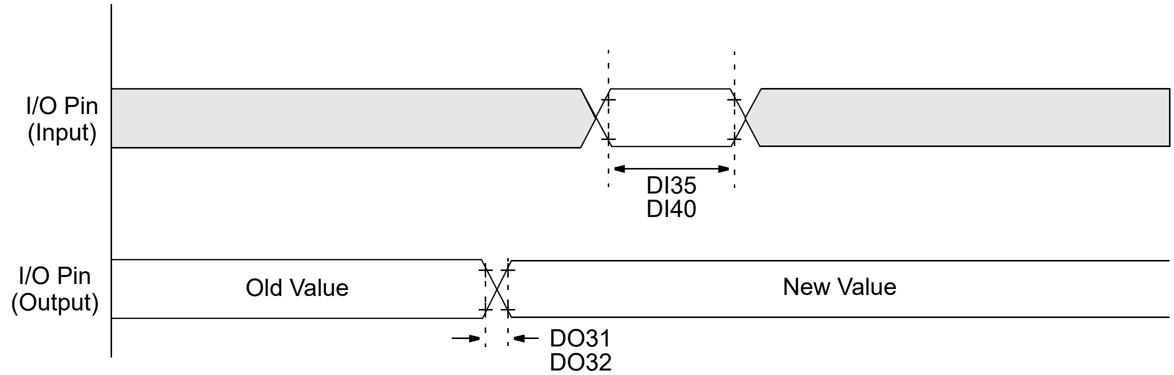
33.2 AC Characteristics and Timing Parameters

FIGURE 33-1: LOAD CONDITIONS FOR I/O SPECIFICATIONS



R_L = 464Ω
C_L = 50 pF

FIGURE 33-2: I/O TIMING CHARACTERISTICS



Note: Refer to Figure 33-1 for load conditions.

TABLE 33-18: I/O TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**
DO31	TIOR	Port Output Rise Time(1)	—	10	ns
DO32	TIOF	Port Output Fall Time(1)	—	10	ns
DI35	TINP	INTx Input Pins High or Low Time	20	—	ns
DI40	TRBP	I/O and CNx Inputs High or Low Time	2	—	TCY

Note 1: This parameter is characterized but not tested in manufacturing.

FIGURE 33-3: EXTERNAL CLOCK TIMING

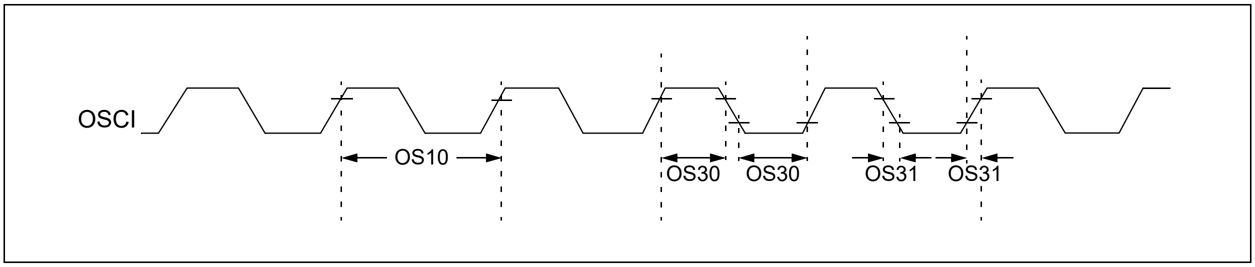


TABLE 33-19: EXTERNAL CLOCK TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended						
Param No.	**Sym**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
OS10	FIN	External CLKI Frequency	DC	64	MHz	EC
		Oscillator Crystal Frequency	3.5	10	MHz	XT
			10	32	MHz	HS
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x OS10	0.55 x OS10	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time(1)	—	10	ns	EC

Note 1: This parameter is characterized but not tested in manufacturing.

TABLE 33-20: PLL CLOCK TIMING SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**
OS50	FPLLI	PLL Input Frequency Range	8	64	MHz
OS51	FPFD	Phase-Frequency Detector Input Frequency (after first divider)	8	FVCO/16	MHz
OS52	FVCO	VCO Output Frequency	400	1600	MHz
OS53	TLOCK	Lock Time for PLL(1)	—	250	µS

Note 1: This parameter is characterized but not tested in manufacturing.

TABLE 33-21: FRC OSCILLATOR SPECIFICATIONS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic	Min	Typ[2]	Max	Units	Conditions
F20	AFRC	FRC Accuracy @ 8 MHz[1]	-2.0	—	2.0	%	-40°C ≤ TA ≤ -5°C
			-1.5	—	1.5	%	-5°C ≤ TA ≤ +85°C
			-2.0	—	2.0	%	+85°C ≤ TA ≤ +125°C
F21	TFRC	FRC Oscillator Start-up Time[3]	—	—	15	μS	
F22	STUNE	OSCTUN Step-Size	—	0.05	—	%/bit	

Note 1: To achieve this accuracy, physical stress applied to the microcontroller package (ex., by flexing the PCB) must be kept to a minimum.

2: Data in the “Typ” column are 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

3: This parameter is characterized but not tested in manufacturing.

TABLE 33-22: BFRC OSCILLATOR SPECIFICATIONS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic	Min	Max	Units
F40	ABFRC	BFRC Accuracy @ 8 MHz	-17	17	%

FIGURE 33-4: BOR AND MASTER CLEAR RESET TIMING CHARACTERISTICS

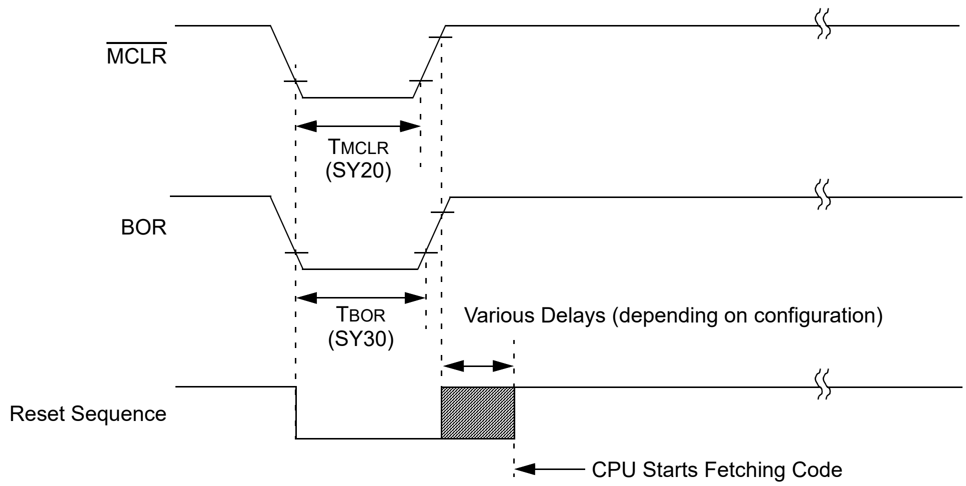


TABLE 33-23: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic[1]	Min.	Typ.[2]	Max.	Units	Conditions
SY00	TPU	Power-up Period	—	200	—	µs	FNOSC[2:0] are FRC
SY10	TOST	Oscillator Start-up	—	1024 TOSC	—	—	TOSC = OSCI period
SY13	TIOZ	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	1.5	—	µs	
SY20	TMCLR	MCLR Pulse Width (low)	2	—	—	µs	
SY30	TBOR	BOR Pulse Width (low)	1	—	—	µs	
SY35	TFSCM	Fail-Safe Clock Monitor Delay	—	—	40	µs	
SY37	TOSCDFRC	FRC Oscillator Start-up Delay	—	—	15	µs	From POR event
SY38	TOSCDLPRC	LPRC Oscillator Start-up Delay	—	—	50	µs	From Reset event

Note 1: These parameters are characterized but not tested in manufacturing.
2: Data in the "Typ." column are at 3.3V, +25°C unless otherwise stated.

FIGURE 33-5: HIGH-SPEED PWMx MODULE TIMING CHARACTERISTICS

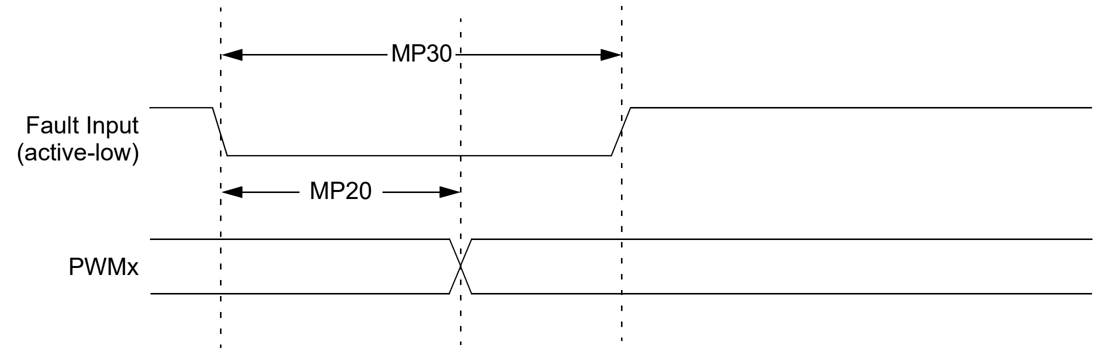


TABLE 33-24: HIGH-SPEED PWMx MODULE TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic[1]	Min.	Max.	Units
MP10	FIN	PWM Input Frequency[2]	—	500	MHz
MP20	TFD	Fault Input ↓ to PWMx I/O Change	—	26	ns
MP30	TFH	Fault Input Pulse Width	8	—	ns

Note 1: These parameters are characterized but not tested in manufacturing.
2: Input frequency of 500 MHz must be used for High-Resolution mode.

FIGURE 33-6: SPIx MODULE MASTER MODE (CKE = 0) TIMING CHARACTERISTICS

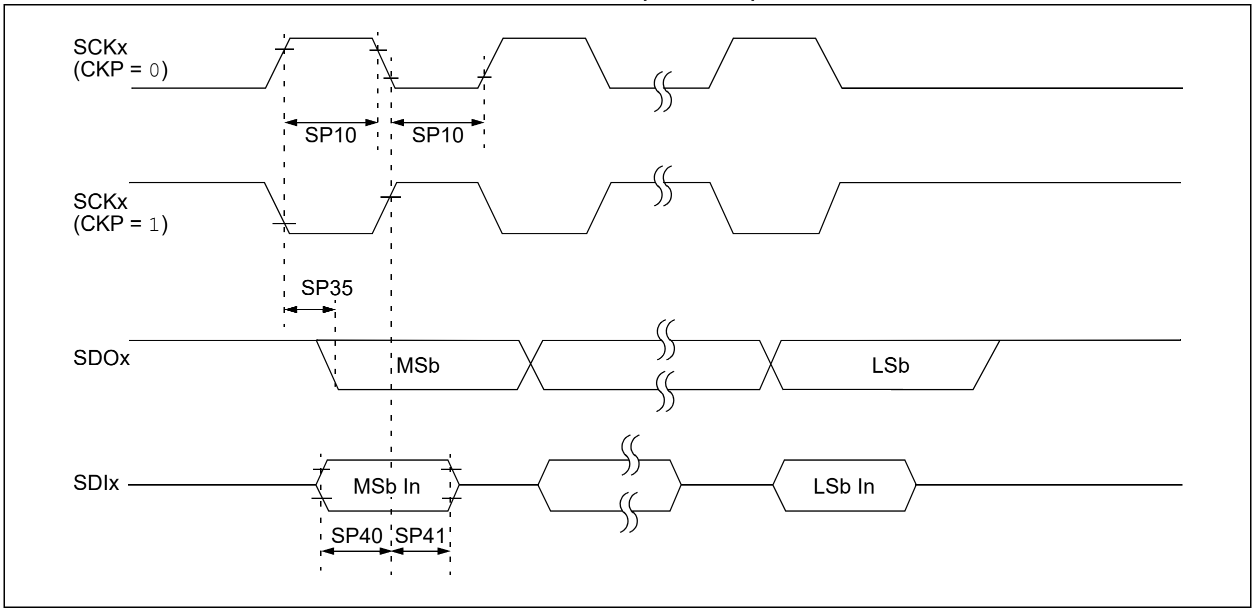


FIGURE 33-7: SPIx MODULE MASTER MODE (CKE = 1) TIMING CHARACTERISTICS

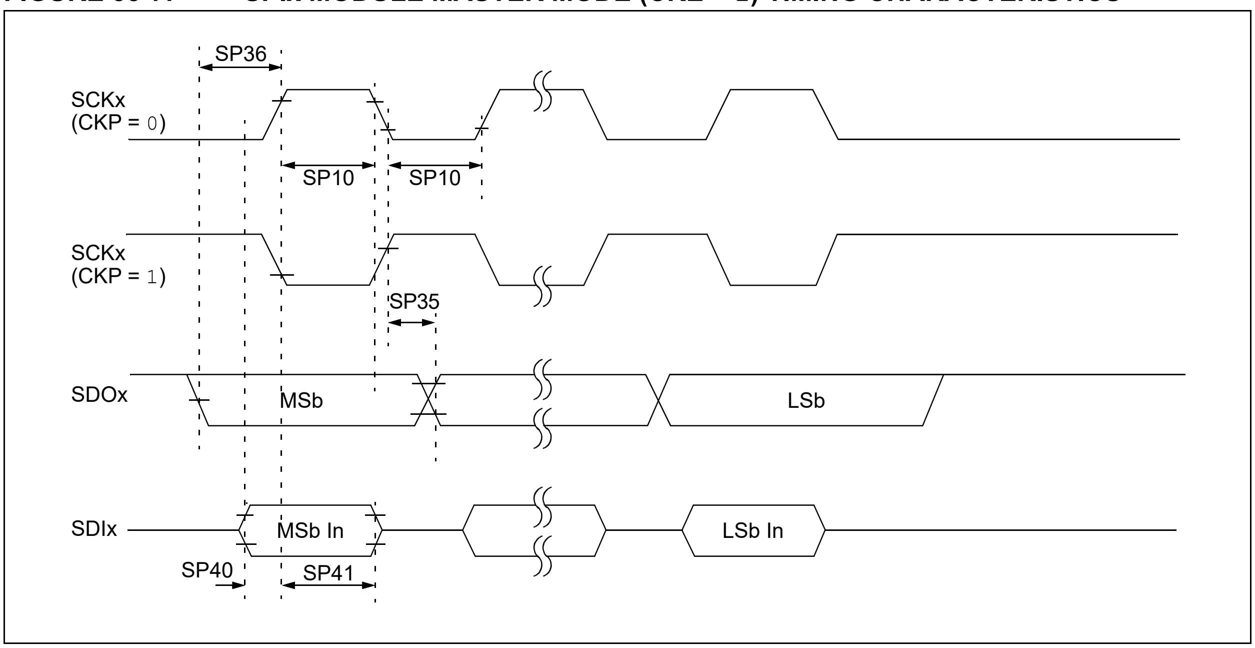


TABLE 33-25: SPIx MODULE MASTER MODE TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param. No.	Symbol	Characteristics[1]	Min	Max	Units
SP10	TscL, TscH	SCKx Output Low or High Time	15	—	ns
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	20	ns
SP36	TdoV2sc, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	3	—	ns
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	10	—	ns
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	15	—	ns

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 33-8: SPIx MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS

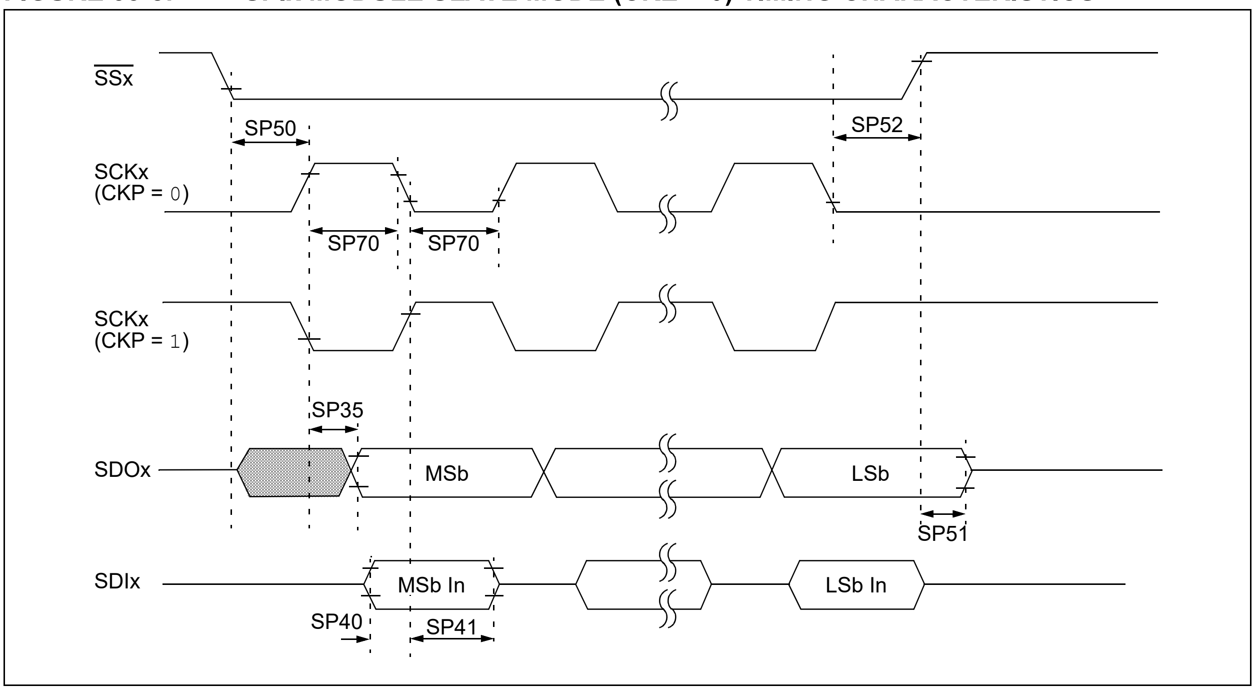


FIGURE 33-9: SPIx MODULE SLAVE MODE (CKE = 1) TIMING CHARACTERISTICS

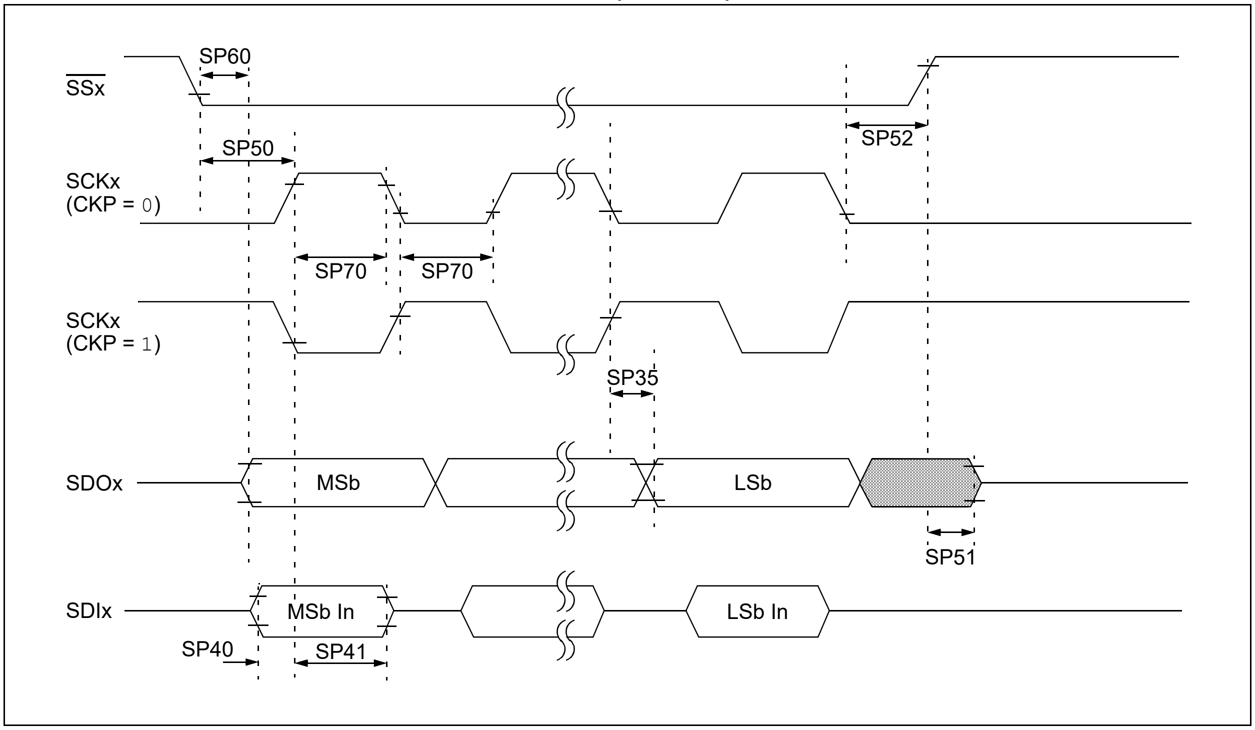


TABLE 33-26: SPIx MODULE SLAVE MODE TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial, -40°C ≤ TA ≤ +125°C for Extended					
Param.No.	**Symbol**	**Characteristics[1]**	**Min**	**Max**	**Units**
SP70	TSCL, TSCH	SCKx Input Low Time or High Time	15	—	ns
SP35	TSCH2DOV, TSCL2DOV	SDOx Data Output Valid after SCKx Edge	—	20	ns
SP40	TDIV2SCH, TDIV2SCL	Setup Time of SDIx Data Input to SCKx Edge	10	—	ns
SP41	TSCH2DIL, TSCL2DIL	Hold Time of SDIx Data Input to SCKx Edge	15	—	ns
SP50	TSSL2SCH, TSSL2SCL	SSx ↓ to SCKx ↓ or SCKx ↑ Input	120	—	ns
SP51	TSSH2DOZ	SSx ↑ to SDOx Output High-Impedance	8	50	ns
SP52	TSCH2SSH TSCL2SSH	SSx ↑ after SCKx Edge	1.5 TCY + 40	—	ns
SP60	TSSL2DOV	SDOx Data Output Valid after SSx Edge	—	50	ns

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 33-10: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)

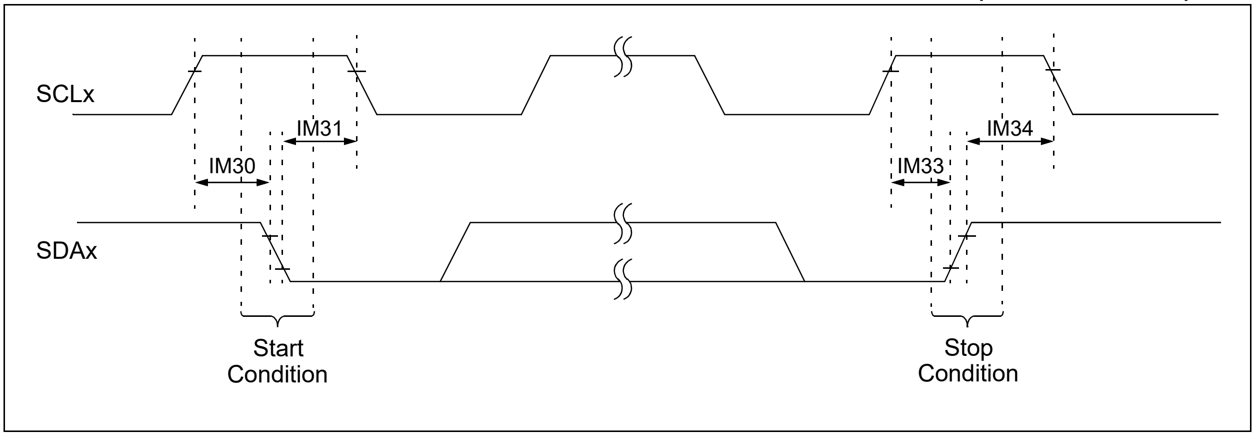


FIGURE 33-11: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

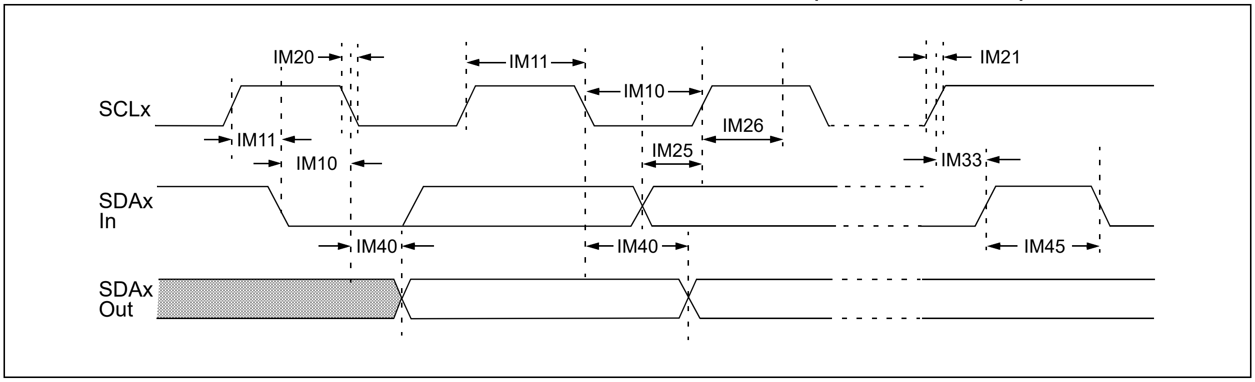


TABLE 33-27: I2Cx BUS DATA TIMING REQUIREMENTS (MASTER MODE)

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristics**		**Min.(1)**	**Max.**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY * (BRG + 1)	—	μs	
			400 kHz mode	TCY * (BRG + 1)	—	μs	
			1 MHz mode	TCY * (BRG + 1)	—	μs	
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY * (BRG + 1)	—	μs	
			400 kHz mode	TCY * (BRG + 1)	—	μs	
			1 MHz mode	TCY * (BRG + 1)	—	μs	
IM20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 x (VDD/5.5V)	300	ns	
			1 MHz mode	20 x (VDD/5.5V)	120	ns	
IM21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode	—	120	ns	
IM25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	
IM26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	μs	
			400 kHz mode	0	0.9	μs	
			1 MHz mode	0	0.3	μs	
IM30	TSU:STA	Start Condition Setup Time	100 kHz mode	TCY * (BRG + 1)	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	TCY * (BRG + 1)	—	μs	
			1 MHz mode	TCY * (BRG + 1)	—	μs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	TCY * (BRG + 1)	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	TCY * (BRG + 1)	—	μs	
			1 MHz mode	TCY * (BRG + 1)	—	μs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	TCY * (BRG + 1)	—	μs	
			400 kHz mode	TCY * (BRG + 1)	—	μs	
			1 MHz mode	TCY * (BRG + 1)	—	μs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	TCY * (BRG + 1)	—	ns	
			400 kHz mode	TCY * (BRG + 1)	—	ns	
			1 MHz mode	TCY * (BRG + 1)	—	ns	
IM40	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3450	ns	
			400 kHz mode	—	900	ns	
			1 MHz mode	—	450	ns	
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	μs	The amount of time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	
IM50	CB	Bus Capacitive Loading	100 kHz mode	—	400	pF	
			400 kHz mode	—	400	pF	
			1 MHz mode	—	10	pF	
IM51	TPGD	Pulse Gobbler Delay		65	390	ns	

Note 1: BRG is the value of the I^2C Baud Rate Generator.

FIGURE 33-12: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

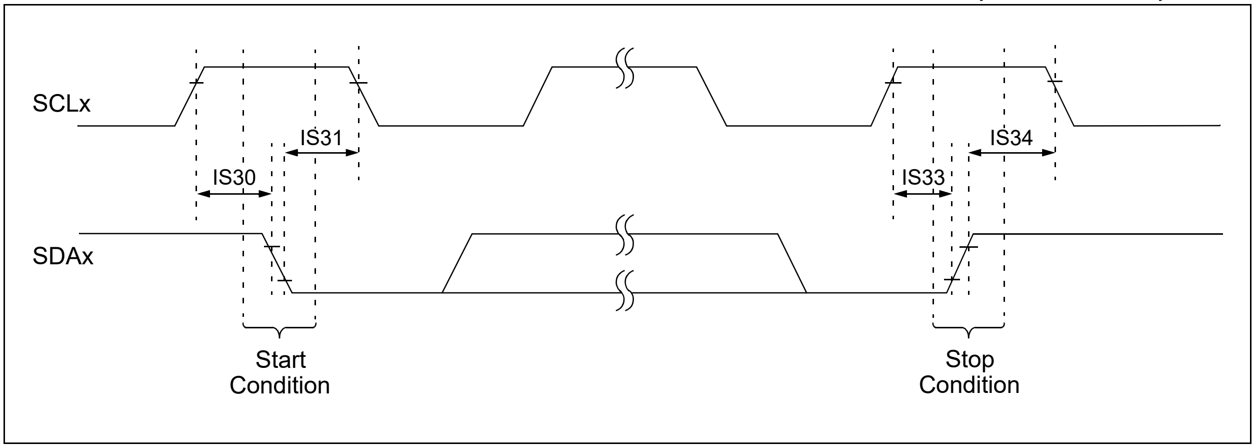


FIGURE 33-13: I2Cx BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)

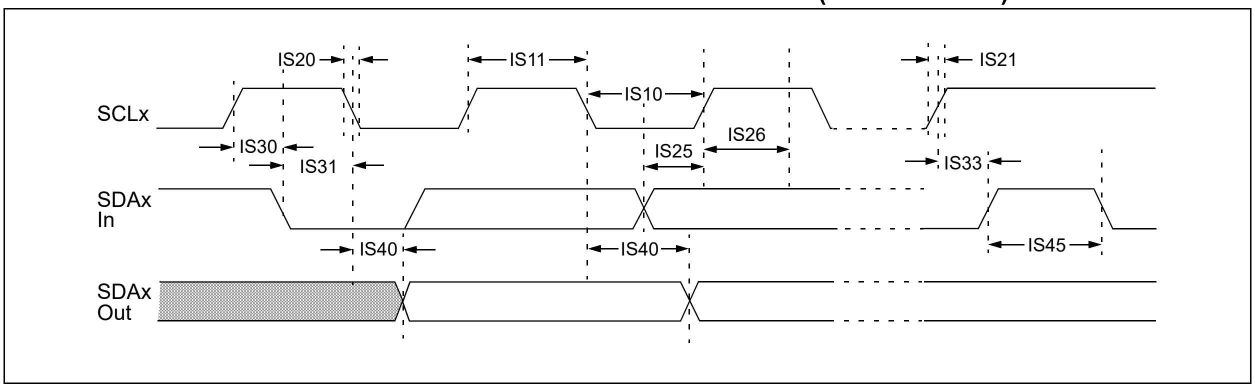


TABLE 33-28: I2Cx BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7	—	μs	CPU clock must be a minimum 800 kHz
			400 kHz mode	1.3	—	μs	CPU clock must be a minimum 3.2 MHz
			1 MHz mode	0.5	—	μs	
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	—	μs	CPU clock must be a minimum 800 kHz
			400 kHz mode	0.6	—	μs	CPU clock must be a minimum 3.2 MHz
			1 MHz mode	0.26	—	μs	
IS20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 x (VDD/5.5V)	300	ns	
			1 MHz mode	20 x (VDD/5.5V)	120	ns	
IS21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode	—	120	ns	
IS25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	
IS26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
			1 MHz mode	0	0.3	μs	
IS30	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	
IS31	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	
IS33	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.0	—	μs	
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	
IS34	THD:STO	Stop Condition Hold Time	100 kHz mode	> 0	—	μs	
			400 kHz mode	> 0	—	μs	
			1 MHz mode	> 0	—	μs	
IS40	TAA:SCL	Output Valid from Clock	100 kHz mode	0	3.45	μs	
			400 kHz mode	0	0.9	μs	
			1 MHz mode	0	0.45	μs	
IS45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	μs	The amount of time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	
IS50	CB	Bus Capacitive Loading	100 kHz mode	—	400	pF	
			400 kHz mode	—	400	pF	
			1 MHz mode	—	10	pF	

FIGURE 33-14: UARTx MODULE TIMING CHARACTERISTICS

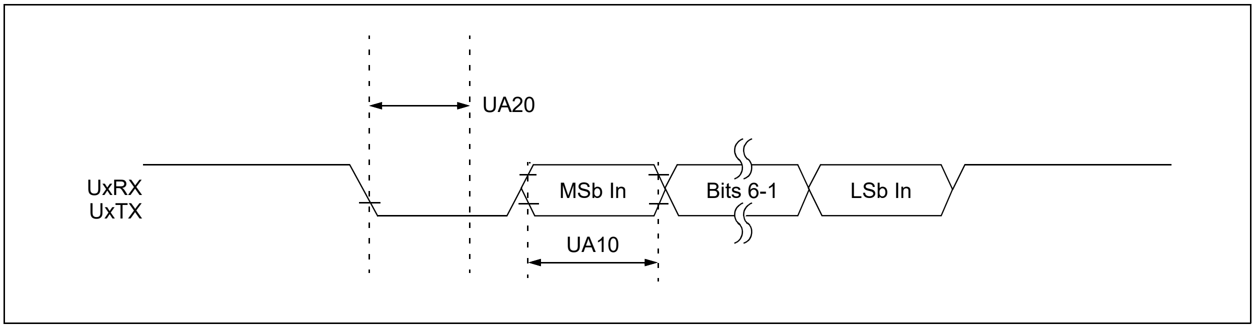


TABLE 33-29: UARTx MODULE TIMING REQUIREMENTS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Symbol**	**Characteristic[1]**	**Min.**	**Max.**	**Units**
UA10	TUABAUD	UARTx Baud Time	40	—	ns
UA11	FBAUD	UARTx Baud Rate	—	40	Mbps
UA20	TCWF	Start Bit Pulse Width to Trigger UARTx Wake-up	50	—	ns

Note 1: These parameters are characterized but not tested in manufacturing.

TABLE 33-30: ADC MODULE SPECIFICATIONS

Standard Operating Conditions: (unless otherwise stated)
3.0V to 3.6V[4]
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristics	Min.	Typical	Max.	Units	Conditions
Clock Requirements							
AD9	FSRC	ADC Module Input Frequency	—	—	500	MHz	Clock frequency selected by the CLKSELx bits
AD10	FCORESRC	ADC Control Clock Frequency	—	—	250	MHz	Clock frequency after the first divider controlled by the CLKDIVx bits
AD11	FADCORE	ADC SAR Core Clock Frequency	—	—	70	MHz	SAR core frequency after the second divider controlled by the ADCSx or SHRADCSx bits
Analog Input							
AD12	VINH-VINL	Full-Scale Input Span	AVSS	—	AVDD	V	
AD14	VIN	Absolute Input Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
AD61	CHOLD	Capacitance	—	18	—	pF	Shared core[1]
AD62	RIC	Input resistance	—	500	1000	Ω	Note 1
AD66	VBG	Internal Voltage Reference Source	1.14	1.2	1.26	V	
ADC Accuracy							
AD20	Nr	Resolution	12 data bits			bits	
AD21b	INL _1S	Shared Core Integral Nonlinearity (1 Active Core)	-3.5	-1.5/+1.5	+3.5	LSb	2.7 Msps[5], TADC = 4 ns (250 MHz), TCORESRC = 8 ns (125 MHz), TADCORE = 16 ns (62.5 MHz), Sampling Time = 10 TADCORE, VDD = 3.3V, AVDD = 3.3V
AD22b	DNL_1S	Shared Core Differential Nonlinearity (1 Active Core)	-1	1.5	+3.5	LSb	
AD23b	GERR_1S	Shared Core Gain Error (1 Active Core)	—	+4	—	LSb	
AD24b	OERR_1S	Shared Core Offset Error (1 Active Core)	—	-4	—	LSb	
AD25c	—	Monotonicity	—	—	—	—	Guaranteed
Dynamic Performance							
AD31b	SINAD	Signal-to-Noise and Distortion	56	—	70	dB	Notes 2, 3
AD34b	ENOB	Effective Number of Bits	9.8	10.2	11.4	bits	Notes 2, 3
AD50	TAD	ADC Clock Period	14.3	—	—	ns	
AD51	FTP	Throughput Rate	—	—	2.7	Msps	Shared core[5]

Note 1: These parameters are not characterized or tested in manufacturing.
2: These parameters are characterized but not tested in manufacturing.
3: Characterized with a 1 kHz sine wave.
4: The ADC module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. Unless otherwise stated, module functionality is ensured, but not characterized.
5: For the shared core, the throughput includes a 10 TADCORE sampling time and 13 TADCORE conversion time.

TABLE 33-31: ANALOG-TO-DIGITAL CONVERSION TIMING REQUIREMENTS[1]

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Symbol**	**Characteristics**	**Min.**	**Max.**	**Units**
AD50	TAD	ADC Clock Period	14.28	—	ns
AD51	FTP	ADC Throughput Rate (for all channels)	—	3.5	Msps

Note 1: The equivalent model of the input stages of the ADC include the Interconnect Resistance (RIC). The RIC value is 1 kOhm (max) and the Sample/Hold Capacitance (CHOLD) value is 14 pF. For additional information, refer to **"12-Bit High-Speed, Multiple SARs A/D Converter (ADC)"** (www.microchip.com/DS70005213).

TABLE 33-32: DIE TEMPERATURE DIODE SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.[1]**	**Max.**	**Units**	**Comments**
TD01	TCOEFF	Temperature Coefficient	—	1.5	—	mV/C	Note 1

Note 1: These parameters are not characterized or tested in manufacturing.

TABLE 33-33: HIGH-SPEED ANALOG COMPARATOR MODULE SPECIFICATIONS

Operating Conditions (unless otherwise stated):[2] 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.[1]**	**Max.**	**Units**	**Comments**
CM09	FIN	Input Frequency	400	—	500	MHz	-40°C ≤ TA ≤ +85°C
			400	—	480		+85°C < TA ≤ +125°C
CM10	VIOFF	Input Offset Voltage	-20	—	20	mV	
CM11	VICM	Input Common-Mode Voltage Range	AVSS	—	AVDD	V	Note 1
CM13	CMRR	Common-Mode Rejection Ratio	65	—	—	dB	Note 1
CM14	TRESP	Large Signal Response	—	15	—	ns	V+ input step of 100 mV while V- input is held at AVDD/2
CM15	VHYST	Input Hysteresis	15	—	45	mV	Depends on HYSSEL[1:0][1]

Note 1: These parameters are for design guidance only and are not tested in manufacturing.
2: The comparator module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. Unless otherwise stated, module functionality is tested but not characterized.

TABLE 33-34: DAC MODULE SPECIFICATIONS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic	Min.	Typ.[1]	Max.	Units	Comments
DA02	CVRES	Resolution	12			bits	
DA03	INL	Integral Nonlinearity Error	-38	—	0	LSb	
DA04	DNL	Differential Nonlinearity Error	-5	—	5	LSb	
DA05	EOFF	Offset Error	-3.5	—	21.5	LSb	
DA06	EG	Gain Error	0	—	41	LSb	
DA07	TSET	Settling Time	600	750	2000	ns	Output with 2% of desired output voltage with a 10-90% or 90-10% step
DA08	VOUT	Voltage Output Range	0.165	—	3.135	V	VDD = 3.3V[1]
DA09	TTR	Transition Time	340	—	—	ns	**Note 1**
DA10	TSS	Steady-State Time	550	—	—	ns	**Note 1**

Note 1: Parameters are for design guidance only and are not tested.

TABLE 33-35: DAC OUTPUT (DACOUT PIN) SPECIFICATIONS

Operating Conditions (unless otherwise stated):
3.0V ≤ VDD ≤ 3.6V,
-40°C ≤ TA ≤ +85°C for Industrial
-40°C ≤ TA ≤ +125°C for Extended

Param No.	Symbol	Characteristic	Min.	Typ.	Max.	Units	Comments
DA11	RLOAD	Resistive Output Load Impedance	10K	—	—	Ohm	
DA11a	CLOAD	Output Load Capacitance	—	—	35	pF	Including output pin capacitance
DA12	IOUT	Output Current Drive Strength	—	3	—	mA	Sink and source

TABLE 33-36: CURRENT BIAS GENERATOR SPECIFICATIONS[1]

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
ParamNo.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**
CC03	I_{10SRC}	10 μA Source Current	8.8	11.2	μA
CC04	I_{50SRC}	50 μA Source Current	44	56	μA
CC05	I_{50SNK}	50 μA Sink Current	-44	-56	μA

Note 1: Parameters are characterized but not tested in manufacturing.

TABLE 33-37: OPERATIONAL AMPLIFIER SPECIFICATIONS[1,2]

Operating Conditions (unless otherwise stated): 3.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
OAMP1	GBWP	Gain Bandwidth Product	—	20	—	MHz	
OAMP2	SR	Slew Rate	—	40	—	V/μs	
OAMP3	V_{IOFF}	Input Offset Voltage	-3/+3	-1/+1	-3/+3	mV	Unity gain configuration
			-8	-3/+3	+8		Open-loop configuration
OAMP4	V_{ICM}	Common-Mode Input Voltage Range	AVSS	—	AVDD	V	NCHDISx = 0
			AVSS	—	AVDD – 1.4V	V	NCHDISx = 1
OAMP5	CMRR	Common-Mode Rejection Ratio	—	68	—	db	
OAMP6	PSRR	Power Supply Rejection Ratio	—	74	—	dB	
OAMP7	V_{OR}	Output Voltage Range	AVSS	—	AVDD	mV	0.5V input overdrive, no output loading
OAMP8	V_{IBC}	Input Bias Current	—	—	—	mV	Note 2
OAMP11	C_{LOAD}	Output Load Capacitance	—	—	30	pF	Including output pin capacitance
OAMP12	I_{OUT}	Output Current Drive Strength	—	3	—	mA	Sink and source
OAMP13	P_{MARGIN}	Phase Margin	44	—	—	degree	Unity gain [1]
OAMP14	G_{MARGIN}	Gain Margin	7	—	—	dB	Unity gain[1]
OAMP15	OLG	Open-Loop Gain	68	75	—	dB	Note 1

Note 1: Parameters are for design guidance only and are not tested in manufacturing.

2: The op amps use CMOS input circuitry with negligible input bias current. The maximum "effective bias current" is the I/O pin leakage specified by electrical Parameter DI50.

3: Parameters are characterized but not tested in manufacturing.

4: The module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. The module functionality is tested, but not characterized.

34.0 HIGH-TEMPERATURE ELECTRICAL CHARACTERISTICS

This section provides an overview of the dsPIC33CDVL64MC106 family devices operating in an ambient temperature range of -40°C to +150°C.

The specifications between -40°C to +150°C are identical to those shown in **Section 33.0 "Electrical Characteristics"** for operation between -40°C to +125°C, with the exception of the parameters listed in this section.

Parameters in this section begin with an H, which denotes High temperature. For example, Parameter DC20 in **Section 33.0 "Electrical Characteristics"** is the Industrial and Extended temperature equivalent of HDC20.

Absolute maximum ratings for the dsPIC33CDVL64MC106 family high-temperature devices are listed below. Exposure to these maximum rating conditions for extended periods can affect device reliability. Functional operation of the device, at these or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings[1]

Ambient temperature under bias .... -40°C to +150°C
Storage temperature .... -65°C to +150°C
Voltage on VDD with respect to VSS .... -0.3V to +4.0V
Voltage on any pin that is not 5V tolerant with respect to VSS[3] .... -0.3V to (VDD + 0.3V)
Voltage on any 5V tolerant pin with respect to VSS when VDD ≥ 3.0V[3] .... -0.3V to +5.5V
Voltage on any 5V tolerant pin with respect to Vss when VDD < 3.0V[3] .... -0.3V to +3.6V
Maximum current out of VSS pin .... 300 mA
Maximum current into VDD pin[2] .... 300 mA
Maximum current sunk/sourced by any regular I/O pin .... 15 mA
Maximum current sunk/sourced by an I/O pin with increased current drive strength (RB1, RC8, RC9 and RD8) .... 25 mA
Maximum current sunk by a group of I/Os between two VSS pins[4] .... 75 mA
Maximum current sourced by a group of I/Os between two VDD pins[4] .... 75 mA
Maximum current sunk by all I/Os[2] .... 200 mA
Maximum current sourced by all I/Os[2] .... 200 mA

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those, or any other conditions above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

2: Maximum allowable current is a function of device maximum power dissipation (see Table 34-2).

3: See the **"Pin Diagrams"** section for the 5V tolerant pins.

4: Not applicable to AVDD and AVSS pins.

34.1 DC Characteristics

TABLE 34-1: OPERATING MIPS vs. VOLTAGE

VDD Range	Temperature Range	Maximum CPU Clock Frequency
3.0V to 3.6V	-40°C to +150°C	70

TABLE 34-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min.	Max.	Unit
High-Temperature Devices				
Operating Junction Temperature Range	T_J	-40	+165	°C
Operating Ambient Temperature Range	T_A	-40	+150	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $I/O = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$		W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$		W

TABLE 34-3: OPERATING VOLTAGE SPECIFICATIONS

Operating Conditions (unless otherwise stated):
-40°C ≤ T_A ≤ +150°C for High

Param No.	Symbol	Characteristic	Min.	Max.	Units	Conditions
HDC10	V_{DD}	**Supply Voltage**	3.0	3.6	V	
HDC16	V_{POR}	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	—	V_{SS}	V	
HDC17	S_{VDD}	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.03	—	V/ms	0V-3V in 100 ms
HBO10	V_{BOR}(1)	**BOR Event** on VDD Transition High-to-Low	2.68	2.99	V	

Note 1: Device is functional at $V_{BORMIN} < V_{DD} < V_{DDMIN}$. Analog modules (ADC and comparators) may have degraded performance. The VBOR parameter is for design guidance only and is not tested in manufacturing.

TABLE 34-4: OPERATING CURRENT (I_{DD})[2]

Parameter No.	Typ.[1]	Max.	Units	Conditions		
HDC20	10.2	24.1	mA	+150°C	3.3V	10 MIPS (N1 = 1, N2 = 5, N3 = 2, M = 50, F_{VCO} = 400 MHz, F_{PLLO} = 40 MHz)
HDC21	12.2	25.4	mA	+150°C	3.3V	20 MIPS (N1 = 1, N2 = 5, N3 = 1, M = 50, F_{VCO} = 400 MHz, F_{PLLO} = 80 MHz)
HDC22	15.5	29.0	mA	+150°C	3.3V	40 MIPS (N1 = 1, N2 = 3, N3 = 1, M = 60, F_{VCO} = 480 MHz, F_{PLLO} = 160 MHz)
HDC23	21.2	34.1	mA	+150°C	3.3V	70 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 70, F_{VCO} = 560 MHz, F_{PLLO} = 280 MHz)

Note 1: Data in the "Typ." column are for design guidance only and are not tested.

2: Base Run current (I_{DD}) is measured as follows:

- Oscillator is switched to EC+PLL mode in software
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to V_{DD} – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFCN (FOSC[2]) = `0`)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = `01`)
- Watchdog Timer is disabled (FWDT[15] = `0` and WDTCONL[15] = `0`)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all '`1`'s)
- JTAG is disabled (JTAGEN (FICD[5]) = `0`)
- `NOP` instructions are executed in `while(1)` loop

TABLE 34-5: IDLE CURRENT (I_{IDLE})(2)

Parameter No.	Typ.(1)	Max.	Units	Conditions		
HDC40	9.0	17.1	mA	+150°C	3.3V	10 MIPS (N1 = 1, N2 = 5, N3 = 2, M = 50, F_{VCO} = 400 MHz, F_{PLLO} = 40 MHz)
HDC41	9.7	22.6	mA	+150°C	3.3V	20 MIPS (N1 = 1, N2 = 5, N3 = 1, M = 50, F_{VCO} = 400 MHz, F_{PLLO} = 80 MHz)
HDC42	11.2	24	mA	+150°C	3.3V	40 MIPS (N1 = 1, N2 = 3, N3 = 1, M = 60, F_{VCO} = 480 MHz, F_{PLLO} = 160 MHz)
HDC43	13.4	25.8	mA	+150°C	3.3V	70 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 70, F_{VCO} = 560 MHz, F_{PLLO} = 280 MHz)

Note 1: Data in the “Typ.” column are for design guidance only and are not tested.

2: Base Idle current (I_{IDLE}) is measured as follows:

- Oscillator is switched to EC+PLL mode in software
- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to V_{DD} – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFCN (FOSC[2]) = 0)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = 01)
- Watchdog Timer is disabled (FWDT[15] = 0 and WDTCONL[15] = 0)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all ‘1’s)
- JTAG is disabled (JTAGEN (FICD[5]) = 0)
- Flash in standby with NVMSIDL (NVMCON[12]) = 1

TABLE 34-6: POWER-DOWN CURRENT (IPD)[2]

Parameter No.	Characteristic	Typ.[1]	Max.	Units	Conditions	
HDC60	Base Power-Down Current	6.3	19.8	mA	+150°C	3.3V

Note 1: Data in the "Typ." column are for design guidance only and are not tested.

2: Base Sleep current (IPD) is measured as follows:

- OSC1 pin is driven with external 8 MHz square wave with levels from 0.3V to VDD – 0.3V
- OSC2 is configured as an I/O in the Configuration Words (OSCIOFCN (FOSC[2]) = `0`)
- FSCM is disabled (FCKSM[1:0] (FOSC[7:6]) = `01`)
- Watchdog Timer is disabled (FWDT[15] = `0` and WDTCONL[15] = `0`)
- All I/O pins (except OSC1) are configured as outputs and driving low
- No peripheral modules are operating or being clocked (defined PMDx bits are all '`1`'s)
- JTAG is disabled (JTAGEN (FICD[5]) = `0`)
- The regulators are in Active mode, VREGS bit = `1` (Standby mode only valid up to +85°C)
- The regulators are in Full-Power mode, LPWREN bit = `0` (Low-Power mode only valid up to +85°C)

TABLE 34-7: DOZE CURRENT (IDOZE)

Parameter No.	Typ.[1]	Max.	Doze Ratio	Units	Conditions		
HDC70	17.9	30.1	1:2	mA	+150°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	13.6	26	1:128	mA			

Note 1: Data in the "Typ." column are for design guidance only and are not tested.

TABLE 34-8: WATCHDOG TIMER DELTA CURRENT (ΔIWDT)[1]

Parameter No.	Typ.	Max.	Units	Conditions	
HDC61	24	—	μA	+150°C	3.3V

Note 1: The ΔIWDT current is the additional current consumed when the module is enabled. This current should be added to the base IPD current. All parameters are characterized but not tested during manufacturing.

TABLE 34-9: PWM DELTA CURRENT

Parameter No.	Typ.	Max.	Units	Conditions		
HDC100	5.48	7.2	mA	+150°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (F_{PLLO} = 500 MHz, VCO = 1000 MHz, PLLFBD = 125)
HDC101	4.44	6.8	mA	+150°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (F_{PLLO} = 400 MHz, VCO = 400 MHz, PLLFBD = 50)
HDC102	2.31	3.7	mA	+150°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (F_{PLLO} = 200 MHz, VCO = 200 MHz, PLLFBD = 50)
HDC103	1.22	2.3	mA	+150°C	3.3V	PWM Output Frequency = 500 kHz, PWM Input (F_{PLLO} = 100 MHz, VCO = 100 MHz, PLLFBD = 50)

TABLE 34-10: ADC DELTA CURRENT(1)

Parameter No.	Typ.	Max.	Units	Conditions		
HDC120	3.76	6.1	mA	+150°C	3.3V	T_{AD} = 14.3 ns (3.5 Msps conversion rate)

Note 1: Shared core continuous conversion. T_{AD} = 14.3 ns (3.5 Msps conversion rate). Listed delta currents are for only one ADC core. All parameters are characterized but not tested during manufacturing.

TABLE 34-11: COMPARATOR + DAC DELTA CURRENT

Parameter No.	Symbol	Characteristic	Min.	Typ.	Max.	Units	Conditions		
HDC130	—	—	—	1.25	1.65	mA	+150°C	3.3V	FPLLO @ 500 MHz[1]
HDCM09	FIN	Input Frequency	400	—	475	MHz	+125°C < TA ≤ +150°C		

Note 1: Listed delta currents are for only one comparator + DAC instance. All parameters are characterized but not tested during manufacturing.

TABLE 34-12: OP AMP DELTA CURRENT[1]

Parameter No.	Typ.	Max.	Units	Conditions	
HDC140	0.58	2.3	mA	+150°C	3.3V

Note 1: Listed delta currents are for only one op amp instance. All parameters are characterized but not tested during manufacturing.

TABLE 34-13: I/O PIN INPUT SPECIFICATIONS

Operating Conditions (unless otherwise stated):
3.0V < VDD < 3.6V
-40°C < TA < +150°C for High

Param No.	Symbol	Characteristic	Min.[3]	Max.[4]	Units
HDI50	IIL	**Input Leakage Current**[1]			
		I/O Pins 5V Tolerant[2]	-800	800	nA
		I/O Pins Not 5V Tolerant[2]	-800	800	nA
		$\overline{\text{MCLR}}$	-800	800	nA
		OSCI	-800	800	nA

Note 1: Negative current is defined as current sourced by the pin.
2: See the **Pin Diagrams** section for the 5V tolerant I/O pins.
3: VPIN = VSS.
4: VPIN = VDD.

TABLE 34-14: INTERNAL FRC ACCURACY

Operating Conditions (unless otherwise stated):
3.0V < VDD < 3.6V
-40°C < TA < +150°C for High

Param No.	Characteristic	Min.	Max.	Units
HF20a	FRC @ 8 MHz[1]	-3	+3	%

Note 1: Frequency is calibrated at +25°C and 3.3V.

TABLE 34-15: HIGH-SPEED ANALOG COMPARATOR MODULE SPECIFICATIONS[1]

Operating Conditions (unless otherwise stated): 3.0V < V_{DD} < 3.6V -40°C < T_A < +150°C for High							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
CM09	F_{IN}	Input Frequency	400	—	475	MHz	

Note 1: These parameters are for design guidance only and are not tested in manufacturing.

TABLE 34-16: DAC MODULE SPECIFICATIONS

Operating Conditions (unless otherwise stated): 3.0V < V_{DD} < 3.6V -40°C < T_A < +150°C for High							
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
HDA03	INL	Integral Nonlinearity Error	-45	—	0	LSb	
HDA04	DNL	Differential Nonlinearity Error	-5	—	5	LSb	
HDA05	EOFF	Offset Error	-21	—	21	LSb	
HDA06	EG	Gain Error	-41	—	41	LSb	

35.0 MOSFET GATE DRIVER ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings†

Input Voltage, HVDD....(GND – 0.3V) to +40V
Internal Power Dissipation....Internally Limited
Operating Junction Temperature(2) ....-40°C to +165°C
Transient Junction Temperature (1)....+170°C
Storage Temperature (2) ....-55°C to +165°C
Digital I/O .... -0.3V to 5.5V
Low-Voltage Analog I/O.... -0.3V to 5.5V
VBx, WAKE ....(GND – 0.3V) to +40V
PHx, HSx....(GND – 5.5V) to +40V
VBOOT, LSx....(GND – 0.3V) to +13.2V
CAP1, CAP2 ....(GND – 0.3V) to +40V

† **Notice:** Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational sections of this specification is not intended. Exposure to maximum rating conditions for extended periods may affect device reliability.

Note 1: Transient junction temperatures should not exceed one second in duration. Sustained junction temperatures above +170°C may impact the device reliability.

2: The maximum allowable power dissipation is a function of ambient temperature, the maximum allowable junction temperature and the thermal resistance from junction to air (i.e., TA, TJ, θJA). Exceeding the maximum allowable power dissipation may cause the device operating junction temperature to exceed the maximum +165°C rating. Sustained junction temperatures above +165°C can impact the device reliability.

TABLE 35-1: AC/DC CHARACTERISTICS

Electrical Specifications: Unless otherwise noted: TJ = -40°C to +150°C; typical values are for +25°C, HVDD = 13.5V, CVBOOT = 4.7 µF, CVREG = 4.7 µF, CCP = 220 nF.

Parameters	Symbol	Min.	Typ.	Max.	Units	Conditions
Power Supply Input						
Input Operating Voltage	HVDD	4.5	—	40	V	VREG active
		6.0	—	29.0		Driver output active
Input Supply Current	ISUP	—	5	15	µA	Sleep mode, TJ = +25°C
		—	180	330		Standby, OE = 0V
		—	500	—		Active, HVDD > 13, 5V, OE > VDIG_HI_TH
		—	1200	—		Active, HVDD = 6V, TJ = +25°C
Input Supply Current	ISUP	—	5	15	µA	Sleep mode, TJ = +25°C
		—	200	350		Standby, OPAMP = 1, OE = 0V
		—	800	1300		Standby, OPAMP = 0, OE = 0V
		—	1000	—		Active, HVDD > 13, 5V, OE > VDIG_HI_TH
		—	1500	—		Active, OE > VDIG_HI_TH, HVDD = 7V, TJ = +25°C

Note 1: Limits based on design, simulation or characterization. Not production tested.

TABLE 35-1: AC/DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise noted: T_J = -40°C to +150°C; typical values are for +25°C, HV_{DD} = 13.5V, CV_{BOOT} = 4.7 µF, CV_{REG} = 4.7 µF, C_{CP} = 220 nF.						
Parameters	**Symbol**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Bias Generator						
+12V Regulated Charge Pump (V_{BOOT})						
Charge Pump Current	I_{CP}	20	—	—	mA	HV_{DD} = 9.0V
Charge Pump Start	CP_{START}	12.50	12.75	—	V	Falling
Charge Pump Stop	CP_{STOP}	—	13.25	14	V	Rising
Charge Pump Frequency	CP_{FSW}	—	76.80	—	kHz	HV_{DD} = 9.0V
		—	0	—		HV_{DD} = 14V
Charge Pump Switch Resistance	CP_{RDSON}	—	14	—	Ω	R_{DSON} sum of high side and low side(1)
Output Voltage	V_{BOOT}	—	12	—	V	$HV_{DD} \geq$ 14V, I_{OUT} = 30 mA
		9	12	—		7V $\leq HV_{DD} <$ 14V, C_{CP} = 150 nF, I_{OUT} = 20 mA
		9	—	—		6.25V $\leq HV_{DD} <$ 7V, C_{CP} = 270 nF, I_{OUT} = 15 mA
Output Voltage Tolerance	$\|TOLV_{OUT12}\|$	—	—	4.0	%	I_{OUT} = 30 mA
Output Capability	I_{BOOT}	30	—	—	mA	Average current
Output Current Limit	$I_{BOOTLIMIT}$	50	60	80	mA	Average current
Output Voltage Temperature Coefficient	TCV_{OUT12}	—	160	—	ppm/°C	Note 1
Line Regulation	$\|\Delta V_{OUT}/(V_{OUT} \times \Delta)\|$	—	0.1	0.5	%/V	14V < HV_{DD} < 19V, I_{OUT} = 30 mA
Load Regulation	$\|\Delta V_{OUT}/V_{OUT}\|$	—	0.2	1.0	%	I_{OUT} = 0.1 mA to 30 mA, HV_{DD} = 14V
Power Supply Rejection Ratio	PSRR	—	60	—	dB	f = 1 kHz, I_{OUT} = 10 mA(1)
Output Capacitor Capacitance Range	CV_{BOOT}	4.7	—	10	µF	Ceramic, Tantalum, Electrolytic(1)
Output Capacitor ESR Range	$CESR_{VBOOT}$	0.010	—	1.0	Ω	Note 1
Flying Capacitor Capacitance Range	C_{CP}	100	220	1000	nF	Note 1
V_{BOOT} Ready Threshold	V_{12SM_PG}	—	50	—	$\%V_{BOOT}$	State machine V_{BOOT} Power Good threshold to move to next state (1)

Note 1: Limits based on design, simulation or characterization. Not production tested.

TABLE 35-1: AC/DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise noted: T_J = -40°C to +150°C; typical values are for +25°C, HV_{DD} = 13.5V, CV_{BOOT} = 4.7 µF, CV_{REG} = 4.7 µF, C_{CP} = 220 nF.

Parameters	Symbol	Min.	Typ.	Max.	Units	Conditions
+3.3V/+5V Linear Regulator (VREG)						
Output Voltage	V_{REG}	—	—	—	V	HV_{DD} = 6V, I_{OUT} = 70 mA
		4.8	5	5.2		V_{REG} = 5V
		3.168	3.3	3.432		V_{REG} = 3.3V
Output Voltage Tolerance	$\|TOLV_{REG}\|$	—	—	4.0	%	
Output Current	I_{OUT}	70	—	—	mA	Average current
Output Foldback Current Corner	I_{FOLD}	80	95	120	mA	Average current
Output Foldback Current Limit	I_{FOLD_LIM}	—	10	—	mA	R_{LOAD} = 10 mΩ
Line Regulation	$\|\Delta V_{OUT}/(V_{OUT} \times \Delta V_{DD})\|$	—	0.1	0.5	%/V	V_{REG} = 3.3V: 6V < HV_{DD} < 19V, I_{OUT} = 70 mA; V_{REG} = 5V: 7.5V < HV_{DD} < 19V, I_{OUT} = 70 mA
Load Regulation	$\|\Delta V_{OUT}/V_{OUT}\|$	—	0.2	1.0	%	I_{OUT} = 0.1 mA to 70 mA
Power Supply Rejection Ratio	PSRR	—	60	—	dB	f = 1 kHz, I_{OUT} = 10 mA [1]
Output Capacitor Capacitance Range	CV_{REG}	4.7	—	30	µF	Ceramic, Tantalum, Electrolytic[1]
Output Capacitor ESR Range	$CESR_{VREG}$	0.010	—	1.0	Ω	Note 1
Voltage Supervisor						
V_{REG} Undervoltage Fault Inactive	$VREGUVF_{INACT}$	—	92	—	$\%V_{REG}$	V_{REG} rising
V_{REG} Undervoltage Fault Active	$VREGUVF_{ACT}$	—	88	—	$\%V_{REG}$	V_{REG} falling
V_{REG} Undervoltage Fault Hysteresis	$VREGUVF_{HYS}$	—	4	—	$\%V_{REG}$	
HV_{DD} Undervoltage Lockout Inactive	$UVLO_{INACT}$	—	6.0	6.25	V	Rising
HV_{DD} Undervoltage Lockout Active	$UVLO_{ACT}$	5.1	5.5	—	V	Falling
HV_{DD} Undervoltage Lockout Hysteresis	$UVLO_{HYS}$	—	0.5	—	V	
HV_{DD} Undervoltage Shutdown Active	$UVSHDN_{ACT}$	4.0	4.25	4.5	V	HV_{DD} < $UVSHDN_{ACT}$
HV_{DD} Undervoltage Shutdown Inactive	$UVSHDN_{INACT}$	$UVLO_{INACT}$			V	HV_{DD} > $UVLO_{INACT}$
HV_{DD} Overvoltage Lockout Active	$OVLO_{ACT}$	—	32.0	33.0	V	HV_{DD} rising
HV_{DD} Overvoltage Lockout Inactive	$OVLO_{INACT}$	29.0	30.0	—	V	HV_{DD} falling
HV_{DD} Overvoltage Lockout Hysteresis	$OVLO_{HYS}$	—	2.0	—	V	
Temperature Supervisor						
Thermal Warning Temperature	T_{WARN}	—	140	—	°C	Rising temperature
Thermal Warning Hysteresis	ΔT_{WARN}	—	15	—	°C	Falling temperature
Thermal Shutdown Temperature	T_{SD}	170	210	—	°C	Rising temperature[1]
Thermal Shutdown Hysteresis	ΔT_{SD}	—	25	—	°C	Falling temperature

Note 1: Limits based on design, simulation or characterization. Not production tested.

TABLE 35-1: AC/DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise noted: TJ = -40°C to +150°C; typical values are for +25°C, HVDD = 13.5V, CVBOOT = 4.7 µF, CVREG = 4.7 µF, CCP = 220 nF.

Parameters	Symbol	Min.	Typ.	Max.	Units	Conditions
Motor Control Unit						
Gate Output Drivers						
Output Driver Source Current	ISOURCE	0.25	0.37	—	A	HS[A:C], LS[A:C] (1)
Output Driver Sink Current	ISINK	0.3	0.49	—	A	HS[A:C], LS[A:C] (1)
Output Driver Source Resistance	RDSONSOURCE	—	14	26	Ω	IOUT = -10 mA, HS[A:C], LS[A:C]
Output Driver Sink Resistance LS	RDSONSINKLS	—	14	26	Ω	IOUT = 10 mA, LS[A:C]
Output Driver Sink Resistance HS Dynamic	RDSONSINKHSDYN	—	14	26	Ω	IOUT = 10 mA, HS[A:C], t < 1 ms
Output Driver Sink Resistance HS	RDSONSINKHS	—	19	31	Ω	IOUT = 10 mA, HS[A:C]
Output Driver Fault Blanking Time (UVLO and OCP); Set in the DRVBL[1:0] bits (CFG2[1:0])	tBLANK	3900	4400	4900	ns	00 – Default (1)
		2000	2200	2400		01(1)
		900	1100	1300		10(1)
		400	550	700		11(1)
Output Driver UVLO Threshold	VDUVLO	4	—	4.5	V	Configuration Register 0 (bit 3 = 0)
Output Driver PWM Dead Time; Set in the DRVDT[2:0] bits (CFG2[4:2])	tPWM_DEAD	1800	2000	2200	ns	000 – Default (1)
		1550	1750	1950		001(1)
		1350	1500	1650		010(1)
		1100	1250	1400		011(1)
		900	1000	1150		100(1)
		650	750	900		101(1)
		450	500	650		110(1)
		200	250	350		111(1)
Output Driver Propagation Delay Time On	tGATE_PROP_ON	—	40	80	ns	From PWMxy active to HSx/LSx > 10% (1)
Output Driver Propagation Delay Time Off	tGATE_PROP_OFF	—	40	80	ns	From PWMxy inactive to HSx/LSx < 90% (1)
Output Driver HS Drive Voltage	VHS	4.5	12	12.5	V	With respect to Phase pin(1)
Output Driver LS Drive Voltage	VLS	4.5	12	12.5	V	With respect to ground (1)
Output Driver Short-Circuit Protection Threshold (High Side: HVDD – VPHX), (Low Side: VPHX – PGND); Set in the EXTOC[1:0] bits (CFG0[1:0])	DSC_THR	0.230	0.250	0.270	V	00 – Default (1)
		0.470	0.500	0.530		01(1)
		0.720	0.750	0.780		10(1)
		0.960	1.000	1.040		11(1)
Output Driver Short-Circuit Filter Time	TSC_DLY	230	—	600	ns	CLOAD = 1000 pF, HVDD = 12V, detection after filtering(1)
Filter Time for All Other Faults	TFLT_DLY	1400	—	3600	ns	Note 1
Power-up or Sleep to Standby	tPOWER	—	5	—	ms	IVREG = 70 mA
Standby to Motor Operational	tMOTOR	—	35	—	µs	OE high-low-high transition < 1 ms Fault clearing pulse(1)
		—	5	10	ms	OE low-high transition, Standby state to operational(1)
		—	—	16	ms	OE low-high transition, Standby state to operational if VBOOT fails to reach V12SM_PG(1)

Note 1: Limits based on design, simulation or characterization. Not production tested.

TABLE 35-1: AC/DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise noted: TJ = -40°C to +150°C; typical values are for +25°C, HVDD = 13.5V, CVBOOT = 4.7 µF, CVREG = 4.7 µF, CCP = 220 nF.

Parameters	Symbol	Min.	Typ.	Max.	Units	Conditions
Fault to Driver Output Turn-Off	TFAULT_OFF	—	—	—	µs	CLOAD = 1000 pF, HVDD = 12V, time after Fault occurs(1)
		—	0.420	1.0		XOCP(1)
		—	2.4	4.0		OVLO(1)
		—	4.2	6.0		All other Faults(1)
OE Low to Driver Output Turn-Off	TDEL_OFF	—	3.2	4.0	µs	CLOAD = 1000 pF, HVDD = 12V, time after OE = Low(1)
OE Low to Standby State	tSTANDBY	0.9	—	1.35	ms	Time after OE = Low, SLEEP bit = 0
OE Low to Sleep State	tSLEEP	0.9	—	1.35	ms	Time after OE = Low, SLEEP bit = 1
OE Fault Clearing Pulse	tFAULT_CLR	1	—	900	µs	OE high-low-high transition time
Operational Amplifiers (DSTEMP)						
Input Offset Voltage	VOS	-10	—	+10	mV	VCM = 0V
Input Offset Temperature Drift	ΔVOS/ΔTA	—	±2.0	—	µV/°C	VCM = 0V(1)
Input Bias Current	IB	-1	—	+1	µA	
Common-Mode Input Range	VCMR	-0.3	—	VREG	V	
Common-Mode Rejection Ratio	CMRR	—	80	—	dB	Freq = 1 kHz, IOUT = 10 µA(1)
Maximum Output Voltage Range	VOL, VOH	0.15	—	VREG – 0.300	V	IOUT = ±200 µA
Slew Rate	SR	—	±7	—	V/µs	Symmetrical, CLOAD = 20 pF(1)
Gain Bandwidth Product	GBWP	4	10.0	—	MHz	**Note 1**
I/O Ports						
Digital Interface						
Digital Input/Output	DIGITALI/O	0	—	5.5	V	VREG = 5.0V version(1)
		0	—	3.3		VREG = 3.3V version(1)
Digital Open-Drain Low Voltage	DIGITALVI/O	—	—	50	mV	ILOAD = 1 mA
Digital Input Rising Threshold	VDIG_HI_TH	—	—	1.26	V	
Digital Input Falling Threshold	VDIG_LO_TH	0.54	—	—	V	
Digital Input Current	IDIG	—	30	100	µA	VDIG = 3.0V
		—	0.2	—		VDIG = 0V
Input Pull-Down Resistance	RPULLDN	—	51	—	kΩ	PWM[A:C]H/L, OE pins

Note 1: Limits based on design, simulation or characterization. Not production tested.

TABLE 35-1: AC/DC CHARACTERISTICS (CONTINUED)

Electrical Specifications: Unless otherwise noted: T_J = -40°C to +150°C; typical values are for +25°C, HV_{DD} = 13.5V, CV_{BOOT} = 4.7 µF, CV_{REG} = 4.7 µF, C_{CP} = 220 nF.

Parameters	Symbol	Min.	Typ.	Max.	Units	Conditions
Analog Interface						
Analog Low-Voltage Input	$ANALOG_{VIN}$	0	—	5.5	V	Excludes high-voltage pins (1)
Analog Low-Voltage Output	$ANALOG_{VOUT}$	0	—	V_{REG}	V	Excludes high-voltage pins (1)
WAKE Input						
Input Voltage	$WAKE_{I/O}$	0	—	HV_{DD}	V	
Input Rising Threshold	$V_{WAKE\_HI\_TH}$	—	—	1.26	V	(Note 1)
Input Falling Threshold	$V_{WAKE\_LO\_TH}$	0.54	—	—	V	
Input Current	I_{WAKE}	—	0.2	—	µA	V_{WAKE} = 0.0V (1)
		—	70	—		V_{WAKE} = 3.3V (1)
		—	106	—		V_{WAKE} = 5.0V (1)
		—	596	—		V_{WAKE} = 28V (1)
Input Pull-Down Resistance	R_{WAKE_PULLDN}	—	51	—	kΩ	
Wake-up Signal Setup Time	t_{WAIT_SETUP}	150	—	—	µs	Minimum time WAKE pin must be logic low before rising edge of wake-up pulse
DE2 Communications						
Baud Rate	BAUD	9030	9600	10170	bps	Half-duplex
Power-up Delay	PU_DELAY	—	6	10	ms	Time from rising HV_{DD} ≥ 6V to DE2 starts sending POR message, C_{VREG} = 1 µF (1)
DE2 Sink Current	I_{DE2_SINK}	1	—	—	mA	V_{DE2} ≤ 50 mV (1)
DE2 Message Response Time	t_{DE2_RSP}	0	—	1	ms	Time from last received Stop bit to response Start bit
DE2 Host Wait Time	t_{DE2_WAIT}	2.8	—	—	ms	Minimum time for host to wait for response; three packets based on 9600 Baud
DE2 Message Receive Time-out	$DE2_{RCVTOUT}$	—	—	1.45	ms	Time after Start bit received to NACK for no Stop bit
Auto-Baud Detection Window (Break)	$ABAUD_{DET}$	1.29	—	2.00	ms	Window for valid detection of continuous logic low on DE2 link
Auto-Baud Response Delay	$ABAUD_{DLY}$	—	1.00	—	ms	Delay from $ABAUD_{DET}$ to start of sending 0x55 byte
Auto-Baud Complete Delay	$ABAUD_{COMP}$	—	2.00	—	ms	Delay after sending 0x55 byte before exiting auto-baud function
Delay Between Bytes of Multibyte Message from Host	$t_{DE2\_HOST\_MULTI\_DLY}$	—	—	1.3	ms	Delay between message bytes arriving from host

Note 1: Limits based on design, simulation or characterization. Not production tested.

36.0 LIN TRANSCEIVER ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(1)

Input Voltage, LIN_VDD ....(LIN_VSS – 0.3V) to +40.0V
Logic Pins (LIN_RXD, LIN_TXD, LIN_EN) Voltage Levels ....-0.3V to +5.5V
Logic Output DC Currents...-5 mA to +5 mA
LIN_BUS DC Voltage .... -27 to +40V
LIN_BUS DC Voltage Transient .... -27 to +43V
LIN_BUS DC Current 43V
LIN_INH DC Voltage .... -0.3V to (LIN_VDD + 0.3V)
LIN_INH DC Current ....-100 mA to +30 mA
LIN_WKIN DC Voltage ....-0.3V to +40V

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 36-1: ELECTRICAL CHARACTERISTICS

Operating Conditions (unless otherwise stated): 5V < V_S < 28V, -40°C < T_J < +150°C; all values refer to LIN_VSS pins								
Parameters	**Test Conditions**	**Pin**	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Type**
LIN_VDD Pin								
Nominal DC Voltage Range		LIN_VDD	V_S	5	13.5	28	V	A
Supply Current in Sleep Mode	Sleep mode, V_{LIN} > V_S – 0.5V, V_S < 14V, T = +27°C	LIN_VDD	$I_{VSsleep}$	3	9	15	µA	B
	Sleep mode, V_{LIN} > V_S – 0.5V, V_S < 14V	LIN_VDD	$I_{VSsleep}$	3	11	18	µA	A
	Sleep mode, V_{LIN} = 0V, bus shorted to LIN_VSS, V_S < 14V	LIN_VDD	$I_{VSsleep\_short}$	20	50	100	µA	A
Supply Current in Normal Mode	Bus recessive, V_S < 14V	LIN_VDD	I_{VSrec}	150	250	320	µA	A
Supply Current in Normal Mode	Bus dominant (internal LIN pull-up resistor active), V_S < 14V	LIN_VDD	I_{VSdom}	200	700	950	µA	A
Supply Current in Fail-Safe Mode	Bus recessive, V_S < 14V	LIN_VDD	I_{VSfail}	40	80	110	µA	A
LIN_VDD Undervoltage Threshold (switching from Normal to Fail-Safe mode)	Decreasing supply voltage	LIN_VDD	$V_{VS\_th\_N\_F\_down}$	3.9	4.3	4.7	V	A
	Increasing supply voltage	LIN_VDD	$V_{VS\_th\_N\_F\_up}$	4.1	4.6	4.9	V	A
LIN_VDD Undervoltage Hysteresis		LIN_VDD	$V_{VS\_hys\_F\_N}$	0.1	0.25	0.4	V	A
LIN_VDD Operation Threshold (switching to Unpowered mode)	Switch to Unpowered mode	LIN_VDD	$V_{VS\_th\_U\_down}$	1.9	2.05	2.3	V	A
	Switch from Unpowered mode to Fail-Safe mode	LIN_VDD	$V_{VS\_th\_U\_F\_up}$	2.0	2.25	2.4	V	A
LIN_VDD Undervoltage Hysteresis		LIN_VDD	$V_{VS\_hys\_U}$	0.1	0.2	0.3	V	A
LIN_RXD Output Pin (open-drain)								
Low-Level Output Sink Capability	Normal mode, V_{LIN} = 0V, I_{RXD} = 2 mA	LIN_RXD	V_{RXDL}	—	0.2	0.4	V	A
High-Level Leakage Current	Normal mode, V_{LIN} = V_S, V_{RXD} = 5V	LIN_RXD	I_{RXDH}	-3	—	+3	µA	A
LIN_TXD Input/Output Pin								
Low-Level Voltage Input		LIN_TXD	V_{TXDL}	-0.3	—	+0.8	V	A
High-Level Voltage Input		LIN_TXD	V_{TXDH}	2	—	5.5	V	A
Pull-Down Resistor	V_{TXD} = 5V	LIN_TXD	R_{TXD}	150	200	300	kΩ	A
Low-Level Leakage Current	V_{TXD} = 0V	LIN_TXD	I_{TXD}	-3	—	+3	µA	A
Low-Level Output Sink Current at Wake-up Request	Fail-Safe mode, V_{TXD} = 0.4V	LIN_RXD	I_{TXD}	2	2.5	8	mA	A

Legend: Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design Parameter

TABLE 36-1: ELECTRICAL CHARACTERISTICS (CONTINUED)

Operating Conditions (unless otherwise stated): 5V < V_S < 28V, -40°C < T_J < +150°C; all values refer to LIN_VSS pins								
Parameters	**Test Conditions**	**Pin**	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Type**
LIN_EN Input Pin								
Low-Level Voltage Input		LIN_EN	V_{ENL}	-0/3	—	+0.8	V	A
High-Level Voltage Input		LIN_EN	V_{ENH}	2	—	5.5	V	A
Pull-Down Resistor	V_{EN} = 5V	LIN_EN	R_{EN}	50	125	200	kΩ	A
Low-Level Input Current	V_{EN} = 0V	LIN_EN	I_{EN}	-3	—	+3	μA	A
LIN_WKIN Input Pin								
High-Level Input Voltage		LIN_WKIN	V_{WKinH}	V_S – 1V	—	V_S + 0.3V	V	A
Low-Level Input Voltage	Initializes a wake-up signal	LIN_WKIN	V_{WKinL}	-1	—	V_S – 3.3V	V	A
LIN_WKIN Pull-up Current	V_S < 28V, V_{WKin} = 0V	LIN_WKIN	I_{WKin}	-30	-10	—	μA	A
High-Level Leakage Current	V_S = 28V, V_{WKin} = 28V	LIN_WKIN	I_{WKinL}	-5	—	+5	μA	A
Debounce Time of Low Pulse for Wake-up via LIN_WKIN	V_{WKin} = 0V	LIN_WKIN	t_{WKin}	50	100	150	μs	A
LIN_INH Output Pin								
Switch-on Resistance Between LIN_VDD and LIN_INH	Normal or Fail-Safe mode, I_{INH} = -15 mA	LIN_INH	$R_{DSon,INH}$	—	12	25	Ω	A
Leakage Current	Transceiver in Sleep mode, V_{INH} = 0V/28V, V_S = 28V	LIN_INH	$I_{leak,INH}$	-3	—	+3	μA	A
High-Level Voltage	Normal or Fail-Safe mode, I_{INH} = -15 mA	LIN_INH	V_{INH}	V_S – 0.375	—	V_S	V	A
LIN Bus Driver: Bus Load Conditions: Load 1 (small): 1 nF, 1 kΩ; Load 2 (large): 10 nF, 500Ω; External Pull-up, R_{RXD} = 4.7 kΩ; C_{RXD} = 20 pF; Load 3 (medium): 6.8 nF, 660Ω characterized on samples. Duty Cycle 1 and Duty Cycle 2 specify the timing parameters for proper operation at 20 kb/s and Duty Cycle 3 and Duty Cycle 4 at 10.4 kb/s.								
Driver Recessive Output Voltage	Load1/Load2	LIN_BUS	V_{BUSrec}	0.9 x V_S	—	V_S	V	A
Driver Dominant Voltage	V_{VS} = 7V, R_{load} = 500Ω	LIN_BUS	$V_{\_LoSUP}$	—	—	1.2	V	A
Driver Dominant Voltage	V_{VS} = 18V, R_{load} = 500Ω	LIN_BUS	$V_{\_HiSUP}$	—	—	2	V	A
Driver Dominant Voltage	V_{VS} = 7V, R_{load} = 1000Ω	LIN_BUS	$V_{\_LoSUP\_1k}$	0.6	—	—	V	A
Driver Dominant Voltage	V_{VS} = 18V, R_{load} = 1000Ω	LIN_BUS	$V_{\_HiSUP\_1k}$	0.8	—	—	V	A
Pull-up Resistor to LIN_VDD	Serial diode is mandatory	LIN_BUS	R_{LIN}	20	30	47	kΩ	A
Voltage Drop at the Serial Diodes	In pull-up path with R_{slave}, $I_{SerDiode}$ = 10 mA	LIN_BUS	$V_{SerDiode}$	0.4	—	1.0	V	D
LIN_BUS Current Limitation	V_{BUS} = V_{Bat_max}	LIN_BUS	I_{BUS_LIM}	40	120	200	mA	A
Input Leakage Current at the Receiver Including Pull-up Resistor as Specified	Input leakage current driver off, V_{BUS} = 0V, V_{BAT} = 12V	LIN_BUS	$I_{BUS\_PAS\_dom}$	-1	-0.35	—	mA	A
Leakage Current LIN_BUS Recessive	Driver off, 8V < V_{BAT} < 18V, 8V < V_{BUS} < 18V, $V_{BUS} \geq V_{BAT}$	LIN_BUS	IBUS_PAS_rec	—	10	20	μA	A

Legend: Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design Parameter

TABLE 36-1: ELECTRICAL CHARACTERISTICS (CONTINUED)

Operating Conditions (unless otherwise stated): 5V < V_S < 28V, -40°C < T_J < +150°C; all values refer to LIN_VSS pins								
Parameters	**Test Conditions**	**Pin**	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Type**
Leakage Current when Control Unit Disconnected from Ground; Loss of Local Ground must not Affect Communication in the Residual Network	$GND_{Device} = V_S$, V_{BAT} = 12V, 0V < V_{BUS} < 18V	LIN_BUS	$I_{BUS\_NO\_gnd}$	-10	+0.5	+10	µA	A
Leakage Current at Disconnected Battery; Node has to Sustain the Current that can Flow Under this Condition; Bus Must Remain Operational Under this Condition	V_{BAT} disconnected, V_{SUP_Device} = LIN_VSS, 0V < V_{BUS} < 18V	LIN_BUS	$I_{BUS\_NO\_bat}$	—	0.1	2	µA	A
Capacitance on Pin LIN_BUS to LIN_VSS		LIN_BUS	C_{LIN}	—	—	20	pF	D
LIN Bus Receiver								
Center of Receiver Threshold	$V_{BUS\_CNT} = (V_{th\_dom} + V_{th\_rec})/2$	LIN_BUS	V_{BUS_CNT}	0.475 x V_S	0.5 x V_S	0.525 x V_S	V	A
Receiver Dominant State	V_{EN} = 5V	LIN_BUS	V_{BUSdom}	-27	—	0.4 x V_S	V	A
Receiver Recessive State	V_{EN} = 5V	LIN_BUS	V_{BUSrec}	0.6 x V_S	—	40	V	A
Receiver Input Hysteresis	$V_{hys} = V_{th\_rec} – V_{th\_dom}$	LIN_BUS	V_{BUShys}	0.028 x V_S	0.1 x V_S	0.175 x V_S	V	A
Pre-Wake Detection LIN_BUS High-Level Input Voltage		LIN_BUS	V_{LINH}	V_S – 2V	—	V_S + 0.3V	V	A
Pre-Wake Detection LIN_BUS Low-Level Input Voltage	Activates the LIN receiver	LIN_BUS	V_{LINL}	-27	—	V_S – 3.3V	V	A
Internal Timers								
Dominant Time for Wake-up via LIN_BUS	V_{LIN} = 0V	LIN_BUS	t_{bus}	50	100	150	µs	A
Time Delay for Mode Change from Fail-Safe into Normal Mode via LIN_EN Pin	V_{EN} = 5V	LIN_EN	t_{norm}	5	15	20	µs	A
Time Delay for Mode Change from Normal Mode to Sleep Mode via LIN_EN Pin	V_{EN} = 0V	LIN_EN	t_{sleep}	5	15	20	µs	A
Time Delay for Mode Change from Sleep Mode to Normal Mode via LIN_EN Pin	V_{EN} = 5V	LIN_EN	t_{s_norm}	—	150	300	µs	A
LIN_TXD Dominant Time-out Time	V_{TXD} = 0V	LIN_TXD	t_{dom}	20	40	60	ms	A
Duty Cycle 1	$TH_{Rec(max)} = 0.744 \times V_S$, $TH_{Dom(max)} = 0.581 \times V_S$, V_S = 7.0V to 18V, t_{Bit} = 50 µs, $D1 = t_{bus\_rec(min)}/(2 \times t_{Bit})$	LIN_BUS	D1	0.396	—	—	—	A

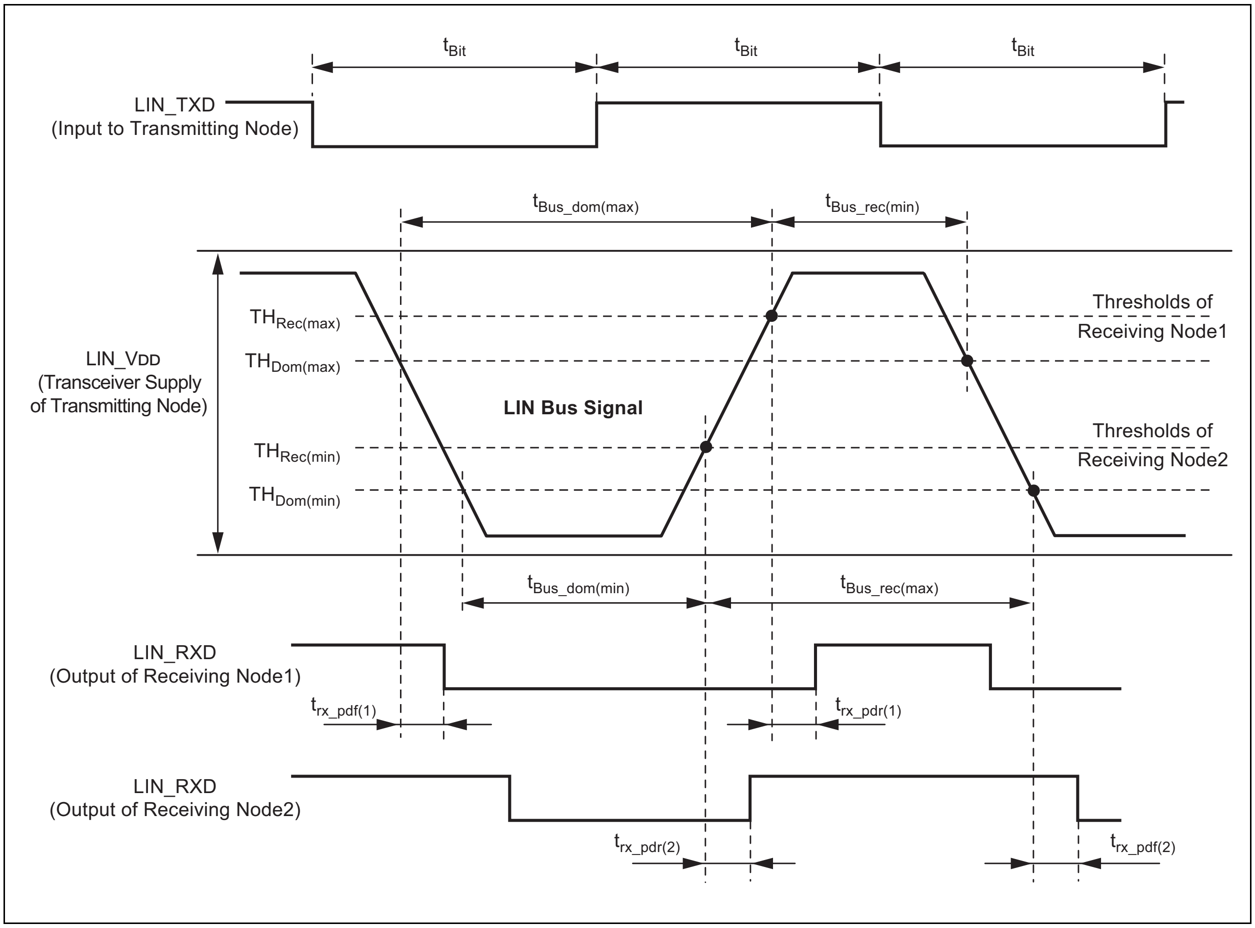
Legend: Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design Parameter

TABLE 36-1: ELECTRICAL CHARACTERISTICS (CONTINUED)

Operating Conditions (unless otherwise stated): 5V < V_S < 28V, -40°C < T_J < +150°C; all values refer to LIN_VSS pins								
Parameters	**Test Conditions**	**Pin**	**Symbol**	**Min.**	**Typ.**	**Max.**	**Unit**	**Type**
Duty Cycle 2	$TH_{Rec(min)}$ = 0.422 x V_S, $TH_{Dom(min)}$ = 0.284 x V_S, V_S = 7.6V to 18V, t_{Bit} = 50 µs, D2 = $t_{bus\_rec(max)}$/(2 x t_{Bit})	LIN_BUS	D2	—	—	0.581	—	A
Duty Cycle 3	$TH_{Rec(max)}$ = 0.778 x V_S, $TH_{Dom(max)}$ = 0.616 x V_S, V_S = 7.0V to 18V, t_{Bit} = 96 µs, D3 = $t_{bus\_rec(min)}$/(2 x t_{Bit})	LIN_BUS	D3	0.417	—	—	—	A
Duty Cycle 4	$TH_{Rec(max)}$ = 0.778 x V_S, $TH_{Dom(max)}$ = 0.616 x V_S, V_S = 7.0V to 18V, t_{Bit} = 96 µs, D3 = $t_{bus\_rec(min)}$/(2 x t_{Bit})	LIN_BUS	D4	—	—	0.590	—	A
Slope Time Falling and Rising Edge at LIN_BUS	V_S = 7.0V to 18V	LIN_BUS	t_{SLOPE_fall}, t_{SLOPE_rise}	3.5	—	22.5	µs	A
Receiver Electrical AC Parameters of the LIN Physical Layer; LIN Receiver, LIN_RXD Load Conditions: C_{RXD} = 20 pF, R_{RXD} = 4.7 kΩ								
Propagation Delay of Receiver	V_S = 7.0V to 18V, t_{rx_pd} = max(t_{rx_pdr}, t_{rx_pdf})	LIN_RXD	t_{rx_pd}	—	—	6	µs	A
Symmetry of Receiver Propagation Delay Rising Edge Minus Falling Edge	V_S = 7.0V to 18V, t_{rx_sym} = t_{rx_pdr} – t_{rx_pdf}	LIN_RXD	t_{rx_sym}	-2	—	+2	µs	A

Legend: Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design Parameter

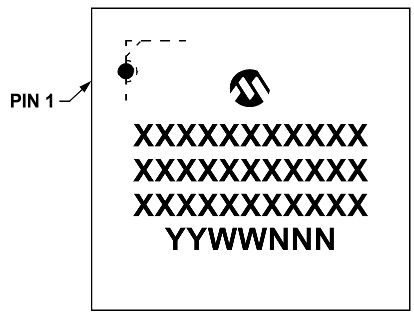
FIGURE 36-1: DEFINITION OF LIN BUS TIMING CHARACTERISTICS



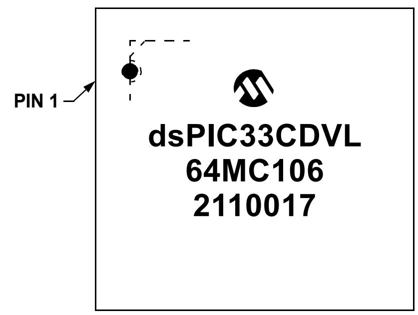
37.0 PACKAGING INFORMATION

37.1 Package Marking Information

64-Lead VGQFN (9x9 mm)



Example

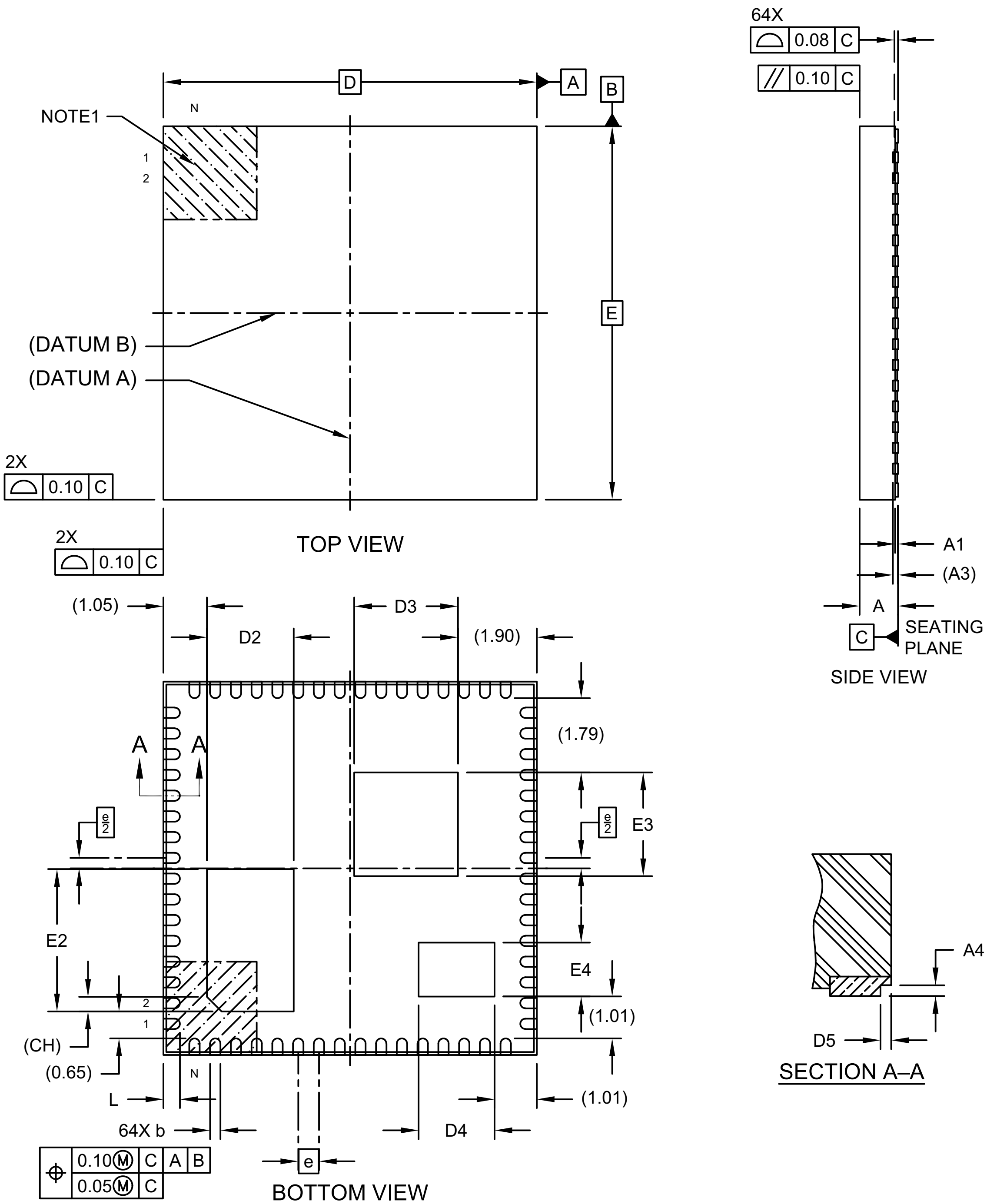


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
Note:		In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

37.2 Package Details

64-Lead Very Thin Grid Array Quad Flat Pack No-Lead (M8) - 9x9x0.927 mm Body [VGQFN] With Multiple Exposed Pads and Stepped Wettable Flanks

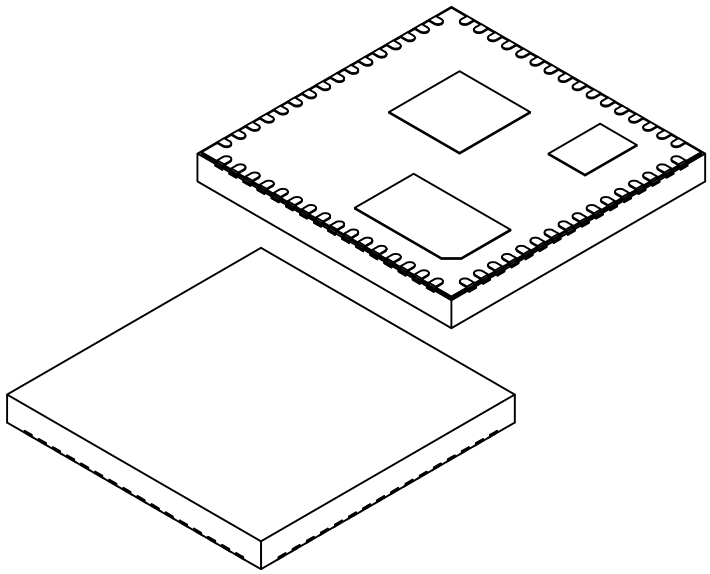
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-530-M8 Rev C Sheet 1 of 2

64-Lead Very Thin Grid Array Quad Flat Pack No-Lead (M8) - 9x9x0.927 mm Body [VGQFN] With Multiple Exposed Pads and Stepped Wettable Flanks

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Terminals	N		64	
Pitch	e		0.50 BSC	
Overall Height	A	0.827	0.877	0.927
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.127 REF	
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	1.99	2.09	2.19
Exposed Pad Length	D3	2.40	2.50	2.60
Exposed Pad Length	D4	1.73	1.83	1.93
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	3.33	3.43	3.53
Exposed Pad Width	E3	2.40	2.50	2.60
Exposed Pad Width	E4	1.20	1.30	1.40
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Exposed Pad Corner Chamfer	CH		0.35 REF	
Wettable Flank Step Cut Length	D5	0.03	0.07	0.11
Wettable Flank Step Cut Height	A4	0.03	-	-

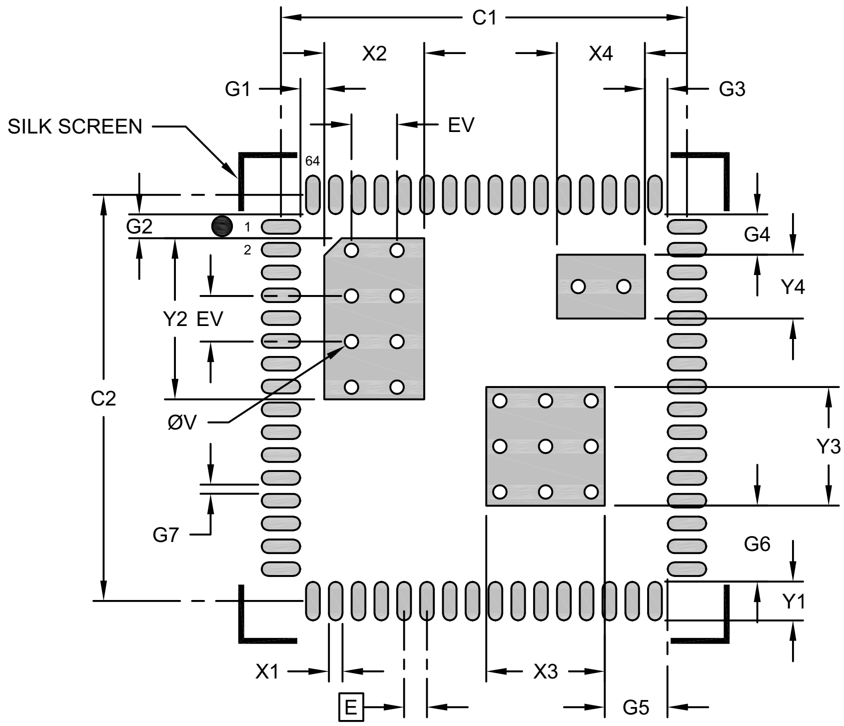
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-530-M8 Rev C Sheet 2 of 2

64-Lead Very Thin Grid Array Quad Flat Pack No-Lead (M8) - 9x9x0.927 mm Body [VGQFN] With Multiple Exposed Pads and Stepped Wettable Flanks

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Center Pad Width	X2			2.19
Center Pad Length	Y2			3.53
Center Pad Width	X3			2.60
Center Pad Length	Y3			2.60
Center Pad Width	X4			1.93
Center Pad Length	Y4			1.40
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (Xnn)	X1			0.30
Contact Pad Length (Xnn)	Y1			0.85
Contact Pad to Center Pad	G1	0.53		
Contact Pad to Center Pad	G2	0.53		
Contact Pad to Center Pad	G3	0.50		
Contact Pad to Center Pad	G4	0.86		
Contact Pad to Center Pad	G5	1.38		
Contact Pad to Center Pad	G6	1.67		
Contact Pad to Contact Pad	G7	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2530-M8 Rev C

APPENDIX A: REVISION HISTORY

Revision A (September 2020)

This is the initial version of the document.

Revision B (October 2021)

- Sections:
  - Updated **"Peripheral Features"**, **"Protection Features"**, **"Pin Diagrams"**, **Section 33.0 "Electrical Characteristics"** and **Section 35.0 "MOSFET Gate Driver Electrical Characteristics"**.
- Tables:
  - Updated Table 1, Table 1-1, Table 1-2, Table 34-1 and Table 34-2.
- Figures:
  - Updated Figure 1-2.
- Registers:
  - Updated Register 26-1 and Register 26-2.

Revision C (March 2023)

- Sections:
  - Updated **Section "Operating Conditions"**, **Section "High-Performance 16-Bit DSP RISC CPU"**, **Section "High-Resolution PWM"**, **Section "High-Speed Analog-to-Digital Converter"**, **Section "Microcontroller Features"**, **Section "Peripheral Features"**, **Section "Analog Features"**, **Section "Debug Features"**, **Section "Safety Features"**, **Section "Functional Safety Collaterals"**, **Section "Qualification"**, **Section 4.3 "BIST Overview"**, **Section 4.3.3 "Fault Simulation"**, **Section 4.5.1.1 "Extended X Data Space"**, **Section 5.4 "Error Correcting Code (ECC)"**, **Section 9.2 "CPU Clocking"**, **Section 9.4 "Internal Fast RC (FRC) Oscillator"**, **Section 11.0 "High-Resolution PWM with Fine Edge Placement"**, **Section 13.2 "Temperature Sensor"**, **Section 16.0 "Universal Asynchronous Receiver Transmitter (UART)"**, **Section 17.0 "LIN Transceiver Module"**, **Section 29.2.1 "Sleep Mode"**, **Section 29.2.2 "Idle Mode"**, **Section 35.0 "MOSFET Gate Driver Electrical Characteristics"**, **Section 36.0 "LIN Transceiver Electrical Characteristics"**, and **Section 37.2 "Package Details"**.
  - Added **"Terminology Cross Reference"**, **Section 9.5 "Low-Power RC (LPRC) Oscillator"**, **Section 13.4 "Differential-Mode"**.
- Tables:
  - Updated Table 4-11, Table 7-1, Table 7-2, Table 9-1, Table 24-3, Table 30-1, Table 33-30, Table 33-7, Table 33-34 and Table 36-1.
  - Added Table 33-32.
- Figures:
  - Updated Figure 7-2, Figure 9-1, Figure 9-2, Figure 9-5, Figure 13-1 and Figure 13-2.
- Registers:
  - Updated Register 5-1, Register 7-5, Register 9-1, Register 9-6, Register 11-21, Register 11-24, Register 11-26, Register 11-32, Register 13-6, Register 13-23, Register 14-7, Register 14-9, Register 15-2, Register 15-7, Register 15-10, Register 15-17, Register 23-3, Register 30-4, Register 30-6, Register 30-13 and Register 30-18.
  - Added Register 3-4

- Equations:
  - Updated Equation 20-2 and Equation 20-3.
- Examples:
  - Added Example 8-1

Revision D (January 2024)

- Sections:
  - Updated **Section 2.1 "Basic Connection Requirements"**, **Section 12.19.3 "Sleep Mode"**, **Section 27.0 "Deadman Timer (DMT)"** and **Section 36.0 "LIN Transceiver Electrical Characteristics"**.
  - Added **Section 2.6 "MOSFET Gate Driver Sleep Mode Requirements"** and **Section 9.3.1 "Primary Oscillator Pin Functionality"**.
- Tables:
  - Updated Table 1-2, and Table 7-2
- Figures:
  - Updated Figure 2-1 and Figure 13-1.
  - Added Figure 12-5.
- Registers:
  - Updated Register 7-5, Register 15-1, and Register 27-1.

NOTES:

INDEX

S

T

U

V

W

NOTES:

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

dsPIC 33C D V L 64 MC 106 T -I / M8 - XXX

- **Microchip Trademark** — dsPIC
- **Architecture** — 33C
- **MOSFET Gate Driver** — D
- **Voltage Regulator** — V
- **LIN Transceiver** — L
- **Program Memory Size (Kbyte)** — 64
- **Product Group** — MC
- **Pin Count** — 106
- **Tape and Reel Flag (if applicable)** — T
- **Temperature Range** — I
- **Package** — M8
- **Pattern** — XXX

Architecture:	33	=	16-Bit Digital Signal Controller
Product Group:	MC	=	Motor Control
Pin Count:	06	=	64-pin
Temperature Range:	I	=	-40°C to +85°C (Industrial)
	E	=	-40°C to +125°C (Extended)
	H	=	-40°C to +150°C (High)
Package:	M8	=	Very Thin Grid Array Quad Flat – (64-pin) 9x9 mm body (VGQFN)

Example:

dsPIC33CDVL64MC106-I/M8:
dsPIC33, 64-Kbyte Program Memory,
Motor Control, 64-Pin,
Industrial Temperature,
VGQFN Package.

NOTES:

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable" Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at https://www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-6683-3741-7

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820