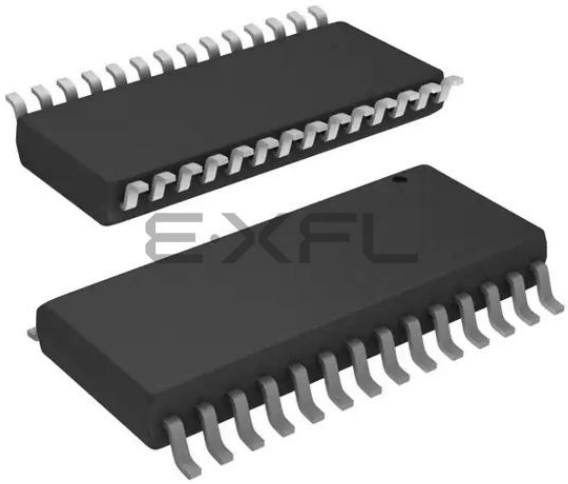


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	60 MIPs
Connectivity	CANbus, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, WDT
Number of I/O	21
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 11x10/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ev256gm102-e-so

TABLE 4-3: INPUT CAPTURE 1 THROUGH INPUT CAPTURE 4 REGISTER MAP

SFR Name	Addr.	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	All Resets
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144	Input Capture 1 Buffer Register																xxxx
IC1TMR	0146	Input Capture 1 Timer Register																0000
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	Input Capture 2 Buffer Register																xxxx
IC2TMR	014E	Input Capture 2 Timer Register																0000
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	Input Capture 3 Buffer Register																xxxx
IC3TMR	0156	Input Capture 3 Timer Register																0000
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC4BUF	015C	Input Capture 4 Buffer Register																xxxx
IC4TMR	015E	Input Capture 4 Timer Register																0000

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-4: I2C1 REGISTER MAP

SFR Name	Addr.	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	All Resets
I2C1CON1	0200	I2CEN	—	I2CSIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1CON2	0202	—	—	—	—	—	—	—	—	—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	1000
I2C1STAT	0204	ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C1ADD	0206	—	—	—	—	—	—	I2C1 Address Register										0000
I2C1MSK	0208	—	—	—	—	—	—	I2C1 Address Mask Register										0000
I2C1BRG	020A	Baud Rate Generator Register																0000
I2C1TRN	020C	—	—	—	—	—	—	—	—	I2C1 Transmit Register								00FF
I2C1RCV	020E	—	—	—	—	—	—	—	—	I2C1 Receive Register								0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-11: CAN1 REGISTER MAP WHEN WIN (C1CTRL<0>) = 1 FOR dsPIC33EVXXXGM10X DEVICES (CONTINUED)

SFR Name	Addr.	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	All Resets
C1RXF11SID	046C	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxxx
C1RXF11EID	046E	EID<15:0>																xxxx
C1RXF12SID	0470	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxxx
C1RXF12EID	0472	EID<15:0>																xxxx
C1RXF13SID	0474	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxxx
C1RXF13EID	0476	EID<15:0>																xxxx
C1RXF14SID	0478	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxxx
C1RXF14EID	047A	EID<15:0>																xxxx
C1RXF15SID	047C	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxxx
C1RXF15EID	047E	EID<15:0>																xxxx

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-12: SENT1 RECEIVER REGISTER MAP

SFR Name	Addr.	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	All Resets
SENT1CON1	0500	SNTEN	—	SNTSIDL	—	RCVEN	TXM	TXPOL	CRCEN	PPP	SPCEN	—	PS	—	NIBCNT2	NIBCNT1	NIBCNT0	0000
SENT1CON2	0504	TICKTIME<15:0> (Transmit modes) or SYNCMAX<15:0> (Receive mode)																FFFF
SENT1CON3	0508	FRAMETIME<15:0> (Transmit modes) or SYNCMIN<15:0> (Receive mode)																FFFF
SENT1STAT	050C	—	—	—	—	—	—	—	—	PAUSE	NIB2	NIB1	NIB0	CRCERR	FRMERR	RXIDLE	SYNCTXEN	0000
SENT1SYNC	0510	Synchronization Time Period Register (Transmit mode)																0000
SENT1DATL	0514	DATA4<3:0>				DATA5<3:0>				DATA6<3:0>				CRC<3:0>				0000
SENT1DATH	0516	STAT<3:0>				DATA1<3:0>				DATA2<3:0>				DATA3<3:0>				0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-13: SENT2 RECEIVER REGISTER MAP

SFR Name	Addr.	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	All Resets
SENT2CON1	0520	SNTEN	—	SNTSIDL	—	RCVEN	TXM	TXPOL	CRCEN	PPP	SPCEN	—	PS	—	NIBCNT2	NIBCNT1	NIBCNT0	0000
SENT2CON2	0524	TICKTIME<15:0> (Transmit modes) or SYNCMAX<15:0> (Receive mode)																FFFF
SENT2CON3	0528	FRAMETIME<15:0> (Transmit modes) or SYNCMIN<15:0> (Receive mode)																FFFF
SENT2STAT	052C	—	—	—	—	—	—	—	—	PAUSE	NIB2	NIB1	NIB0	CRCERR	FRMERR	RXIDLE	SYNCTXEN	0000
SENT2SYNC	0530	Synchronization Time Period Register (Transmit mode)																0000
SENT2DATL	0534	DATA4<3:0>				DATA5<3:0>				DATA6<3:0>				CRC<3:0>				0000
SENT2DATH	0536	STAT<3:0>				DATA1<3:0>				DATA2<3:0>				DATA3<3:0>				0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Allocating different Page registers for read and write access allows the architecture to support data movement between different pages in the data memory. This is accomplished by setting the DSRPAG register value to the page from which you want to read, and configure the DSWPAG register to the page to which it needs to be written. Data can also be moved from different PSV to EDS pages by configuring the DSRPAG and DSWPAG registers to address PSV and EDS space, respectively. The data can be moved between pages by a single instruction.

When an EDS or 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 EDS or PSV pages can occur at the page boundaries when:

- The initial address, prior to modification, addresses an EDS or a 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 DSxPAG register is incremented and the EA<15> bit is set to keep the base address within the EDS or PSV window. When an underflow is detected, the DSxPAG register is decremented and the EA<15> bit is set to keep the base address within the EDS or PSV window. This creates a linear EDS and PSV address space, but only when using the Register Indirect Addressing modes.

Exceptions to the operation described above arise when entering and exiting the boundaries of Page 0, EDS and PSV spaces. Table 4-43 lists the effects of overflow and underflow scenarios at different boundaries.

In the following cases, when an overflow or underflow occurs, the EA<15> bit is set and the DSxPAG 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 4-43: OVERFLOW AND UNDERFLOW SCENARIOS AT PAGE 0, EDS AND PSV SPACE BOUNDARIES(2,3,4)

O/U, R/W	Operation	Before			After		
		DSxPAG	DS EA<15>	Page Description	DSxPAG	DS EA<15>	Page Description
O, Read	[++Wn] or [Wn++]	DSRPAG = 0x1FF	1	EDS: Last Page	DSRPAG = 0x1FF	0	See **Note 1**
O, Read		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**
O, Write		DSWPAG = 0x1FF	1	EDS: Last Page	DSWPAG = 0x1FF	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 DSxPAG = 0x000 will generate an address error trap.

3: Only reads from PS are supported using DSRPAG. An attempt to write to PS using DSWPAG will generate an address error trap.

4: Pseudolinear Addressing is not supported for large offsets.

5.4 Error Correcting Code (ECC)

In order to improve program memory performance and durability, these devices include Error Correcting Code functionality (ECC) 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 is written to program memory, ECC generates a 7-bit Hamming code parity value for every two (24-bit) instruction words. The data is stored in blocks of 48 data bits and 7 parity bits; parity data is not memory-mapped and is inaccessible. When the data is read back, the ECC calculates the parity on it and compares it to the previously stored parity value. If a parity mismatch occurs, there are two possible outcomes:

- Single-bit errors are automatically identified and corrected on read-back. An optional device-level interrupt (ECCSBEIF) is also generated.
- Double-bit errors will generate a generic hard trap and the read data is not changed. If special exception handling for the trap is not implemented, a device Reset will also occur.

To use the single-bit error interrupt, set the ECC Single-Bit Error Interrupt Enable bit (ECCSBEIE) and configure the ECCSBEIP bits to set the appropriate interrupt priority.

Except for the single-bit error interrupt, error events are not captured or counted by hardware. This functionality can be implemented in the software application, but it is the user's responsibility to do so.

5.5 Flash Memory Resources

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

5.5.1 KEY RESOURCES

- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

5.6 Control Registers

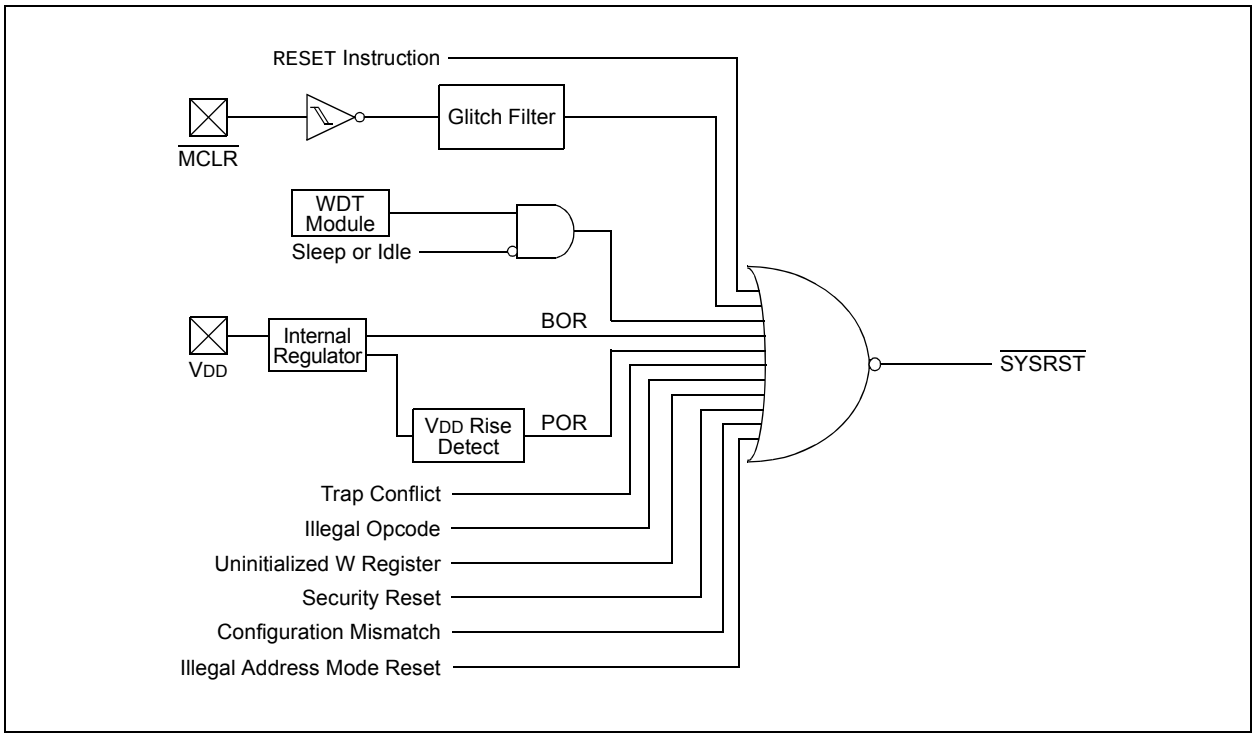
The following five SFRs are used to read and write the program Flash memory: NVMCON, NVMKEY, NVMADR, NVMADRU and NVMSRCADR.

The NVMCON register (Register 5-1) selects the operation to be performed (page erase, word/row program, inactive panel erase) and initiates the program/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 8 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 is written into data memory space (RAM) at an address defined by the NVMSRCADR register (location of the first element in row programming data).

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 6-1: RCON: RESET CONTROL REGISTER(1)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR	SWR	SWDTEN(2)	WDTO	SLEEP	IDLE	BOR	POR
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 **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 or an Illegal Address mode, or Uninitialized W register used as an Address Pointer caused a Reset
0 = An Illegal Opcode Reset or Uninitialized W Register Reset has not occurred

bit 13-12 **Unimplemented:** Read as '0'

bit 11 **VREGSF:** Flash Voltage Regulator Standby During Sleep bit
1 = Flash voltage regulator is active during Sleep mode
0 = Flash voltage regulator goes into Standby mode during Sleep mode

bit 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 ($\overline{MCLR}$) 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 **SWDTEN:** Software Enable/Disable of WDT bit(2)
1 = WDT is enabled
0 = WDT is disabled

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
1 = WDT time-out has occurred
0 = WDT time-out 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.

2: If the FWDTEN<1:0> Configuration bits are '11' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

REGISTER 13-2: TyCON (T3CON AND T5CON) CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(2)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1,3)	—
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 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Timery Stop in Idle Mode bit(2)
1 = Discontinues module operation when the device enters an Idle mode
0 = Continues module operation in an Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit(1,3)
1 = External clock is from pin, TyCK (on the rising edge)
0 = Internal clock (FP)

bit 0 **Unimplemented:** Read as '0'

Note 1: When 32-bit operation is enabled (T2CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through TxCON.

2: When 32-bit timer operation is enabled (T32 = 1) in the Timerx Control register (TxCON<3>), the TSIDL bit must be cleared to operate the 32-bit timer in Idle mode.

3: The TyCK pin is not available on all timers. See the **"Pin Diagrams"** section for the available pins.

REGISTER 14-7: DMTPSCNTL: DMT POST CONFIGURE COUNT 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
PSCNT<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
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 14-8: DMTPSCNTH: DMT POST CONFIGURE COUNT STATUS 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
PSCNT<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
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 17-5: CHOP: PWMx CHOP CLOCK GENERATOR REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CHPCLKEN	—	—	—	—	—	CHOPCLK9	CHOPCLK8
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
CHOPCLK7	CHOPCLK6	CHOPCLK5	CHOPCLK4	CHOPCLK3	CHOPCLK2	CHOPCLK1	CHOPCLK0
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 **CHPCLKEN:** Enable Chop Clock Generator bit
1 = Chop clock generator is enabled
0 = Chop clock generator is disabled

bit 14-10 **Unimplemented:** Read as '0'

bit 9-0 **CHOPCLK<9:0>:** Chop Clock Divider bits
The frequency of the chop clock signal is given by the following expression:
Chop Frequency = (FP/PCLKDIV<2:0>)/(CHOPCLK<9:0> + 1)

REGISTER 17-6: MDC: PWMx 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>							
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>							
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>:** PWMx Master Duty Cycle Value bits

REGISTER 17-16: LEBCONx: PWMx LEADING-EDGE BLANKING CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	—	—
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
—	—	BCH(1)	BCL(1)	BPHH	BPHL	BPLH	BPLL
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 **PHR:** PWMxH Rising Edge Trigger Enable bit
1 = Rising edge of PWMxH will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the rising edge of PWMxH

bit 14 **PHF:** PWMxH Falling Edge Trigger Enable bit
1 = Falling edge of PWMxH will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the falling edge of PWMxH

bit 13 **PLR:** PWMxL Rising Edge Trigger Enable bit
1 = Rising edge of PWMxL will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the rising edge of PWMxL

bit 12 **PLF:** PWMxL Falling Edge Trigger Enable bit
1 = Falling edge of PWMxL will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the falling edge of PWMxL

bit 11 **FLTLEBEN:** Fault Input Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to the selected Fault input
0 = Leading-Edge Blanking is not applied to the selected Fault input

bit 10 **CLLEBEN:** Current-Limit Input Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to the selected current-limit input
0 = Leading-Edge Blanking is not applied to the selected current-limit input

bit 9-6 **Unimplemented:** Read as '0'

bit 5 **BCH:** Blanking in Selected Blanking Signal High Enable bit(1)
1 = State blanking (of current-limit and/or Fault input signals) when selected blanking signal is high
0 = No blanking when the selected blanking signal is high

bit 4 **BCL:** Blanking in Selected Blanking Signal Low Enable bit(1)
1 = State blanking (of current-limit and/or Fault input signals) when selected blanking signal is low
0 = No blanking when the selected blanking signal is low

bit 3 **BPHH:** Blanking in PWMxH High Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxH output is high
0 = No blanking when the PWMxH output is high

bit 2 **BPHL:** Blanking in PWMxH Low Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxH output is low
0 = No blanking when the PWMxH output is low

bit 1 **BPLH:** Blanking in PWMxL High Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxL output is high
0 = No blanking when the PWMxL output is high

bit 0 **BPLL:** Blanking in PWMxL Low Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxL output is low
0 = No blanking when the PWMxL output is low

Note 1: The blanking signal is selected through the BLANKSEL<3:0> bits in the AUXCONx register.

18.1 SPI Helpful Tips

1. In Frame mode, if there is a possibility that the master may not be initialized before the slave:
   a) If FRMPOL (SPIxCON2<13>) = 1, use a pull-down resistor on $\overline{SSx}$.
   b) If FRMPOL = 0, use a pull-up resistor on $\overline{SSx}$.

Note:	This insures that the first frame transmission after initialization is not shifted or corrupted.

2. In Non-Framed 3-Wire mode (i.e., not using $\overline{SSx}$ from a master):
   a) If CKP (SPIxCON1<6>) = 1, always place a pull-up resistor on $\overline{SSx}$.
   b) If CKP = 0, always place a pull-down resistor on $\overline{SSx}$.

Note:	This will insure that during power-up and initialization, the master/slave will not lose sync due to an errant SCKx transition that would cause the slave to accumulate data shift errors, for both transmit and receive, appearing as corrupted data.

3. FRMEN (SPIxCON2<15>) = 1 and SSEN (SPIxCON1<7>) = 1 are exclusive and invalid. In Frame mode, SCKx is continuous and the Frame Sync pulse is active on the $\overline{SSx}$ pin, which indicates the start of a data frame.

Note:	Not all third-party devices support Frame mode timing. For more information, refer to the SPI specifications in **Section 30.0 "Electrical Characteristics"**.

4. In Master mode only, set the SMP bit (SPIxCON1<9>) to a '1' for the fastest SPI data rate possible. The SMP bit can only be set at the same time or after the MSTEN bit (SPIxCON1<5>) is set.

To avoid invalid slave read data to the master, the user's master software must ensure enough time for slave software to fill its write buffer before the user application initiates a master write/read cycle. It is always advisable to preload the SPIxBUF Transmit register in advance of the next master transaction cycle. SPIxBUF is transferred to the SPIx Shift register and is empty once the data transmission begins.

REGISTER 22-18: CxFMSKSEL1: CANx FILTERS 7-0 MASK SELECTION 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
F7MSK1	F7MSK0	F6MSK1	F6MSK0	F5MSK1	F5MSK0	F4MSK1	F4MSK0
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
F3MSK1	F3MSK0	F2MSK1	F2MSK0	F1MSK1	F1MSK0	F0MSK1	F0MSK0
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 **F7MSK<1:0>:** Mask Source for Filter 7 bit
11 = Reserved
10 = Acceptance Mask 2 registers contain the mask
01 = Acceptance Mask 1 registers contain the mask
00 = Acceptance Mask 0 registers contain the mask

bit 13-12 **F6MSK<1:0>:** Mask Source for Filter 6 bit (same values as bits 15-14)

bit 11-10 **F5MSK<1:0>:** Mask Source for Filter 5 bit (same values as bits 15-14)

bit 9-8 **F4MSK<1:0>:** Mask Source for Filter 4 bit (same values as bits 15-14)

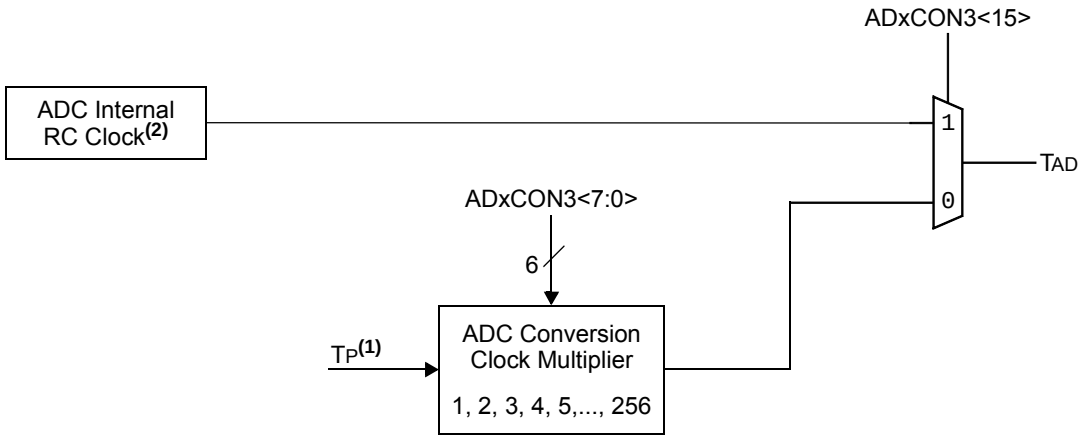
bit 7-6 **F3MSK<1:0>:** Mask Source for Filter 3 bit (same values as bits 15-14)

bit 5-4 **F2MSK<1:0>:** Mask Source for Filter 2 bit (same values as bits 15-14)

bit 3-2 **F1MSK<1:0>:** Mask Source for Filter 1 bit (same values as bits 15-14)

bit 1-0 **F0MSK<1:0>:** Mask Source for Filter 0 bit (same values as bits 15-14)

FIGURE 24-2: ADCx CONVERSION CLOCK PERIOD BLOCK DIAGRAM



Note 1: TP = 1/FP.

2: Refer to the ADC electrical specifications in **Section 30.0 "Electrical Characteristics"** for the exact RC clock value.

REGISTER 24-3: ADxCON3: ADCx CONTROL REGISTER 3

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC4(1)	SAMC3(1)	SAMC2(1)	SAMC1(1)	SAMC0(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
ADCS7(2)	ADCS6(2)	ADCS5(2)	ADCS4(2)	ADCS3(2)	ADCS2(2)	ADCS1(2)	ADCS0(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 **ADRC:** ADCx Conversion Clock Source bit
1 = ADCx internal RC clock
0 = Clock derived from system clock

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits(1)
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** ADCx Conversion Clock Select bits(2)
11111111 = TP • (ADCS<7:0> + 1) = TP • 256 = TAD
•
•
•
00000010 = TP • (ADCS<7:0> + 1) = TP • 3 = TAD
00000001 = TP • (ADCS<7:0> + 1) = TP • 2 = TAD
00000000 = TP • (ADCS<7:0> + 1) = TP • 1 = TAD

Note 1: These bits are only used if SSRC<2:0> (ADxCON1<7:5>) = 111 and SSRCG (ADxCON1<4>) = 0.
2: These bits are not used if ADRC (ADxCON3<15>) = 1.

TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
25	CTXTSWP	CTXTSWP #lit3	Switch CPU register context to context defined by lit3	1	2	None
		CTXTSWP Wn	Switch CPU register context to context defined by Wn	1	2	None
26	DAW	DAW Wn	Wn = decimal adjust Wn	1	1	C
27	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
28	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
29	DISI	DISI #lit14	Disable Interrupts for k instruction cycles	1	1	None
30	DIV	DIV.S Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N,Z,C,OV
		DIV.SD Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N,Z,C,OV
		DIV.U Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N,Z,C,OV
		DIV.UD Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N,Z,C,OV
31	DIVF	DIVF Wm,Wn	Signed 16/16-bit Fractional Divide	1	18	N,Z,C,OV
32	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
33	ED	ED Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance (no accumulate)	1	1	OA,OB,OAB, SA,SB,SAB
34	EDAC	EDAC Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance	1	1	OA,OB,OAB, SA,SB,SAB
35	EXCH	EXCH Wns,Wnd	Swap Wns with Wnd	1	1	None
36	FBCL	FBCL Ws,Wnd	Find Bit Change from Left (MSb) Side	1	1	C
37	FF1L	FF1L Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
38	FF1R	FF1R Ws,Wnd	Find First One from Right (LSb) Side	1	1	C
39	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
40	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
41	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
42	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
43	LAC	LAC Wso,#Slit4,Acc	Load Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
44	LNK	LNK #lit14	Link Frame Pointer	1	1	SFA
45	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

Note: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.

29.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

29.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

29.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

29.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

TABLE 30-4: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions (see Note 3): 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.(1)**	**Max.**	**Units**	**Conditions**
Operating Voltage							
DC10	VDD	**Supply Voltage(3)**	VBOR	—	5.5	V	
DC12	VDR	**RAM Data Retention Voltage(2)**	1.8	—	—	V	
DC16	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	—	—	VSS	V	
DC17	SVDD	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	1.0	—	—	V/ms	0V-5.0V in 5 ms
DC18	VCORE	**VDD Core** Internal Regulator Voltage	1.62	1.8	1.98	V	Voltage is dependent on load, temperature and VDD

Note 1: Data in "Typ." column is at 5.0V, +25°C unless otherwise stated.

2: This is the limit to which VDD may be lowered without losing RAM data.

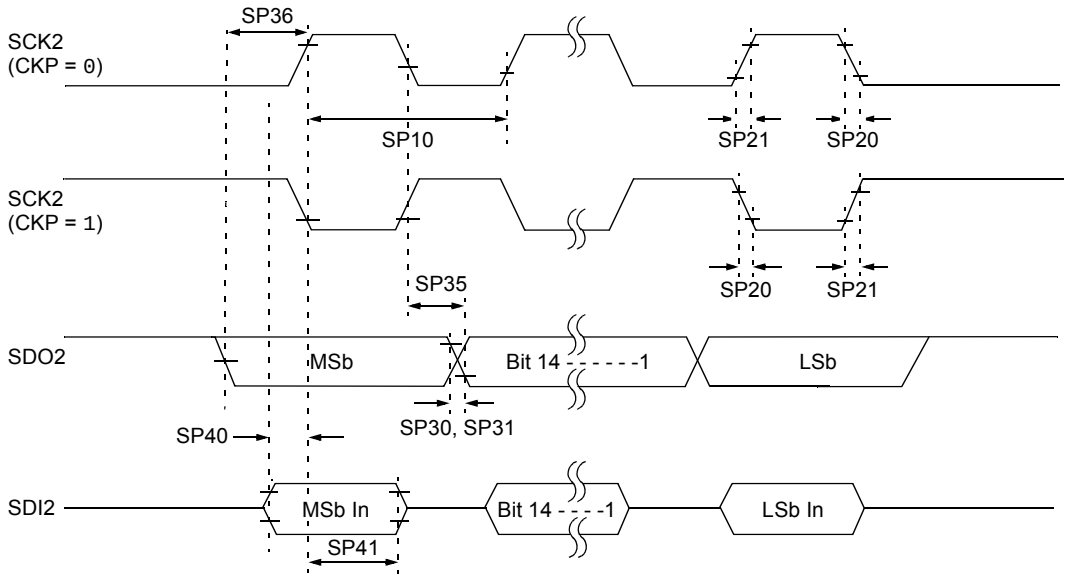
3: VDD voltage must remain at VSS for a minimum of 200 μs to ensure POR.

TABLE 30-5: FILTER CAPACITOR (CEFC) SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated): Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended							
Param No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
	CEFC	External Filter Capacitor Value(1)	4.7	10	—	μF	Capacitor must have a low series resistance (< 1Ω)

Note 1: Typical VCAP Voltage = 1.8 volts when VDD ≥ VDDMIN.

FIGURE 30-14: SPI2 MASTER MODE (FULL-DUPLEX, CKE = 1, CKP = x, SMP = 1) TIMING CHARACTERISTICS



Note: Refer to Figure 30-1 for load conditions.

TABLE 30-32: SPI2 MASTER MODE (FULL-DUPLEX, CKE = 1, CKP = x, SMP = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param.	**Symbol**	**Characteristic(1)**	**Min.**	**Typ.(2)**	**Max.**	**Units**	**Conditions**
SP10	FscP	Maximum SCK2 Frequency	—	—	9	MHz	See **Note 3**
SP20	TscF	SCK2 Output Fall Time	—	—	—	ns	See Parameter DO32 and **Note 4**
SP21	TscR	SCK2 Output Rise Time	—	—	—	ns	See Parameter DO31 and **Note 4**
SP30	TdoF	SDO2 Data Output Fall Time	—	—	—	ns	See Parameter DO32 and **Note 4**
SP31	TdoR	SDO2 Data Output Rise Time	—	—	—	ns	See Parameter DO31 and **Note 4**
SP35	TscH2doV, TscL2doV	SDO2 Data Output Valid after SCK2 Edge	—	6	20	ns	
SP36	TdoV2sc, TdoV2scL	SDO2 Data Output Setup to First SCK2 Edge	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDI2 Data Input to SCK2 Edge	30	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDI2 Data Input to SCK2 Edge	30	—	—	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ." column is at 5.0V, +25°C unless otherwise stated.

3: The minimum clock period for SCK2 is 111 ns. The clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPI2 pins.

TABLE 31-13: PLL CLOCK TIMING SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +150°C				
Param No.	Symbol	Characteristic	Min.	Typ.(1)	Max.	Units	Conditions
HOS50	FPLLI	PLL Voltage Controlled Oscillator (VCO) Input Frequency Range	0.8	—	8.0	MHz	ECPLL, XTPLL modes
HOS51	FSYS	On-Chip VCO System Frequency	120	—	340	MHz	
HOS52	TLOCK	PLL Start-up Time (Lock Time)	0.9	1.5	3.1	ms	
HOS53	DCLK	CLKO Stability (Jitter)(2)	-3	0.5	3	%	

Note 1: Data in "Typ." column is at 5.0V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: This jitter specification is based on clock cycle-by-clock cycle measurements. To get the effective jitter for individual time bases or communication clocks used by the application, use the following formula:

$$\text{Effective Jitter} = \frac{DCLK}{\sqrt{\frac{FOSC}{\text{Time Base or Communication Clock}}}}$$

For example, if FOSC = 120 MHz and the SPI bit rate = 10 MHz, the effective jitter is as follows:

$$\text{Effective Jitter} = \frac{DCLK}{\sqrt{\frac{120}{10}}} = \frac{DCLK}{\sqrt{12}} = \frac{DCLK}{3.464}$$

TABLE 31-14: INTERNAL FRC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +150°C					
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
Internal FRC Accuracy @ FRC Frequency = 7.3728 MHz							
HF20C	FRC	-3	1	+3	%	-40°C ≤ TA ≤ +150°C	VDD = 4.5V to 5.5V

TABLE 31-15: INTERNAL LPRC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 4.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +150°C					
Param No.	Characteristic	Min	Typ	Max	Units	Conditions	
LPRC @ 32.768 kHz(1,2)							
HF21C	LPRC	-30	10	+30	%	-40°C ≤ TA ≤ +150°C	VDD = 4.5V to 5.5V

Note 1: Change of LPRC frequency as VDD changes.

2: LPRC accuracy impacts the Watchdog Timer Time-out Period (TWDT1). See **Section 27.5 "Watchdog Timer (WDT)"** for more information.

NOTES: