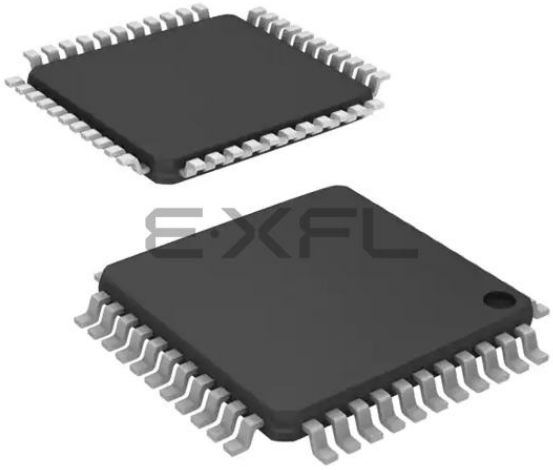
Microchip Technology - DSPIC33FJ16GS504-50I/PT Datasheet



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	50 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	35
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10b; D/A 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj16gs504-50i-pt

TABLE 4-7: INTERRUPT CONTROLLER REGISTER MAP FOR dsPIC33FJ06G202 DEVICES ONLY

File Name	SFR 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
INTCON1	0080	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	—	T2IF	—	—	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	AC1IF	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	AC2IF	—	—	—	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	ADCP6IF	—	—	—	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	—	T2IE	—	—	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	AC1IE	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	AC2IE	—	—	—	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	ADCP6IE	—	—	—	ADCP2IE	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPI1EIP2	SPI1EIP1	SPI1EIP0	—	—	—	—	4440
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP2	ADIP1	ADIP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	AC1IP2	AC1IP1	AC1IP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP2	INT2IP1	INT2IP0	—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP2	PSEMIP1	PSEMIP0	—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP2	U1EIP1	U1EIP0	—	—	—	—	0040
IPC23	00D2	—	PWM2IP2	PWM2IP1	PWM2IP0	—	PWM1IP2	PWM1IP1	PWM1IP0	—	—	—	—	—	—	—	—	4400
IPC25	00D6	—	AC2IP2	AC2IP1	AC2IP0	—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC27	00DA	—	ADCP1IP2	ADCP1IP1	ADCP1IP0	—	ADCP0IP2	ADCP0IP1	ADCP0IP0	—	—	—	—	—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IP2	ADCP2IP1	ADCP2IP0	0004
IPC29	00DE	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP6IP2	ADCP6IP1	ADCP6IP0	0004
INTTREG	00E0	—	—	—	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

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

4.6.2 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the lower word of any address within the program space without going through data space. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

The PC is incremented by two for each successive 24-bit program word. 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 and TBLWTL access the space that contains the least significant data word. TBLRDH and TBLWTH access the space that contains the upper data byte.

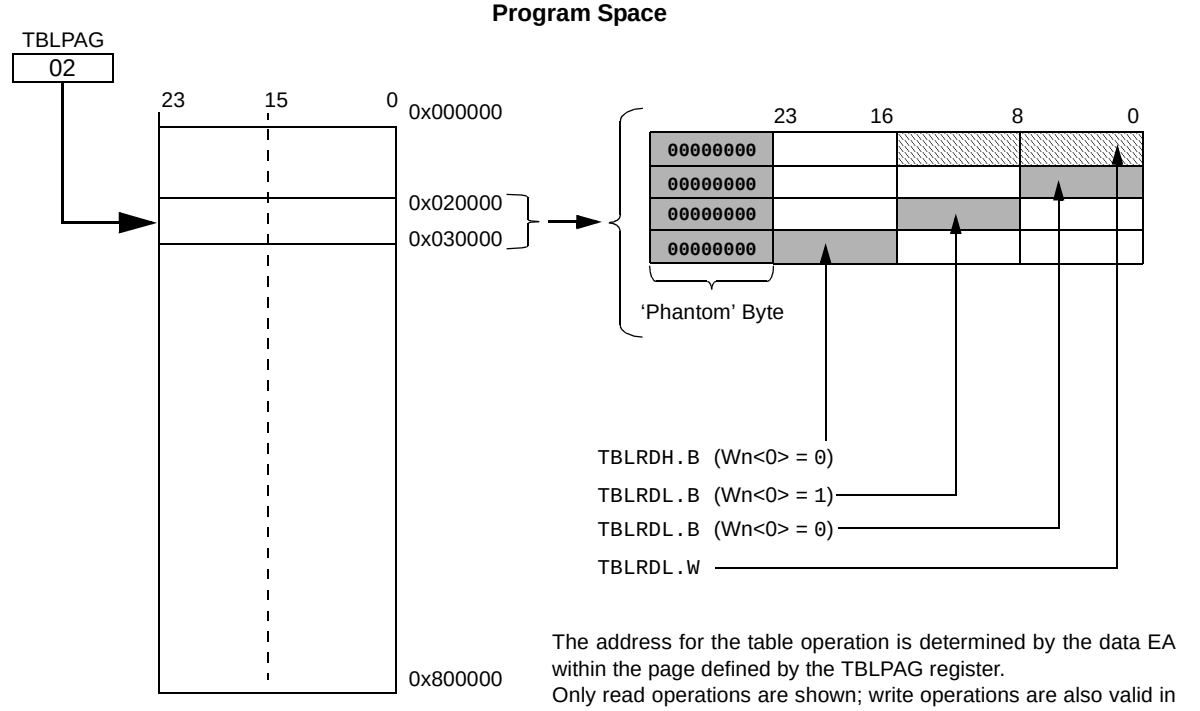
Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

- 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. Note that D<15:8>, the 'phantom byte', will always be '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 is always '0' when the upper 'phantom' byte is selected (Byte Select = 1).

Similarly, two table instructions, TBLWTH and TBLWTL, are used to write individual bytes or words to a program space address. The details of their operation are explained in **Section 5.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Page register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = 0, the table page is located in the user memory space. When TBLPAG<7> = 1, the page is located in configuration space.

FIGURE 4-10: 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.

5.2 RTSP Operation

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user application to erase a page of memory, which consists of eight rows (512 instructions) at a time, and to program one row or one word at a time. Table 24-12 shows typical erase and programming times. The 8-row erase pages and single row write rows are edge-aligned from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

The program memory implements holding buffers that can contain 64 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the buffers sequentially. The instruction words loaded must always be from a group of 64 boundary.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register. A total of 64 TBLWTL and TBLWTH instructions are required to load the instructions.

All of the Table Write operations are single-word writes (two instruction cycles) because only the buffers are written. A programming cycle is required for programming each row.

5.3 Programming Operations

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.

The programming time depends on the FRC accuracy (see Table 24-20) and the value of the FRC Oscillator Tuning register (see Register 8-4). Use the following formula to calculate the minimum and maximum values for the Row Write Time, Page Erase Time, and Word Write Cycle Time parameters (see Table 24-12).

EQUATION 5-1: PROGRAMMING TIME

$$\frac{T}{7.37\ MHz \times (FRC\ Accuracy)\% \times (FRC\ Tuning)\%}$$

For example, if the device is operating at +125°C, the FRC accuracy will be ±5%. If the TUN<5:0> bits (see Register 8-4) are set to 'b111111, the minimum row write time is equal to Equation 5-2.

EQUATION 5-2: MINIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 + 0.05) \times (1 - 0.00375)} = 1.435ms$$

The maximum row write time is equal to Equation 5-3.

EQUATION 5-3: MAXIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 - 0.05) \times (1 - 0.00375)} = 1.586ms$$

Setting the WR bit (NVMCON<15>) starts the operation, and the WR bit is automatically cleared when the operation is finished.

5.4 Control Registers

Two SFRs are used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and the start of the programming cycle.

NVMKEY 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. Refer to **Section 5.3 "Programming Operations"** for further details.

EXAMPLE 5-2: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV     #0x4001, W0               ;
       MOV     W0, NVMCON                ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0               ;
       MOV     W0, TBLPAG                ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0               ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2           ;
       MOV     #HIGH_BYTE_0, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2           ;
       MOV     #HIGH_BYTE_1, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2           ;
       MOV     #HIGH_BYTE_2, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV     #LOW_WORD_31, W2          ;
       MOV     #HIGH_BYTE_31, W3         ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
```

EXAMPLE 5-3: INITIATING A PROGRAMMING SEQUENCE

```
       DISI    #5                        ; Block all interrupts with priority <7
                                         ; for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                ; Write the 55 key
       MOV     #0xAA, W1                 ;
       MOV     W1, NVMKEY                ; Write the AA key
       BSET    NVMCON, #WR               ; Start the erase sequence
       NOP                               ; Insert two NOPs after the
       NOP                               ; erase command is asserted
```

6.1 System Reset

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 families of devices have two types of Reset:

- Cold Reset
- Warm Reset

A Cold Reset is the result of a Power-on Reset (POR) or a Brown-out Reset (BOR). On a Cold Reset, the FNOSCx Configuration bits in the FOSC Configuration register select the device clock source.

A Warm Reset is the result of all the other Reset sources, including the RESET instruction. On Warm Reset, the device will continue to operate from the current clock source as indicated by the Current Oscillator Selection (COSC<2:0>) bits in the Oscillator Control (OSCCON<14:12>) register.

The device is kept in a Reset state until the system power supplies have stabilized at appropriate levels and the oscillator clock is ready. The sequence in which this occurs is detailed in Figure 6-2.

TABLE 6-1: OSCILLATOR DELAY

Oscillator Mode	Oscillator Startup Delay	Oscillator Startup Timer	PLL Lock Time	Total Delay
FRC, FRCDIV16, FRCDIVN	T_{OSCD}(1)	—	—	T_{OSCD}(1)
FRCPLL	T_{OSCD}(1)	—	T_{LOCK}(3)	$T_{OSCD} + T_{LOCK}$(1,3)
XT	T_{OSCD}(1)	T_{OST}(2)	—	$T_{OSCD} + T_{OST}$(1,2)
HS	T_{OSCD}(1)	T_{OST}(2)	—	$T_{OSCD} + T_{OST}$(1,2)
EC	—	—	—	—
XTPLL	T_{OSCD}(1)	T_{OST}(2)	T_{LOCK}(3)	$T_{OSCD} + T_{OST} + T_{LOCK}$(1,2,3)
HSPLL	T_{OSCD}(1)	T_{OST}(2)	T_{LOCK}(3)	$T_{OSCD} + T_{OST} + T_{LOCK}$(1,2,3)
ECPLL	—	—	T_{LOCK}(3)	T_{LOCK}(3)
LPRC	T_{OSCD}(1)	—	—	T_{OSCD}(1)

Note 1: T_{OSCD} = Oscillator start-up delay (1.1 μs max for FRC, 70 μs max for LPRC). Crystal oscillator start-up times vary with crystal characteristics, load capacitance, etc.

2: T_{OST} = Oscillator Start-up Timer delay (1024 oscillator clock period). For example, T_{OST} = 102.4 μs for a 10 MHz crystal and T_{OST} = 32 ms for a 32 kHz crystal.

3: T_{LOCK} = PLL lock time (1.5 ms nominal) if PLL is enabled.

REGISTER 7-12: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0 (CONTINUED)

bit 1 **IC1IE:** Input Capture Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **INT0IE:** External Interrupt 0 Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-24: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

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-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
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 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 7-25: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT2IP2	INT2IP1	INT2IP0	—	—	—	—
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 6-4 **INT2IP<2:0>:** External Interrupt 2 Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

8.1.3 PLL CONFIGURATION

The primary oscillator and internal FRC oscillator can optionally use an on-chip PLL to obtain higher speeds of operation. The PLL provides significant flexibility in selecting the device operating speed. A block diagram of the PLL is shown in Figure 8-2.

The output of the primary oscillator or FRC, denoted as 'FIN', is divided down by a prescale factor (N1) of 2, 3, ... or 33 before being provided to the PLL's Voltage Controlled Oscillator (VCO). The input to the VCO must be selected in the range of 0.8 MHz to 8 MHz. The prescale factor 'N1' is selected using the PLLPRE<4:0> bits (CLKDIV<4:0>).

The PLL Feedback Divisor, selected using the PLLDIV<8:0> bits (PLLFBD<8:0>), provides a factor, 'M', by which the input to the VCO is multiplied. This factor must be selected such that the resulting VCO output frequency is in the range of 100 MHz to 200 MHz.

The VCO output is further divided by a postscale factor, 'N2'. This factor is selected using the PLLPOST<1:0> bits (CLKDIV<7:6>). 'N2' can be either 2, 4, or 8, and must be selected such that the PLL output frequency (FOSC) is in the range of 12.5 MHz to 80 MHz, which generates device operating speeds of 6.25-40 MIPS.

For a primary oscillator or FRC oscillator, output 'FIN', the PLL output 'FOSC' is given by Equation 8-2.

EQUATION 8-2: FOSC CALCULATION

$$FOSC = FIN * \left(\frac{M}{N1*N2}\right)$$

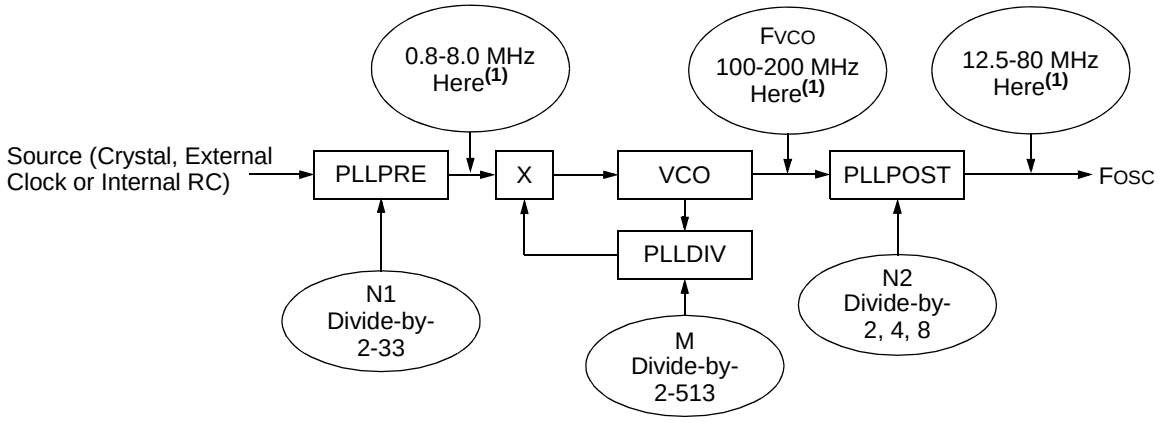
For example, suppose a 10 MHz crystal is being used with the selected oscillator mode of XT with PLL (see Equation 8-3).

- If PLLPRE<4:0> = 0, then N1 = 2. This yields a VCO input of 10/2 = 5 MHz, which is within the acceptable range of 0.8-8 MHz.
- If PLLDIV<8:0> = 0x1E, then M = 32. This yields a VCO output of 5 x 32 = 160 MHz, which is within the 100-200 MHz ranged needed.
- If PLLPOST<1:0> = 0, then N2 = 2. This provides a Fosc of 160/2 = 80 MHz. The resultant device operating speed is 80/2 = 40 MIPS.

EQUATION 8-3: XT WITH PLL MODE EXAMPLE

$$FCY = \frac{FOSC}{2} = \frac{1}{2}\left(\frac{10000000 * 32}{2 * 2}\right) = 40 \text{ MIPS}$$

FIGURE 8-2: dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 PLL BLOCK DIAGRAM



Note 1: This frequency range must be satisfied at all times.

8.4 Oscillator Control Registers

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER(1,2)

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC2	COSC1	COSC0	—	NOSC2(3)	NOSC1(3)	NOSC0(3)
bit 15							bit 8

R/W-0	R/W-0	R-0	U-0	R/C-0	U-0	U-0	R/W-0
CLKLOCK	IOLOCK	LOCK	—	CF	—	—	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
110 = Fast RC oscillator (FRC) with divide-by-16
101 = Low-Power RC oscillator (LPRC)
100 = Reserved
011 = Primary oscillator (XT, HS, EC) with PLL
010 = Primary oscillator (XT, HS, EC)
001 = Fast RC oscillator (FRC) with PLL
000 = Fast RC oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(3)
111 = Fast RC oscillator (FRC) with divide-by-n
110 = Fast RC oscillator (FRC) with divide-by-16
101 = Low-Power RC oscillator (LPRC)
100 = Reserved
011 = Primary oscillator (XT, HS, EC) with PLL
010 = Primary oscillator (XT, HS, EC)
001 = Fast RC oscillator (FRC) with PLL
000 = Fast RC oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
If Clock Switching is Enabled and FSCM is Disabled, (FOSC<FCKSM> = 0b01):
1 = Clock switching is disabled, system clock source is locked
0 = Clock switching is enabled, system clock source can be modified by clock switching

bit 6 **IOLOCK:** Peripheral Pin Select Lock bit
1 = Peripheral Pin Select is locked, write to Peripheral Pin Select registers not allowed
0 = Peripheral Pin Select is not locked, write to Peripheral Pin Select registers allowed

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. Refer to **"Oscillator (Part IV)"** (DS70307) in the *"dsPIC33F/PIC24H Family Reference Manual"* (available from the Microchip web site) for details.

2: This register is reset only on a Power-on Reset (POR).

3: 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 transition clock source between the two PLL modes.

REGISTER 9-2: PMD2: PERIPHERAL MODULE DISABLE CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	IC2MD	IC1MD
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	OC2MD	OC1MD
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 **IC2MD**: Input Capture 2 Module Disable bit
1 = Input Capture 2 module is disabled
0 = Input Capture 2 module is enabled

bit 8 **IC1MD**: Input Capture 1 Module Disable bit
1 = Input Capture 1 module is disabled
0 = Input Capture 1 module is enabled

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **OC2MD**: Output Compare 2 Module Disable bit
1 = Output Compare 2 module is disabled
0 = Output Compare 2 module is enabled

bit 0 **OC1MD**: Output Compare 1 Module Disable bit
1 = Output Compare 1 module is disabled
0 = Output Compare 1 module is enabled

REGISTER 10-14: RPINR34: PERIPHERAL PIN SELECT INPUT REGISTER 34

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-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SYNCI2R5	SYNCI2R4	SYNCI2R3	SYNCI2R2	SYNCI2R1	SYNCI2R0
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 **SYNCI2R<5:0>:** Assign PWM Master Time Base External Synchronization Signal to the Corresponding RPn Pin bits

111111 = Input tied to Vss
100011 = Input tied to RP35
100010 = Input tied to RP34
100001 = Input tied to RP33
100000 = Input tied to RP32
•
•
•
00000 = Input tied to RP0

REGISTER 10-15: 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
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
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
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
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 **RP1R<5:0>:** Peripheral Output Function is Assigned to RP1 Output Pin bits (see Table 10-2 for peripheral function numbers)

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

bit 5-0 **RP0R<5:0>:** Peripheral Output Function is Assigned to RP0 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 15-14: IOCONx: PWMx I/O CONTROL REGISTER (CONTINUED)

bit 3-2 **CLDAT<1:0>:** State for PWMxH and PWMxL Pins if CLMODE is Enabled bits(2)

FCLCONx<IFLTMOD> = 0: Normal Fault mode:
If current-limit is active, then CLDAT<1> provides the state for PWMxH
If current-limit is active, then CLDAT<0> provides the state for PWMxL

FCLCONx<IFLTMOD> = 1: Independent Fault mode:
CLDAT<1:0> bits are ignored.

bit 1 **SWAP<1:0>:** Swap PWMxH and PWMxL pins

1 = PWMxH output signal is connected to the PWMxL pin and the PWMxL signal is connected to the PWMxH pins
0 = PWMxH and PWMxL pins are mapped to their respective pins

bit 0 **OSYNC:** Output Override Synchronization bit

1 = Output overrides via the OVRDAT<1:0> bits are synchronized to the PWM time base
0 = Output overrides via the OVDDAT<1:0> bits occur on the next CPU clock boundary

Note 1: These bits should be changed only when PTEN = 0. Changing the clock selection during operation will yield unpredictable results.

2: The state represents the active/inactive state of the PWM module depending on the POLH and POLL bit settings.

17.2 I²C Registers

I2CxCON and I2CxSTAT are control and status registers. The I2CxCON register is readable and writable. The lower six bits of I2CxSTAT are read-only. The remaining bits of the I2CxSTAT are read/write:

- I2CxRSR is the shift register used for shifting data internal to the module and the user application has no access to it
- I2CxRCV is the receive buffer and the register to which data bytes are written, or from which data bytes are read
- I2CxTRN is the transmit register to which bytes are written during a transmit operation
- The I2CxADD register holds the slave address
- A status bit, ADD10, indicates 10-Bit Addressing mode
- The I2CxBRG acts as the Baud Rate Generator (BRG) reload value

In receive operations, I2CxRSR and I2CxRCV together form a double-buffered receiver. When I2CxRSR receives a complete byte, it is transferred to I2CxRCV, and an interrupt pulse is generated.

TABLE 22-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	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	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if Greater Than or Equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if Unsigned Greater Than or Equal	1	1 (2)	None
		BRA	GT,Expr	Branch if Greater Than	1	1 (2)	None
		BRA	GTU,Expr	Branch if Unsigned Greater Than	1	1 (2)	None
		BRA	LE,Expr	Branch if Less Than or Equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if Unsigned Less Than or Equal	1	1 (2)	None
		BRA	LT,Expr	Branch if Less Than	1	1 (2)	None
		BRA	LTU,Expr	Branch if Unsigned Less Than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	OA,Expr	Branch if Accumulator A Overflow	1	1 (2)	None
		BRA	OB,Expr	Branch if Accumulator B Overflow	1	1 (2)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
		BRA	SA,Expr	Branch if Accumulator A Saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if Accumulator B Saturated	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	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
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None

23.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® Digital Signal Controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

23.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

FIGURE 24-8: OCx/PWMx MODULE TIMING CHARACTERISTICS

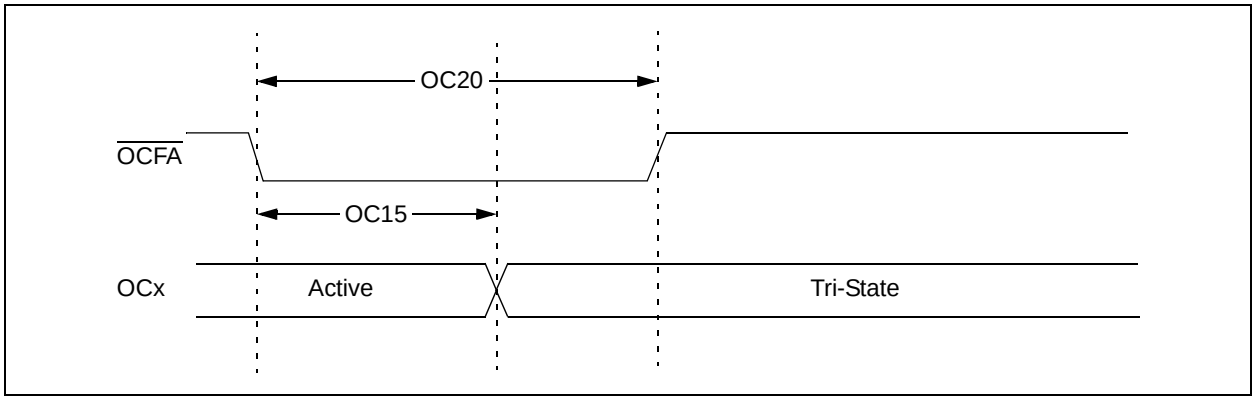
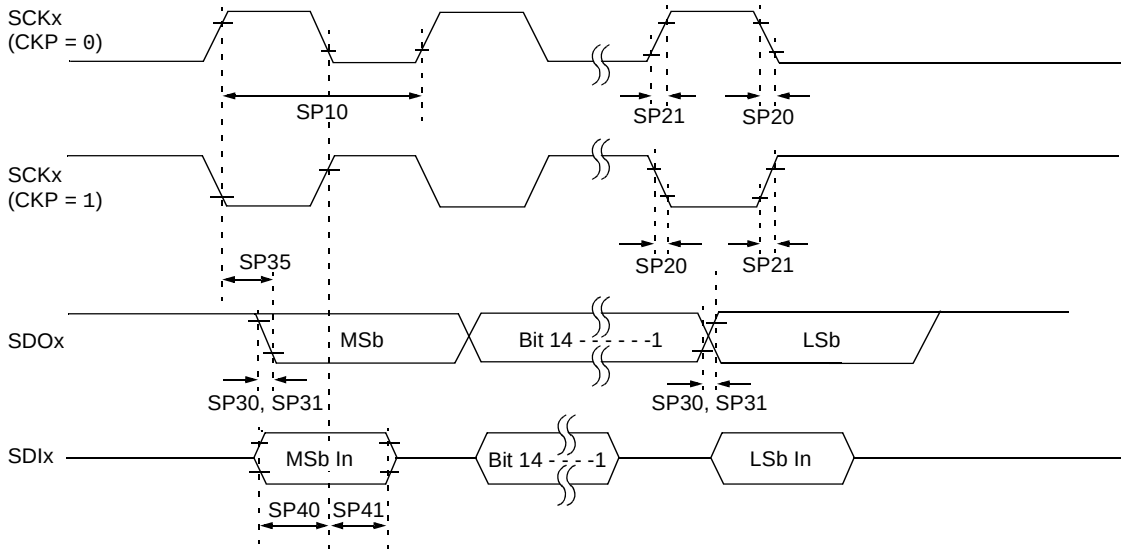


TABLE 24-28: SIMPLE OCx/PWMx MODE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic(1)**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
OC15	TFD	Fault Input to PWMx I/O Change	—	—	TCY + 20	ns	
OC20	TFLT	Fault Input Pulse Width	TCY + 20	—	—	ns	

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

FIGURE 24-14: SPIx MASTER MODE (FULL-DUPLEX, CKE = 0, CKP = x, SMP = 1) TIMING CHARACTERISTICS



Note: Refer to Figure 24-1 for load conditions.

TABLE 24-33: SPIx MASTER MODE (FULL-DUPLEX, CKE = 0, CKP = x, SMP = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -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**
SP10	TscP	Maximum SCKx Frequency	—	—	9	MHz	-40ºC to +125ºC and see **Note 3**
SP20	TscF	SCKx Output Fall Time	—	—	—	ns	See Parameter DO32 and **Note 4**
SP21	TscR	SCKx Output Rise Time	—	—	—	ns	See Parameter DO31 and **Note 4**
SP30	TdoF	SDOx Data Output Fall Time	—	—	—	ns	See Parameter DO32 and **Note 4**
SP31	TdoR	SDOx Data Output Rise Time	—	—	—	ns	See Parameter DO31 and **Note 4**
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	6	20	ns	
SP36	TdoV2scH, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	30	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	30	—	—	ns	

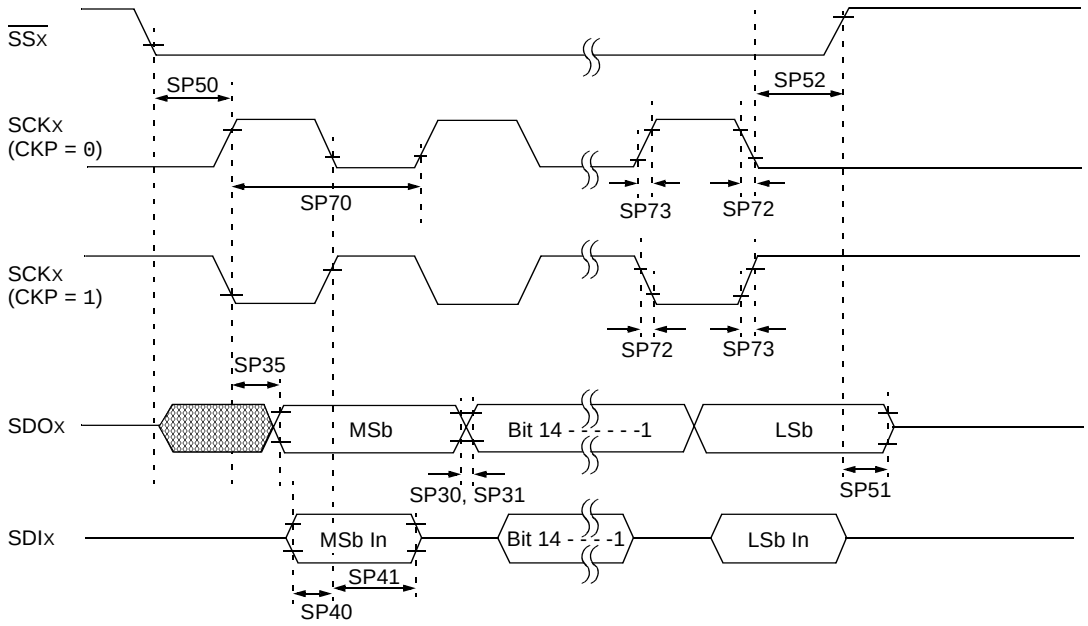
Note 1: These parameters are characterized, but are not tested in manufacturing.

2: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated.

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

4: Assumes 50 pF load on all SPIx pins.

FIGURE 24-17: SPIx SLAVE MODE (FULL-DUPLEX CKE = 0, CKP = 1, SMP = 0) TIMING CHARACTERISTICS



Note: Refer to Figure 24-1 for load conditions.

25.1 High-Temperature DC Characteristics

TABLE 25-1: OPERATING MIPS VS. VOLTAGE

Characteristic	VDD Range (in Volts)	Temperature Range (in °C)	Max MIPS
			dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04
—	3.0V to 3.6V(1)	-40°C to +150°C	20

Note 1: Overall functional device operation at VBORMIN < VDD < VDDMIN is tested but not characterized. All device analog modules such as the ADC, etc., will function but with degraded performance below VDDMIN. Refer to Parameter BO10 in Table 24-11 for BOR values.

TABLE 25-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
High-Temperature Devices					
Operating Junction Temperature Range	TJ	-40	—	+155	°C
Operating Ambient Temperature Range	TA	-40	—	+150	°C
Power Dissipation: Internal chip power dissipation: PINT = VDD x (IDD - Σ IOH) I/O Pin Power Dissipation: I/O = Σ ({VDD - VOH} x IOH) + Σ (VOL x IOL)	PD	PINT + PI/O			W
Maximum Allowed Power Dissipation	PDMAX	(TJ - TA)/θJA			W

TABLE 25-3: DC TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +150°C for High Temperature				
Parameter No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
Operating Voltage							
HDC10	**Supply Voltage**						
	VDD	—	3.0	3.3	3.6	V	-40°C to +150°C

TABLE 26-3: DC CHARACTERISTICS: IDLE CURRENT (I_{IDLE})

DC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Parameter No.	**Typical**	**Max**	**Units**	**Conditions**		
Idle Current (I_{IDLE}): Core Off, Clock On Base Current(1)						
MDC45d	64	105	mA	-40°C	3.3V	50 MIPS
MDC45a	64	105	mA	+25°C		
MDC45b	64	105	mA	+85°C		

Note 1: Base Idle current (I_{IDLE}) is measured as follows:

- CPU core is off, oscillator is configured in EC mode and external clock active, OSC1 is driven with external square wave from rail-to-rail (EC clock overshoot/undershoot < 250 mV required)
- CLKO is configured as an I/O input pin in the Configuration Word
- All I/O pins are configured as inputs and pulled to VSS
- $\overline{MCLR}$ = VDD, WDT and FSCM are disabled
- No peripheral modules are operating; however, every peripheral is being clocked (all PMDx bits are zeroed)
- The NVMSIDL bit (NVMCON<12>) = 1 (i.e., Flash regulator is set to standby while the device is in Idle mode)
- JTAG is disabled