

Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18c452-e-l">https://www.e-xfl.com/product-detail/microchip-technology/pic18c452-e-l</a>

# PIC18CXX2

---

NOTES:

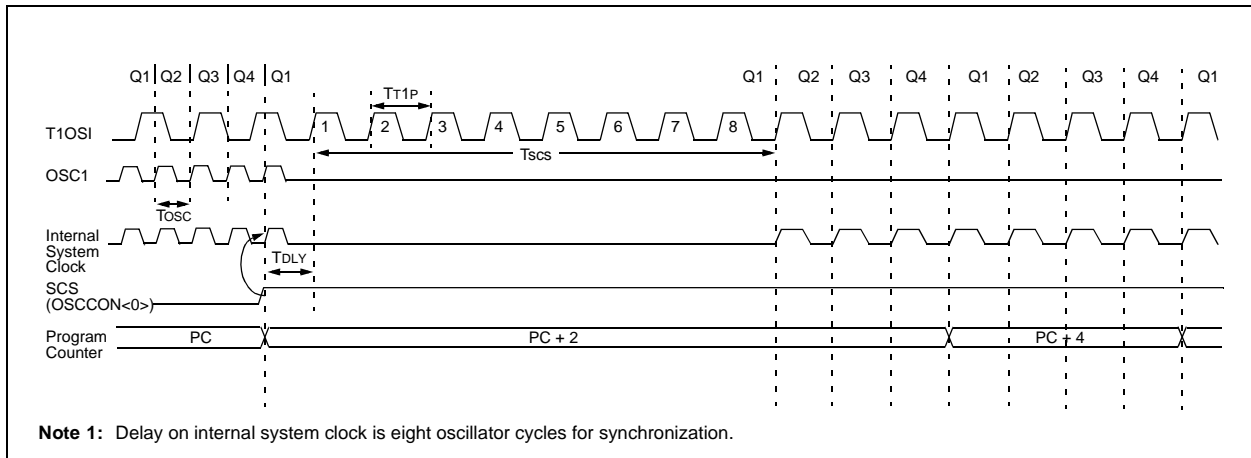


## 2.6.2 OSCILLATOR TRANSITIONS

The PIC18CXX2 devices contain circuitry to prevent "glitches" when switching between oscillator sources. Essentially, the circuitry waits for eight rising edges of the clock source that the processor is switching to. This ensures that the new clock source is stable and that its pulse width will not be less than the shortest pulse width of the two clock sources.

A timing diagram indicating the transition from the main oscillator to the Timer1 oscillator is shown in Figure 2-8. The Timer1 oscillator is assumed to be running all the time. After the SCS bit is set, the processor is frozen at the next occurring Q1 cycle. After eight synchronization cycles are counted from the Timer1 oscillator, operation resumes. No additional delays are required after the synchronization cycles.

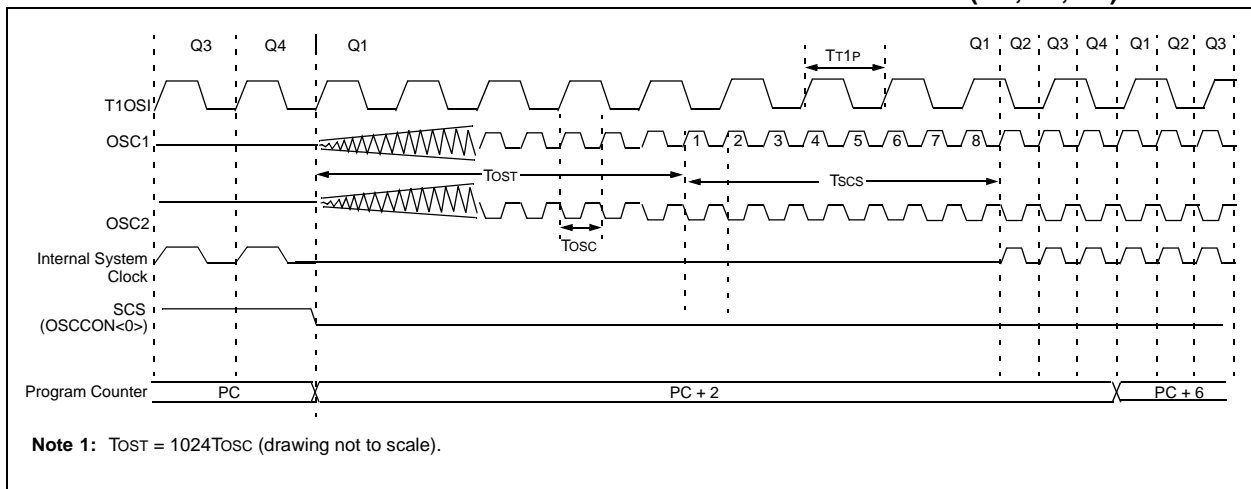
**FIGURE 2-8: TIMING DIAGRAM FOR TRANSITION FROM OSC1 TO TIMER1 OSCILLATOR**



The sequence of events that takes place when switching from the Timer1 oscillator to the main oscillator will depend on the mode of the main oscillator. In addition to eight clock cycles of the main oscillator, additional delays may take place.

If the main oscillator is configured for an external crystal (HS, XT, LP), then the transition will take place after an oscillator start-up time ( $T_{OST}$ ) has occurred. A timing diagram indicating the transition from the Timer1 oscillator to the main oscillator for HS, XT and LP modes is shown in Figure 2-9.

**FIGURE 2-9: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (HS, XT, LP)**

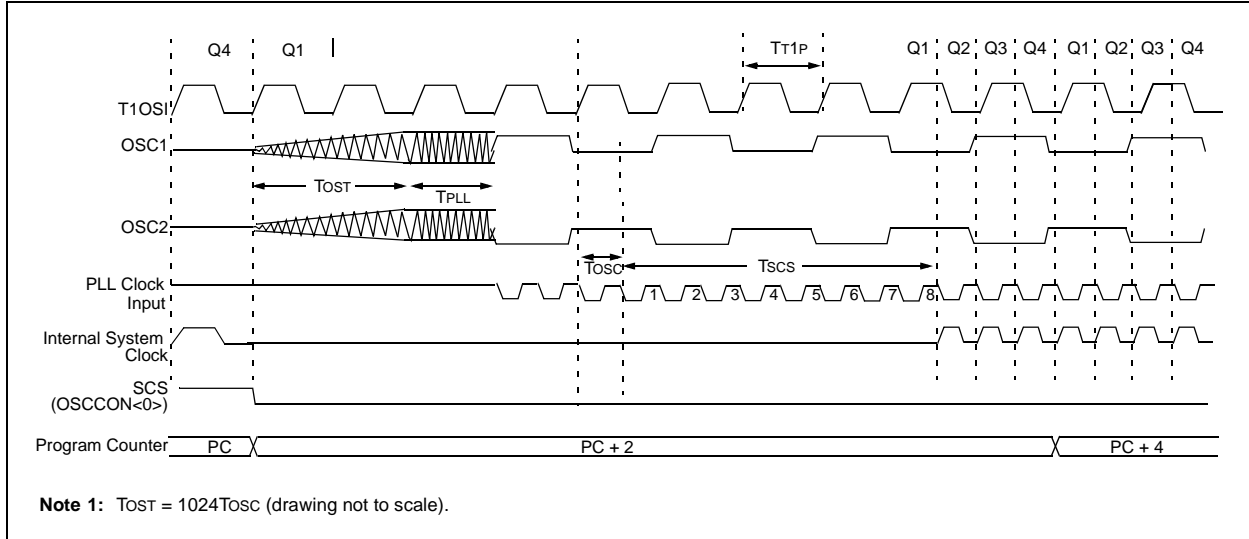


# PIC18CXX2

If the main oscillator is configured for HS-PLL mode, an oscillator start-up time ( $T_{OST}$ ) plus an additional PLL time-out ( $T_{PLL}$ ) will occur. The PLL time-out is typically 2 ms and allows the PLL to lock to the main oscillator

frequency. A timing diagram, indicating the transition from the Timer1 oscillator to the main oscillator for HS-PLL mode, is shown in Figure 2-10.

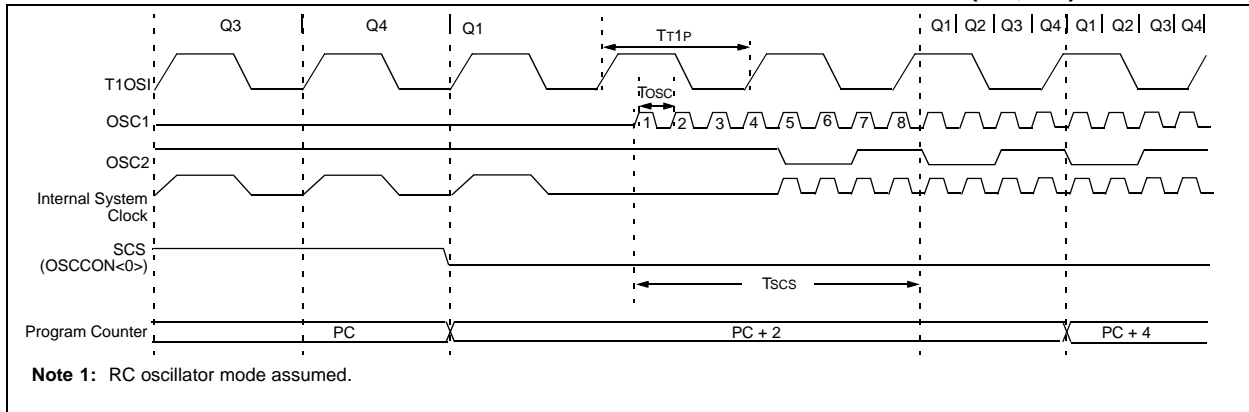
**FIGURE 2-10: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (HS WITH PLL)**



If the main oscillator is configured in the RC, RCIO, EC or ECIO modes, there is no oscillator start-up time-out. Operation will resume after eight cycles of the main oscillator have been counted. A timing diagram, indi-

cating the transition from the Timer1 oscillator to the main oscillator for RC, RCIO, EC and ECIO modes, is shown in Figure 2-11.

**FIGURE 2-11: TIMING FOR TRANSITION BETWEEN TIMER1 AND OSC1 (RC, EC)**



**TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

Register	Applicable Devices				Power-on Reset, Brown-out Reset	MCLR Resets	Wake-up via WDT or Interrupt
						WDT Reset RESET Instruction Stack Resets	
FSR1H	242	442	252	452	---- 0000	---- 0000	---- uuuu
FSR1L	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	242	442	252	452	---- 0000	---- 0000	---- uuuu
INDF2	242	442	252	452	N/A	N/A	N/A
POSTINC2	242	442	252	452	N/A	N/A	N/A
POSTDEC2	242	442	252	452	N/A	N/A	N/A
PREINC2	242	442	252	452	N/A	N/A	N/A
PLUSW2	242	442	252	452	N/A	N/A	N/A
FSR2H	242	442	252	452	---- 0000	---- 0000	---- uuuu
FSR2L	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	242	442	252	452	---x xxxx	---u uuuu	---u uuuu
TMR0H	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0L	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	242	442	252	452	1111 1111	1111 1111	uuuu uuuu
OSCCON	242	442	252	452	---- --0	---- --0	---- --u
LVDCON	242	442	252	452	--00 0101	--00 0101	--uu uuuu
WDTCON	242	442	252	452	---- --0	---- --0	---- --u
RCON <sup>(4, 6)</sup>	242	442	252	452	00-1 11q0	00-1 qquu	uu-u qquu
TMR1H	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	242	442	252	452	0-00 0000	u-uu uuuu	u-uu uuuu
TMR2	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR2	242	442	252	452	1111 1111	1111 1111	1111 1111
T2CON	242	442	252	452	-000 0000	-000 0000	-uuu uuuu
SSPBUF	242	442	252	452	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	242	442	252	452	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	242	442	252	452	0000 0000	0000 0000	uuuu uuuu
SSPCON1	242	442	252	452	0000 0000	0000 0000	uuuu uuuu
SSPCON2	242	442	252	452	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

**Note 1:** One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

**2:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

**3:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

**4:** See Table 3-2 for RESET value for specific condition.

**5:** Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

**6:** The long write enable is only reset on a POR or MCLR Reset.

**7:** Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they are read as '0'.

## 4.7.1 TWO-WORD INSTRUCTIONS

The PIC18CXX2 devices have four two-word instructions: `MOVFF`, `CALL`, `GOTO` and `LFSR`. The second word of these instructions has the 4 MSBs set to 1's and is a special kind of `NOP` instruction. The lower 12-bits of the second word contain data to be used by the instruction. If the first word of the instruction is executed, the data in the second word is accessed. If the

second word of the instruction is executed by itself (first word was skipped), it will execute as a `NOP`. This action is necessary when the two-word instruction is preceded by a conditional instruction that changes the PC. A program example that demonstrates this concept is shown in Example 4-3. Refer to Section 19.0 for further details of the instruction set.

### EXAMPLE 4-3: TWO-WORD INSTRUCTIONS

CASE 1:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; No, execute 2-word instruction
1111 0100 0101 0110	; 2nd operand holds address of REG2
0010 0100 0000 0000	ADDWF REG3 ; continue code
CASE 2:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; Yes
1111 0100 0101 0110	; 2nd operand becomes NOP
0010 0100 0000 0000	ADDWF REG3 ; continue code

## 4.8 Lookup Tables

Lookup tables are implemented two ways. These are:

- Computed `GOTO`
- Table Reads

### 4.8.1 COMPUTED GOTO

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`).

A lookup table can be formed with an `ADDWF PCL` instruction and a group of `RETLW 0xnn` instructions. `WREG` is loaded with an offset into the table, before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next instruction executed will be one of the `RETLW 0xnn` instructions that returns the value `0xnn` to the calling function.

The offset value (value in `WREG`) specifies the number of bytes that the program counter should advance.

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

### 4.8.2 TABLE READS/TABLE WRITES

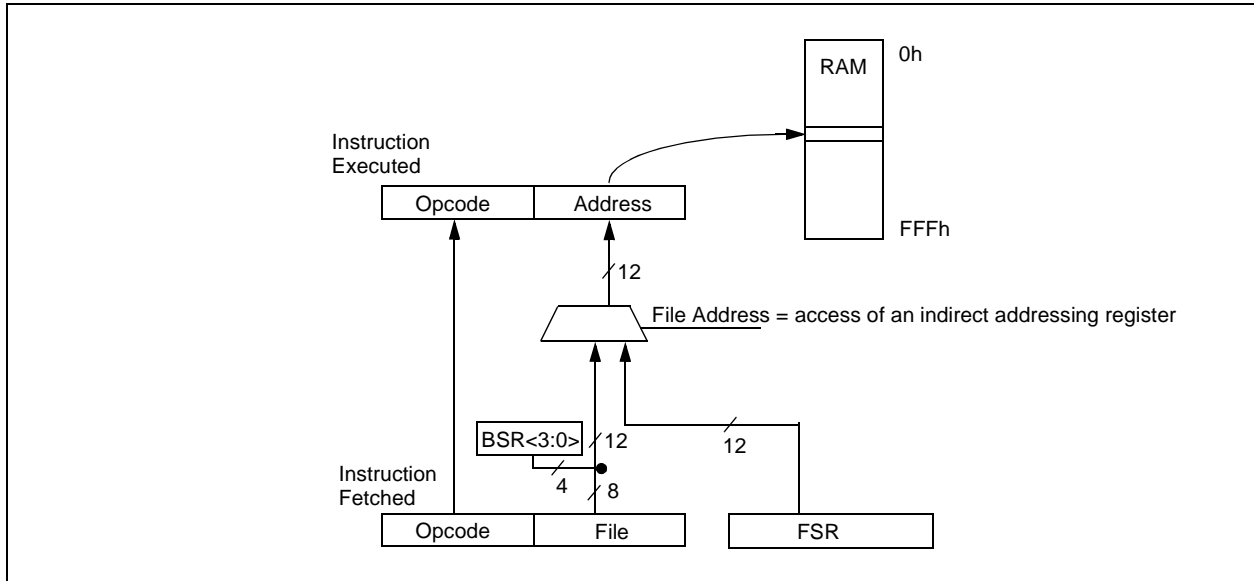
A better method of storing data in program memory allows 2 bytes of data to be stored in each instruction location.

Lookup table data may be stored 2 bytes per program word by using table reads and writes. The table pointer (`TBLPTR`) specifies the byte address and the table latch (`TABLAT`) contains the data that is read from, or written to program memory. Data is transferred to/from program memory one byte at a time.

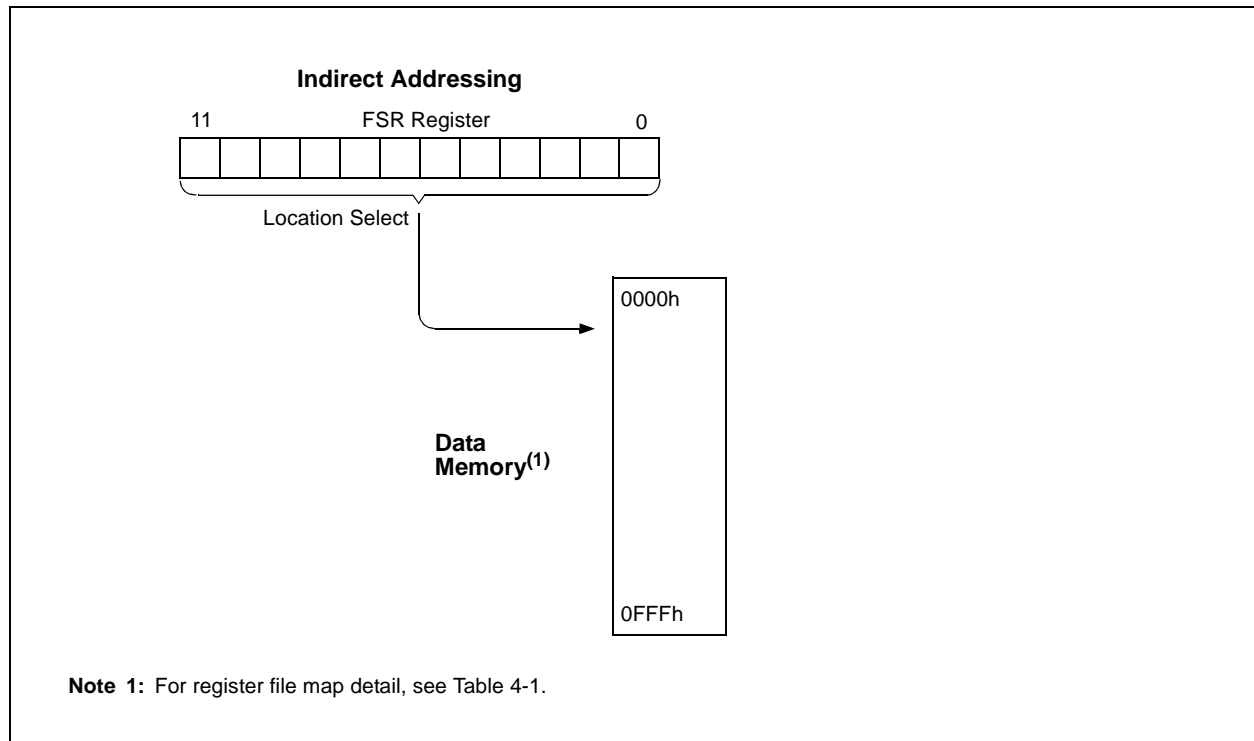
A description of the Table Read/Table Write operation is shown in Section 5.0.

If an indirect addressing operation is done where the target address is an FSRnH or FSRnL register, the write operation will dominate over the pre- or post-increment/decrement functions.

**FIGURE 4-9: INDIRECT ADDRESSING OPERATION**



**FIGURE 4-10: INDIRECT ADDRESSING**





## 8.4 PORTD, TRISD and LATD Registers

This section is only applicable to the PIC18C4X2 devices.

PORTD is an 8-bit wide, bi-directional port. The corresponding Data Direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

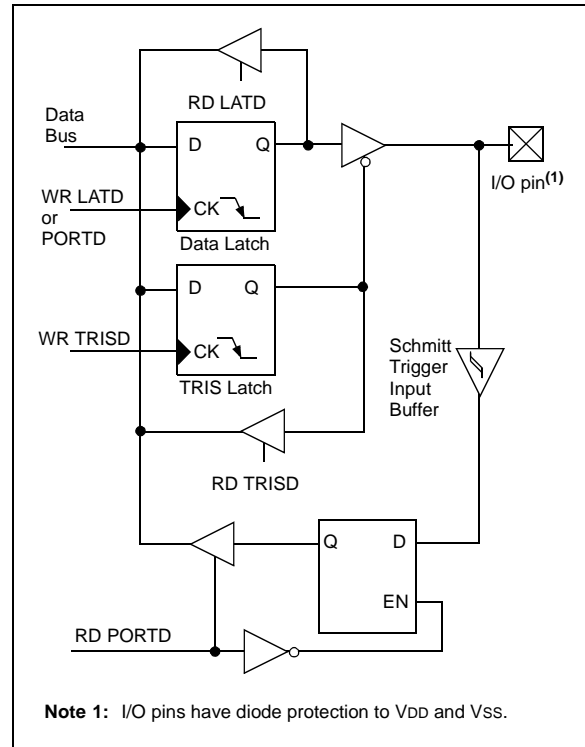
**Note:** On a Power-on Reset, these pins are configured as digital inputs.

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register reads and writes the latched output value for PORTD.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL. See Section 8.6 for additional information on the Parallel Slave Port (PSP).

**FIGURE 8-8: PORTD BLOCK DIAGRAM IN I/O PORT MODE**



### EXAMPLE 8-4: INITIALIZING PORTD

```

CLRF   PORTD   ; Initialize PORTD by
               ; clearing output
               ; data latches
CLRF   LATD    ; Alternate method
               ; to clear output
               ; data latches
MOVLW 0xCF    ; Value used to
               ; initialize data
               ; direction
MOVWF  TRISD  ; Set RD<3:0> as inputs
               ; RD<5:4> as outputs
               ; RD<7:6> as inputs
    
```

## 10.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter (two 8-bit registers: TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt-on-overflow from FFFFh to 0000h
- Reset from CCP module special event trigger

Figure 10-1 is a simplified block diagram of the Timer1 module.

Register 10-1 details the Timer1 control register. This register controls the operating mode of the Timer1 module, and contains the Timer1 oscillator enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit TMR1ON (T1CON<0>).

### REGISTER 10-1: T1CON: TIMER1 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	
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	
bit 7								bit 0

- bit 7 **RD16:** 16-bit Read/Write Mode Enable bit  
 1 = Enables register Read/Write of Timer1 in one 16-bit operation  
 0 = Enables register Read/Write of Timer1 in two 8-bit operations
- bit 6 **Unimplemented:** Read as '0'
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits  
 11 = 1:8 Prescale value  
 10 = 1:4 Prescale value  
 01 = 1:2 Prescale value  
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit  
 1 = Timer1 Oscillator is enabled  
 0 = Timer1 Oscillator is shut-off  
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit  
When TMR1CS = 1:  
 1 = Do not synchronize external clock input  
 0 = Synchronize external clock input  
When TMR1CS = 0:  
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit  
 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)  
 0 = Internal clock (FOSC/4)
- bit 0 **TMR1ON:** Timer1 On bit  
 1 = Enables Timer1  
 0 = Stops Timer1

**Legend:**

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

## 14.4.16.1 Bus Collision During a START Condition

During a START condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the START condition (Figure 14-24).
- SCL is sampled low before SDA is asserted low (Figure 14-25).

During a START condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

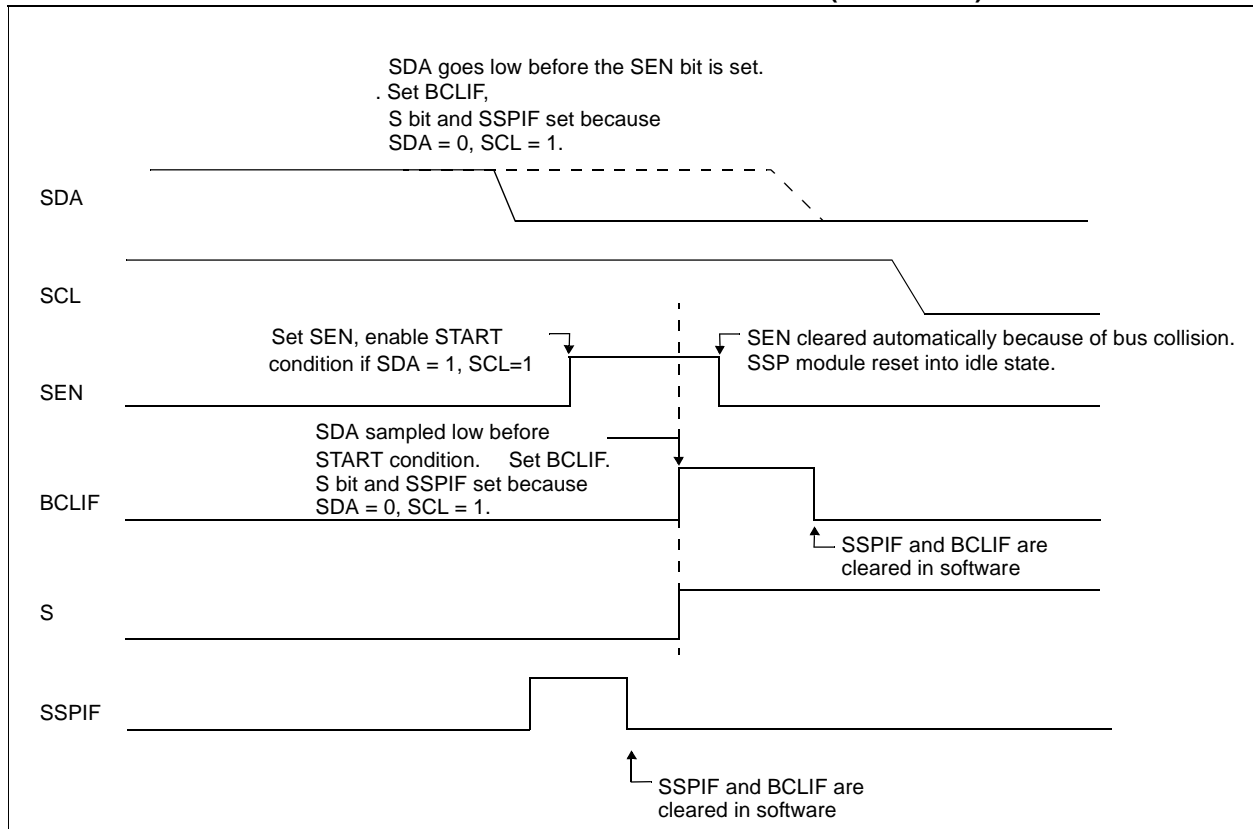
- the START condition is aborted,
- the BCLIF flag is set, and
- the MSSP module is reset to its IDLE state (Figure 14-24).

The START condition begins with the SDA and SCL pins de-asserted. When the SDA pin is sampled high, the baud rate generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the START condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 14-26). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The baud rate generator is then reloaded and counts down to 0, and during this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a START condition, is that no two bus masters can assert a START condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision, because the two masters must be allowed to arbitrate the first address following the START condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated START, or STOP conditions.

**FIGURE 14-24: BUS COLLISION DURING START CONDITION (SDA ONLY)**



# PIC18CXX2

## 15.4 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

### 15.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical, except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- a) The first word will immediately transfer to the TSR register and transmit.
- b) The second word will remain in TXREG register.
- c) Flag bit TXIF will not be set.
- d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
- e) If enable bit TXIE is set, the interrupt will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. Clear bits CREN and SREN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting enable bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.

**TABLE 15-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other RESETS
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	USART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'.

Shaded cells are not used for Synchronous Slave Transmission.

**Note 1:** The PSPIF, PSPIE and PSPPIP bits are reserved on the PIC18C2X2 devices. Always maintain these bits clear.

## 18.2 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO/RA6 pins of the device has been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The  $\overline{TO}$  bit in the RCON register will be cleared upon a WDT time-out.

The Watchdog Timer is enabled/disabled by a device configuration bit. If the WDT is enabled, software execution may not disable this function. When the WDTEN configuration bit is cleared, the SWDTEN bit enables/disables the operation of the WDT.

The WDT time-out period values may be found in the Electrical Specifications section under parameter #31. Values for the WDT postscaler may be assigned using the configuration bits.

**Note:** The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

**Note:** When a CLRWDT instruction is executed and the postscaler is assigned to the WDT, the postscaler count will be cleared, but the postscaler assignment is not changed.

### 18.2.1 CONTROL REGISTER

Register 18-7 shows the WDTCON register. This is a readable and writable register, which contains a control bit that allows software to override the WDT enable configuration bit, only when the configuration bit has disabled the WDT.

#### REGISTER 18-7: WDTCON REGISTER

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

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit

1 = Watchdog Timer is on

0 = Watchdog Timer is turned off if the WDTEN configuration bit in the configuration register = '0'

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR Reset

# PIC18CXX2

**TABLE 19-2: PIC18CXXX INSTRUCTION SET (CONTINUED)**

Mnemonic, Operands	Description	Cycles	16-bit Instruction Word				Status Affected	Notes	
			MSb	LSb					
<b>LITERAL OPERATIONS</b>									
ADDLW	k	Add literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW	k	AND literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW	k	Inclusive OR literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR	f, k	Move literal (12-bit) 2nd word to FSRx 1st word	2	1110	1110	00ff	kkkk	None	
MOVLB	k	Move literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW	k	Move literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW	k	Multiply literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW	k	Subtract WREG from literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW	k	Exclusive OR literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
<b>DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS</b>									
TBLRD*		Table Read	2	0000	0000	0000	1000	None	
TBLRD*+		Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-		Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD+*		Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*		Table Write	2 (5)	0000	0000	0000	1100	None	
TBLWT*+		Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*-		Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT+*		Table Write with pre-increment		0000	0000	0000	1111	None	

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable,  $d = 1$ ), the prescaler will be cleared if assigned.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOOP`.
- 4:** Some instructions are 2 word instructions. The second word of these instructions will be executed as a `NOOP`, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
- 5:** If the table write starts the write cycle to internal memory, the write will continue until terminated.

**CLRF**                    **Clear f**

---

Syntax:                    *[label]* CLRF f [,a]

Operands:                 $0 \leq f \leq 255$   
 $a \in [0,1]$

Operation:                 $000h \rightarrow f$   
 $1 \rightarrow Z$

Status Affected:        Z

Encoding:                

0110	101a	ffff	ffff
------	------	------	------

Description:             Clears the contents of the specified register. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words:                    1

Cycles:                   1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

**Example:**                CLRF                FLAG\_REG, 1

Before Instruction

FLAG\_REG = 0x5A

After Instruction

FLAG\_REG = 0x00

**CLRWDT**                **Clear Watchdog Timer**

---

Syntax:                    [*label*] CLRWDT

Operands:                None

Operation:                 $000h \rightarrow$  WDT,  
 $000h \rightarrow$  WDT postscaler,  
 $1 \rightarrow \overline{TO}$ ,  
 $1 \rightarrow \overline{PD}$

Status Affected:         $\overline{TO}$ ,  $\overline{PD}$

Encoding:                

0000	0000	0000	0100
------	------	------	------

Description:             CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits  $\overline{TO}$  and  $\overline{PD}$  are set.

Words:                    1

Cycles:                   1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

**Example:**                CLRWDT

Before Instruction

WDT counter = ?

After Instruction

WDT counter = 0x00

WDT Postscaler = 0

$\overline{TO}$  = 1

$\overline{PD}$  = 1

**LFSR**                      **Load FSR**

---

Syntax:                    [ *label* ] LFSR f,k

Operands:                 $0 \leq f \leq 2$   
 $0 \leq k \leq 4095$

Operation:                 $k \rightarrow \text{FSRf}$

Status Affected:        None

Encoding:                

1110	1110	00ff	$k_{11}kkk$
1111	0000	$k_7kkk$	kkkk

Description:             The 12-bit literal 'k' is loaded into the file select register pointed to by 'f'.

Words:                    2

Cycles:                    2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSRfH
Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSRfL

**Example:**                      LFSR 2, 0x3AB

After Instruction

```
FSR2H = 0x03
FSR2L = 0xAB
```

**MOVF**                      **Move f**

---

Syntax:                    [ *label* ] MOVF f[,d[,a]]

Operands:                 $0 \leq f \leq 255$   
 $d \in [0,1]$   
 $a \in [0,1]$

Operation:                 $f \rightarrow \text{dest}$

Status Affected:        N,Z

Encoding:                

0101	00da	ffff	ffff
------	------	------	------

Description:             The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256 byte bank. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words:                    1

Cycles:                    1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write WREG

**Example:**                      MOVF REG, 0, 0

Before Instruction

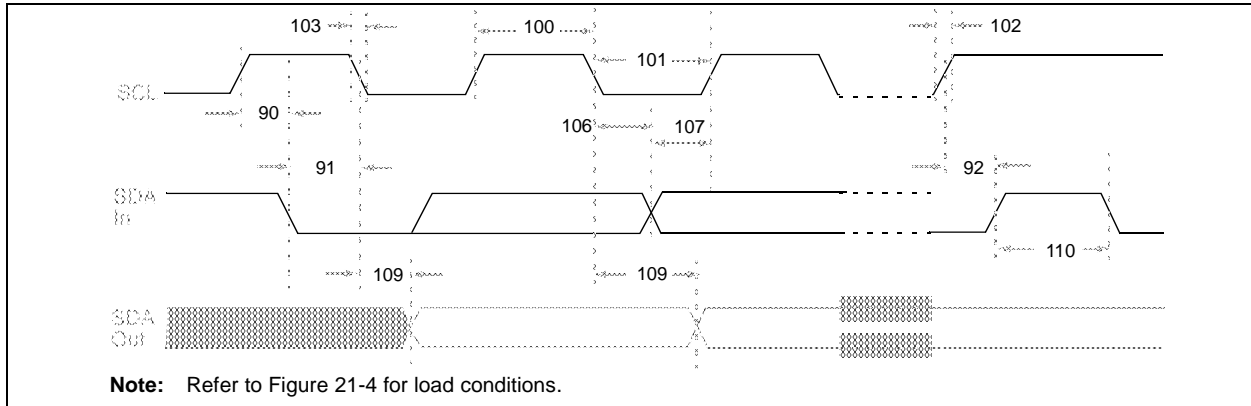
```
REG = 0x22
WREG = 0xFF
```

After Instruction

```
REG = 0x22
WREG = 0x22
```



**FIGURE 21-17: I<sup>2</sup>C BUS DATA TIMING**



**TABLE 21-16: I<sup>2</sup>C BUS DATA REQUIREMENTS (SLAVE MODE)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions	
100	THIGH	Clock high time	100 kHz mode	4.0	—	μs	PIC18CXXX must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	PIC18CXXX must operate at a minimum of 10 MHz
			SSP Module	1.5T <sub>CY</sub>	—		
101	TLOW	Clock low time	100 kHz mode	4.7	—	μs	PIC18CXXX must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	PIC18CXXX must operate at a minimum of 10 MHz
			SSP Module	1.5T <sub>CY</sub>	—		
102	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 to 400 pF
103	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 to 400 pF
90	TSU:STA	START condition setup time	100 kHz mode	4.7	—	μs	Only relevant for Repeated START condition
			400 kHz mode	0.6	—	μs	
91	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	
106	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	<b>(Note 2)</b>
			400 kHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output valid from clock	100 kHz mode	—	3500	ns	<b>(Note 1)</b>
			400 kHz mode	—	—	ns	
110	TBUF	Bus free time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
D102	CB	Bus capacitive loading	—	400	pF		

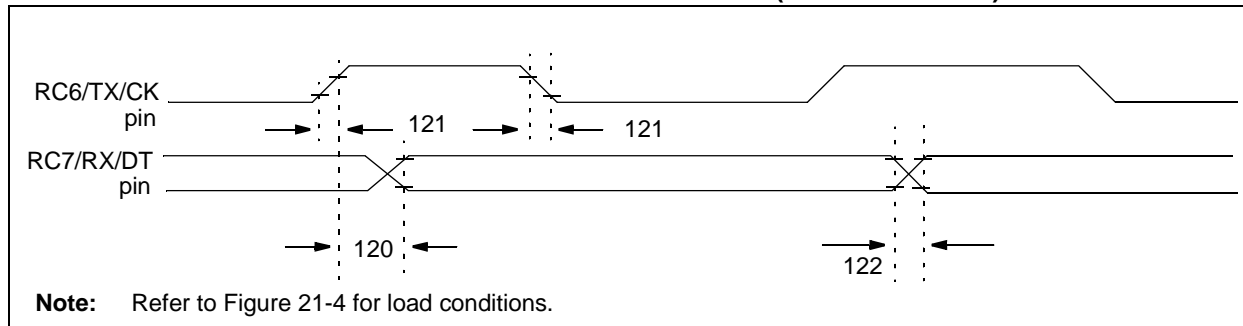
**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

**2:** A fast mode I<sup>2</sup>C bus device can be used in a standard mode I<sup>2</sup>C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line.

TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the standard mode I<sup>2</sup>C bus specification) before the SCL line is released.

# PIC18CXX2

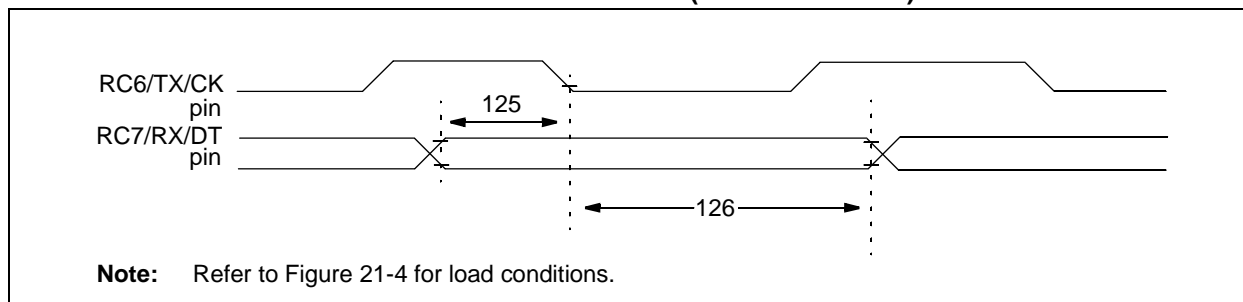
**FIGURE 21-20: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**



**TABLE 21-19: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions	
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE)					
		Clock high to data out valid	PIC18CXXX	—	40	ns	
			PIC18LCXXX	—	100	ns	
121	Tckrf	Clock out rise time and fall time (Master mode)	PIC18CXXX	—	25	ns	
			PIC18LCXXX	—	50	ns	
122	TdtV	Data out rise time and fall time	PIC18CXXX	—	25	ns	
			PIC18LCXXX	—	50	ns	

**FIGURE 21-21: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**

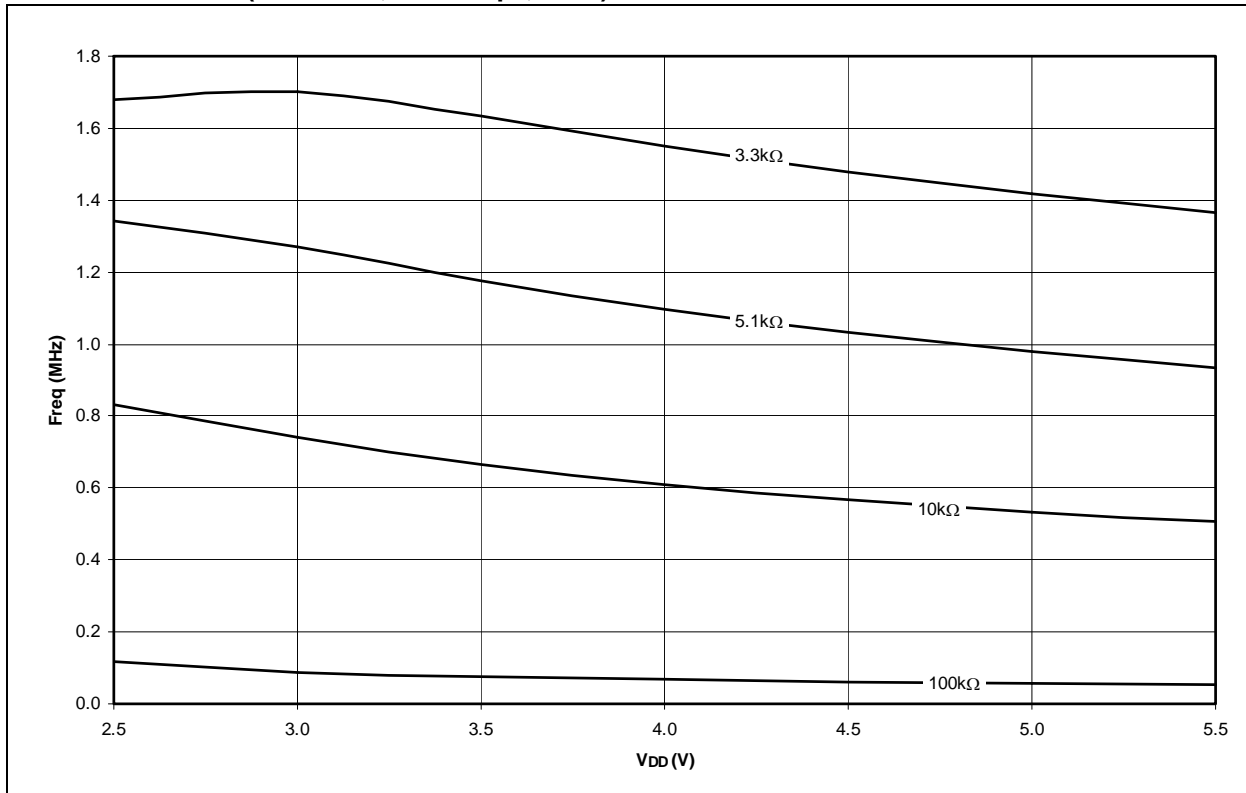


**TABLE 21-20: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

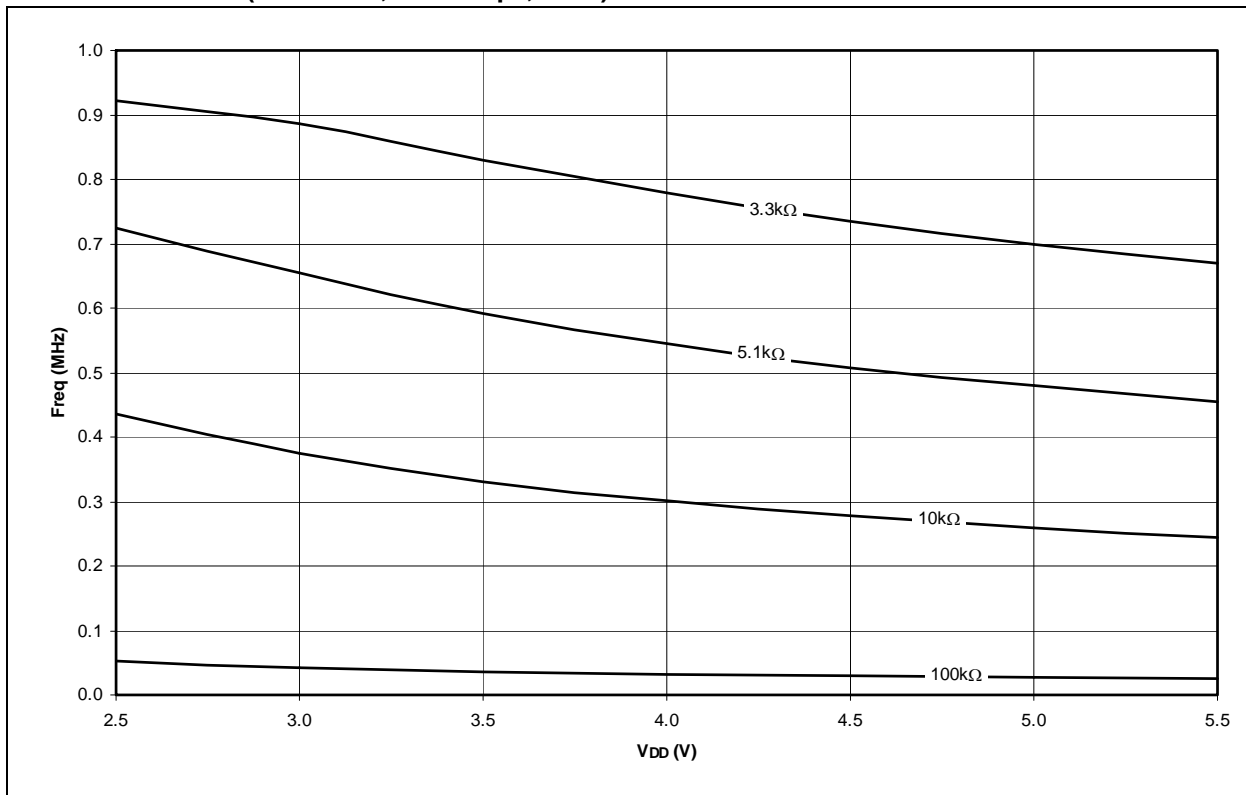
Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TdtV2ckI	SYNC RCV (MASTER & SLAVE)				
		Data hold before CK ↓ (DT hold time)	10	—	ns	
126	TckL2dtI	Data hold after CK ↓ (DT hold time)	15	—	ns	

# PIC18CXX2

**FIGURE 22-11: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, 25°C)**



**FIGURE 22-12: AVERAGE  $F_{osc}$  vs.  $V_{DD}$  FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, 25°C)**



# PIC18CXX2

Code Examples	
16 x 16 Signed Multiply Routine	62
16 x 16 Unsigned Multiply Routine	62
8 x 8 Signed Multiply Routine	61
8 x 8 Unsigned Multiply Routine	61
Changing Between Capture Prescalers	109
Fast Register Stack	39
Initializing PORTA	77
Initializing PORTB	80
Initializing PORTC	83
Initializing PORTD	85
Initializing PORTE	87
Loading the SSPBUF Register	122
Saving STATUS, WREG and BSR Registers in RAM	75
Code Protection	179, 186
COMF	204
Compare (CCP Module)	110
Associated Registers	111
Block Diagram	110
CCP Pin Configuration	110
CCPR1H:CCPR1L Registers	110
Software Interrupt	110
Special Event Trigger	99, 105, 110, 171
Timer1 Mode Selection	110
Configuration Bits	179
Context Saving During Interrupts	75
Example Code	75
Conversion Considerations	288
CPFSEQ	204
CPFSGT	205
CPFSLT	205
<b>D</b>	
Data Memory	42
General Purpose Registers	42
Special Function Registers	42
DAW	206
DC Characteristics	237, 240
DECF	206
DECFSNZ	207
DECFSZ	207
Device Differences	287
Direct Addressing	51
<b>E</b>	
Electrical Characteristics	235
Errata	5
<b>F</b>	
Firmware Instructions	187
<b>G</b>	
General Call Address Sequence	133
General Call Address Support	133
GOTO	208
<b>I</b>	
I/O Ports	77
I <sup>2</sup> C (SSP Module)	128
ACK Pulse	128, 129
Addressing	129
Block Diagram	128
Read/Write Bit Information (R/W Bit)	129
Reception	129
Serial Clock (RC3/SCK/SCL)	129
Slave Mode	128
Timing Diagram, Data	257
Timing Diagram, START/STOP Bits	256
Transmission	129
I <sup>2</sup> C Master Mode Reception	139
I <sup>2</sup> C Master Mode Repeated START Condition	138
I <sup>2</sup> C Module	
Acknowledge Sequence Timing	142
Baud Rate Generator	
Block Diagram	
Baud Rate Generator	136
BRG Reset Due to SDA Collision	146
BRG Timing	136
Bus Collision	
Acknowledge	144
Repeated START Condition	147
Repeated START Condition Timing (Case 1)	147
Repeated START Condition Timing (Case 2)	147
START Condition	145
START Condition Timing	145, 146
STOP Condition	148
STOP Condition Timing (Case 1)	148
STOP Condition Timing (Case 2)	148
Transmit Timing	144
Bus Collision Timing	144
Clock Arbitration	143
Clock Arbitration Timing (Master Transmit)	143
General Call Address Support	133
Master Mode 7-bit Reception Timing	141
Master Mode Operation	135
Master Mode START Condition	137
Master Mode Transmission	139
Master Mode Transmit Sequence	135
Multi-Master Mode	144
Repeat START Condition Timing	138
STOP Condition Receive or Transmit Timing	143
STOP Condition Timing	142
Waveforms for 7-bit Reception	130
Waveforms for 7-bit Transmission	130
ICEPIC In-Circuit Emulator	230
ID Locations	179, 186
INCF	208
INCFSZ	209
In-Circuit Serial Programming (ICSP)	179, 186
Indirect Addressing	51
FSR Register	50
INFSNZ	209
Instruction Cycle	39
Instruction Flow/Pipelining	40
Instruction Format	189

# PIC18CXX2

---