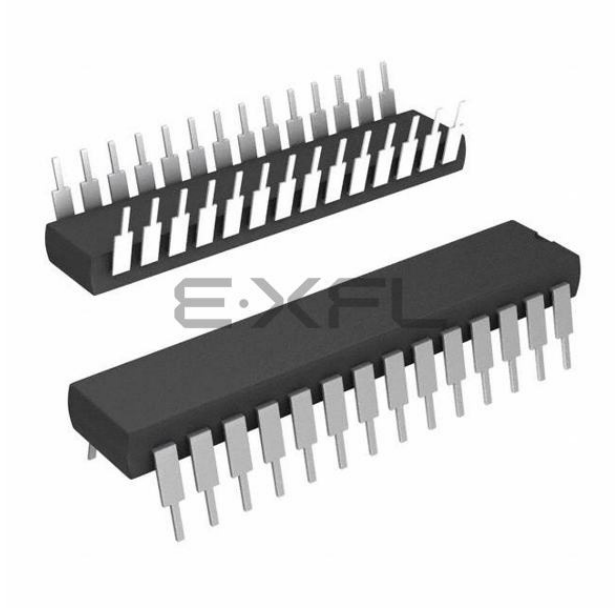


Welcome to E-XFL.COM

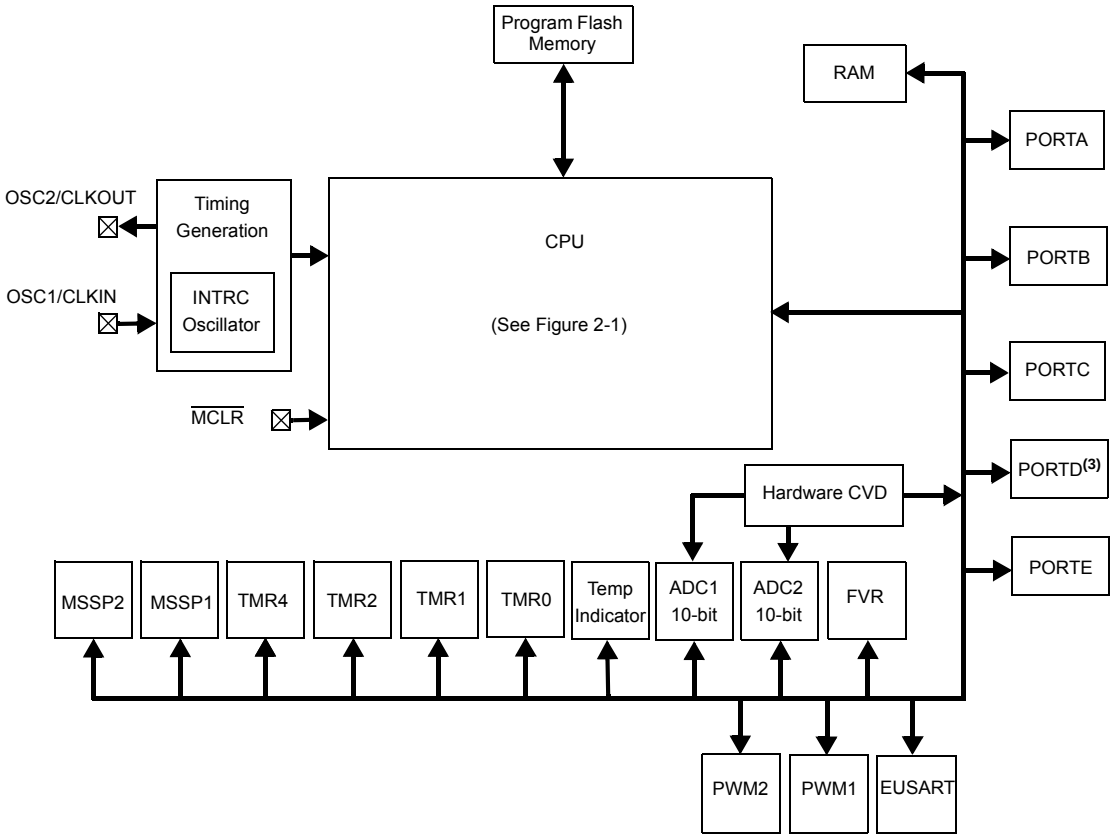
What is "Embedded - Microcontrollers"?

"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	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 23x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1566-e-sp

FIGURE 1-1: PIC16LF1566/1567 BLOCK DIAGRAM(1,2)



Note 1: See applicable chapters for more information on peripherals.
2: See Table 1-1 for peripherals available on specific devices.
3: PIC16LF1567 only.

TABLE 1-3: PIC16LF1567 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB3/AN28/PWM23	RB3	TTL	CMOS	General Purpose I/O with IOC and WPU.
	AN28	AN	—	ADC Channel Input for ADC2.
	PWM23	—	CMOS	PWM Output for PWM2.
RB4/AN18/AD1GRDA(1)/AD2GRDA(1)	RB4	TTL	CMOS	General Purpose I/O with IOC and WPU.
	AN18	AN	—	ADC Channel Input for ADC1.
	AD1GRDA	—	CMOS	ADC1 Guard Ring Output A.
	AD2GRDA	—	CMOS	ADC2 Guard Ring Output A.
RB5/AN29/AD1GRDA(1)/AD2GRDA(1)/ T1G	RB5	TTL	CMOS	General Purpose I/O with IOC and WPU.
	AN29	AN	—	ADC Channel Input for ADC2.
	AD1GRDA	—	CMOS	ADC1 Guard Ring Output A.
	AD2GRDA	—	CMOS	ADC2 Guard Ring Output A.
	T1G	ST	—	Timer1 Gate Input
RB6/AN19/AD1GRDB(1)/AD2GRDB(1)/ ICSPCLK/ICDCLK	RB6	TTL	CMOS	General Purpose I/O with IOC and WPU.
	AN19	AN	—	ADC Channel Input for ADC1.
	AD1GRDB	—	CMOS	ADC1 Guard Ring Output B.
	AD2GRDB	—	CMOS	ADC2 Guard Ring Output B.
	ICSPCLK	ST	CMOS	ICSP™ Programming Clock.
	ICDCLK	ST	CMOS	In-Circuit Debug Clock.
RB7/AN40/AD1GRDB(1)/AD2GRDB(1)/ ICSPDAT/ICDDAT	RB7	TTL	CMOS	General Purpose I/O with IOC and WPU.
	AN40	AN	—	ADC Channel Input for ADC2.
	AD1GRDB	—	CMOS	ADC1 Guard Ring Output B.
	AD2GRDB	—	CMOS	ADC2 Guard Ring Output B.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
	ICDDAT	ST	CMOS	In-Circuit Debug Data.
RC0/AN12/T1CKI/SDO2	RC0	TTL	CMOS	General Purpose I/O.
	AN12	AN	—	ADC Channel Input for ADC1.
	T1CKI	ST	—	Timer1 Clock Input.
	SDO2	—	CMOS	SPI Data Output for MSSP2.
RC1/AN23/PWM2/SCL2/SCK2	RC1	TTL	CMOS	General Purpose I/O.
	AN23	AN	—	ADC Channel Input for ADC2.
	PWM2	—	CMOS	PWM Output for PWM2.
	SCL2	I^2C	OD	I^2C Clock for MSSP2.
	SCK2	ST	CMOS	SPI Clock for MSSP2.
RC2/AN13/PWM1/SDA2/SDI2	RC2	TTL	CMOS	General Purpose I/O.
	AN13	AN	—	ADC Channel Input for ADC1.
	PWM1	—	CMOS	PWM Output for PWM1.
	SDA2	I^2C	OD	I^2C Data for MSSP2.
	SDI2	CMOS	—	SPI Data Input for MSSP2.
RC3/AN24/SCL1/SCK1	RC3	TTL	CMOS	General Purpose I/O.
	AN24	AN	—	ADC Channel Input for ADC2.
	SCL1	I^2C	OD	I^2C Clock for MSSP1.
	SCK1	ST	CMOS	SPI Clock for MSSP1.

TABLE 3-5: PIC16LF1566/1567 MEMORY MAP, BANKS 8-15

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	CPU Core Register, see Table 3-2 for specifics															
401h																
402h																
403h																
404h																
405h																
406h																
407h																
408h																
409h																
40Ah																
40Bh																
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—	
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—	
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—	
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—	
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—	
411h	—	491h	—	511h	—	591h	—	611h	PWM1DCL	691h	ADCTX	711h	AD1CON0	791h	AD2CON0	
412h	—	492h	—	512h	—	592h	—	612h	PWM1DCH	692h	AD1TX0	712h	ADCOMCON	792h	—	
413h	—	493h	—	513h	—	593h	—	613h	PWM1CON	693h	AD1TX1	713h	AD1CON2	793h	AD2CON2	
414h	—	494h	—	514h	—	594h	—	614h	PWM2DCL	694h	AD2TX0	714h	AD1CON3	794h	AD2CON3	
415h	TMR4	495h	—	515h	—	595h	—	615h	PWM2DCH	695h	AD2TX1	715h	ADSTAT	795h	—	
416h	PR4	496h	—	516h	—	596h	—	616h	PWM2CON	696h	—	716h	AD1PRECON	796h	AD2PRECON	
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	AD1ACQCON	797h	AD2ACQCON	
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	AD1GRD	798h	AD2GRD	
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	AD1CAPCON	799h	AD2CAPCON	
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	AAD1RES0L	79Ah	AAD2RES0L	
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	AAD1RES0H	79Bh	AAD2RES0H	
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	AAD1RES1L	79Ch	AAD2RES1L	
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	PWMTMRS	69Dh	—	71Dh	AAD1RES1H	79Dh	AAD2RES1H	
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	PWM1AOE	69Eh	—	71Eh	AD1CH0	79Eh	AD2CH0	
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	PWM2AOE	69Fh	—	71Fh	AD1CH1	79Fh	AD2CH1	
420h	General Purpose Register 80 Bytes	4A0h	General Purpose Register 80 Bytes	520h	General Purpose Register 80 Bytes	5A0h	General Purpose Register 80 Bytes	620h 64Fh	General Purpose Register 48 Bytes	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'	
46Fh		4EFh		56Fh		5EFh		650h 66Fh	Unimplemented Read as '0'	6EFh		76Fh		7EFh		
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh	
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh		

Note 1: These ADC registers are the same as the registers in Bank 1.

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	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
Bank 31											
F80h	INDF0(1)	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
F81h	INDF1(1)	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
F82h	PCL(1)	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
F83h	STATUS(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
F84h	FSR0L(1)	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
F85h	FSR0H(1)	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
F86h	FSR1L(1)	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
F87h	FSR1H(1)	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
F88h	BSR(1)	—	—	—	BSR<4:0>					---0 0000	---0 0000
F89h	WREG(1)	Working Register								0000 0000	uuuu uuuu
F8Ah	PCLATH(1)	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
F8Bh	INTCON(1)	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
F8Ch	ICDIO	PORT_ ICDDAT	PORT_ ICDCLK	LAT_ ICDDAT	LAT_ ICDCLK	TRIS_ ICDDAT	TRIS_ ICDCLK	—	—	xxxxxx--	
F8Dh	ICDCON0	INBUG	FREEZ	SSTEP	—	DBGINEX	—	—	RSTVEC	xxx-x--x	
F8Eh to F90h	—	Unimplemented								—	—
F91h	ICDSTAT	TRP1HLTF	TRP0HLTF	—	—	—	—	USRHLTF	—	xx----x-	
F92h to F95h	—	Unimplemented								—	—
F96h	ICDINSTL	DBGIN7	DBGIN6	DBGIN5	DBGIN4	DBGIN3	DBGIN2	DBGIN1	DBGIN0	xxxxxxxx	
F97h	ICDINSTH	—	—	DBGIN13	DBGIN12	DBGIN11	DBGIN10	DBGIN9	DBGIN8	--xxxxxx	
F98h to F9Bh	—	Unimplemented								—	—
F9Ch	ICDBK0CON	BKEN	—	—	—	—	—	—	BKHLT	x------x	
F9Dh	ICDBK0L	BKA7	BKA6	BKA5	BKA4	BKA3	BKA2	BKA1	BKA0	xxxxxxxx	
F9Eh	ICDBK0H	—	BKA14	BKA13	BKA12	BKA11	BKA10	BKA9	BKA8	-xxxxxxx	
F9Fh	—	Unimplemented								—	—
FA0h to FBFh	—	Unimplemented								—	—
FC0h to FCFh	—	Unimplemented								—	—
FD0h to FE2h	—	Unimplemented								—	—
FE3h	BSRICDSHAD	—	—	—	BSR_ICDSHAD					---xxxxx	–
FE4h	STATUS_- SHAD	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -xxx	---- -uuu
FE5h	WREG_SHAD	WREG_SHAD								xxxx xxxx	uuuu uuuu
FE6h	BSR_SHAD	—	—	—	BSR_SHAD					---x xxxx	---u uuuu
FE7h	PCLATH_- SHAD	—	PCLATH_SHAD							-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	FSR0L_SHAD								xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD	FSR0H_SHAD								xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD	FSR1L_SHAD								xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD	FSR1H_SHAD								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: These registers can be accessed from any bank.
2: PIC16LF1567.
3: These registers/bits are available at two address locations, in Bank 1 and Bank 14.
4: PIC16LF1566 only.
5: Unimplemented, read as '1'.

REGISTER 4-4: REVISIONID: REVISION ID REGISTER(1)

R	R	R	R	R	R
REV<13:8>					
bit 13					bit 8

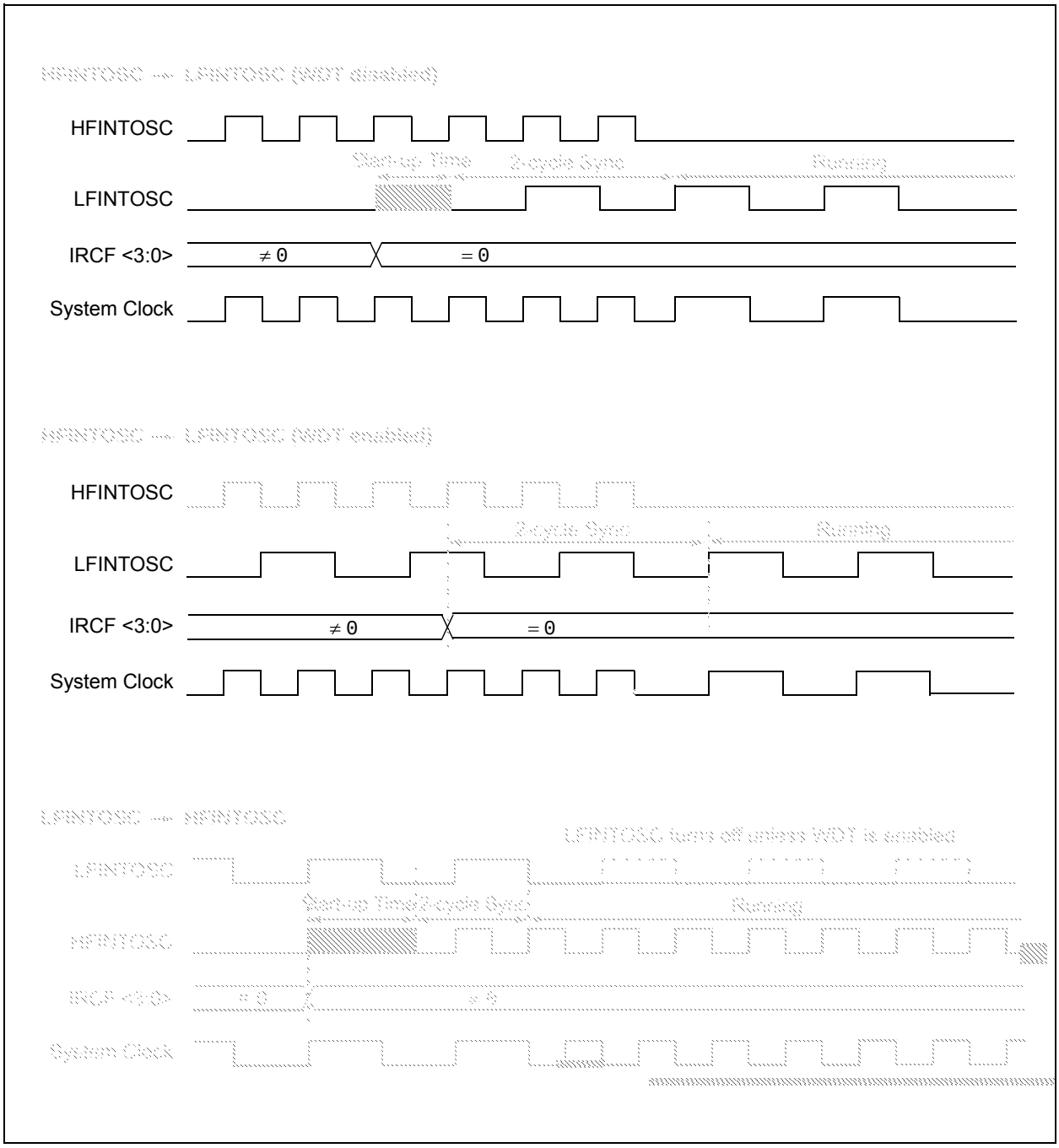
R	R	R	R	R	R	R	R
REV<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit		
'0' = Bit is cleared	'1' = Bit is set	x = Bit is unknown

bit 13-0 **REV<13:0>:** Revision ID bits
These bits are used to identify the device revision.

Note 1: This location cannot be written.

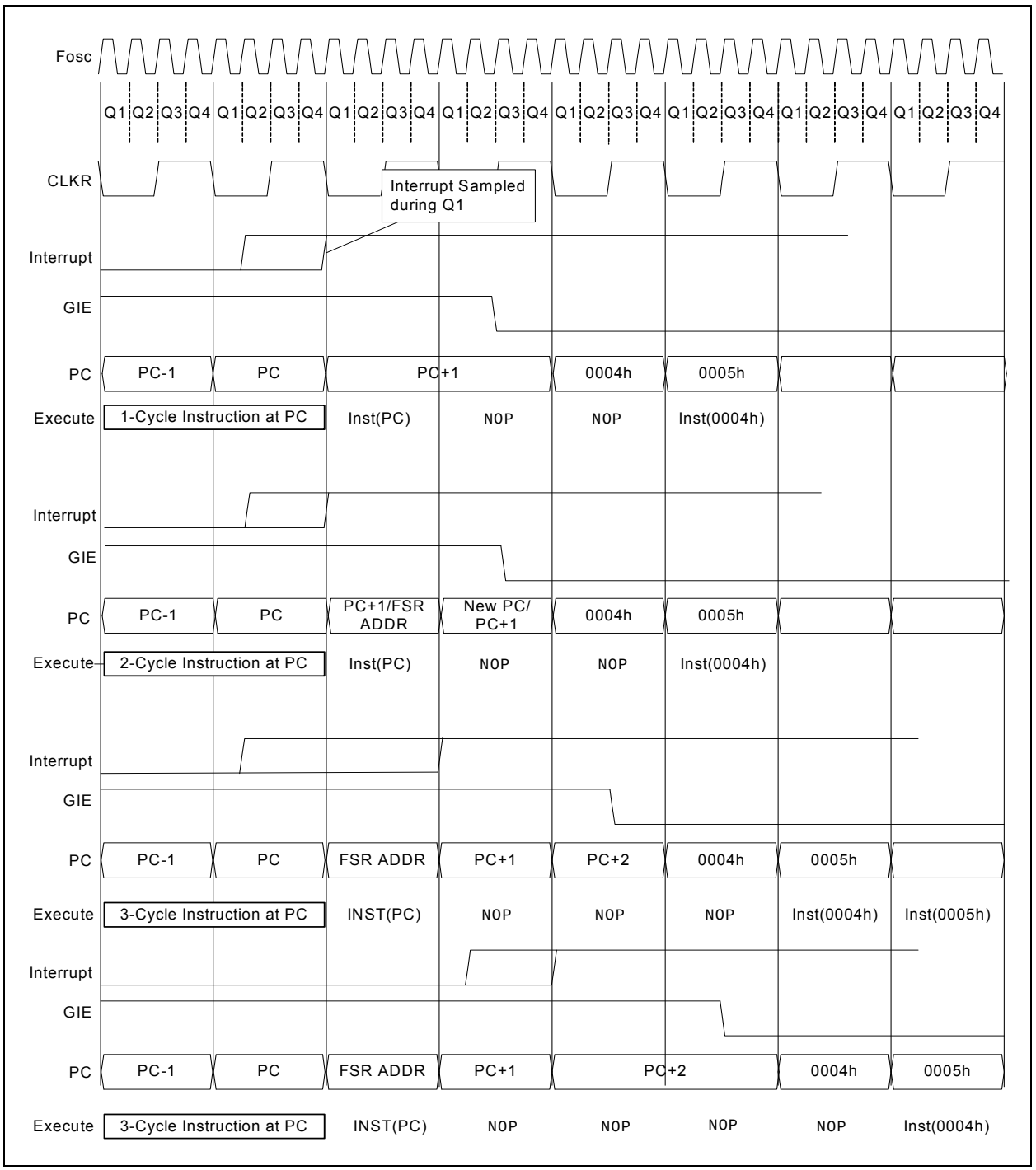
FIGURE 5-3: INTERNAL OSCILLATOR SWITCH TIMING



7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

FIGURE 7-2: INTERRUPT LATENCY



EXAMPLE 10-2: ERASING ONE ROW OF PROGRAM MEMORY

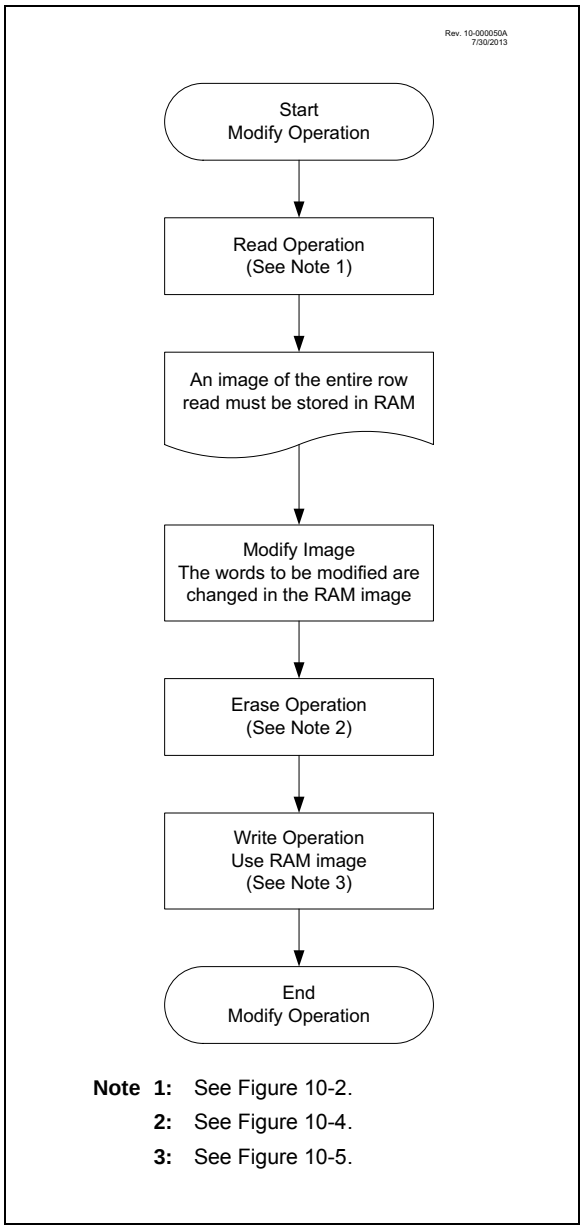
```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)


        BCF         INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL
        MOVF        ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W         ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS     ; Not configuration space
        BSF         PMCON1,FREE     ; Specify an erase operation
        BSF         PMCON1,WREN     ; Enable writes

Required
Sequence
        MOVLW       55h             ; Start of required sequence to initiate erase
        MOVWF       PMCON2          ; Write 55h
        MOVLW       0AAh            ;
        MOVWF       PMCON2          ; Write AAh
        BSF         PMCON1,WR       ; Set WR bit to begin erase
        NOP                         ; NOP instructions are forced as processor starts
        NOP                         ; row erase of program memory.
                                    ;
                                    ; The processor stalls until the erase process is complete
                                    ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN     ; Disable writes
        BSF         INTCON,GIE      ; Enable interrupts
```


10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 10-2.
2: See Figure 10-4.
3: See Figure 10-5.

11.7 PORTC Registers

11.7.1 DATA REGISTER

PORTC is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 11-12). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTC register (Register 11-11) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

11.7.2 DIRECTION CONTROL

The TRISC register (Register 11-12) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.7.3 ANALOG CONTROL

The ANSELC register (Register 11-14) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELC bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELC bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSELx bits must be initialized to '0' by user software.

11.7.4 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-8.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input and some digital input functions are not included in the output priority list. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the output priority list.

TABLE 11-8: PORTC OUTPUT PRIORITY

Pin Name	Function Priority(1)
RC0	T1CKI SDO2 RC0
RC1	SCK2 SCL2 PWM2 RC1
RC2	SDA2 SDI2 PWM1 RC2
RC3	SCK1 SCL1 RC3
RC4	SDA1 SDI1 RC4
RC5	I2CLVL SDO1 RC5
RC6	TX CK RC6
RC7	RX DT RC7

Note 1: Priority listed from highest to lowest.

REGISTER 15-3: ADCON1(1)/ADCOMCON(2): ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			ADNREF	GO/$\overline{\text{DONE}}$_ALL	ADPREF<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ADFM:** ADC Result Format Select bit
- 1 = Right justified. Six Most Significant bits of ADxRESxH are set to '0' when the conversion result is loaded.
- 0 = Left justified. Six Least Significant bits of ADxRESxL are set to '0' when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
- 111 = FRC (clock supplied from an internal RC oscillator)
- 110 = FOSC/64
- 101 = FOSC/16
- 100 = FOSC/4
- 011 = FRC (clock supplied from an internal RC oscillator)
- 010 = FOSC/32
- 001 = FOSC/8
- 000 = FOSC/2

bit 3 **ADNREF:** ADC Negative Voltage Reference Configuration bit
- 1 = VREFL is connected to external VREF- pin(4)
- 0 = VREFL is connected to AVSS.

bit 2 **GO/$\overline{\text{DONE}}$_ALL(3):** Synchronized ADC Conversion Status bit
- 1 = Synchronized ADC conversion in progress. Setting this bit starts conversion in any ADC with ADxON = 1.
- 0 = Synchronized ADC conversion completed/ not in progress.

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
- 11 = VREFH is connected to internal Fixed Voltage Reference.
- 10 = VREFH is connected to external VREF+ pin(4)
- 01 = Reserved
- 00 = VREFH is connected to VDD

Note 1: Bank 1 name is ADCON1.

2: Bank 14 name is ADCOMCON.

3: Setting this bit triggers the GO/$\overline{\text{DONE}}$x bits in both ADCs. Each ADC will run a conversion according to its control register settings. This bit reads as an OR of the individual GO/$\overline{\text{DONE}}$x bits.

4: When selecting the VREF+ or VREF- pin as the source of the positive or negative reference, be aware that a minimum voltage specification exists. See **Section 25.0 "Electrical Specifications"** for details.

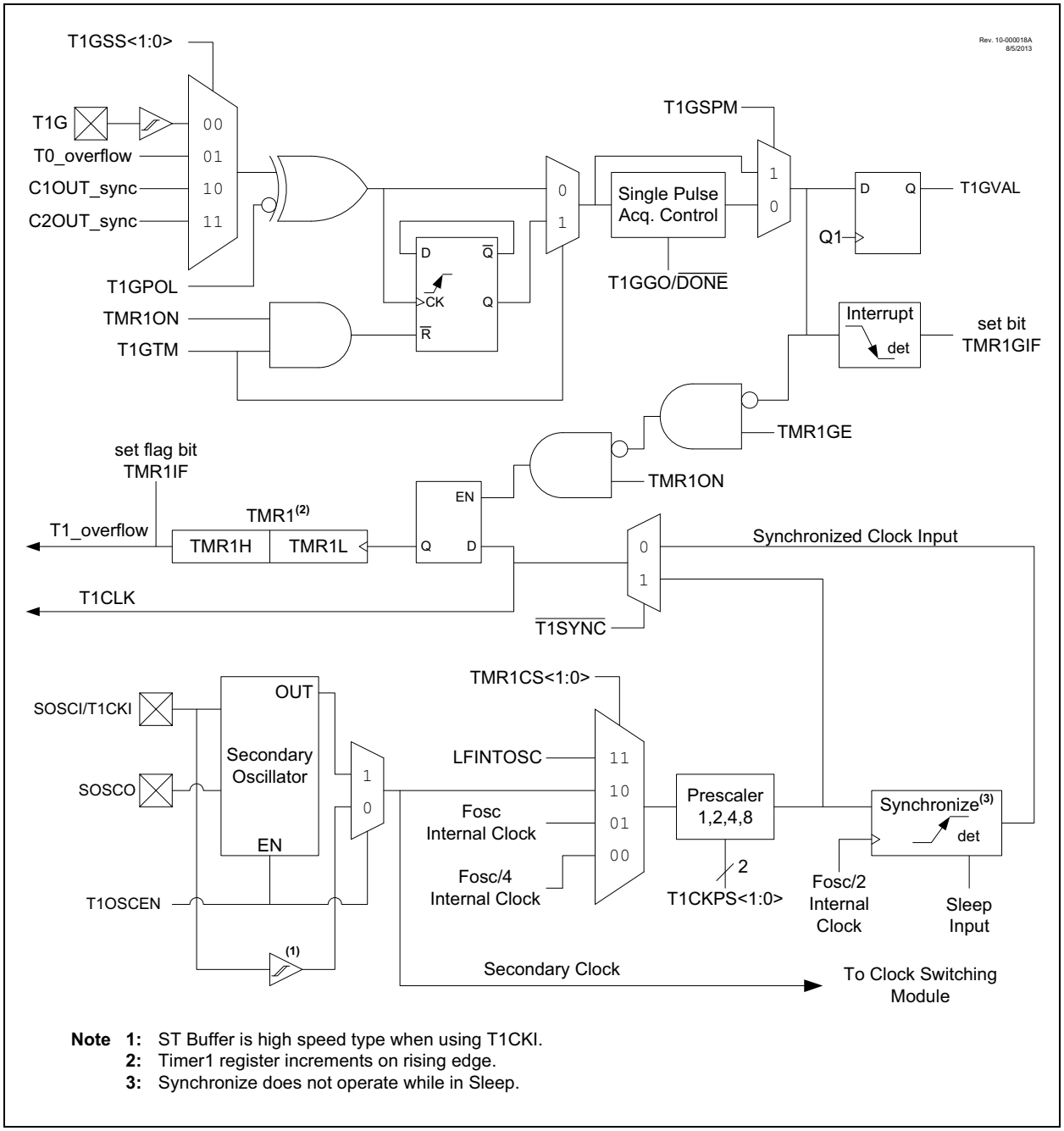
18.0 TIMER1 MODULE WITH GATE CONTROL

The Timer1 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Optionally synchronized comparator out
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- ADC Auto-Conversion Trigger(s)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 18-1 is a block diagram of the Timer1 module.

FIGURE 18-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using T1CKI.
2: Timer1 register increments on rising edge.
3: Synchronize does not operate while in Sleep.

20.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a chip select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCKx)
- Serial Data Out (SDOx)
- Serial Data In (SDIx)
- Slave Select ($\overline{SS}$x)

Figure 20-1 shows the block diagram of the MSSPx module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 20-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 20-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDOx output pin which is connected to, and received by, the slave's SDIx input pin. The slave device transmits information out on its SDOx output pin, which is connected to, and received by, the master's SDIx input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDOx pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDOx pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

20.4.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDAx from a high to a low state while SCLx line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 20-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDAx line low before asserting it low. This does not conform to the I^2C Specification that states no bus collision can occur on a Start.

20.4.6 STOP CONDITION

A Stop condition is a transition of the SDAx line from low-to-high state while the SCLx line is high.

Note: At least one SCLx low time must appear before a Stop is valid, therefore, if the SDAx line goes low then high again while the SCLx line stays high, only the Start condition is detected.

20.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 20-13 shows the wave form for a Restart condition.

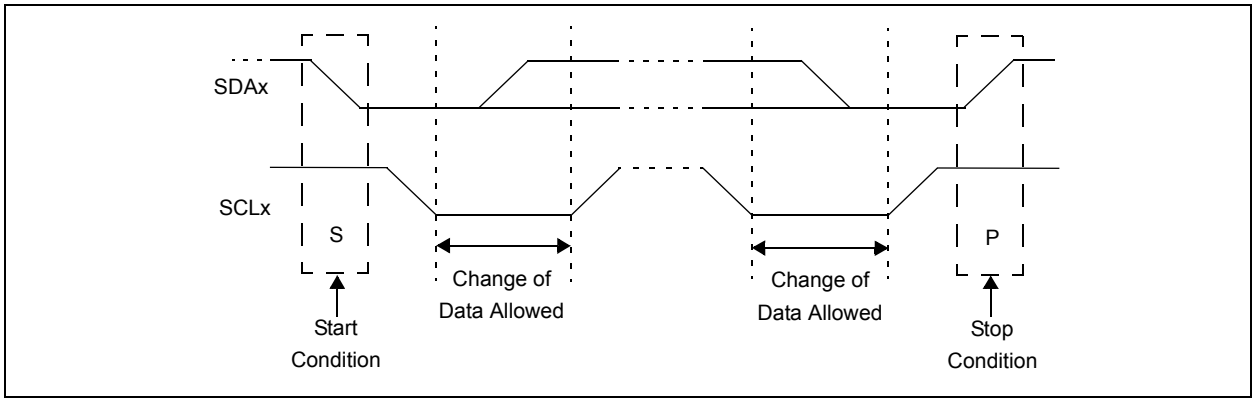
In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained. Until a Stop condition, a high address with R/$\overline{W}$ clear, or high address match fails.

20.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

FIGURE 20-12: I^2C START AND STOP CONDITIONS



20.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCLx line low effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCLx.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. Setting CKP will release SCLx and allow more communication.

20.5.6.1 Normal Clock Stretching

Following an $\overline{ACK}$ if the R/$\overline{W}$ bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the $\overline{ACK}$ sequence. Once the slave is ready; CKP is set by software and communication resumes.

Note 1: The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCLx.

2: Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCLx. It is now always cleared for read requests.

20.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCLx is stretched without CKP being cleared. SCLx is released immediately after a write to SSPxADD.

Note: Previous versions of the module did not stretch the clock if the second address byte did not match.

20.5.6.3 Byte NACKing

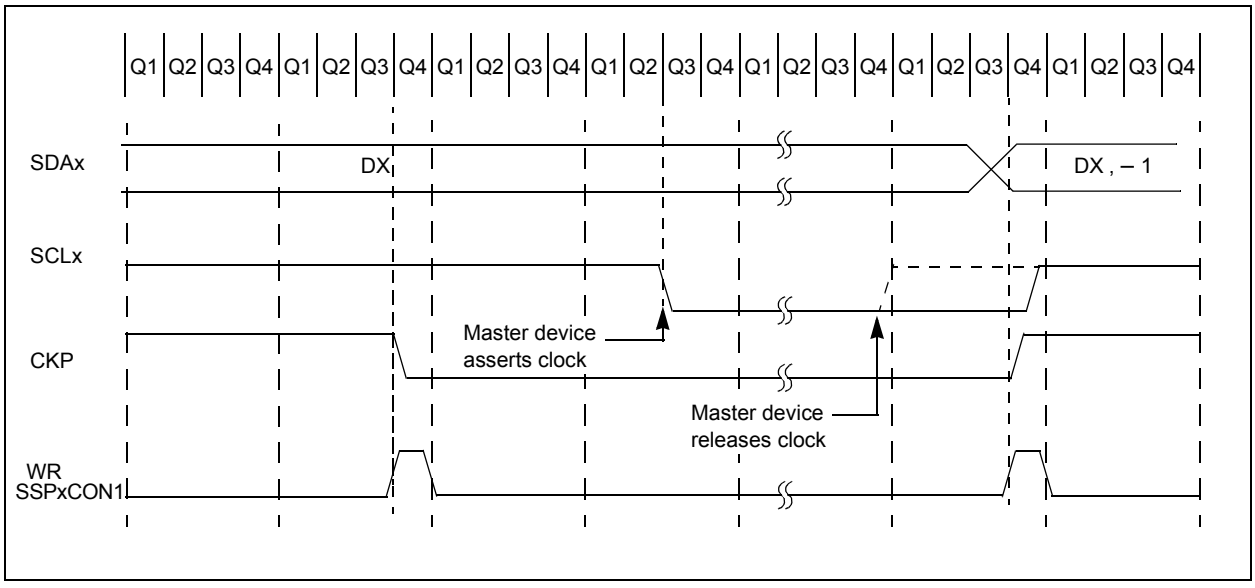
When the AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCLx for a received matching address byte. When the DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCLx for received data.

Stretching after the eighth falling edge of SCLx allows the slave to look at the received address or data and decide if it wants to ACK the received data.

20.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I^2C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 20-23).

FIGURE 20-23: CLOCK SYNCHRONIZATION TIMING



20.6 I²C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDAx and SCLx lines.

The following events will cause the SSPx Interrupt Flag bit, SSPxIF, to be set (SSPx interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSPx module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

2: Master mode suspends Start/Stop detection when sending the Start/Stop condition by means of the SEN/PEN control bits. The SSPIF bit is set at the end of the Start/Stop generation when hardware clears the control bit.

20.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDAx, while SCLx outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDAx, while SCLx outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCLx. See **Section 20.7 "Baud Rate Generator"** for more detail.

FIGURE 21-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

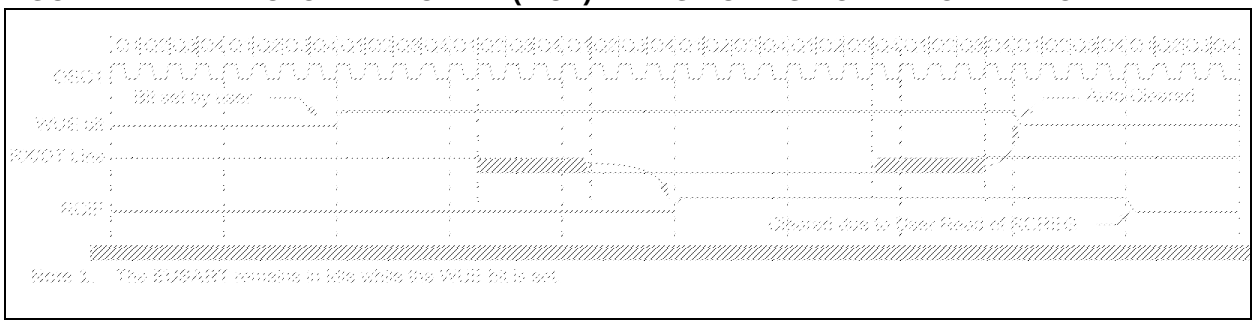


FIGURE 21-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP

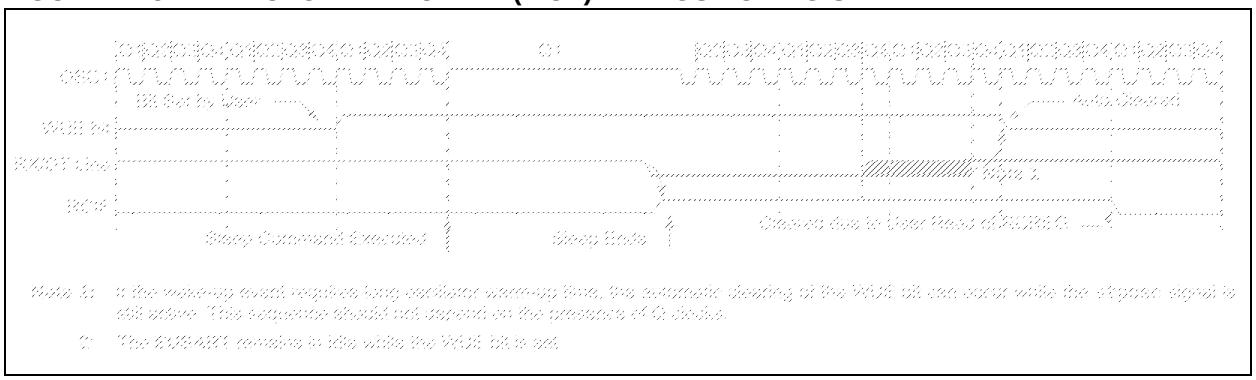


TABLE 24-3: ENHANCED MID-RANGE INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	–	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	–	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	–	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
INHERENT OPERATIONS									
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
NOP	–	No Operation	1	00	0000	0000	0000		
OPTION	–	Load OPTION_REG register with W	1	00	0000	0110	0010		
RESET	–	Software device Reset	1	00	0000	0000	0001		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C-COMPILER OPTIMIZED									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm kkkk	Z	2, 3
	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	1nmm	Z	2
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00	0000	0001	kkkk		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk			2

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See Table in the MOVIW and MOVWI instruction descriptions.

TABLE 25-4: I/O PORTS

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
		Input Low Voltage					
		I/O PORT:					
D030		with TTL buffer	—	—	0.15 VDD	V	1.8V ≤ VDD ≤ 3.6V
D031	VIL	with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 3.6V
		with SMBus levels	—	—	0.8	V	3.0V ≤ VDD ≤ 3.6V
		with I2CLVL enabled	—	—	0.3 VI2CLVL	V	TBD ≤ VI2CLVL ≤ VDD
D032		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:		—	—		
D040		with TTL buffer	0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD ≤ 3.6V
D041		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 3.6V
		with SMBus levels	2.1	—	—	V	3.0V ≤ VDD ≤ 3.6V
		with I2CLVL enabled	0.7 VI2CLVL	—	—	V	TBD ≤ VI2CLVL ≤ VDD
D042		$\overline{MCLR}$	0.8 VDD	—	—	V	
	IIL	**Input Leakage Current(1)**					
D060		I/O ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance at 85°C
				± 5	± 1000	nA	125°C
D061		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD at 85°C
	IPUR	**Weak Pull-up Current**					
D070*			25	100	200	µA	VDD = 3.3V, VPIN = VSS
	VOL	**Output Low Voltage(3)**					
D080		I/O ports	—	—	0.6	V	IOL = 6mA, VDD = 3.3V IOL = 1.8mA, VDD = 1.8V
	VOH	**Output High Voltage(3)**					
D090		I/O ports	VDD - 0.7	—	—	V	IOH = 3mA, VDD = 3.3V IOH = 1mA, VDD = 1.8V
		Capacitive Loading Specs on Output Pins					
D101A*	CIO	All I/O pins	—	—	50	pF	

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

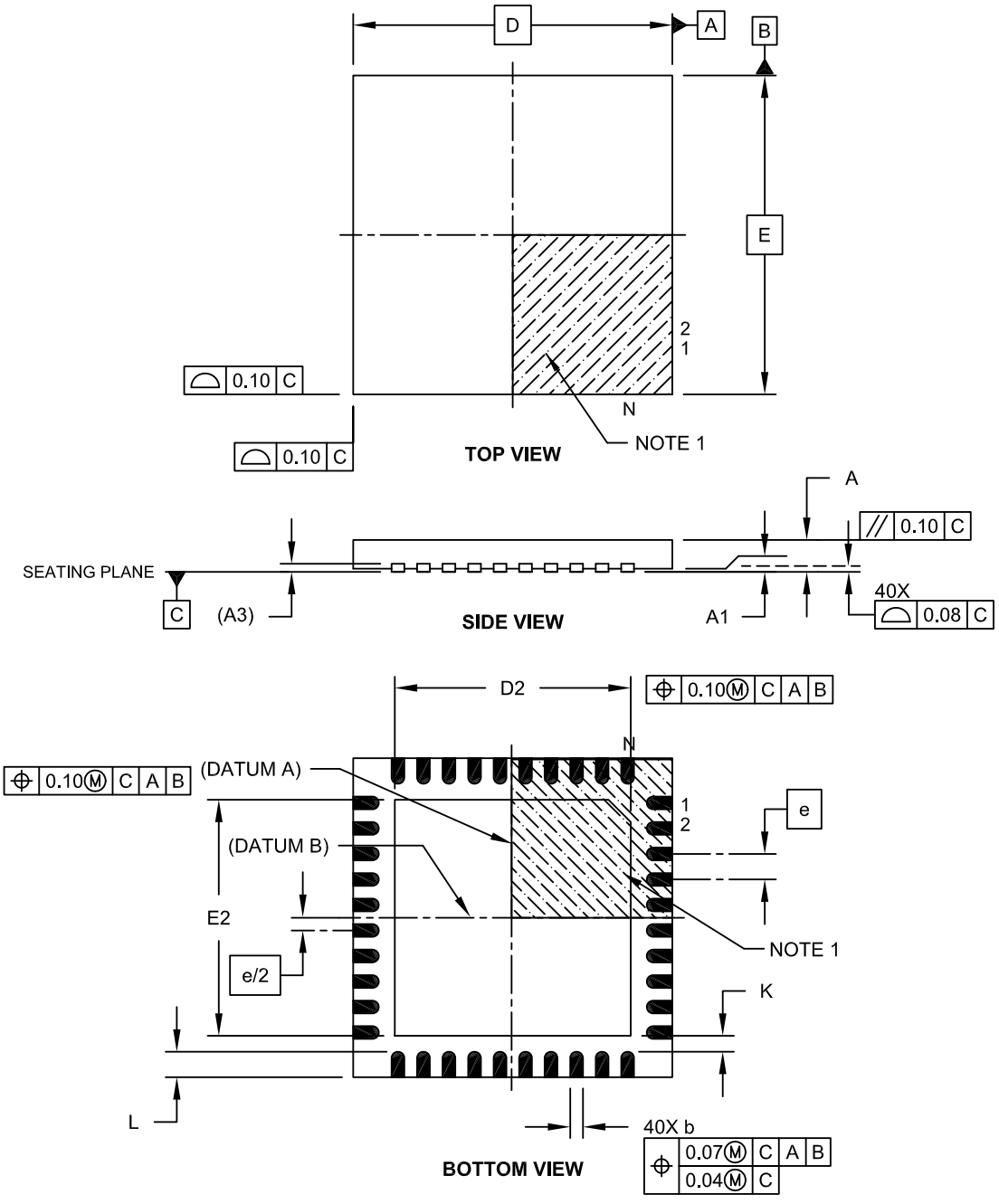
Note 1: Negative current is defined as current sourced by the pin.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Including OSC2 in CLKOUT mode.

40-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) – 5x5x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-156A Sheet 1 of 2

