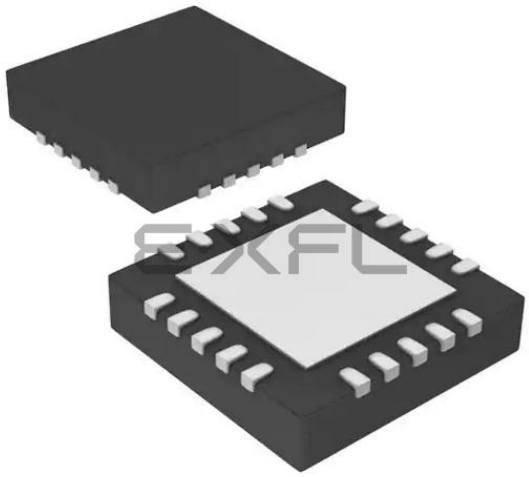


Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1508-e-ml

3.3 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.6 "Indirect Addressing"** for more information.

Data memory uses a 12-bit address. The upper five bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

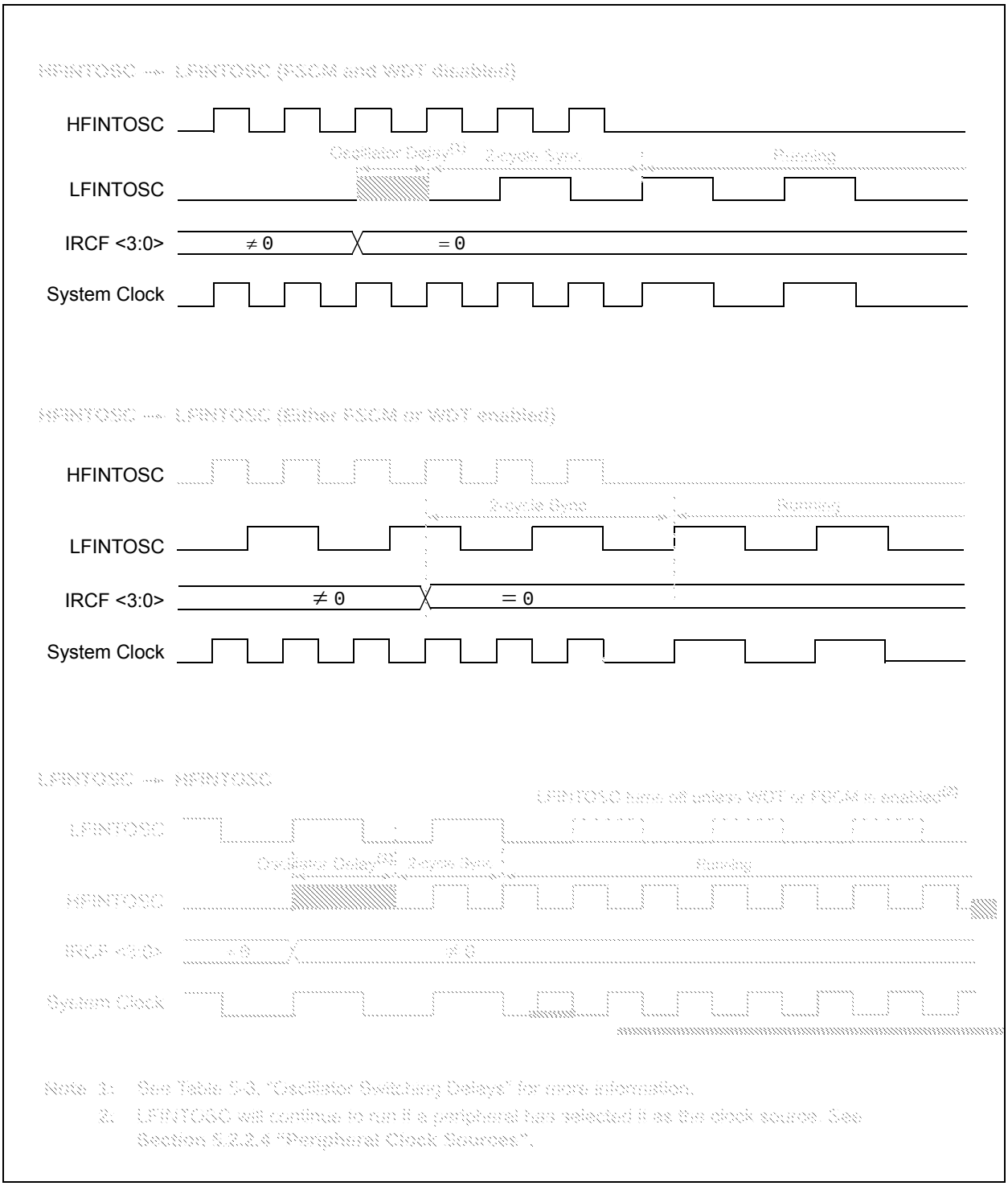
3.3.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-8.

TABLE 3-2: CORE REGISTERS

Addresses	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

FIGURE 5-7: INTERNAL OSCILLATOR SWITCH TIMING



Note 1: See Table 5-3, "Oscillator Switching Delays" for more information.
2. LFINTOSC will continue to run if a peripheral has selected it as the clock source. See Section 5.2.2.4 "Peripheral Clock Sources".

5.6 Register Definitions: Oscillator Control

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
—	IRCF<3:0>				—	SCS<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 **Unimplemented:** Read as '0'

bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
1111 = 16 MHz
1110 = 8 MHz
1101 = 4 MHz
1100 = 2 MHz
1011 = 1 MHz
1010 = 500 kHz(1)
1001 = 250 kHz(1)
1000 = 125 kHz(1)
0111 = 500 kHz (default upon Reset)
0110 = 250 kHz
0101 = 125 kHz
0100 = 62.5 kHz
001x = 31.25 kHz
000x = 31 kHz LF

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

bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 = Secondary oscillator
00 = Clock determined by FOSC<2:0> in Configuration Words.

Note 1: Duplicate frequency derived from HFINTOSC.

REGISTER 5-2: OSCSTAT: OSCILLATOR STATUS REGISTER

R-1/q	U-0	R-q/q	R-0/q	U-0	U-0	R-0/q	R-0/q
SOSCR	—	OSTS	HFIOFR	—	—	LFIOFR	HFIOFS
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	q = Conditional

bit 7 **SOSCR:** Secondary Oscillator Ready bit
If T1OSCEN = 1:
1 = Secondary oscillator is ready
0 = Secondary oscillator is not ready
If T1OSCEN = 0:
1 = Timer1 clock source is always ready

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

bit 5 **OSTS:** Oscillator Start-up Timer Status bit
When the FOSC<2:0> bits select HS, XT or LP oscillator:
1 = OST has counted 1024 clocks, device is clocked by the FOSC<2:0> bit selection
0 = OST is counting, device is clocked from the internal oscillator (INTOSC) selected by the IRCF<3:0> bits.
For all other FOSC<2:0> bit selections:
See Table 5-2, "OSTS Bit Definition".

bit 4 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = HFINTOSC is ready
0 = HFINTOSC is not ready

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

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit
1 = LFINTOSC is ready
0 = LFINTOSC is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit
1 = HFINTOSC 16 MHz Oscillator is stable and is driving the INTOSC
0 = HFINTOSC 16 MHz is not stable, the Start-up Oscillator is driving INTOSC

REGISTER 10-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q[2]	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—[1]	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HC = Bit is cleared by hardware

bit 7 **Unimplemented:** Read as ‘1’

bit 6 **CFGS:** Configuration Select bit
- 1 = Access Configuration, User ID and Device ID Registers
- 0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit[3]
- 1 = Only the addressed program memory write latch is loaded/updated on the next WR command
- 0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
- 1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
- 0 = Performs a write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write ‘1’) of the WR bit).
- 0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash program/erase operation.
  The operation is self-timed and the bit is cleared by hardware once operation is complete.
  The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash is complete and inactive.

bit 0 **RD:** Read Control bit
- 1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash read.

Note 1: Unimplemented bit, read as ‘1’.
2: The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

11.0 I/O PORTS

Each port has three standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

TABLE 11-1: PORT AVAILABILITY PER DEVICE

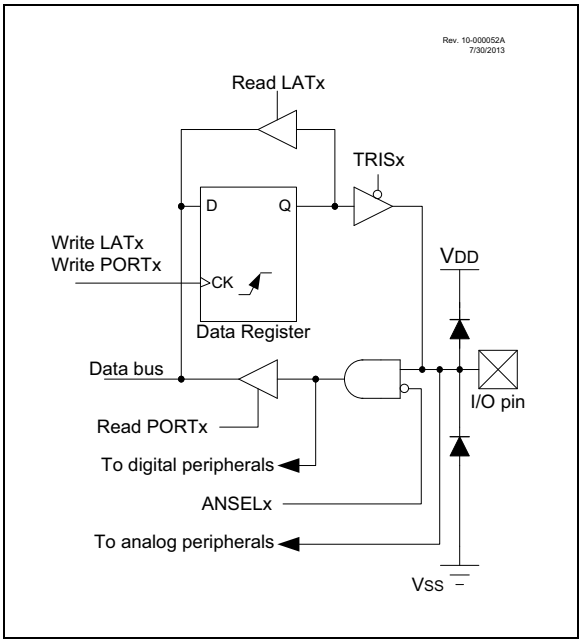
Device	PORTA	PORTB	PORTC
PIC16(L)F1508/9	●	●	●
PIC16(L)F1508/9	●	●	●

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



REGISTER 11-15: ANSELC: PORTC ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0
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 7-6 **ANSC<7:6>**: Analog Select between Analog or Digital Function on pins RC<7:6>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **ANSC<3:0>**: Analog Select between Analog or Digital Function on pins RC<3:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

TABLE 11-9: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	118
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	117
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	117
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	117

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTC.

REGISTER 15-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES<9:8>	
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 7-2 **Reserved**: Do not use.

bit 1-0 **ADRES<9:8>**: ADC Result Register bits
Upper two bits of 10-bit conversion result

REGISTER 15-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
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 7-0 **ADRES<7:0>**: ADC Result Register bits
Lower eight bits of 10-bit conversion result

REGISTER 17-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	CxNCH<2:0>		
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 **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit

bit 6 **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit

bit 5-4 **CxPCH<1:0>:** Comparator Positive Input Channel Select bits
11 = CxVP connects to VSS
10 = CxVP connects to FVR Voltage Reference
01 = CxVP connects to DAC Voltage Reference
00 = CxVP connects to CxIN+ pin

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

bit 2-0 **CxNCH<2:0>:** Comparator Negative Input Channel Select bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = CxVN connects to FVR Voltage reference
011 = CxVN connects to CxIN3- pin
010 = CxVN connects to CxIN2- pin
001 = CxVN connects to CxIN1- pin
000 = CxVN connects to CxIN0- pin

REGISTER 17-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
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 7-2 **Unimplemented:** Read as '0'

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

21.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCKx. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCKx pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCKx pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCKx pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

21.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 21-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

21.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SSx}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100).

When the $\overline{SSx}$ pin is low, transmission and reception are enabled and the SDOx pin is driven.

When the $\overline{SSx}$ pin goes high, the SDOx pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SSx}$ pin is set to VDD.
2:	When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SSx}$ pin control.
3:	While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SSx}$ pin to a high level or clearing the SSPEN bit.

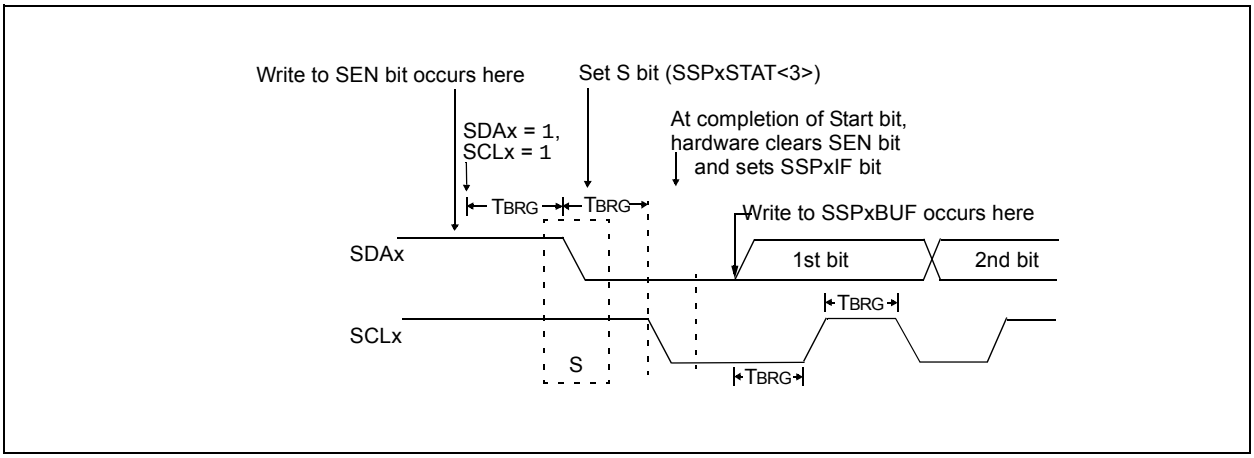
21.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 21-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C Specification states that a bus collision cannot occur on a Start.

FIGURE 21-26: FIRST START BIT TIMING



22.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH mark state which represents a '1' data bit, and a VOL space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 22-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

22.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 22-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

22.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note:	The TXIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

22.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

22.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section22.5.1.2 "Clock Polarity"**.

22.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

REGISTER 22-3: BAUDCON: BAUD RATE CONTROL REGISTER

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is idle
0 = Start bit has been received and the receiver is receiving
Synchronous mode:
Don't care

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

bit 4 **SCKP**: Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Transmit inverted data to the TX/CK pin
0 = Transmit non-inverted data to the TX/CK pin
Synchronous mode:
1 = Data is clocked on rising edge of the clock
0 = Data is clocked on falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used
0 = 8-bit Baud Rate Generator is used

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received, RCIF bit will be set. WUE will automatically clear after RCIF is set.
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care

22.5 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

22.5.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for synchronous master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART.

22.5.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

22.5.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCON register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

22.5.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

22.5.1.4 Synchronous Master Transmission Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section22.4 “EUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

24.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

24.1.1 DATA SELECTION

There are 16 signals available as inputs to the configurable logic. Four 8-input multiplexers are used to select the inputs to pass on to the next stage. The 16 inputs to the multiplexers are arranged in groups of four. Each group is available to two of the four multiplexers, in each case, paired with a different group. This arrangement makes possible selection of up to two from a group without precluding a selection from another group.

Data selection is through four multiplexers as indicated on the left side of Figure 24-2. Data inputs in the figure are identified by a generic numbered input name.

Table 24-1 correlates the generic input name to the actual signal for each CLC module. The columns labeled lcxd1 through lcxd4 indicate the MUX output for the selected data input. D1S through D4S are abbreviations for the MUX select input codes: LCxD1S<2:0> through LCxD4S<2:0>, respectively. Selecting a data input in a column excludes all other inputs in that column.

Data inputs are selected with CLCxSEL0 and CLCxSEL1 registers (Register 24-3 and Register 24-5, respectively).

Note:	Data selections are undefined at power-up.

TABLE 24-1: CLCx DATA INPUT SELECTION

Data Input	lcxd1 D1S	lcxd2 D2S	lcxd3 D3S	lcxd4 D4S	CLC 1	CLC 2	CLC 3	CLC 4
LCx_in[0]	000	—	—	100	CLC1IN0	CLC2IN0	CLC3IN0	CLC4IN0
LCx_in[1]	001	—	—	101	CLC1IN1	CLC2IN1	CLC3IN1	CLC4IN1
LCx_in[2]	010	—	—	110	C1OUT_sync	C1OUT_sync	C1OUT_sync	C1OUT_sync
LCx_in[3]	011	—	—	111	C2OUT_sync	C2OUT_sync	C2OUT_sync	C2OUT_sync
LCx_in[4]	100	000	—	—	FOSC	FOSC	FOSC	FOSC
LCx_in[5]	101	001	—	—	T0_overflow	T0_overflow	T0_overflow	T0_overflow
LCx_in[6]	110	010	—	—	T1_overflow	T1_overflow	T1_overflow	T1_overflow
LCx_in[7]	111	011	—	—	T2_match	T2_match	T2_match	T2_match
LCx_in[8]	—	100	000	—	LC1_out	LC1_out	LC1_out	LC1_out
LCx_in[9]	—	101	001	—	LC2_out	LC2_out	LC2_out	LC2_out
LCx_in[10]	—	110	010	—	LC3_out	LC3_out	LC3_out	LC3_out
LCx_in[11]	—	111	011	—	LC4_out	LC4_out	LC4_out	LC4_out
LCx_in[12]	—	—	100	000	NCO1_out	LFINTOSC	TX_out (EUSART)	SCK_out (MSSP)
LCx_in[13]	—	—	101	001	HFINTOSC	FRC	LFINTOSC	SDO_out (MSSP)
LCx_in[14]	—	—	110	010	PWM3_out	PWM1_out	PWM2_out	PWM1_out
LCx_in[15]	—	—	111	011	PWM4_out	PWM2_out	PWM3_out	PWM4_out

REGISTER 24-2: CLCxPOL: SIGNAL POLARITY CONTROL REGISTER

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
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 **LCxPOL:** LCOUT Polarity Control bit
- 1 = The output of the logic cell is inverted
- 0 = The output of the logic cell is not inverted

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **LCxG4POL:** Gate 4 Output Polarity Control bit
- 1 = The output of gate 4 is inverted when applied to the logic cell
- 0 = The output of gate 4 is not inverted

bit 2 **LCxG3POL:** Gate 3 Output Polarity Control bit
- 1 = The output of gate 3 is inverted when applied to the logic cell
- 0 = The output of gate 3 is not inverted

bit 1 **LCxG2POL:** Gate 2 Output Polarity Control bit
- 1 = The output of gate 2 is inverted when applied to the logic cell
- 0 = The output of gate 2 is not inverted

bit 0 **LCxG1POL:** Gate 1 Output Polarity Control bit
- 1 = The output of gate 1 is inverted when applied to the logic cell
- 0 = The output of gate 1 is not inverted

26.10 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep, provided that the CWG module is enabled, the input source is active, and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, the CPU will go idle during Sleep, but the CWG will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

26.11 Configuring the CWG

The following steps illustrate how to properly configure the CWG to ensure a synchronous start:

1. Ensure that the TRIS control bits corresponding to CWGxA and CWGxB are set so that both are configured as inputs.
2. Clear the GxEN bit, if not already cleared.
3. Set desired dead-band times with the CWGxDBR and CWGxDBF registers.
4. Setup the following controls in CWGxCON2 auto-shutdown register:
   - Select desired shutdown source.
   - Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   - Set the GxASE bit and clear the GxARSEN bit.
5. Select the desired input source using the CWGxCON1 register.
6. Configure the following controls in CWGxCON0 register:
   - Select desired clock source.
   - Select the desired output polarities.
   - Set the output enables for the outputs to be used.
7. Set the GxEN bit.
8. Clear TRIS control bits corresponding to CWGxA and CWGxB to be used to configure those pins as outputs.
9. If auto-restart is to be used, set the GxARSEN bit and the GxASE bit will be cleared automatically. Otherwise, clear the GxASE bit to start the CWG.

26.11.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the GxASDLA and GxASDLB bits of the CWGxCON1 register (Register 26-3). GxASDLA controls the CWG1A override level and GxASDLB controls the CWG1B override level. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not apply to the override level.

26.11.2 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to have resume operation:

- Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 26-5 and Figure 26-6.

26.11.2.1 Software Controlled Restart

When the GxARSEN bit of the CWGxCON2 register is cleared, the CWG must be restarted after an auto-shutdown event by software.

Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the GxASE bit will remain set. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

26.11.2.2 Auto-Restart

When the GxARSEN bit of the CWGxCON2 register is set, the CWG will restart from the auto-shutdown state automatically.

The GxASE bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

29.3 DC Characteristics

TABLE 29-1: SUPPLY VOLTAGE

PIC16LF1508/9			Standard Operating Conditions (unless otherwise stated)				
PIC16F1508/9							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
D001	VDD	**Supply Voltage**					
			VDDMIN		VDDMAX		
			1.8	—	3.6	V	FOSC ≤ 16 MHz
			2.5	—	3.6	V	FOSC ≤ 20 MHz
D001			2.3	—	5.5	V	FOSC ≤ 16 MHz
			2.5	—	5.5	V	FOSC ≤ 20 MHz
D002*	VDR	**RAM Data Retention Voltage(1)**					
			1.5	—	—	V	Device in Sleep mode
D002*			1.7	—	—	V	Device in Sleep mode
D002A*	VPOR	**Power-on Reset Release Voltage(2)**					
			—	1.6	—	V	
D002A*			—	1.6	—	V	
D002B*	VPORR*	**Power-on Reset Rearm Voltage(2)**					
			—	0.8	—	V	
D002B*			—	1.5	—	V	
D003	VFVR	**Fixed Voltage Reference Voltage**					
		1x gain (1.024V nominal)					VDD ≥ 2.5V, -40°C ≤ TA ≤ +85°C
		2x gain (2.048V nominal)	-4	—	+4	%	VDD ≥ 2.5V, -40°C ≤ TA ≤ +85°C
		4x gain (4.096V nominal)	-3	—	+7	%	VDD ≥ 4.75V, -40°C ≤ TA ≤ +85°C
D004*	SVDD	**VDD Rise Rate(2)**	0.05	—	—	V/ms	Ensures that the Power-on Reset signal is released properly.

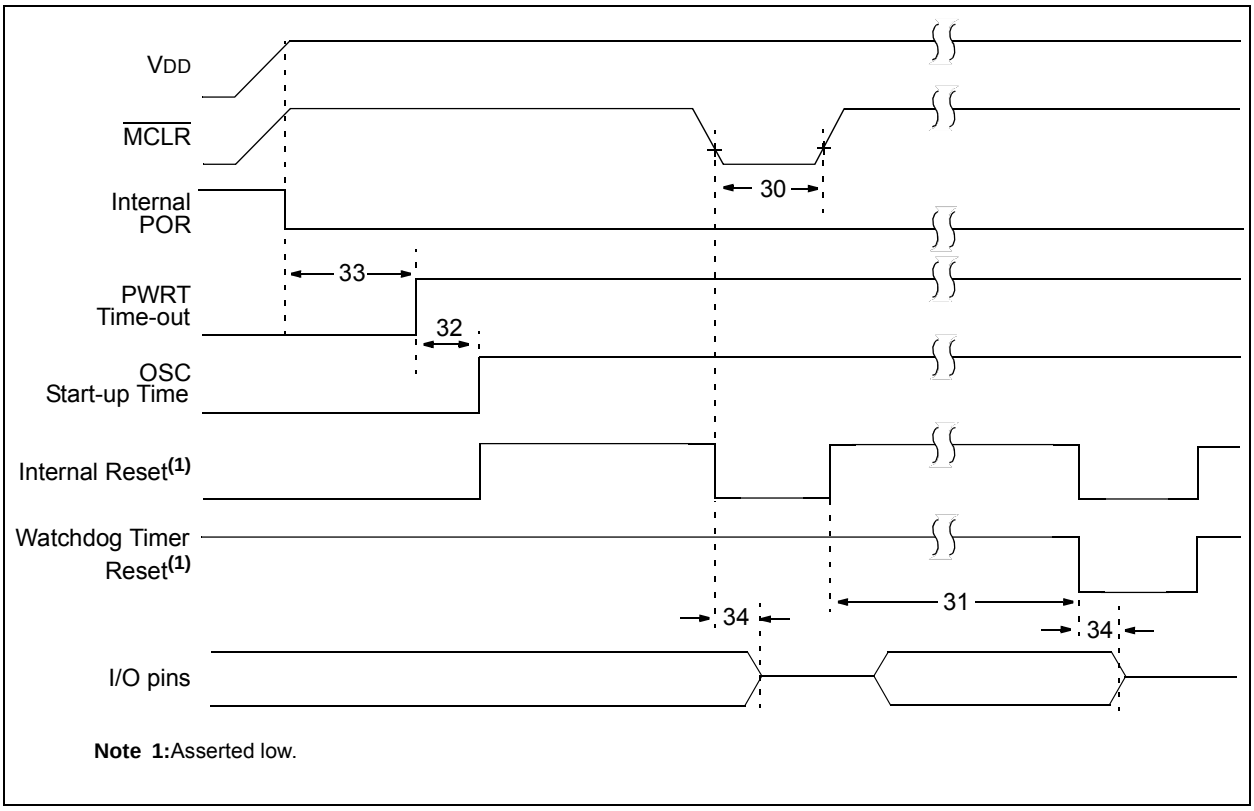
* These parameters are characterized but not tested.

† 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: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: See Figure 29-3, POR and POR REARM with Slow Rising VDD.

FIGURE 29-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

31.0 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

31.1 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