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	40MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1220-i-ss

3.3.1 PRI_IDLE MODE

This mode is unique among the three Low-Power Idle modes, in that it does not disable the primary system clock. For timing sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to "warm up" or transition from another oscillator.

PRI_IDLE mode is entered by setting the IDLEN bit, clearing the SCS bits and executing a SLEEP instruction. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified in Configuration Register 1H. The OSTS bit remains set in PRI_IDLE mode (see Figure 3-3).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of approximately 10 μs is required between the wake event and code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OSTS bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 3-4).

FIGURE 3-3: TRANSITION TIMING TO PRI_IDLE MODE

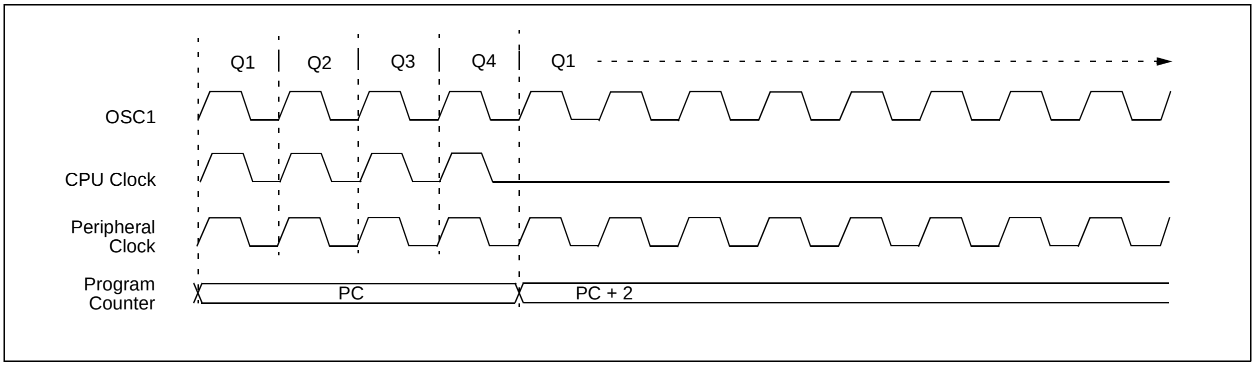


FIGURE 3-4: TRANSITION TIMING FOR WAKE FROM PRI_IDLE MODE

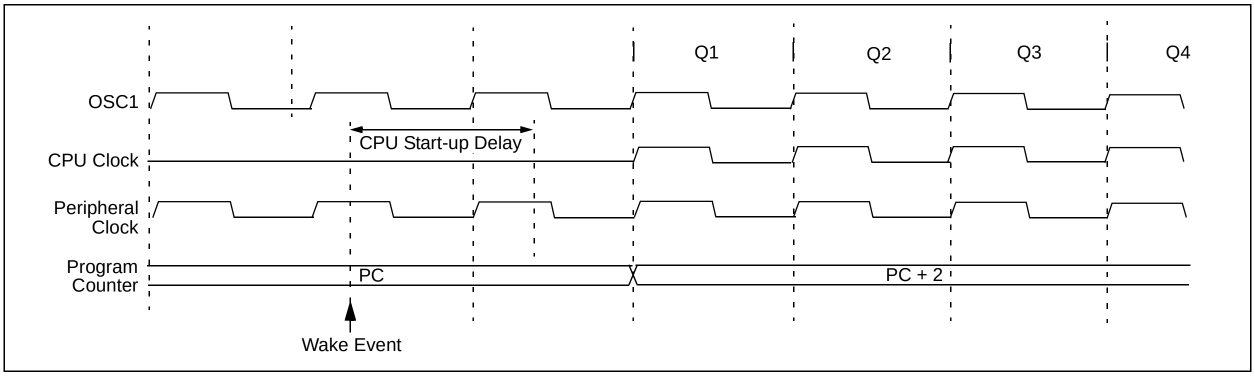


TABLE 3-3: ACTIVITY AND EXIT DELAY ON WAKE FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)

Clock in Power Managed Mode	Primary System Clock	Power Managed Mode Exit Delay	Clock Ready Status Bit (OSCCON)	Activity during Wake-up from Power Managed Mode	
				Exit by Interrupt	Exit by Reset
Primary System Clock (PRI_IDLE mode)	LP, XT, HS	5-10 μs(5)	OSTS	CPU and peripherals clocked by primary clock and executing instructions.	Not clocked or Two-Speed Start-up (if enabled)(3).
	HSPLL				
	EC, RC, INTRC(1)		—		
	INTOSC(2)		IOFS		
T1OSC or INTRC(1)	LP, XT, HS	OST	OSTS	CPU and peripherals clocked by selected power managed mode clock and executing instructions until primary clock source becomes ready.	
	HSPLL	OST + 2 ms			
	EC, RC, INTRC(1)	5-10 μs(5)	—		
	INTOSC(2)	1 ms(4)	IOFS		
INTOSC(2)	LP, XT, HS	OST	OSTS		
	HSPLL	OST + 2 ms			
	EC, RC, INTRC(1)	5-10 μs(5)	—		
	INTOSC(2)	None	IOFS		
Sleep mode	LP, XT, HS	OST	OSTS	Not clocked or Two-Speed Start-up (if enabled) until primary clock source becomes ready(3).	
	HSPLL	OST + 2 ms			
	EC, RC, INTRC(1)	5-10 μs(5)	—		
	INTOSC(2)	1 ms(4)	IOFS		

Note 1: In this instance, refers specifically to the INTRC clock source.

2: Includes both the INTOSC 8 MHz source and postscaler derived frequencies.

3: Two-Speed Start-up is covered in greater detail in **Section 19.3 "Two-Speed Start-up"**.

4: Execution continues during the INTOSC stabilization period.

5: Required delay when waking from Sleep and all Idle modes. This delay runs concurrently with any other required delays (see **Section 3.3 "Idle Modes"**).

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
BSR	1220	1320	---- 0000	---- 0000	---- uuuu
INDF2	1220	1320	N/A	N/A	N/A
POSTINC2	1220	1320	N/A	N/A	N/A
POSTDEC2	1220	1320	N/A	N/A	N/A
PREINC2	1220	1320	N/A	N/A	N/A
PLUSW2	1220	1320	N/A	N/A	N/A
FSR2H	1220	1320	---- 0000	---- 0000	---- uuuu
FSR2L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	1220	1320	---x xxxx	---u uuuu	---u uuuu
TMR0H	1220	1320	0000 0000	0000 0000	uuuu uuuu
TMR0L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	1220	1320	1111 1111	1111 1111	uuuu uuuu
OSCCON	1220	1320	0000 q000	0000 q000	uuuu qquu
LVDCON	1220	1320	--00 0101	--00 0101	--uu uuuu
WDTCON	1220	1320	---- ---0	---- ---0	---- ---u
RCON(4)	1220	1320	0--1 11q0	0--q qquu	u--u qquu
TMR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	1220	1320	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	1220	1320	0000 0000	0000 0000	uuuu uuuu
PR2	1220	1320	1111 1111	1111 1111	1111 1111
T2CON	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADRESH	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1220	1320	00-0 0000	00-0 0000	uu-u uuuu
ADCON1	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADCON2	1220	1320	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
PWM1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
ECCPAS	1220	1320	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

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 4-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the Oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 5 of PORTA is enabled if MCLR is disabled.

5.2 Return Address Stack

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, with the Stack Pointer initialized to 00000B after all Resets. There is no RAM associated with Stack Pointer, 00000B. This is only a Reset value. During a CALL type instruction, causing a push onto the stack, the Stack Pointer is first incremented and the RAM location pointed to by the Stack Pointer (STKPTR) register is written with the contents of the PC (already pointing to the instruction following the CALL). During a RETURN type instruction, causing a pop from the stack, the contents of the RAM location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the top-of-stack Special File Registers. Data can also be pushed to or popped from the stack using the top-of-stack SFRs. Status bits indicate if the stack is full, has overflowed or underflowed.

5.2.1 TOP-OF-STACK ACCESS

The top of the stack is readable and writable. Three register locations, TOSU, TOSH and TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 5-3). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU, TOSH and TOSL registers. These values can be placed on a user defined software stack. At return time, the software can replace the TOSU, TOSH and TOSL and do a return.

The user must disable the Global Interrupt Enable bits while accessing the stack to prevent inadvertent stack corruption.

5.2.2 RETURN STACK POINTER (STKPTR)

The STKPTR register (Register 5-1) contains the Stack Pointer value, the STKFUL (Stack Full) Status bit and the STKUNF (Stack Underflow) Status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. At Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

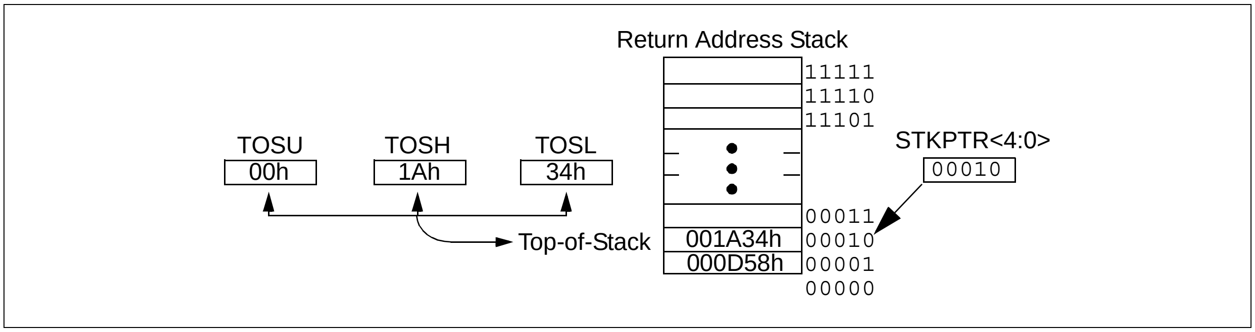
The action that takes place when the stack becomes full depends on the state of the STVR (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 19.1 “Configuration Bits”** for a description of the device Configuration bits.) If STVR is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVR is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and sets the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or a POR occurs.

Note:	Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

FIGURE 5-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
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 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

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

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 9-9: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1/1	U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	U-0
OSCFIP	—	—	EEIP	—	LVDIP	TMR3IP	—
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 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **LVDIP:** Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **Unimplemented:** Read as ‘0’

FIGURE 10-6: MCLR/VPP/RA5 PIN BLOCK DIAGRAM

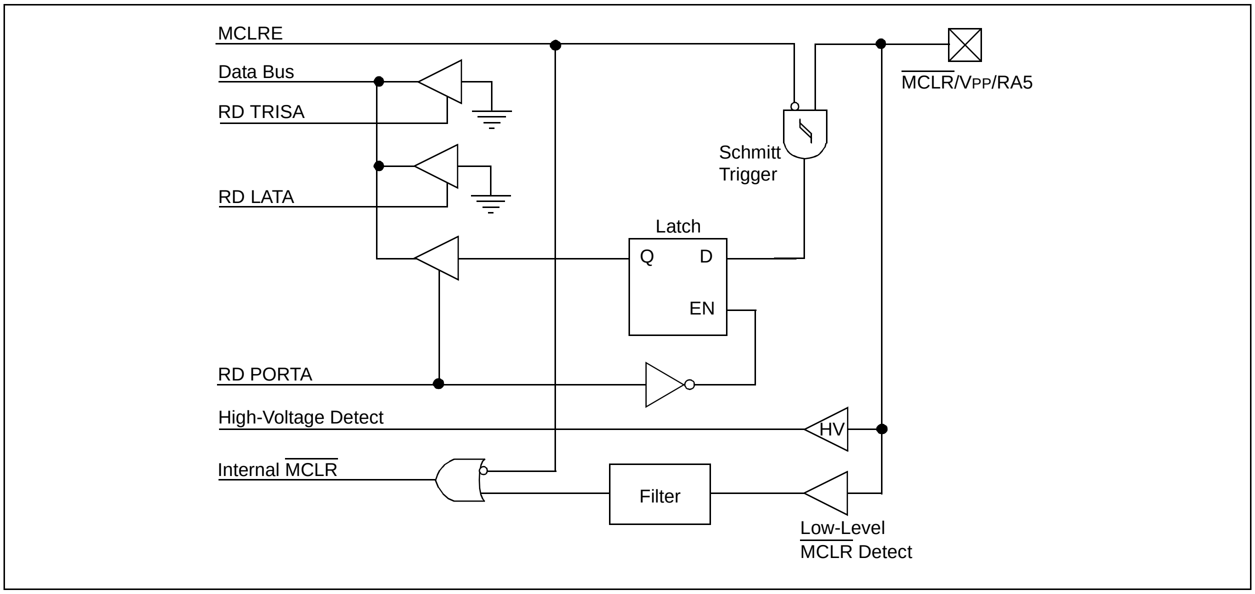


TABLE 10-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	ST	Input/output port pin or analog input.
RA1/AN1/LVDIN	bit 1	ST	Input/output port pin, analog input or Low-Voltage Detect input.
RA2/AN2/VREF-	bit 2	ST	Input/output port pin, analog input or VREF-.
RA3/AN3/VREF+	bit 3	ST	Input/output port pin, analog input or VREF+.
RA4/T0CKI	bit 4	ST	Input/output port pin or external clock input for Timer0. Output is open-drain type.
MCLR/VPP/RA5	bit 5	ST	Master Clear input or programming voltage input (if MCLR is enabled); input only port pin or programming voltage input (if MCLR is disabled).
OSC2/CLKO/RA6	bit 6	ST	OSC2, clock output or I/O pin.
OSC1/CLKI/RA7	bit 7	ST	OSC1, clock input or I/O pin.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 10-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

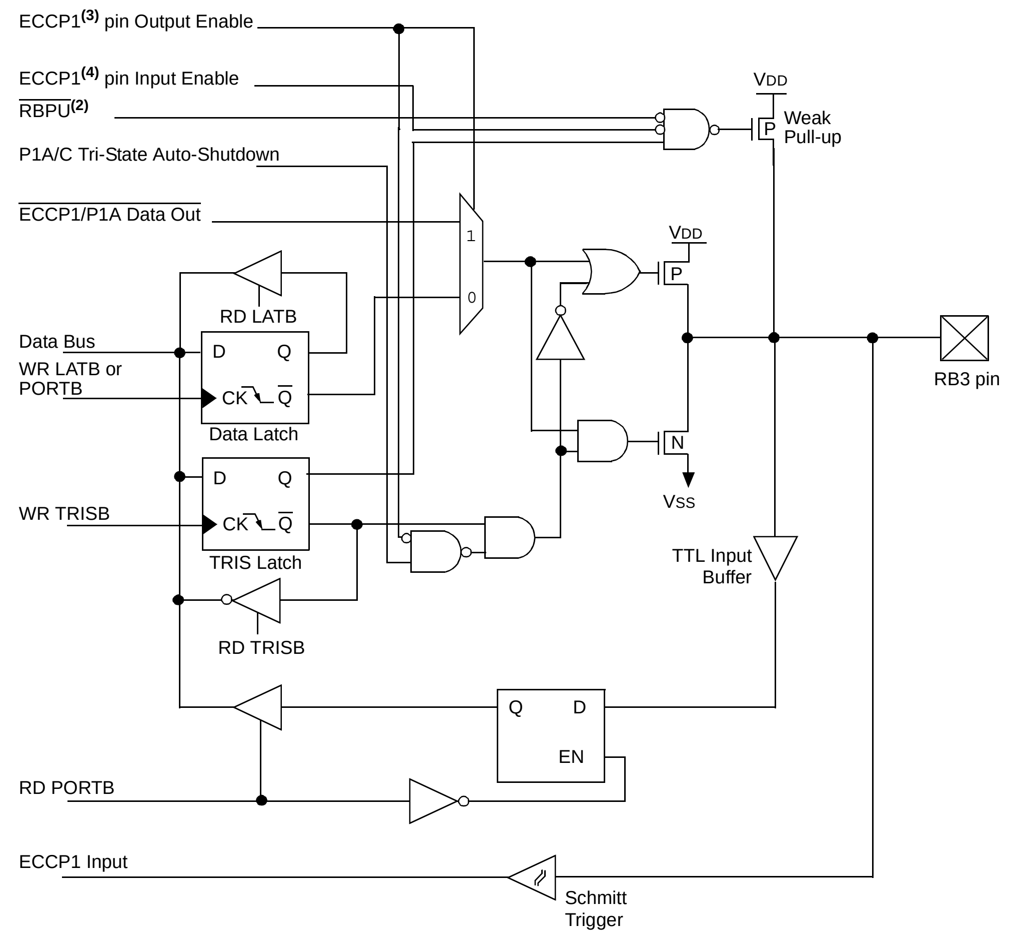
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
PORTA	RA7(1)	RA6(1)	RA5(2)	RA4	RA3	RA2	RA1	RA0	xx0x 0000	uu0u 0000
LATA	LATA7(1)	LATA6(1)	—	LATA Data Output Register					xx-x xxxx	uu-u uuuu
TRISA	TRISA7(1)	TRISA6(1)	—	PORTA Data Direction Register					11-1 1111	11-1 1111
ADCON1	—	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	-000 0000	-000 0000

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

Note 1: RA7:RA6 and their associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.

2: RA5 is an input only if MCLR is disabled.

FIGURE 10-10: BLOCK DIAGRAM OF RB3/CCP1/P1A PIN



Note 1: I/O pins have diode protection to VDD and VSS.

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (INTCON2<7>).

3: ECCP1 pin output enable active for any PWM mode and Compare mode, where CCP1M<3:0> = `1000` or `1001`.

4: ECCP1 pin input enable active for Capture mode only.

12.0 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
- Status of system clock operation

Figure 12-1 is a simplified block diagram of the Timer1 module.

Register 12-1 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>).

The Timer1 oscillator can be used as a secondary clock source in power managed modes. When the T1RUN bit is set, the Timer1 oscillator is providing the system clock. If the Fail-Safe Clock Monitor is enabled and the Timer1 oscillator fails while providing the system clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications, with only a minimal addition of external components and code overhead.

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0/0	R-1/1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
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 **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 **T1RUN:** Timer1 System Clock Status bit
1 = System clock is derived from Timer1 oscillator
0 = System clock is derived from another source

bit 5-4 **T1CKPS<1:0>:** 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 **$\overline{\text{T1SYNC}}$:** 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 RB6/PGC/T1OSO/T13CKI/P1C/KBI2 (on the rising edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

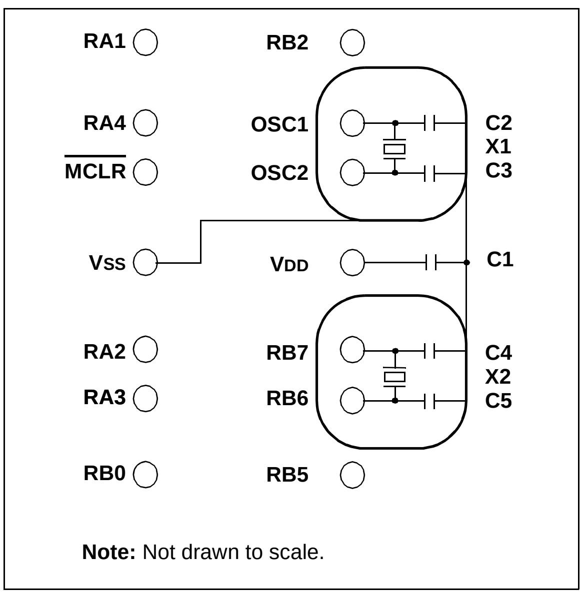
12.3 Timer1 Oscillator Layout Considerations

The Timer1 oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity.

The oscillator circuit, shown in Figure 12-3, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than VSS or VDD.

If a high-speed circuit must be located near the oscillator (such as the CCP1 pin in output compare or PWM mode, or the primary oscillator using the OSC2 pin), a grounded guard ring around the oscillator circuit, as shown in Figure 12-4, may be helpful when used on a single sided PCB, or in addition to a ground plane.

FIGURE 12-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING



12.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing Timer1 Interrupt Enable bit, TMR1IE (PIE1<0>).

12.5 Resetting Timer1 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = `1011`), this signal will reset Timer1 and start an A/D conversion, if the A/D module is enabled (see **Section 15.4.4 “Special Event Trigger”** for more information).

Note:	The special event triggers from the CCP1 module will not set interrupt flag bit, TMR1IF (PIR1<0>).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

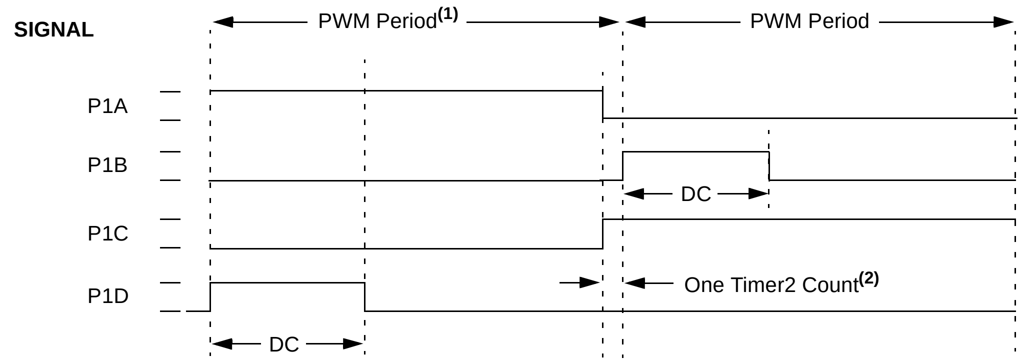
12.6 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 high byte buffer. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, is valid, due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

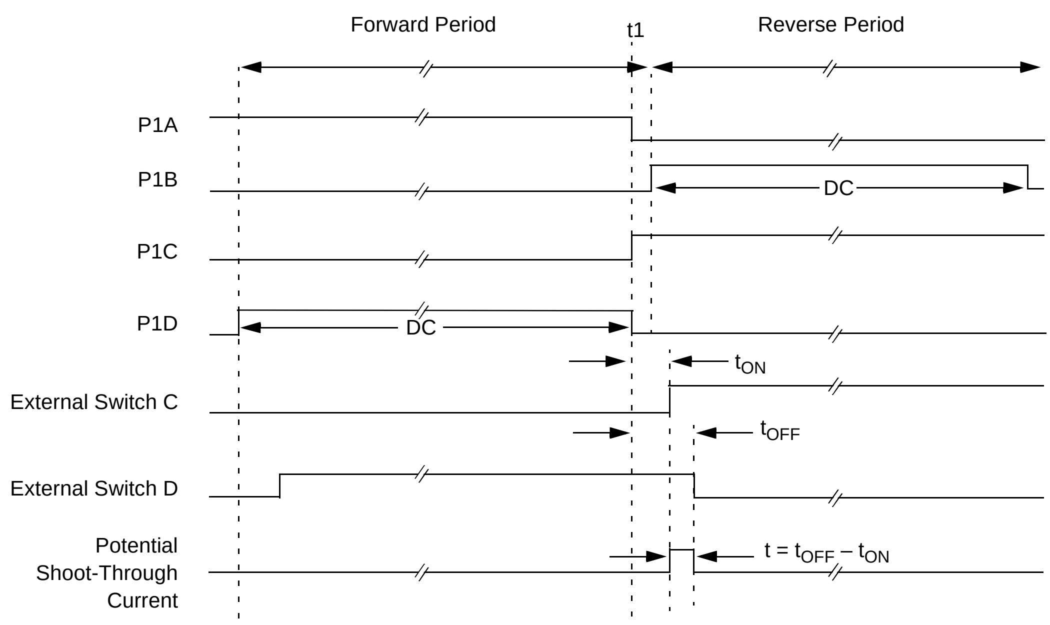
The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

FIGURE 15-10: PWM DIRECTION CHANGE (ACTIVE-HIGH)



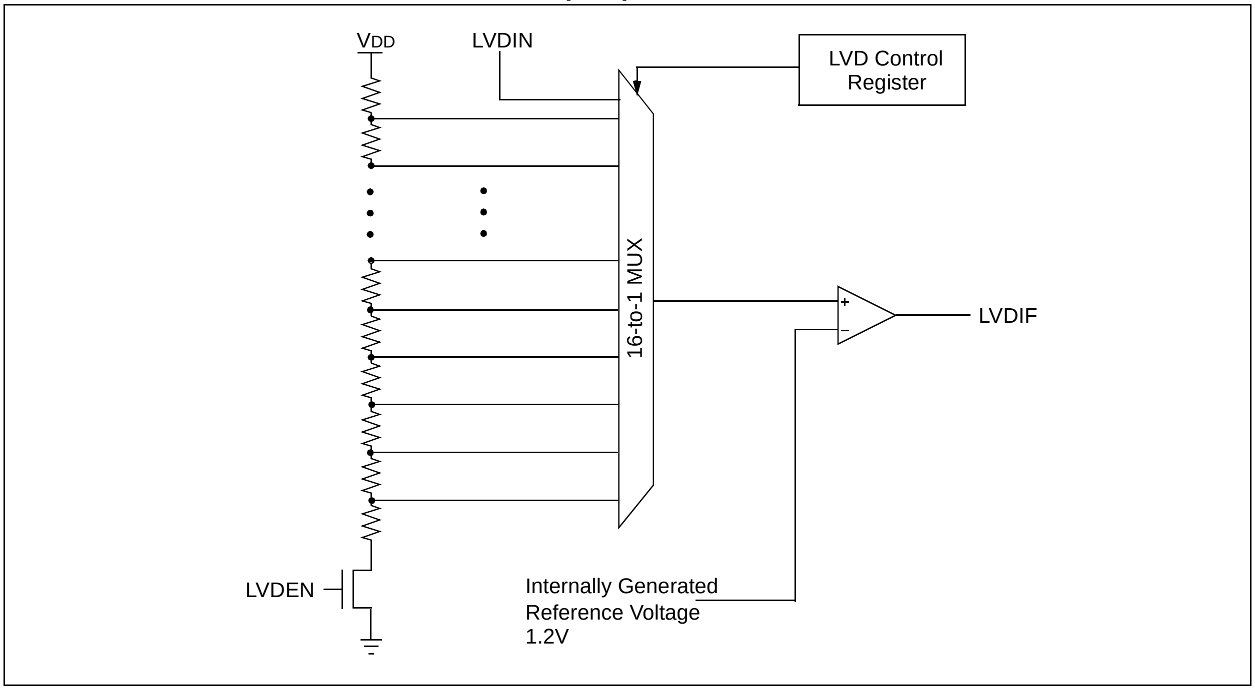
Note 1: The direction bit in the CCP1 Control register (CCP1CON<7>) is written any time during the PWM cycle.
2: When changing directions, the P1A and P1C toggle one Timer2 count before the end of the current PWM cycle. The modulated P1B and P1D signals are inactive at this time.

FIGURE 15-11: PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE (ACTIVE-HIGH)



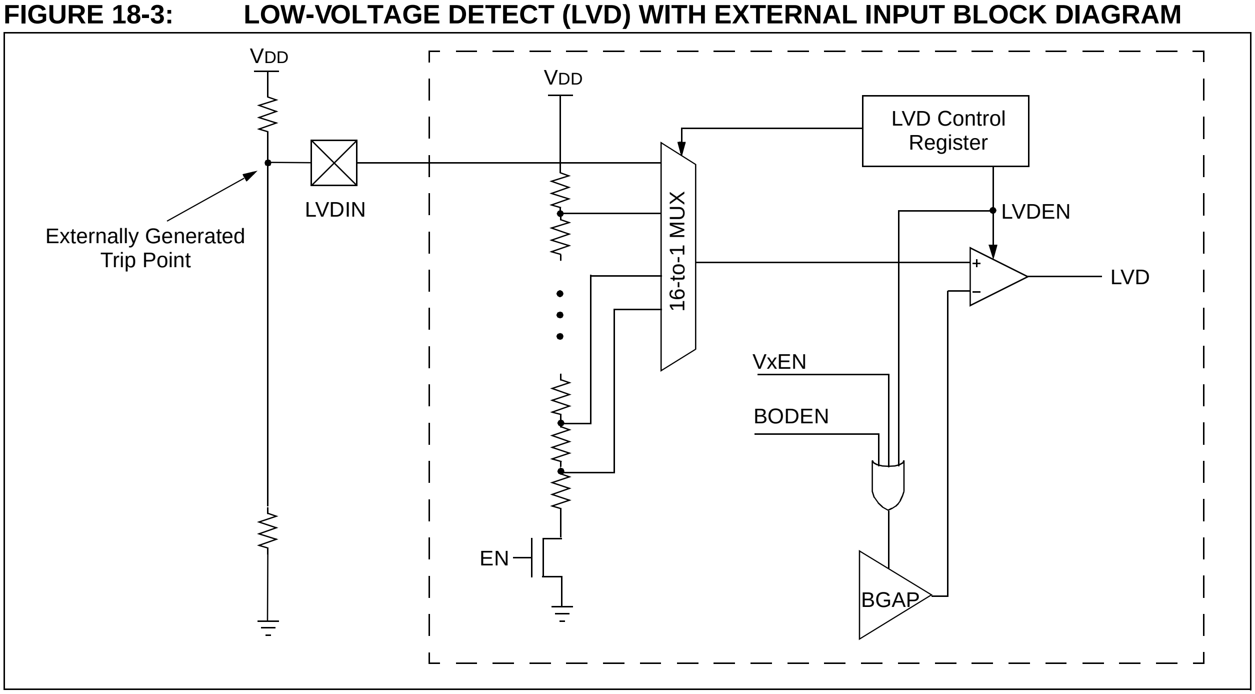
Note 1: t_{ON} is the turn-on delay of power switch QC and its driver.
2: t_{OFF} is the turn-off delay of power switch QD and its driver.

FIGURE 18-2: LOW-VOLTAGE DETECT (LVD) BLOCK DIAGRAM



The LVD module has an additional feature that allows the user to supply the trip voltage to the module from an external source. This mode is enabled when bits, LVDL3:LVDL0, are set to '`1111`'. In this state, the comparator input is multiplexed from the external input pin, LVDIN (Figure 18-3). This gives users flexibility, because it allows them to configure the Low-Voltage Detect interrupt to occur at any voltage in the valid operating range.

FIGURE 18-3: LOW-VOLTAGE DETECT (LVD) WITH EXTERNAL INPUT BLOCK DIAGRAM



18.1 Control Register

The Low-Voltage Detect Control register controls the operation of the Low-Voltage Detect circuitry.

REGISTER 18-1: LVDCON: LOW-VOLTAGE DETECT CONTROL REGISTER

U-0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0
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 **Unimplemented:** Read as ‘0’

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the Low-Voltage Detect logic will generate the interrupt flag at the specified voltage range
0 = Indicates that the Low-Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the LVD interrupt should not be enabled

bit 4 **LVDEN:** Low-Voltage Detect Power Enable bit
1 = Enables LVD, powers up LVD circuit
0 = Disables LVD, powers down LVD circuit

bit 3-0 **LVDL<3:0>:** Low-Voltage Detection Limit bits(1)
1111 = External analog input is used (input comes from the LVDIN pin)
1110 = 4.04V-5.15V
1101 = 3.76V-4.79V
1100 = 3.58V-4.56V
1011 = 3.41V-4.34V
1010 = 3.23V-4.11V
1001 = 3.14V-4.00V
1000 = 2.96V-3.77V
0111 = 2.70V-3.43V
0110 = 2.53V-3.21V
0101 = 2.43V-3.10V
0100 = 2.25V-2.86V
0011 = 2.16V-2.75V
0010 = 1.99V-2.53V
0001 = Reserved
0000 = Reserved

Note 1: LVDL<3:0> modes, which result in a trip point below the valid operating voltage of the device, are not tested.

ADDWFC **ADD W and Carry bit to f**

Syntax:	[*label*] ADDWFC f [,d [,a]]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(W) + (f) + (C) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	`0010` `00da` `ffff` `ffff`
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. If 'a' is '0', the Access Bank will be selected. If 'a' is '1', the BSR will not be overridden.
Words:	1
Cycles:	1

Q Cycle Activity:

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

Example: `ADDWFC REG, W`

Before Instruction
- Carry bit = 1
- REG = 0x02
- W = 0x4D

After Instruction
- Carry bit = 0
- REG = 0x02
- W = 0x50

ANDLW **AND literal with W**

Syntax:	[*label*] ANDLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .AND. k → W
Status Affected:	N, Z
Encoding:	`0000` `1011` `kkkk` `kkkk`
Description:	The contents of W are AND'ed with the 8-bit literal 'k'. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: `ANDLW 0x5F`

Before Instruction
- W = 0xA3

After Instruction
- W = 0x03

LFSR — Load FSR

Syntax: [*label*] LFSR f,k

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

Operation: $k \rightarrow FSRf$

Status Affected: None

Encoding:

1110	1110	00ff	k_{11}kkk
1111	0000	$k_7$kkk	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: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: $f \rightarrow dest$

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 'f', the result is placed in W. If 'd' is 'f', 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 W

Example: `MOVF REG, W`

Before Instruction
REG = 0x22
W = 0xFF

After Instruction
REG = 0x22
W = 0x22

TSTFSZ — Test f, skip if 0

Syntax:	[*label*] TSTFSZ f [,a]
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	skip if f = 0
Status Affected:	None
Encoding:	0110 011a ffff ffff
Description:	If 'f' = 0, the next instruction, fetched during the current instruction execution is discarded and a NOP is executed, making this a 2-cycle instruction. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).
Words:	1
Cycles:	1(2) **Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    TSTFSZ CNT
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
If CNT = 0x00,
PC = Address (ZERO)
If CNT ≠ 0x00,
PC = Address (NZERO)

XORLW — Exclusive OR literal with W

Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → W
Status Affected:	N, Z
Encoding:	0000 1010 kkkk kkkk
Description:	The contents of W are XOR'ed with the 8-bit literal 'k'. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: XORLW 0xAF

Before Instruction
W = 0xB5

After Instruction
W = 0x1A

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial) (Continued)

PIC18LF1220/1320 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial
PIC18F1220/1320 (Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended

Param No.	Device	Typ.	Max.	Units	Conditions		
	Supply Current (IDD)(2,3)						
	PIC18LF1220/1320	140	275	μA	-40°C	VDD = 2.0V	FOSC = 4 MHz (**RC_IDLE** mode, Internal oscillator source)
		140	275	μA	+25°C		
		150	275	μA	+85°C		
	PIC18LF1220/1320	220	375	μA	-40°C	VDD = 3.0V	
		220	375	μA	+25°C		
		220	375	μA	+85°C		
	All devices	390	800	μA	-40°C	VDD = 5.0V	
		400	800	μA	+25°C		
		380	800	μA	+85°C		
	Extended devices	410	800	μA	+125°C		
	PIC18LF1220/1320	150	250	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz (**PRI_RUN** mode, EC oscillator)
		150	250	μA	+25°C		
		160	250	μA	+85°C		
	PIC18LF1220/1320	340	350	μA	-40°C	VDD = 3.0V	
		300	350	μA	+25°C		
		280	350	μA	+85°C		
	All devices	0.72	1.0	mA	-40°C	VDD = 5.0V	
		0.63	1.0	mA	+25°C		
		0.58	1.0	mA	+85°C		
	Extended devices	0.53	1.0	mA	+125°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

22.5 High Temperature Operation

This section outlines the specifications for the following devices operating in the high temperature range between -40°C and 150°C.(6)

- PIC18F1220
- PIC18F1320

When the value of any parameter is identical for both the 125°C Extended and the 150°C High Temp. temperature ranges, then that value will be found in the standard specification tables shown earlier in this chapter, under the fields listed for the 125°C Extended temperature range. If the value of any parameter is unique to the 150°C High Temp. temperature range, then it will be listed here, in this section of the data sheet.

If a Silicon Errata exists for the product and it lists a modification to the 125°C Extended temperature range value, one that is also shared at the 150°C High Temp. temperature range, then that modified value will apply to both temperature ranges.

Note 1: Data contained in this section is applicable to the following devices: PIC18F1220 and PIC18F1320.

2: Writes are **not allowed** for Flash program memory above 125°C.

3: All AC timing specifications are increased by 30%.

4: Figure 22-3; The frequency range is decreased to 20 MHz.

5: The temperature range indicator in the catalog part number and device marking is “H” for -40°C to 150°C.

Example: PIC18F1220T-H/SO indicates the device is shipped in a Tape and Reel configuration, in the SOIC package, and is rated for operation from -40°C to 150°C.

6: AEC-Q100 reliability testing for devices intended to operate at 150°C is 1,000 hours. Any design in which the total operating time from 125°C to 150°C will be greater than 1,000 hours is not warranted without prior written approval from Microchip Technology Inc.

TABLE 22-15: ABSOLUTE MAXIMUM RATINGS

Parameter	Source/Sink	Value	Units
Max. Current: VSS	Sink	300	mA
Max. Current: VDD	Source	250	mA
Max. Current: Pin	Sink	20	mA
Max. Current: Pin	Source	15	mA
Max. Port Current: All ports combined	Sink	20	mA
Max. Port Current: All ports combined	Source	15	mA
Max. Junction Temperature		155	°C

Note: Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

FIGURE 23-13: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} PRI_IDLE, EC MODE, +25°C

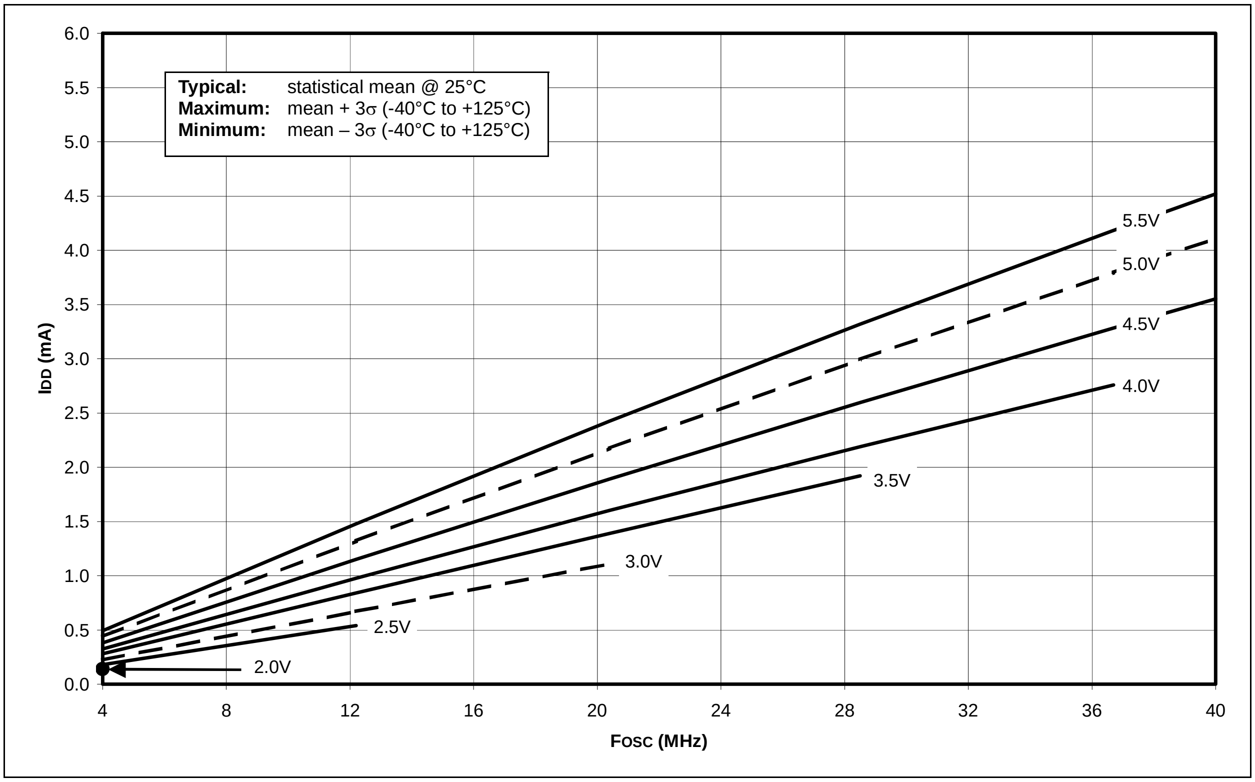
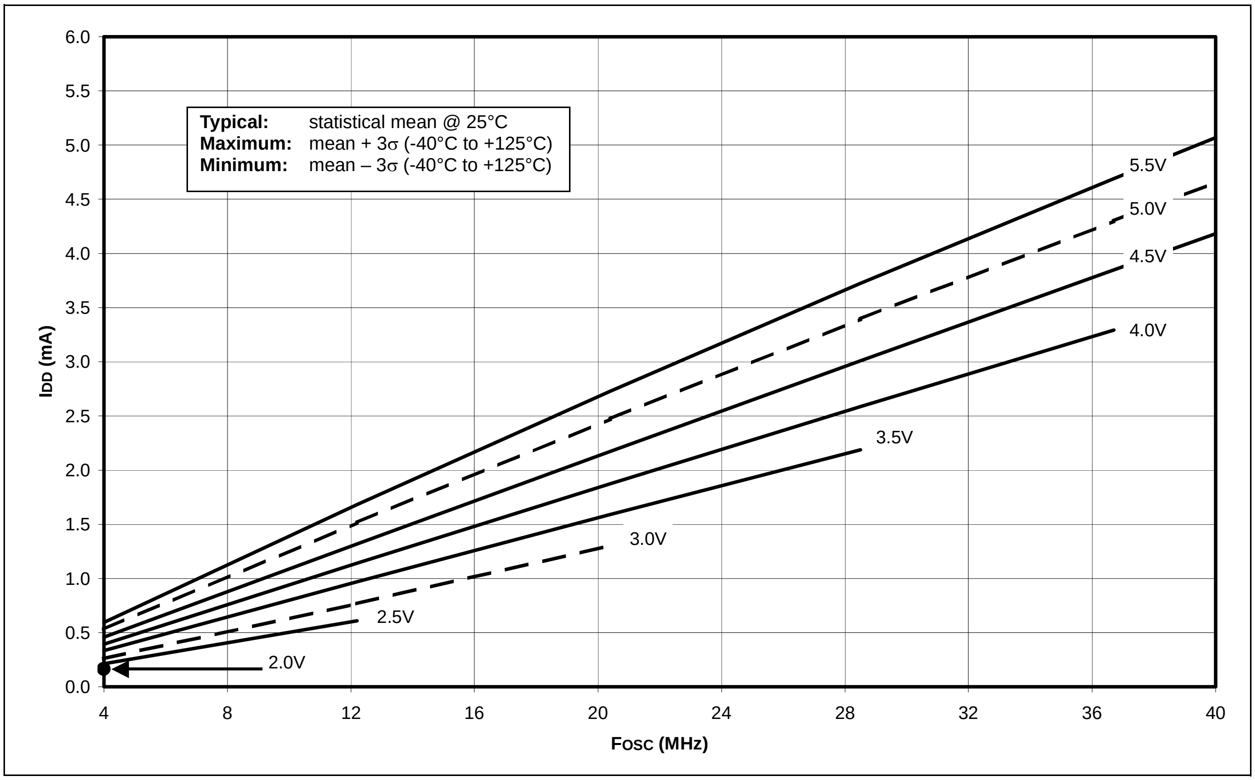


FIGURE 23-14: MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} PRI_IDLE, EC MODE, -40°C TO +125°C



APPENDIX C: CONVERSION CONSIDERATIONS

This appendix discusses the considerations for converting from previous versions of a device to the ones listed in this data sheet. Typically, these changes are due to the differences in the process technology used. An example of this type of conversion is from a PIC16C74A to a PIC16C74B.

Not Applicable

APPENDIX D: MIGRATION FROM BASELINE TO ENHANCED DEVICES

This section discusses how to migrate from a baseline device (i.e., PIC16C5X) to an enhanced MCU device (i.e., PIC18FXXX).

The following are the list of modifications over the PIC16C5X microcontroller family:

Not Currently Available