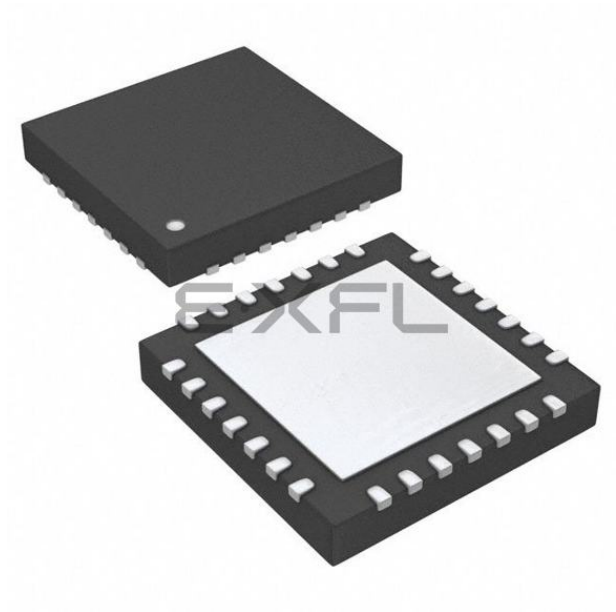


Welcome to E-XFL.COM

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	128 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1230-i-ml

4.4.3 RC_IDLE MODE

In RC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator block using the INTOSC multiplexer. This mode allows for controllable power conservation during Idle periods.

From RC_RUN, this mode is entered by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, first set IDLEN, then set the SCS1 bit and execute SLEEP. Although its value is ignored, it is recommended that SCS0 also be cleared; this is to maintain software compatibility with future devices. The INTOSC multiplexer may be used to select a higher clock frequency by modifying the IRCF bits before executing the SLEEP instruction. When the clock source is switched to the INTOSC multiplexer, the primary oscillator is shut down and the OSTS bit is cleared.

If the IRCF bits are set to any non-zero value, or the INTSRC bit is set, the INTOSC output is enabled. The IOFS bit becomes set, after the INTOSC output becomes stable, after an interval of TIOBST (parameter 39, Table 23-10). Clocks to the peripherals continue while the INTOSC source stabilizes. If the IRCF bits were previously at a non-zero value, or INTSRC was set before the SLEEP instruction was executed and the INTOSC source was already stable, the IOFS bit will remain set. If the IRCF bits and INTSRC are all clear, the INTOSC output will not be enabled, the IOFS bit will remain clear and there will be no indication of the current clock source.

When a wake event occurs, the peripherals continue to be clocked from the INTOSC multiplexer. After a delay of TCSD following the wake event, the CPU begins executing code being clocked by the INTOSC multiplexer. The IDLEN and SCS bits are not affected by the wake-up. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

4.5 Exiting Idle and Sleep Modes

An exit from Sleep mode or any of the Idle modes is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed modes (see **Section 4.2 “Run Modes”, Section 4.3 “Sleep Mode”** and **Section 4.4 “Idle Modes”**).

4.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode or the Sleep mode to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 11.0 “Interrupts”**).

A fixed delay of interval TCSD following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

4.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 4.2 “Run Modes”** and **Section 4.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 20.2 “Watchdog Timer (WDT)”**).

The WDT timer and postscaler are cleared by executing a SLEEP or CLRWDT instruction, the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled) and modifying the IRCF bits in the OSCCON register if the internal oscillator block is the device clock source.

4.5.3 EXIT BY RESET

Normally, the device is held in Reset by the Oscillator Start-up Timer (OST) until the primary clock becomes ready. At that time, the OSTS bit is set and the device begins executing code. If the internal oscillator block is the new clock source, the IOFS bit is set instead.

The exit delay time from Reset to the start of code execution depends on both the clock sources before and after the wake-up and the type of oscillator if the new clock source is the primary clock. Exit delays are summarized in Table 4-2.

Code execution can begin before the primary clock becomes ready. If either the Two-Speed Start-up (see **Section 20.3 “Two-Speed Start-up”**) or Fail-Safe Clock Monitor (see **Section 20.4 “Fail-Safe Clock Monitor”**) is enabled, the device may begin execution as soon as the Reset source has cleared. Execution is clocked by the INTOSC multiplexer driven by the internal oscillator block. Execution is clocked by the internal oscillator block until either the primary clock becomes ready or a power-managed mode is entered before the primary clock becomes ready; the primary clock is then shut down.

5.6 Reset State of Registers

Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a “Reset state” depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register, $\overline{RI}$, $\overline{TO}$, $\overline{PD}$, $\overline{POR}$ and $\overline{BOR}$, are set or cleared differently in different Reset situations, as indicated in Table 5-3. These bits are used in software to determine the nature of the Reset.

Table 5-4 describes the Reset states for all of the Special Function Registers. These are categorized by Power-on and Brown-out Resets, Master Clear and WDT Resets and WDT wake-ups.

TABLE 5-3: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter	RCON Register						STKPTR Register	
		SBOREN	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	STKFUL	STKUNF
Power-on Reset	0000h	1	1	1	1	0	0	0	0
RESET Instruction	0000h	u(2)	0	u	u	u	u	u	u
Brown-out Reset	0000h	u(2)	1	1	1	u	0	u	u
$\overline{MCLR}$ during Power-Managed Run Modes	0000h	u(2)	u	1	u	u	u	u	u
$\overline{MCLR}$ during Power-Managed Idle Modes and Sleep Mode	0000h	u(2)	u	1	0	u	u	u	u
WDT Time-out during Full Power or Power-Managed Run Mode	0000h	u(2)	u	0	u	u	u	u	u
$\overline{MCLR}$ during Full Power Execution	0000h	u(2)	u	u	u	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u(2)	u	u	u	u	u	1	u
Stack Underflow Reset (STVREN = 1)	0000h	u(2)	u	u	u	u	u	u	1
Stack Underflow Error (not an actual Reset, STVREN = 0)	0000h	u(2)	u	u	u	u	u	u	1
WDT Time-out during Power-Managed Idle or Sleep Modes	PC + 2	u(2)	u	0	0	u	u	u	u
Interrupt Exit from Power-Managed Modes	PC + 2(1)	u(2)	u	u	0	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bit is set, the PC is loaded with the interrupt vector (008h or 0018h).

2: Reset state is ‘1’ for POR and unchanged for all other Resets when software BOR is enabled (BOREN1:BOREN0 Configuration bits = 01 and SBOREN = 1); otherwise, the Reset state is ‘0’.

6.1.2.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 6-1) 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. On 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 (RTOS) 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 STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 20.1 "Configuration Bits"** for a description of the device Configuration bits.) If STVREN 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 STVREN 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 until 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.

6.1.2.3 PUSH and POP Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable feature. The PIC18 instruction set includes two instructions, PUSH and POP, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The PUSH instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

REGISTER 6-1: STKPTR: STACK POINTER REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

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

bit 7 **STKFUL:** Stack Full Flag bit(1)
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit(1)
1 = Stack underflow occurred
0 = Stack underflow did not occur

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

bit 4-0 **SP4:SP0:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

6.4.3.1 FSR Registers and the INDF Operand

At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. The four upper bits of the FSRnH register are not used so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as "virtual" registers: they are mapped in the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction's target. The INDF operand is just a convenient way of using the pointer.

Because Indirect Addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are "virtual" registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by 1 afterwards
- POSTINC: accesses the FSR value, then automatically increments it by 1 afterwards
- PREINC: increments the FSR value by 1, then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation.

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by that in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, rollovers of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

FIGURE 6-7: INDIRECT ADDRESSING

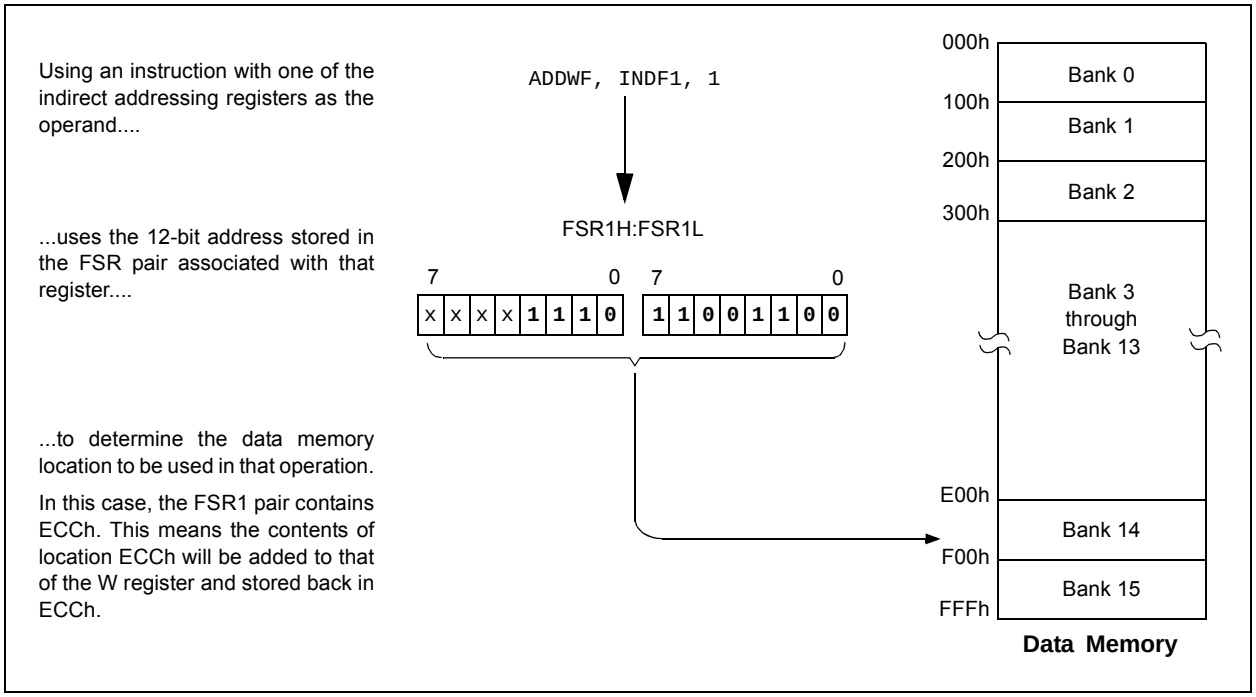


TABLE 10-1: PORTA I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RA0/AN0/INT0/ KBI0/CMP0	RA0	0	O	DIG	LATA<0> data output; not affected by analog input.
		1	I	TTL	PORTA<0> data input; disabled when analog input enabled.
	AN0	1	I	ANA	Analog input 0.
	INT0	1	I	ST	External interrupt 0.
	KBI0	1	I	TTL	Interrupt-on-change pin.
	CMP0	1	I	ANA	Comparator 0 input.
RA1/AN1/INT1/ KBI1	RA1	0	O	DIG	LATA<1> data output; not affected by analog input.
		1	I	TTL	PORTA<1> data input; disabled when analog input enabled.
	AN1	1	I	ANA	Analog input 1.
	INT1	1	I	ST	External interrupt 1.
	KBI1	1	I	TTL	Interrupt-on-change pin.
RA2/TX/CK	RA2	0	O	DIG	LATA<2> data output; not affected by analog input. Disabled when CV_{REF} output enabled.
		1	I	TTL	PORTA<2> data input. Disabled when analog functions enabled; disabled when CV_{REF} output enabled.
	TX	0	0	DIG	EUSART asynchronous transmit.
	CK	0	O	DIG	EUSART synchronous clock.
		1	I	ST	
RA3/RX/DT	RA3	0	O	DIG	LATA<3> data output; not affected by analog input.
		1	I	TTL	PORTA<3> data input; disabled when analog input enabled.
	RX	1	I	ANA	EUSART asynchronous receive.
	DT	0	O	DIG	EUSART synchronous data.
		1	I	TTL	
RA4/T0CKI/AN2/ $V_{REF}+$	RA4	0	O	DIG	LATA<4> data output.
		1	I	ST	PORTA<4> data input; default configuration on POR.
	T0CKI	1	I	ST	Timer0 external clock input.
	AN2	1	I	ANA	Analog input 2.
	$V_{REF}+$	1	I	ANA	A/D reference voltage (high) input.
$\overline{MCLR}/V_{PP}$/RA5/ $\overline{FLTA}$	$\overline{MCLR}$	1	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
	V_{PP}	1	I	ANA	Programming voltage input.
	RA5	1	I	ST	Digital input.
	$\overline{FLTA}$(1)	1	I	ST	Fault detect input for PWM.
RA6/OSC2/CLKO/ T1OSO/T1CKI/AN3	RA6	0	O	DIG	LATA<6> data output. Enabled in RCIO, INTIO2 and ECIO modes only.
		1	I	ST	PORTA<6> data input. Enabled in RCIO, INTIO2 and ECIO modes only.
	OSC2	0	O	ANA	Oscillator crystal output or external clock source output.
	CLKO	0	O	ANA	Oscillator crystal output.
	T1OSO(2)	0	O	ANA	Timer1 oscillator output.
	T1CKI(2)	1	I	ST	Timer1 clock input.
	AN3	1	I	ANA	Analog input 3.
RA7/OSC1/CLKI/ T1OSI/$\overline{FLTA}$	RA7	0	O	DIG	LATA<7> data output. Disabled in external oscillator modes.
		1	I	TTL	PORTA<7> data input. Disabled in external oscillator modes.
	OSC1	1	I	ANA	Oscillator crystal input or external clock source input.
	CLKI	1	I	ANA	External clock source input.
	T1OSI(2)	1	I	ANA	Timer1 oscillator input.
	$\overline{FLTA}$(1)	1	I	ST	Fault detect input for PWM.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Placement of $\overline{FLTA}$ depends on the value of Configuration bit, FLTAMX, of CONFIG3H.

2: Placement of T1OSI and T1OSO/T1CKI depends on the value of Configuration bit, T1OSCMX, of CONFIG3H.

REGISTER 11-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7							bit 0

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

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RBIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

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 11-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	U-0	U-0	R/W-1	U-0	R/W-1	U-0	U-0
OSCFIP	—	—	EEIP	—	LVDIP	—	—
bit 7							bit 0

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

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

REGISTER 11-12: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

U-0	U-0	U-0	R/W-1	U-0	U-0	U-0	U-0
—	—	—	PTIP	—	—	—	—
bit 7							bit 0

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **PTIP:** PWM Time Base Interrupt Priority bit
1 = High priority
0 = Low priority

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

REGISTER 14-3: PWMCON0: PWM CONTROL REGISTER 0

U-0	R/W-1(1)	R/W-1(1)	R/W-1(1)	U-0	R/W-0	R/W-0	R/W-0
—	PWMEN2	PWMEN1	PWMEN0	—	PMOD2	PMOD1	PMOD0
bit 7							bit 0

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

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

bit 6-4 **PWMEN2:PWMEN0:** PWM Module Enable bits(1)

111 = All odd PWM I/O pins enabled for PWM output
110 = PWM1, PWM3 pins enabled for PWM output
10x = All PWM I/O pins enabled for PWM output
011 = PWM0, PWM1, PWM2 and PWM3 I/O pins enabled for PWM output
010 = PWM0 and PWM1 pins enabled for PWM output
001 = PWM1 pin is enabled for PWM output
000 = PWM module disabled; all PWM I/O pins are general purpose I/O

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

bit 2-0 **PMOD2:PMOD0:** PWM Output Pair Mode bits

For PMOD0:
1 = PWM I/O pin pair (PWM0, PWM1) is in the Independent mode
0 = PWM I/O pin pair (PWM0, PWM1) is in the Complementary mode

For PMOD1:
1 = PWM I/O pin pair (PWM2, PWM3) is in the Independent mode
0 = PWM I/O pin pair (PWM2, PWM3) is in the Complementary mode

For PMOD2:
1 = PWM I/O pin pair (PWM4, PWM5) is in the Independent mode
0 = PWM I/O pin pair (PWM4, PWM5) is in the Complementary mode

Note 1: Reset condition of PWMEN bits depends on the PWMPIN Configuration bit of CONFIG3L.

14.6 PWM Duty Cycle

PWM duty cycle is defined by the PDCx (PDCxL and PDCxH) registers. There are a total of three PWM Duty Cycle registers for four pairs of PWM channels. The Duty Cycle registers have 14-bit resolution by combining the six LSbs of PDCxH with the 8 bits of PDCxL. PDCx is a double-buffered register used to set the counting period for the PWM time base.

14.6.1 PWM DUTY CYCLE REGISTERS

There are three 14-bit Special Function Registers used to specify duty cycle values for the PWM module:

- PDC0 (PDC0L and PDC0H)
- PDC1 (PDC1L and PDC1H)
- PDC2 (PDC2L and PDC2H)

The value in each Duty Cycle register determines the amount of time that the PWM output is in the active state. The upper 12 bits of PDCx hold the actual duty cycle value from PTMRH/L<11:0>, while the lower two bits control which internal Q clock the duty cycle match will occur. This 2-bit value is decoded from the Q clocks, as shown in Figure 14-11, when the prescaler is 1:1 (PTCKPS<1:0> = 00).

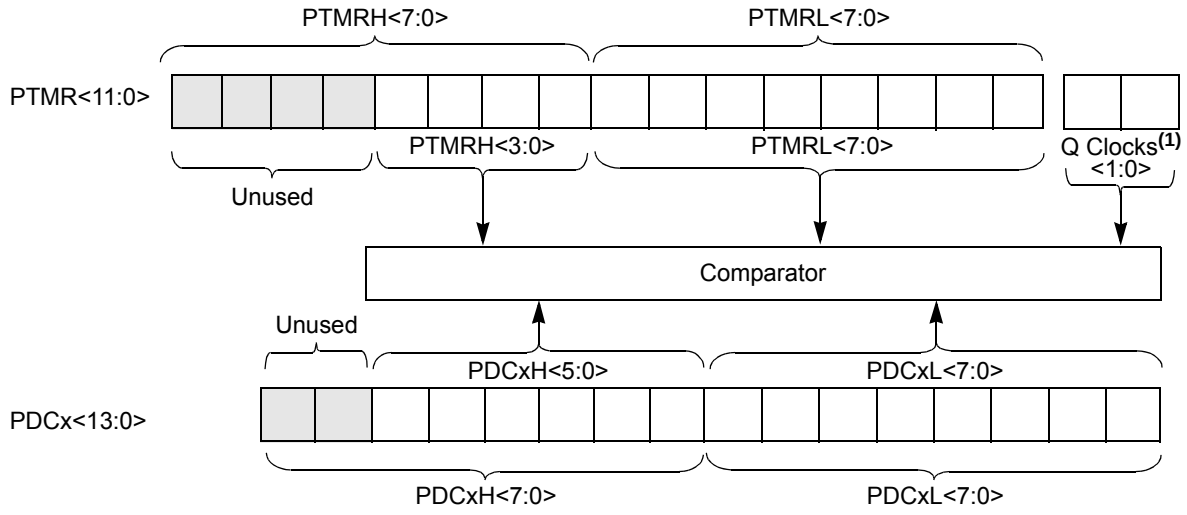
In Edge-Aligned mode, the PWM period starts at Q1 and ends when the Duty Cycle register matches the PTMR register as follows. The duty cycle match is considered when the upper 12 bits of the PDCx are equal to the PTMR and the lower 2 bits are equal to Q1, Q2, Q3 or Q4, depending on the lower two bits of the PDCx (when the prescaler is 1:1 or PTCKPS<1:0> = 00).

Note:	When the prescaler is not 1:1 (PTCKPS<1:0> ≠ ~00), the duty cycle match occurs at the Q1 clock of the instruction cycle when the PTMR and PDCx match occurs.

Each compare unit has logic that allows override of the PWM signals. This logic also ensures that the PWM signals will complement each other (with dead-time insertion) in Complementary mode (see **Section 14.7 "Dead-Time Generators"**).

Note:	To get the correct PWM duty cycle, always multiply the calculated PWM duty cycle value by four before writing it to the PWM Duty Cycle registers. This is due to the two additional LSBs in the PWM Duty Cycle registers which are compared against the internal Q clock for the PWM duty cycle match.

FIGURE 14-11: DUTY CYCLE COMPARISON



Note 1: This value is decoded from the Q clocks:
00 = duty cycle match occurs on Q1
01 = duty cycle match occurs on Q2
10 = duty cycle match occurs on Q3
11 = duty cycle match occurs on Q4

14.6.2 DUTY CYCLE REGISTER BUFFERS

The three PWM Duty Cycle registers are double-buffered to allow glitchless updates of the PWM outputs. For each duty cycle block, there is a Duty Cycle Buffer register that is accessible by the user and a second Duty Cycle register that holds the actual compare value used in the present PWM period.

In Edge-Aligned PWM Output mode, a new duty cycle value will be updated whenever a PTMR match with the PTPER register occurs and PTMR is reset, as shown in Figure 14-12. Also, the contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers when the PWM time base is disabled (PTEN = 0).

When the PWM time base is in the Continuous Up/Down Count mode, new duty cycle values will be updated when the value of the PTMR register is zero and the PWM time base begins to count upwards. The contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers when the PWM time base is disabled (PTEN = 0). Figure 14-13 shows the timings when the duty cycle update occurs for the Continuous Up/Down Count mode. In this mode, up to one entire PWM period is available for calculating and loading the new PWM duty cycle before changes take effect.

When the PWM time base is in the Continuous Up/Down Count mode with double updates, new duty cycle values will be updated when the value of the PTMR register is zero and when the value of the PTMR register matches the value in the PTPER register. The contents of the duty cycle buffers are automatically loaded into the Duty Cycle registers during both of the previously described conditions. Figure 14-14 shows the duty cycle updates for Continuous Up/Down Count mode with double updates. In this mode, up to half of a PWM period is available for calculating and loading the new PWM duty cycle before changes take effect.

14.6.3 EDGE-ALIGNED PWM

Edge-aligned PWM signals are produced by the module when the PWM time base is in the Free-Running mode or the Single-Shot mode. For edge-aligned PWM outputs, the output for a given PWM channel has a period specified by the value loaded in PTPER and a duty cycle specified by the appropriate Duty Cycle register (see Figure 14-12). The PWM output is driven active at the beginning of the period (PTMR = 0) and is driven inactive when the value in the Duty Cycle register matches PTMR. A new cycle is started when PTMR matches the PTPER, as explained in the PWM period section.

If the value in a particular Duty Cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the Duty Cycle register is greater than the value held in the PTPER register.

FIGURE 14-12: EDGE-ALIGNED PWM

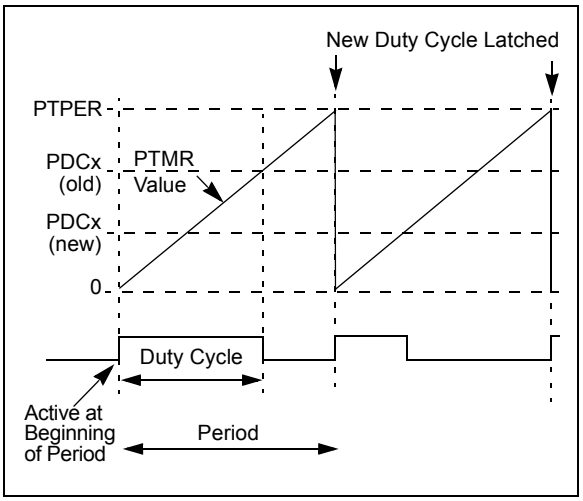


FIGURE 14-13: DUTY CYCLE UPDATE TIMES IN CONTINUOUS UP/DOWN COUNT MODE

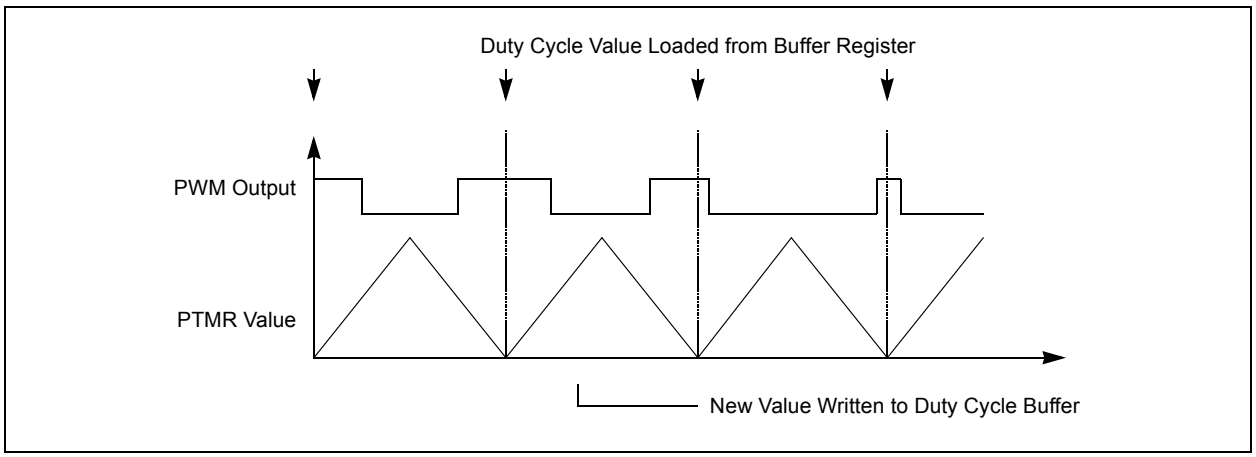
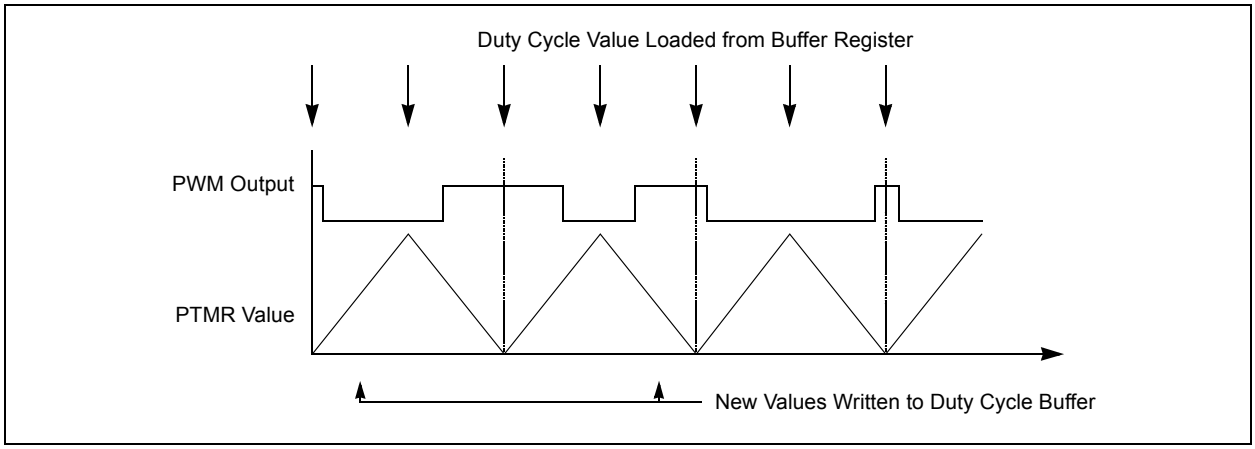


FIGURE 14-14: DUTY CYCLE UPDATE TIMES IN CONTINUOUS UP/DOWN COUNT MODE WITH DOUBLE UPDATES



14.6.4 CENTER-ALIGNED PWM

Center-aligned PWM signals are produced by the module when the PWM time base is configured in a Continuous Up/Down Count mode (see Figure 14-15). The PWM compare output is driven to the active state when the value of the Duty Cycle register matches the value of PTMR and the PWM time base is counting downwards (PTDIR = 1). The PWM compare output will be driven to the inactive state when the PWM time base is counting upwards (PTDIR = 0) and the value in the PTMR register matches the duty cycle value. If the value in a particular Duty Cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the Duty Cycle register is equal to or greater than the value in the PTPER register.

Note:	When the PWM is started in Center-Aligned mode, the PWM Time Base Period register (PTPER) is loaded into the PWM Time Base register (PTMR) and the PTMR is configured automatically to start down counting. This is done to ensure that all the PWM signals don't start at the same time.

FIGURE 14-15: START OF CENTER-ALIGNED PWM

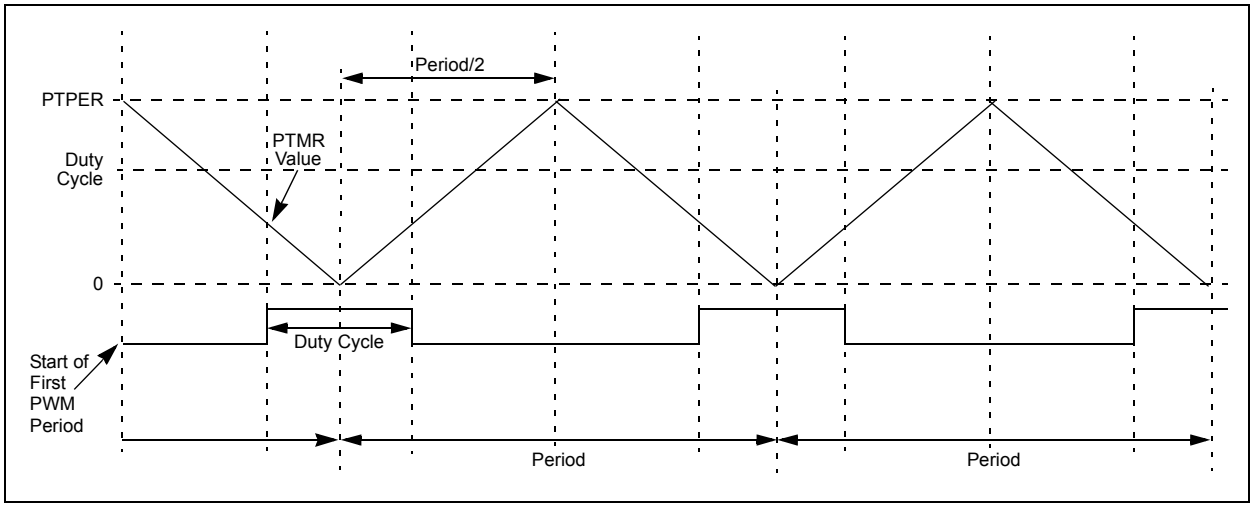


TABLE 15-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

21.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICkit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

21.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

SUBLW **Subtract W from Literal**

Syntax: SUBLW k

Operands: $0 \le k \le 255$

Operation: $k - (W) \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1000	kkkk	kkkk

Description: W is subtracted from the eight-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 1: SUBLW 02h

Before Instruction
W = 01h
C = ?
After Instruction
W = 01h
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBLW 02h

Before Instruction
W = 02h
C = ?
After Instruction
W = 00h
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBLW 02h

Before Instruction
W = 03h
C = ?
After Instruction
W = FFh ; (2's complement)
C = 0 ; result is negative
Z = 0
N = 1

SUBWF **Subtract W from f**

Syntax: SUBWF f {,d {,a}}

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

Operation: $(f) - (W) \rightarrow$ dest

Status Affected: N, OV, C, DC, Z

Encoding:

0101	11da	ffff	ffff

Description: Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'. If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 22.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: SUBWF REG, 1, 0

Before Instruction
REG = 3
W = 2
C = ?
After Instruction
REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2: SUBWF REG, 0, 0

Before Instruction
REG = 2
W = 2
C = ?
After Instruction
REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: SUBWF REG, 1, 0

Before Instruction
REG = 1
W = 2
C = ?
After Instruction
REG = FFh ;(2's complement)
W = 2
C = 0 ; result is negative
Z = 0
N = 1

TABLE 23-2: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +125°C (unless otherwise stated).							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	V_{IOFF}	Input Offset Voltage	—	±5.0	±10	mV	
D301	V_{ICM}	Input Common Mode Voltage	0	—	V_{DD} – 1.5	V	
D302	CMRR	Common Mode Rejection Ratio	55	—	—	dB	
D303	T_{RESP}	Response Time(1)	—	150	400	ns	PIC18**F**XXXX
D303A			—	150	600	ns	PIC18**LF**XXXX, V_{DD} = 2.0V
D304	T_{MC2OV}	Comparator Mode Change to Output Valid	—	—	10	µs	

Note 1: Response time measured with one comparator input at (V_{DD} – 1.5)/2, while the other input transitions from V_{SS} to V_{DD}.

TABLE 23-3: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < V_{DD} < 5.5V, -40°C < T_A < +125°C (unless otherwise stated).							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D310	V_{RES}	Resolution	V_{DD}/24	—	V_{DD}/32	LSb	
D311	V_{RAA}	Absolute Accuracy	—	—	1/2	LSb	
D312	V_{RUR}	Unit Resistor Value (R)	—	2k	—	Ω	
D310	T_{SET}	Settling Time(1)	—	—	10	µs	

Note 1: Settling time measured while CVRR = 1 and CVR3:CVR0 transitions from '0000' to '1111'.

FIGURE 23-4: LOW-VOLTAGE DETECT CHARACTERISTICS

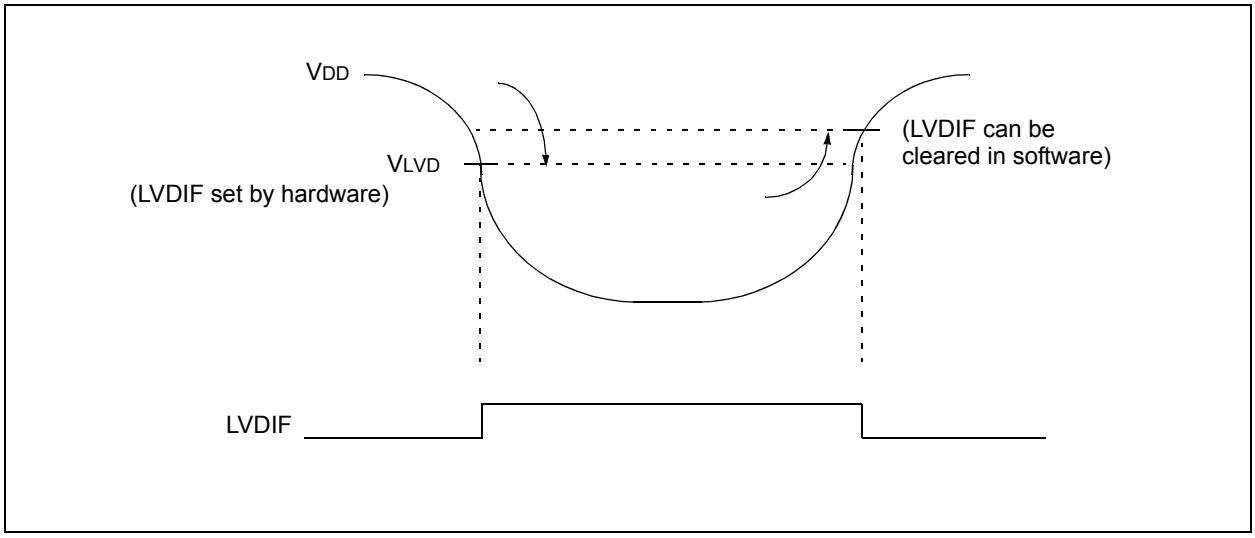


TABLE 23-4: LOW-VOLTAGE DETECT CHARACTERISTICS

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**
D420		LVD Voltage on VDD Transition High-to-Low	LVDL<3:0> = 0000	2.06	2.17	2.28	V	
			LVDL<3:0> = 0001	2.12	2.23	2.34	V	
			LVDL<3:0> = 0010	2.24	2.36	2.48	V	
			LVDL<3:0> = 0011	2.32	2.44	2.56	V	
			LVDL<3:0> = 0100	2.47	2.60	2.73	V	
			LVDL<3:0> = 0101	2.65	2.79	2.93	V	
			LVDL<3:0> = 0110	2.74	2.89	3.04	V	
			LVDL<3:0> = 0111	2.96	3.12	3.28	V	
			LVDL<3:0> = 1000	3.22	3.39	3.56	V	
			LVDL<3:0> = 1001	3.37	3.55	3.73	V	
			LVDL<3:0> = 1010	3.52	3.71	3.90	V	
			LVDL<3:0> = 1011	3.70	3.90	4.10	V	
			LVDL<3:0> = 1100	3.90	4.11	4.32	V	
			LVDL<3:0> = 1101	4.11	4.33	4.55	V	
			LVDL<3:0> = 1110	4.36	4.59	4.82	V	

23.4.2 TIMING CONDITIONS

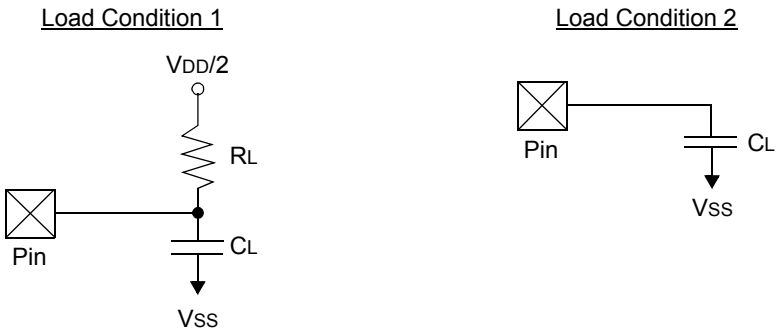
The temperature and voltages specified in Table 23-5 apply to all timing specifications unless otherwise noted. Figure 23-5 specifies the load conditions for the timing specifications.

Note: Because of space limitations, the generic terms "PIC18FXXXX" and "PIC18LFXXXX" are used throughout this section to refer to the PIC18F1230/1330 and PIC18LF1230/1330 families of devices specifically and only those devices.

TABLE 23-5: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended Operating voltage VDD range as described in DC spec **Section 23.1** and **Section 23.3**. LF parts operate for industrial temperatures only.

FIGURE 23-5: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



RL = 464Ω

CL = 50 pF for all pins except OSC2/CLKO and including D and E outputs as ports

TABLE 23-7: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 4.2V TO 5.5V)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	10	MHz	HS mode only
F11	FSYS	On-Chip VCO System Frequency	16	—	40	MHz	HS mode only
F12	t_{rc}	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13	ΔCLK	CLKO Stability (Jitter)	-2	—	+2	%	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 23-8: AC CHARACTERISTICS: INTERNAL RC ACCURACY

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	Min	Typ	Max	Units	Conditions	
	INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, 31 kHz(1)						
	PIC18LF1230/1330	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V
		-5	—	5	%	-10°C to +85°C	VDD = 2.7-3.3V
		-10	+/-1	10	%	-40°C to +85°C	VDD = 2.7-3.3V
	PIC18F1230/1330	-2	+/-1	2	%	+25°C	VDD = 4.5-5.5V
		-5	—	5	%	-10°C to +85°C	VDD = 4.5-5.5V
		-10	+/-1	10	%	-40°C to +85°C	VDD = 4.5-5.5V
	INTRC Accuracy @ Freq = 31 kHz(2,3)						
	PIC18LF1230/1330	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 2.7-3.3V
	PIC18F1230/1330	26.562	—	35.938	kHz	-40°C to +85°C	VDD = 4.5-5.5V

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

Note 1: Frequency calibrated at 25°C. OSCTUNE register can be used to compensate for temperature drift.

2: INTRC frequency after calibration.

3: Change of INTRC frequency as VDD changes.

NOTES: