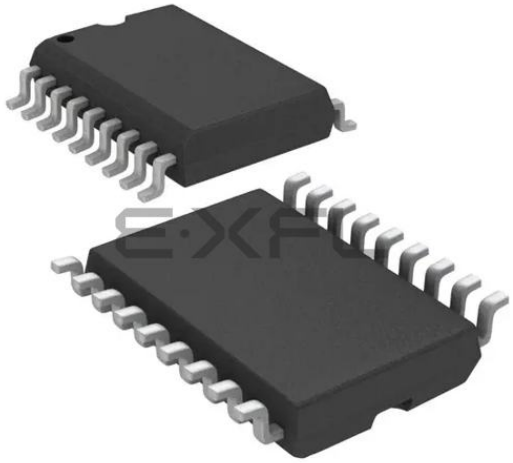


Welcome to E-XFL.COM

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	Obsolete
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	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1230t-i-so

TABLE 1-1: DEVICE FEATURES

Features	PIC18F1230	PIC18F1330
Operating Frequency	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	4096	8192
Program Memory (Instructions)	2048	4096
Data Memory (Bytes)	256	256
Data EEPROM Memory (Bytes)	128	128
Interrupt Sources	17	17
I/O Ports	Ports A, B	Ports A, B
Timers	2	2
Power Control PWM Module	6 Channels	6 Channels
Serial Communications	Enhanced USART	Enhanced USART
10-Bit Analog-to-Digital Module	4 Input Channels	4 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{MCLR}$ (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{MCLR}$ (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes
Programmable Brown-out Reset	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	18-Pin PDIP 18-Pin SOIC 20-Pin SSOP 28-Pin QFN	18-Pin PDIP 18-Pin SOIC 20-Pin SSOP 28-Pin QFN

TABLE 1-2: PIC18F1230/1330 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP, SOIC	SSOP	QFN			
MCLR/VPP/RA5/FLTA	4	4	1			Master Clear (input), programming voltage (input) or Fault detect input.
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				I	Analog	Programming voltage input.
RA5				I	ST	Digital input.
FLTA(1)				I	ST	Fault detect input for PWM.
RA7/OSC1/CLKI/ T1OSI/FLTA	16	18	21			Oscillator crystal, external clock input, Timer1 oscillator input or Fault detect input.
RA7				I/O	ST	Digital I/O.
OSC1				I	Analog	Oscillator crystal input or external clock source input.
CLKI				I	—	External clock source input.
T1OSI(2)				I	Analog	Timer1 oscillator input.
FLTA(1)				I	ST	Fault detect input for PWM.
RA6/OSC2/CLKO/ T1OSO/T1CKI/AN3	15	17	20			Oscillator crystal, clock output, Timer1 oscillator output or analog input.
RA6				I/O	ST	Digital I/O.
OSC2				O	—	Oscillator crystal output or external clock source input.
CLKO				O	—	External clock source output.
T1OSO(2)				O	—	Timer1 oscillator output.
TICKI(2)				I	ST	Timer1 clock input.
AN3				I	Analog	Analog input 3.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Placement of 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.

3.4 RC Oscillator

For timing insensitive applications, the “RC” and “RCIO” device options offer additional cost savings. The actual oscillator frequency is a function of several factors:

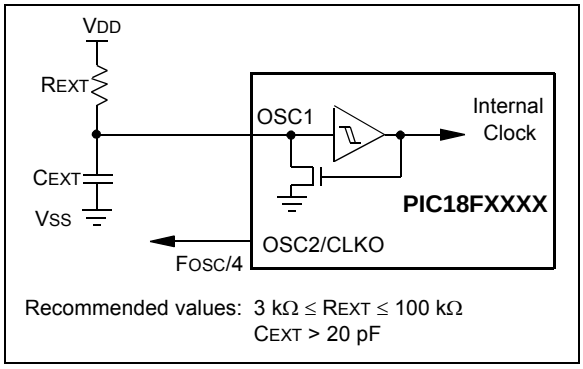
- supply voltage
- values of the external resistor (REXT) and capacitor (CEXT)
- operating temperature

Given the same device, operating voltage and temperature and component values, there will also be unit-to-unit frequency variations. These are due to factors such as:

- normal manufacturing variation
- difference in lead frame capacitance between package types (especially for low CEXT values)
- variations within the tolerance of limits of REXT and CEXT

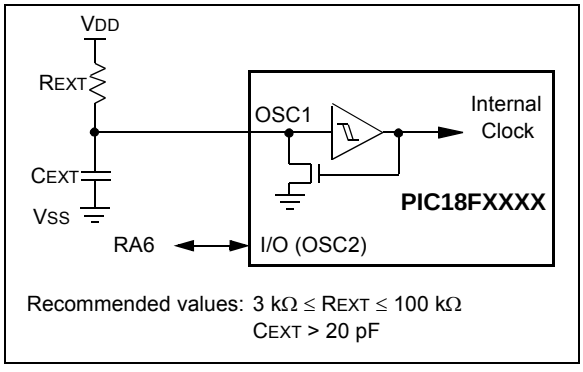
In the RC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 3-5 shows how the R/C combination is connected.

FIGURE 3-5: RC OSCILLATOR MODE



The RCIO Oscillator mode (Figure 3-6) functions like the RC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6).

FIGURE 3-6: RCIO OSCILLATOR MODE



3.5 PLL Frequency Multiplier

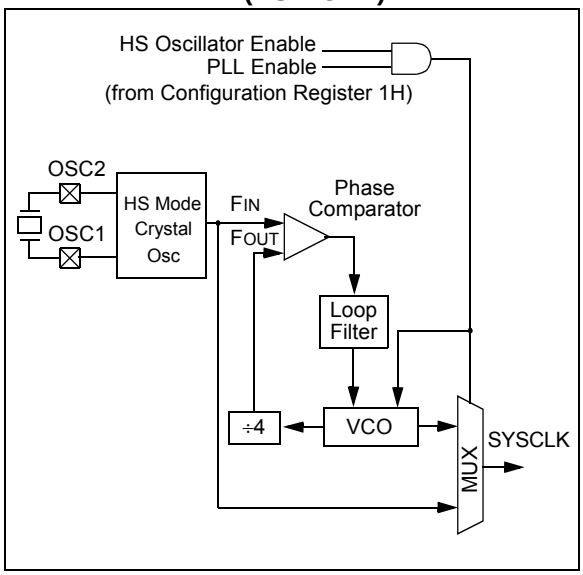
A Phase Locked Loop (PLL) circuit is provided as an option for users who wish to use a lower frequency oscillator circuit or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals or users who require higher clock speeds from an internal oscillator.

3.5.1 HSPLL OSCILLATOR MODE

The HSPLL mode makes use of the HS mode oscillator for frequencies up to 10 MHz. A PLL then multiplies the oscillator output frequency by 4 to produce an internal clock frequency up to 40 MHz. The PLLEN bit is not available in this oscillator mode.

The PLL is only available to the crystal oscillator when the FOSC3:FOSC0 Configuration bits are programmed for HSPLL mode (= 0110).

FIGURE 3-7: PLL BLOCK DIAGRAM (HS MODE)



3.5.2 PLL AND INTOSC

The PLL is also available to the internal oscillator block in selected oscillator modes. In this configuration, the PLL is enabled in software and generates a clock output of up to 32 MHz. The operation of INTOSC with the PLL is described in **Section 3.6.4 “PLL in INTOSC Modes”**.

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.

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

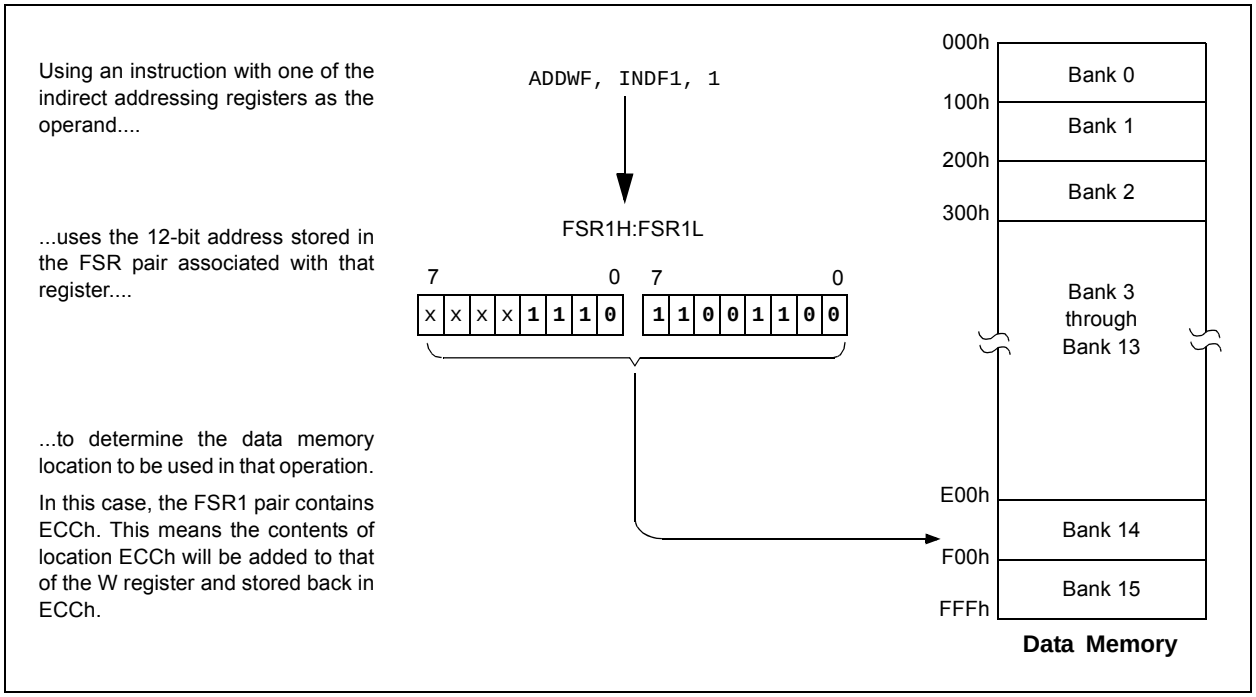
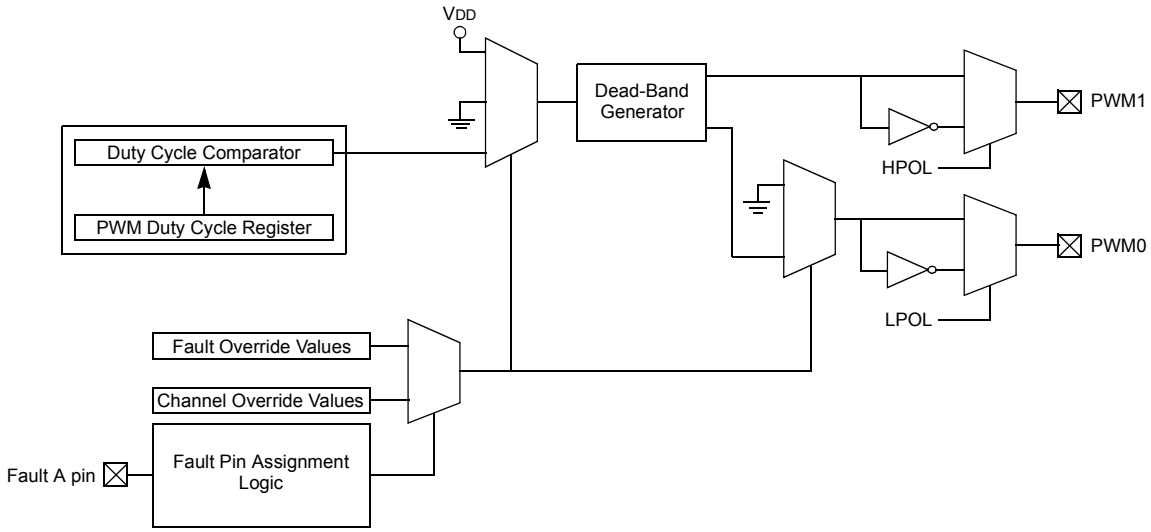
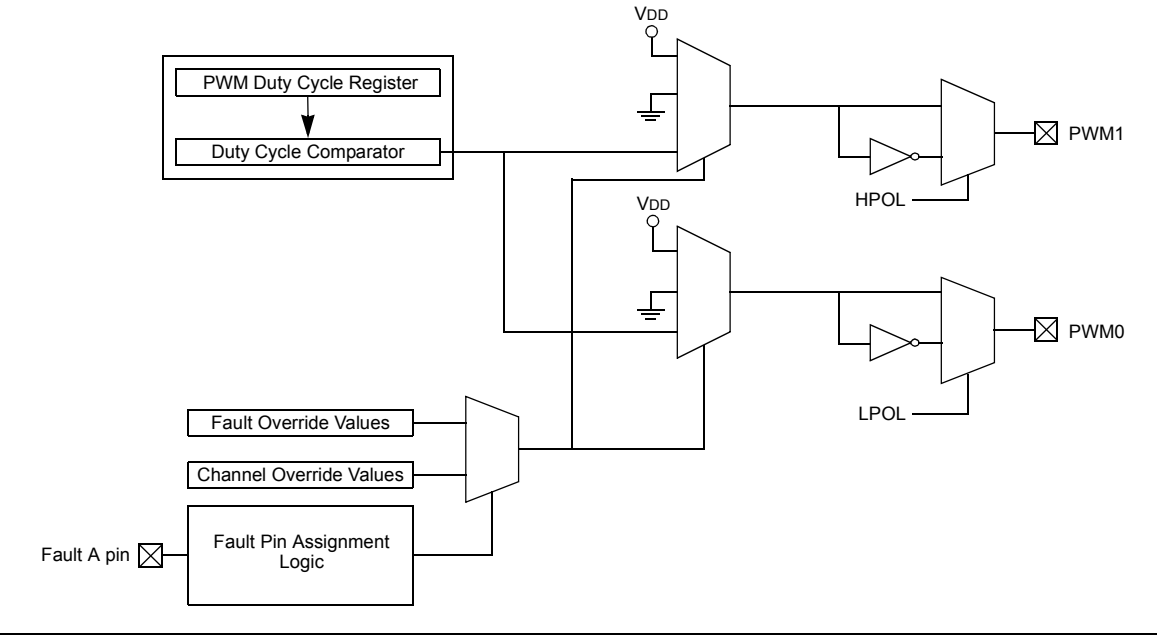


FIGURE 14-2: PWM MODULE BLOCK DIAGRAM, ONE OUTPUT PAIR, COMPLEMENTARY MODE



Note: In the Complementary mode, the even channel cannot be forced active by a Fault or override event when the odd channel is active. The even channel is always the complement of the odd channel and is inactive, with dead time inserted, before the odd channel is driven to its active state.

FIGURE 14-3: PWM MODULE BLOCK DIAGRAM, ONE OUTPUT PAIR, INDEPENDENT MODE



This module contains three duty cycle generators, numbered 0 through 2. The module has six PWM output pins, numbered 0 through 5. The six PWM outputs are grouped into output pairs of even and odd numbered outputs. In Complementary modes, the even PWM pins must always be the complement of the corresponding odd PWM pins. For example, PWM0 will be the complement of PWM1 and PWM2 will be the complement of PWM3. The dead-time generator inserts an OFF period called “dead time” between the going OFF of one pin to the going ON of the complementary pin of the paired pins. This is to prevent damage to the power switching devices that will be connected to the PWM output pins.

The time base for the PWM module is provided by its own 12-bit timer, which also incorporates selectable prescaler and postscaler options.

REGISTER 14-4: PWMCON1: PWM CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC
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 7-4 **SEVOPS3:SEVOPS0:** PWM Special Event Trigger Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
.
.
.
1111 = 1:16 Postscale

bit 3 **SEVTDIR:** Special Event Trigger Time Base Direction bit
1 = A Special Event Trigger will occur when the PWM time base is counting downwards
0 = A Special Event Trigger will occur when the PWM time base is counting upwards

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

bit 1 **UDIS:** PWM Update Disable bit
1 = Updates from Duty Cycle and Period Buffer registers are disabled
0 = Updates from Duty Cycle and Period Buffer registers are enabled

bit 0 **OSYNC:** PWM Output Override Synchronization bit
1 = Output overrides via the OVDCON register are synchronized to the PWM time base
0 = Output overrides via the OVDCON register are asynchronous

14.3.1 FREE-RUNNING MODE

In the Free-Running mode, the PWM time base (PTMRL and PTMRH) will begin counting upwards until the value in the PWM Time Base Period register, PTPER (PTPERL and PTPERH), is matched. The PTMR registers will be reset on the following input clock edge and the time base will continue counting upwards as long as the PTEN bit remains set.

14.3.2 SINGLE-SHOT MODE

In the Single-Shot mode, the PWM time base will begin counting upwards when the PTEN bit is set. When the value in the PTMR register matches the PTPER register, the PTMR register will be reset on the following input clock edge and the PTEN bit will be cleared by the hardware to halt the time base.

14.3.3 CONTINUOUS UP/DOWN COUNT MODES

In Continuous Up/Down Count modes, the PWM time base counts upwards until the value in the PTPER register matches the PTMR register. On the following input clock edge, the timer counts downwards. The PTDIR bit in the PTCON1 register is read-only and indicates the counting direction. The PTDIR bit is set when the timer counts downwards.

Note:	Since the PWM compare outputs are driven to the active state when the PWM time-base is counting downwards and matches the duty cycle value, the PWM outputs are held inactive during the first half of the first period of the Continuous Up/Down Count mode until the PTMR begins to count down from the PTPER value.

14.3.4 PWM TIME BASE PRESCALER

The input clock to PTMR (FOSC/4) has prescaler options of 1:1, 1:4, 1:16 or 1:64. These are selected by control bits, PTCKPS<1:0>, in the PTCON0 register. The prescaler counter is cleared when any of the following occurs:

- Write to the PTMR register
- Write to the PTCON (PTCON0 or PTCON1) register
- Any device Reset

Note:	The PTMR register is not cleared when PTCONx is written.

Table 14-1 shows the minimum PWM frequencies that can be generated with the PWM time base and the prescaler. An operating frequency of 40 MHz (FCYC = 10 MHz) and PTPER = 0xFFF are assumed in the table. The PWM module must be capable of generating PWM signals at the line frequency (50 Hz or 60 Hz) for certain power control applications.

TABLE 14-1: MINIMUM PWM FREQUENCY

Minimum PWM Frequencies vs. Prescaler Value for FCYC = 10 MIPS (PTPER = 0FFFh)		
Prescale	**PWM Frequency Edge-Aligned**	**PWM Frequency Center-Aligned**
1:1	2441 Hz	1221 Hz
1:4	610 Hz	305 Hz
1:16	153 Hz	76 Hz
1:64	38 Hz	19 Hz

14.3.5 PWM TIME BASE POSTSCALER

The match output of PTMR can optionally be postscaled through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate an interrupt. The postscaler counter is cleared when any of the following occurs:

- Write to the PTMR register
- Write to the PTCONx register
- Any device Reset

The PTMR register is not cleared when PTCONx is written.

14.4 PWM Time Base Interrupts

The PWM timer can generate interrupts based on the modes of operation selected by the PTMOD<1:0> bits and the postscaler bits (PTOPS<3:0>).

14.4.1 INTERRUPTS IN FREE-RUNNING MODE

When the PWM time base is in the Free-Running mode (PTMOD<1:0> = 00), an interrupt event is generated each time a match with the PTPER register occurs. The PTMR register is reset to zero in the following clock edge.

Using a postscaler selection other than 1:1 will reduce the frequency of interrupt events.

14.5 PWM Period

The PWM period is defined by the PTPER register pair (PTPERL and PTPERH). The PWM period has 12-bit resolution by combining 4 LSBs of PTPERH and 8 bits of PTPERL. PTPER is a double-buffered register used to set the counting period for the PWM time base.

The PTPER buffer contents are loaded into the PTPER register at the following times:

- Free-Running and Single-Shot modes: When the PTMR register is reset to zero after a match with the PTPER register.
- Continuous Up/Down Count modes: When the PTMR register is zero. The value held in the PTPER buffer is automatically loaded into the PTPER register when the PWM time base is disabled (PTEN = 0). Figure 14-9 and Figure 14-10 indicate the times when the contents of the PTPER buffer are loaded into the actual PTPER register.

The PWM period can be calculated from the following formulas:

EQUATION 14-1: PWM PERIOD FOR FREE-RUNNING MODE

$$T_{PWM} = \frac{(PTPER + 1) \times PTMRPS}{F_{OSC}/4}$$

EQUATION 14-2: PWM PERIOD FOR CONTINUOUS UP/DOWN COUNT MODE

$$T_{PWM} = \frac{(2 \times PTPER) \times PTMRPS}{\frac{F_{OSC}}{4}}$$

The PWM frequency is the inverse of period; or

EQUATION 14-3: PWM FREQUENCY

$$\text{PWM Frequency} = \frac{1}{\text{PWM Period}}$$

The maximum resolution (in bits) for a given device oscillator and PWM frequency can be determined from the following formula:

EQUATION 14-4: PWM RESOLUTION

$$\text{Resolution} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)}$$

The PWM resolutions and frequencies are shown for a selection of execution speeds and PTPER values in Table 14-2. The PWM frequencies in Table 14-2 are calculated for Edge-Aligned PWM mode. For Center-Aligned mode, the PWM frequencies will be approximately one-half the values indicated in this table.

TABLE 14-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS

PWM Frequency = 1/TPWM				
FOSC	**MIPS**	**PTPER Value**	**PWM Resolution**	**PWM Frequency**
40 MHz	10	0FFFh	14 bits	2.4 kHz
40 MHz	10	07FFh	13 bits	4.9 kHz
40 MHz	10	03FFh	12 bits	9.8 kHz
40 MHz	10	01FFh	11 bits	19.5 kHz
40 MHz	10	FFh	10 bits	39.0 kHz
40 MHz	10	7Fh	9 bits	78.1 kHz
40 MHz	10	3Fh	8 bits	156.2 kHz
40 MHz	10	1Fh	7 bits	312.5 kHz
40 MHz	10	0Fh	6 bits	625 kHz
25 MHz	6.25	0FFFh	14 bits	1.5 kHz
25 MHz	6.25	03FFh	12 bits	6.1 kHz
25 MHz	6.25	FFh	10 bits	24.4 kHz
10 MHz	2.5	0FFFh	14 bits	610 Hz
10 MHz	2.5	03FFh	12 bits	2.4 kHz
10 MHz	2.5	FFh	10 bits	9.8 kHz
5 MHz	1.25	0FFFh	14 bits	305 Hz
5 MHz	1.25	03FFh	12 bits	1.2 kHz
5 MHz	1.25	FFh	10 bits	4.9 kHz
4 MHz	1	0FFFh	14 bits	244 Hz
4 MHz	1	03FFh	12 bits	976 Hz
4 MHz	1	FFh	10 bits	3.9 kHz

Note: For center-aligned operation, PWM frequencies will be approximately 1/2 the value indicated in the table.

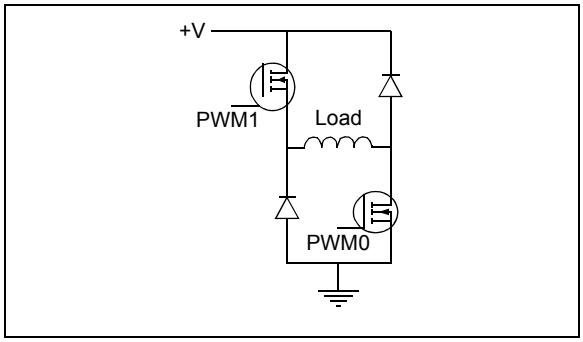
14.8.2 PWM CHANNEL OVERRIDE

PWM output may be manually overridden for each PWM channel by using the appropriate bits in the OVDCOND and OVDCONS registers. The user may select the following signal output options for each PWM output pin operating in the Independent PWM mode:

- I/O pin outputs PWM signal
- I/O pin inactive
- I/O pin active

Refer to **Section 14.10 "PWM Output Override"** for details for all the override functions.

FIGURE 14-19: CENTER CONNECTED LOAD



14.9 Single-Pulse PWM Operation

The single-pulse PWM operation is available only in Edge-Aligned mode. In this mode, the PWM module will produce single-pulse output. Single-pulse operation is configured when the PTMOD1:PTMOD0 bits are set to '01' in the PTCON0 register. This mode of operation is useful for driving certain types of ECMs.

In Single-Pulse mode, the PWM I/O pin(s) are driven to the active state when the PTEN bit is set. When the PWM timer match with Duty Cycle register occurs, the PWM I/O pin is driven to the inactive state. When the PWM timer match with the PTPER register occurs, the PTMR register is cleared, all active PWM I/O pins are driven to the inactive state, the PTEN bit is cleared and an interrupt is generated if the corresponding interrupt bit is set.

Note:	PTPER and PDCx values are held as they are after the single-pulse output. To have another cycle of single pulse, only PTEN has to be enabled.

14.10 PWM Output Override

The PWM output override bits allow the user to manually drive the PWM I/O pins to specified logic states, independent of the duty cycle comparison units. The PWM override bits are useful when controlling various types of ECMs, like a BLDC motor.

OVDCOND and OVDCONS registers are used to define the PWM override options. The OVDCOND register contains six bits, POVD5:POVD0, that determine which PWM I/O pins will be overridden. The OVDCONS register contains six bits, POUT5:POUT0, that determine the state of the PWM I/O pins when a particular output is overridden via the POVD bits.

The POVD bits are active-low control bits. When the POVD bits are set, the corresponding POUT bit will have no effect on the PWM output. In other words, the pins corresponding to POVD bits that are set will have the duty PWM cycle set by the PDCx registers. When one of the POVD bits is cleared, the output on the corresponding PWM I/O pin will be determined by the state of the POUT bit. When a POUT bit is set, the PWM pin will be driven to its active state. When the POUT bit is cleared, the PWM pin will be driven to its inactive state.

14.10.1 COMPLEMENTARY OUTPUT MODE

The even numbered PWM I/O pins have override restrictions when a pair of PWM I/O pins are operating in the Complementary mode (PMODx = 0). In Complementary mode, if the even numbered pin is driven active by clearing the corresponding POVD bit and by setting the POUT bits in the OVDCOND and OVDCONS registers, the output signal is forced to be the complement of the odd numbered I/O pin in the pair (see Figure 14-2 for details).

14.10.2 OVERRIDE SYNCHRONIZATION

If the OSYNC bit in the PWMCON1 register is set, all output overrides performed via the OVDCOND and OVDCONS registers will be synchronized to the PWM time base. Synchronous output overrides will occur on the following conditions:

- When the PWM is in Edge-Aligned mode, synchronization occurs when PTMR is zero.
- When the PWM is in Center-Aligned mode, synchronization occurs when PTMR is zero and when the value of PTMR matches PTPER.

Note 1:	In the Complementary mode, the even channel cannot be forced active by a Fault or override event when the odd channel is active. The even channel is always the complement of the odd channel, with dead-time inserted, before the odd channel can be driven to its active state as shown in Figure 14-20.
2:	Dead time inserted in the PWM channels even when they are in Override mode.

15.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

The EUSART can be configured in the following modes:

- Asynchronous (full-duplex) with:
  - Auto-Wake-up on Character Reception
  - Auto-Baud Calibration
  - 12-Bit Break Character Transmission
- Synchronous – Master (half-duplex) with Selectable Clock Polarity
- Synchronous – Slave (half-duplex) with Selectable Clock Polarity

The pins of the Enhanced USART are multiplexed with PORTA. In order to configure RA2/TX/CK and RA3/RX/DT as an EUSART:

- bit SPEN (RCSTA<7>) must be set (= 1)
- bit TRISA<3> must be set (= 1)
- bit TRISA<2> must be set (= 1)

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These are detailed on the following pages in Register 15-1, Register 15-2 and Register 15-3, respectively.

EXAMPLE 15-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGH:SPBRG] + 1))

Solving for SPBRGH:SPBRG:

X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

TABLE 15-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	48
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	48
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	48
SPBRGH	EUSART Baud Rate Generator Register High Byte								48
SPBRG	EUSART Baud Rate Generator Register Low Byte								48

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the BRG.

TABLE 15-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	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	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	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 = 0, BRG16 = 1								
	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	0.04	832	0.300	-0.16	415	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	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	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	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.647	-2.12	16

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1								
	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	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

17.0 COMPARATOR MODULE

The analog comparator module contains three comparators. The inputs can be selected from the analog inputs multiplexed with pins RA0, RB2 and RB3, as well as the on-chip voltage reference (see **Section 18.0 "Comparator Voltage Reference Module"**). The digital outputs are not available at the pin level and can only be read through the control register, CMCON (Register 17-1). CMCON also selects the comparator input.

REGISTER 17-1: CMCON: COMPARATOR CONTROL REGISTER

R-0	R-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C0OUT	—	—	CMEN2	CMEN1	CMEN0
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 **C2OUT**: Comparator 2 Output bit
1 = C2 VIN+ > C2 VIN- (CVREF)
0 = C2 VIN+ < C2 VIN- (CVREF)

bit 6 **C1OUT**: Comparator 1 Output bit
1 = C1 VIN+ > C1 VIN- (CVREF)
0 = C1 VIN+ < C1 VIN- (CVREF)-

bit 5 **C0OUT**: Comparator 0 Output bit
1 = C0 VIN+ > C0 VIN- (CVREF)
0 = C0 VIN+ < C0 VIN- (CVREF)

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **CMEN2**: Comparator 2 Enable bit
1 = Comparator 2 is enabled
0 = Comparator 2 is disabled

bit 1 **CMEN1**: Comparator 1 Enable bit
1 = Comparator 1 is enabled
0 = Comparator 1 is disabled

bit 0 **CMEN0**: Comparator 0 Enable bit
1 = Comparator 0 is enabled
0 = Comparator 0 is disabled

20.5 Program Verification and Code Protection

The overall structure of the code protection on the PIC18 Flash devices differs significantly from other PIC® devices.

The user program memory is divided into three blocks. One of these is a Boot Block of variable size (maximum 2 Kbytes). The remainder of the memory is divided into two blocks on binary boundaries.

Each of the three blocks has three code protection bits associated with them. They are:

- Code-Protect bit (CPx)
- Write-Protect bit (WRTx)
- External Block Table Read bit (EBTRx)

Figure 20-5 shows the program memory organization for 4 and 8-Kbyte devices and the specific code protection bit associated with each block. The actual locations of the bits are summarized in Table 20-3.

FIGURE 20-5: CODE-PROTECTED PROGRAM MEMORY FOR PIC18F1230/1330

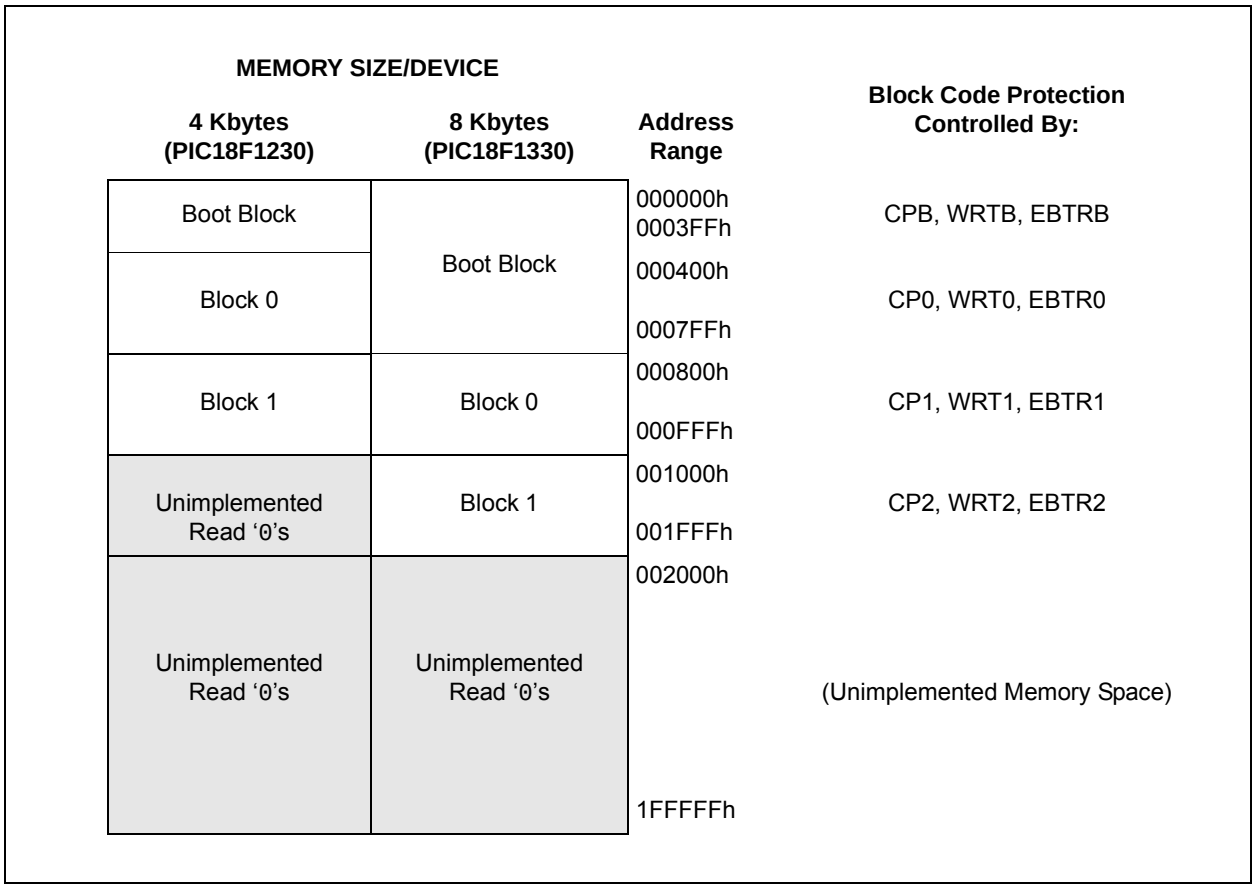


TABLE 20-3: SUMMARY OF CODE PROTECTION REGISTERS

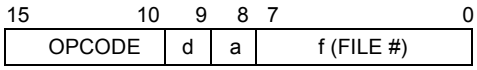
File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	—	—	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	—	—	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	—	—	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded cells are unimplemented.

FIGURE 22-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

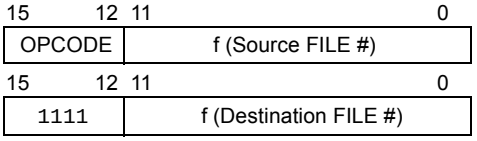
Example Instruction



ADDWF MYREG, W, B

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

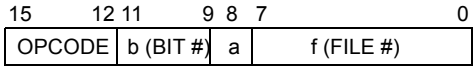
Byte to Byte move operations (2-word)



MOVFF MYREG1, MYREG2

f = 12-bit file register address

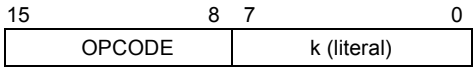
Bit-oriented file register operations



BSF MYREG, bit, B

b = 3-bit position of bit in file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

Literal operations

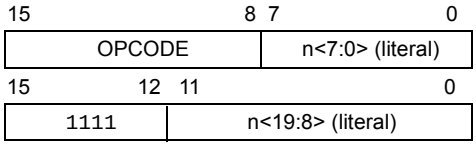


MOVLW 7Fh

k = 8-bit immediate value

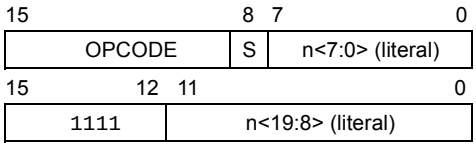
Control operations

CALL, GOTO and Branch operations



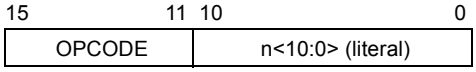
GOTO Label

n = 20-bit immediate value

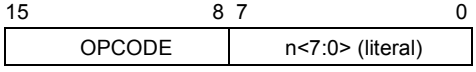


CALL MYFUNC

S = Fast bit



BRA MYFUNC



BC MYFUNC

SUBWFB **Subtract W from f with Borrow**

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

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

Operation: (f) – (W) – ($\overline{C}$) → dest

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

Encoding:

0101	10da	ffff	ffff

Description: Subtract W and the Carry flag (borrow) 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 ≤ 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:
```
SUBWFB  REG, 1, 0
Before Instruction
    REG  =  19h   (0001 1001)
    W    =  0Dh   (0000 1101)
    C    =  1
After Instruction
    REG  =  0Ch   (0000 1011)
    W    =  0Dh   (0000 1101)
    C    =  1
    Z    =  0
    N    =  0     ; result is positive
```

Example 2:
```
SUBWFB  REG, 0, 0
Before Instruction
    REG  =  1Bh   (0001 1011)
    W    =  1Ah   (0001 1010)
    C    =  0
After Instruction
    REG  =  1Bh   (0001 1011)
    W    =  00h
    C    =  1
    Z    =  1     ; result is zero
    N    =  0
```

Example 3:
```
SUBWFB  REG, 1, 0
Before Instruction
    REG  =  03h   (0000 0011)
    W    =  0Eh   (0000 1101)
    C    =  1
After Instruction
    REG  =  F5h   (1111 0100)
                  ; [2's comp]
    W    =  0Eh   (0000 1101)
    C    =  0
    Z    =  0
    N    =  1     ; result is negative
```

SWAPF **Swap f**

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

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

Operation: (f<3:0>) → dest<7:4>,
(f<7:4>) → dest<3:0>

Status Affected: None

Encoding:

0011	10da	ffff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed 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 ≤ 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:
```
SWAPF   REG, 1, 0
Before Instruction
    REG  =  53h
After Instruction
    REG  =  35h
```

XORWF **Exclusive OR W with f**

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

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

Operation: (W) .XOR. (f) → dest

Status Affected: N, Z

Encoding:

0001	10da	ffff	ffff

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the 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: `XORWF REG, 1, 0`

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

APPENDIX E: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the Enhanced devices (i.e., PIC18FXXX) is provided in *AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442"*. The changes discussed, while device specific, are generally applicable to all mid-range to Enhanced device migrations.

This Application Note is available as Literature Number DS00716.

APPENDIX F: MIGRATION FROM HIGH-END TO ENHANCED DEVICES

A detailed discussion of the migration pathway and differences between the high-end MCU devices (i.e., PIC17CXXX) and the Enhanced devices (i.e., PIC18FXXX) is provided in *AN726, "PIC17CXXX to PIC18CXXX Migration"*.

This Application Note is available as Literature Number DS00726.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com