

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	36
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4331-i-pt

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18F MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18F2331/2431/4331/4431 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and Vss pins (see Section 2.2 "Power Supply Pins")
- All AVDD and AVss pins, regardless of whether or not the analog device features are used (see Section 2.2 "Power Supply Pins")
- MCLR pin
 (see Section 2.3 "Master Clear (MCLR) Pin")

These pins must also be connected if they are being used in the end application:

- PGC/PGD pins used for In-Circuit Serial Programming[™] (ICSP[™]) and debugging purposes (see **Section 2.4 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used

(see Section 2.5 "External Oscillator Pins")

Additionally, the following pins may be required:

• VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

Note: The AVDD and AVss pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.



4.3 Sleep Mode

The power-managed Sleep mode in the PIC18F2331/ 2431/4331/4431 devices is identical to the legacy Sleep mode offered in all other PIC devices. It is entered by clearing the IDLEN bit (the default state on device Reset) and executing the SLEEP instruction. This shuts down the selected oscillator (Figure 4-5). All clock source status bits are cleared.

Entering the Sleep mode from any other mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source, selected by the SCS<1:0> bits, becomes ready (see Figure 4-6), or it will be clocked from the internal oscillator block if either the Two-Speed Start-up or the Fail-Safe Clock Monitor is enabled (see **Section 23.0 "Special Features of the CPU"**). In either case, the OSTS bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

4.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

If the IDLEN bit is set to a '1' when a SLEEP instruction is executed, the peripherals will be clocked from the clock source selected using the SCS<1:0> bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a SLEEP instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of TCSD (Parameter 38, Table 26-8) while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC_IDLE mode, the internal oscillator block will clock the CPU and peripherals (in other words, RC_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or Sleep mode, a WDT timeout will result in a WDT wake-up to the Run mode currently specified by the SCS<1:0> bits.

FIGURE 4-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE







4.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- PRI_IDLE mode, where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval, TCSD, following the wake event, is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

TABLE 4-2:EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE
(BY CLOCK SOURCES)

Clock Source Before Wake-up	Clock Source After Wake-up	Exit Delay	Clock Ready Status Bit (OSCCON)
	LP, XT, HS		
Primary Device Clock	HSPLL	тер (1)	OSTS
(PRI_IDLE mode)	EC, RC	105007	
	INTOSC ⁽²⁾		IOFS
	LP, XT, HS	Tost ⁽³⁾	
11050	HSPLL	Tost + t _{rc} (3)	OSTS
TIOSC	EC, RC	TCSD ⁽¹⁾	
	INTOSC ⁽²⁾	TIOBST ⁽⁴⁾	IOFS
	LP, XT, HS	Tost ⁽³⁾	
	HSPLL	Tost + t _{rc} ⁽³⁾	OSTS
	EC, RC	Tcsd ⁽¹⁾	
	INTOSC ⁽²⁾	None	IOFS
	LP, XT, HS	Tost ⁽³⁾	
None	HSPLL	Tost + t _{rc} ⁽³⁾	OSTS
(Sleep mode)	EC, RC	TCSD ⁽¹⁾	
	INTOSC ⁽²⁾	TIOBST ⁽⁴⁾	IOFS

Note 1: TCSD (Parameter 38) is a required delay when waking from Sleep and all Idle modes, and runs concurrently with any other required delays (see **Section 4.4 "Idle Modes"**).

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

3: TOST is the Oscillator Start-up Timer (Parameter 32). t_{rc} is the PLL Lock-out Timer (Parameter F12); it is also designated as TPLL.

4: Execution continues during TIOBST (Parameter 39), the INTOSC stabilization period.

6.1.4.2 Table Reads and Table Writes

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location. Look-up table data may be stored, two bytes per program word, by using table reads and writes.

The Table Pointer register (TBLPTR) specifies the byte address and the Table Latch register (TABLAT) contains the data that is read from or written to program memory. Data is transferred to or from program memory, one byte at a time.

Table read and table write operations are discussed further in Section 8.1 "Table Reads and Table Writes".

6.2 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the Program Counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the Instruction Register (IR) in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 6-4.



FIGURE 6-4: CLOCK/INSTRUCTION CYCLE

6.3 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 6-3).

A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" (IR) in cycle, Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

EXAMPLE 6-3: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline, while the new instruction is being fetched and then executed.

6.5.1 BANK SELECT REGISTER (BSR)

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes.

Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a four-bit Bank Pointer. Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the four Most Significant bits of a location's address; the instruction itself includes the eight Least Significant bits. Only the four lower bits of the BSR are implemented (BSR<3:0>). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the MOVLB instruction.

The value of the BSR indicates the bank in data memory. The eight bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 6-6.

Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to the eight-bit address of F9h, while the BSR is 0Fh, will end up resetting the program counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 6-5 indicates which banks are implemented.

In the core PIC18 instruction set, only the MOVFF instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

6.5.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected; otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 128 bytes of memory (00h-7Fh) in Bank 0 and the last 128 bytes of memory (80h-FFh) in Block 15. The lower half is known as the "Access RAM" and is composed of GPRs. This upper half is also where the device's SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 6-6).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the 'a' parameter in the instruction). When 'a' is equal to '1', the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When 'a' is '0', however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this "forced" addressing allows the instruction to operate on a data address in a single cycle, without updating the BSR first. For 8-bit addresses of 80h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 80h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

6.5.3 GENERAL PURPOSE REGISTER (GPR) FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM, which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

8.2.2 TABLAT – TABLE LATCH REGISTER

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch is used to hold 8-bit data during data transfers between program memory and data RAM.

8.2.3 TBLPTR – TABLE POINTER REGISTER

The Table Pointer (TBLPTR) addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. Setting the 22nd bit allows access to the Device ID, the User ID and the Configuration bits.

The TBLPTR is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 8-1. These operations on the TBLPTR only affect the low-order 21 bits.

8.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the Table Pointer determine which byte is read from program or configuration memory into TABLAT.

When a TBLWT is executed, the three LSbs of the Table Pointer (TBLPTR<2:0>) determine which of the eight program memory holding registers is written to. When the timed write to program memory (long write) begins, the 19 MSbs of the Table Pointer, TBLPTR (TBLPTR<21:3>), will determine which program memory block of 8 bytes is written to (TBLPTR<2:0> are ignored). For more detail, see **Section 8.5 "Writing to Flash Program Memory"**.

When an erase of program memory is executed, the 16 MSbs of the Table Pointer (TBLPTR<21:6>) point to the 64-byte block that will be erased. The Least Significant bits (TBLPTR<5:0>) are ignored.

Figure 8-3 describes the relevant boundaries of TBLPTR based on Flash program memory operations.

TABLE 8-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 8-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



10.0 INTERRUPTS

The PIC18F2331/2431/4331/4431 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 000008h and the low-priority interrupt vector is at 000018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are thirteen registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB[®] IDE be used for the symbolic bit names in these registers. This allows the assembler/ compiler to automatically take care of the placement of these bits within the specified register.

In general, each interrupt source has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority (most interrupt sources have priority bits)

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 00008h or 000018h depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC[®] mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit, which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit, which enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a lowpriority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (00008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the Interrupt Control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior. The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the CCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation:

EQUATION 16-3:



Note:	If the PWM duty cycle value is longer than
	the PWM period, the CCP1 pin will not be
	cleared.

16.5.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP1 module for PWM operation:

- 1. Set the PWM period by writing to the PR2 register.
- 2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
- 3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
- 4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
- 5. Configure the CCP1 module for PWM operation.

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

TABLE 16-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	54
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	57
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	57
IPR1	_	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	57
TRISC	PORTC Da	ta Direction	Register						57
TMR2	Timer2 Reg	gister							55
PR2	Timer2 Per	iod Register							55
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	55
CCPR1L	Capture/Co	Capture/Compare/PWM Register 1 Low Byte 56					56		
CCPR1H	Capture/Co	mpare/PWN	1 Register 1	High Byte					56
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	56
CCPR2L	Capture/Co	mpare/PWN	1 Register 2	Low Byte					56
CCPR2H	Capture/Co	mpare/PWN	1 Register 2	High Byte					56
CCP2CON		_	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	56

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

PIC18F2331/2431/4331/4431



FIGURE 17-1: MOTION FEEDBACK MODULE BLOCK DIAGRAM

17.1 Input Capture

The Input Capture (IC) submodule implements the following features:

- Three channels of independent input capture (16-bits/channel) on the CAP1, CAP2 and CAP3 pins
- Edge-Trigger, Period or Pulse-Width Measurement Operating modes for each channel
- Programmable prescaler on every input capture channel
- Special Event Trigger output (IC1 only)
- · Selectable noise filters on each capture input

Input Channel 1 (IC1) includes a Special Event Trigger that can be configured for use in Velocity Measurement mode. Its block diagram is shown in Figure 17-2. IC2 and IC3 are similar, but lack the Special Event Trigger features or additional velocity measurement logic. A representative block diagram is shown in Figure 17-3. Please note that the time base is Timer5.



FIGURE 17-2: INPUT CAPTURE BLOCK DIAGRAM FOR IC1

18.0 POWER CONTROL PWM MODULE

The Power Control PWM module simplifies the task of generating multiple, synchronized Pulse-Width Modulated (PWM) outputs for use in the control of motor controllers and power conversion applications. In particular, the following power and motion control applications are supported by the PWM module:

- Three-Phase and Single-Phase AC Induction Motors
- Switched Reluctance Motors
- Brushless DC (BLDC) Motors
- Uninterruptible Power Supplies (UPS)
- Multiple DC Brush Motors

The PWM module has the following features:

- Up to eight PWM I/O pins with four duty cycle generators. Pins can be paired to get a complete half-bridge control.
- Up to 14-bit resolution, depending upon the PWM period.
- "On-the-fly" PWM frequency changes.
- Edge and Center-Aligned Output modes.
- Single-Pulse Generation mode.
- Programmable dead-time control between paired PWMs.
- Interrupt support for asymmetrical updates in Center-Aligned mode.
- Output override for Electrically Commutated Motor (ECM) operation; for example, BLDC.
- Special Event Trigger comparator for scheduling other peripheral events.
- PWM outputs disable feature sets PWM outputs to their inactive state when in Debug mode.

The Power Control PWM module supports three PWM generators and six output channels on PIC18F2331/2431 devices, and four generators and eight channels on PIC18F4331/4431 devices. A simplified block diagram of the module is shown in Figure 18-1. Figure 18-2 and Figure 18-3 show how the module hardware is configured for each PWM output pair for the Complementary and Independent Output modes.

Each functional unit of the PWM module will be discussed in subsequent sections.

Table 18-1 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 is 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.

Minimum for F	PWM Frequencies V FCYC = 10 MIPS (PTF	vs. Prescaler Value PER = 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

TABLE 18-1: MINIMUM PWM FREQUENCY

18.3.5 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 PTCON register
- Any device Reset

The PTMR register is not cleared when PTCON is written.

18.4 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>).

18.4.1 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.

FIGURE 18-5: PWM TIME BASE INTERRUPT TIMING, FREE-RUNNING MODE







18.6.4 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 18-15). 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.





REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0	R/W-0	U-0	R/W-0	R-0	R-0	R-0	R-0
VCFG1	VCFG0	—	FIFOEN	BFEMT	BFOVL	ADPNT1	ADPNT0
bit 7			•				bit 0

W = Writable bit	U = Unimplemented bit,	read as '0'
'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
	W = Writable bit '1' = Bit is set	W = Writable bitU = Unimplemented bit,'1' = Bit is set'0' = Bit is cleared

bit 7-6	VCFG<1:0>: A/D VREF+ and A/D VREF- Source Selection bits 00 = VREF+ = AVDD, VREF- = AVSS (AN2 and AN3 are analog inputs or digital I/O) 01 = VREF+ = External VREF+, VREF- = AVSS (AN2 is an analog input or digital I/O) 10 = VREF+ = AVDD, VREF- = External VREF- (AN3 is an analog input or digital I/O) 11 = VREF+ = External VREF-, VREF- = External VREF-
bit 5	Unimplemented: Read as '0'
bit 4	FIFOEN: FIFO Buffer Enable bit
	1 = FIFO is enabled0 = FIFO is disabled
bit 3	BFEMT: Buffer Empty bit
	 1 = FIFO is empty 0 = FIFO is not empty (at least one of four locations has unread A/D result data)
bit 2	BFOVFL: Buffer Overflow bit
	 1 = A/D result has overwritten a buffer location that has unread data 0 = A/D result has not overflowed
bit 1-0	ADPNT<1:0>: Buffer Read Pointer Location bits
	Designates the location to be read next. 00 = Buffer Address 0 01 = Buffer Address 1 10 = Buffer Address 2 11 = Buffer Address 3

23.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation, in the event of an external oscillator failure, by automatically switching the system clock to the internal oscillator block. The FSCM function is enabled by setting the Fail-Safe Clock Monitor Enable bit, FCMEN (CONFIG1H<6>).

When FSCM is enabled, the INTRC oscillator runs at all times to monitor clocks to peripherals and provide an instant backup clock in the event of a clock failure. Clock monitoring (shown in Figure 23-3) is accomplished by creating a sample clock signal, which is the INTRC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral system clock and the sample clock are presented as inputs to the Clock Monitor latch (CM). The CM is set on the falling edge of the system clock source, but cleared on the rising edge of the sample clock.



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while the CM is still set, a clock failure has been detected (Figure 23-4). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- the system clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- the WDT is reset.

Since the postscaler frequency from the internal oscillator block may not be sufficiently stable, it may be desirable to select another clock configuration and enter an alternate power-managed mode (see Section 23.3.1 "Special Considerations for Using Two-Speed Start-up" and Section 4.1.4 "Multiple Sleep Commands" for more details). This can be done to attempt a partial recovery or execute a controlled shutdown.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> bits prior to entering Sleep mode.

Adjustments to the internal oscillator block using the OSCTUNE register also affect the period of the FSCM by the same factor. This can usually be neglected, as the clock frequency being monitored is generally much higher than the sample clock frequency.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

23.4.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF<2:0> bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock Monitor events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

23.4.2 EXITING FAIL-SAFE OPERATION

The fail-safe condition is terminated by either a device Reset, or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTOSC multiplexer provides the system clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock system source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

Entering a power-managed mode by loading the OSCCON register and executing a SLEEP instruction will clear the fail-safe condition. When the fail-safe condition is cleared, the clock monitor will resume monitoring the peripheral clock.

PIC18F2331/2431/4331/4431

MUL	LW	Multiply L	iteral with	w			
Synta	ax:	[label] N	IULLW k				
Operands:		$0 \le k \le 255$					
Oper	ation:	(W) x k \rightarrow f	PRODH:PROI	DL			
Statu	is Affected:	None					
Enco	oding:	0000	1101 kk	kk kkkk			
Desc	ription:	An unsigne out betwee the 8-bit lite is placed in pair. PROD W is uncha None of the Note that n is possible result is pos	d multiplication n the contents ral, 'k'. The 1 PRODH:PRC H contains the nged. Status flags a either Overflo in this operation ssible but not	in is carried s of W and 6-bit result DDL register e high byte. are affected. w nor Carry on. A Zero detected.			
Word	ds:	1					
Cycle	es:	1	1				
QC	ycle Activity:						
	Q1	Q2	Q3	Q4			
	Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL			
Exan	nple:	MULLW	0xC4				
Before Instruction W PRODH PRODL After Instruction W PRODH PRODL		tion = 0x = ? = ?	E2				
		n = 0x = 0x = 0x	E2 AD 08				

MULWF	Multiply W with f				
Syntax:	[label] M	IULWF f[,a]			
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]	0 ≤ f ≤ 255 a ∈ [0,1]			
Operation:	(W) x (f) \rightarrow	(W) x (f) \rightarrow PRODH:PRODL			
Status Affected:	None	None			
Encoding:	0000	001a fff	f ffff		
	out between the register 16-bit result PRODH:PF PRODH co Both W and None of the Note that nu is possible result is pos 'a' is '0', the selected, ov 'a'= 1, then as per the E	in the contents file location, "I t is stored in th RODL register ntains the high I 'f are unchar Status flags a either Overflow in this operation sible but not of Access Bank verriding the B the bank will the BSR value.	of W and of W and i. The he pair. hyte.		
Words:	1				
Cycles:	1				
Q Cycle Activity:					
Q1	Q2	Q3	Q4		
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL		
Example:	MULWF H	REG			
Before Instruc W REG PRODH PRODL	tion = 0x0 = 0x1 = ? = ?	C4 35			

After Instruction		
W	=	0xC4
REG	=	0xB5
PRODH	=	0x8A
PRODL	=	0x94

PIC18F2331/2431/4331/4431

Before Instruction W

After Instruction W

=

=

0xB5

0x1A

тѕт	FSZ	Test f, Ski	Test f, Skip if 0			
Synta	ax:	[label] T	STFSZ f[,a]			
Operands:		0 ≤ f ≤ 255 a ∈ [0,1]	0 ≤ f ≤ 255 a ∈ [0,1]			
Oper	ation:	skip if f = 0				
Statu	s Affected:	None				
Enco	ding:	0110	011a fff	f ffff		
Description:		If 'f' = 0, the during the c is discarded making this is '0', the Ad overriding the then the bar the BSR va	If $f' = 0$, the next instruction, fetched during the current instruction execution, is discarded and a NOP is executed, making this a two-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.			
Word	ls:	1				
Cycles:		1(2) Note: 3 cy by a	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.			
QC	ycle Activity:					
	Q1	Q2	Q3	Q4		
	Decode	Read register 'f'	Process Data	No operation		
lf sk	ip:					
	Q1	Q2	Q3	Q4		
	No	No	No	No		
	operation	operation	operation operation			
If sk	ip and followed	d by 2-word in:	struction:	.		
	Q1	Q2	Q3	Q4		
	No	No	No	No		
	No	No	No	No		
	operation	operation	operation	operation		
Example: HERE TSTFSZ CNT NZERO : ZERO :						
Before Instruction PC = Address (HERE)						
	After Instructic If CNT PC If CNT PC	on = 0x0 = Ad ≠ 0x0 = Ad	DO, dress (zero DO, dress (nzer) 0)		

XORLW	Exclusiv	Exclusive OR Literal with W				
Syntax:	[label]	[label] XORLW k				
Operands:	$0 \le k \le 25$	$0 \le k \le 255$				
Operation:	(W) .XOR	(W) .XOR. $k \rightarrow W$				
Status Affected:	N, Z	N, Z				
Encoding:	0000	0000 1010 kkkk kkkk				
Description:	ription: The contents of W are XORed with the 8-bit literal, 'k'. The result is placed in W.			ed with is placed		
Words:	1					
Cycles:	1					
Q Cycle Activity:						
Q1	Q2	Q3		Q4		
Decode	Read literal 'k'	Proces Data	ss Wr	ite to W		
Example:	XORLW)xAF				

TABLE 26-20: A/D CONVERTER CHARACTERISTICS

PIC18LF2331/2431/4331/4431 (Industrial)		Standard Operating Conditions (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial					
PIC18F2331/2431/4331/4431 (Industrial)		$ \begin{array}{ll} \mbox{Standard Operating Conditions (unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq TA \leq +85^{\circ}C \mbox{ for industrial} \\ -40^{\circ}C \leq TA \leq +125^{\circ}C \mbox{ for extended} \end{array} $					
Param No.	Symbol	Characteristic	Min	Тур	Max	Units	Conditions
Device	Supply	·					•
	AVDD	Analog VDD Supply	VDD - 0.3	_	VDD + 0.3	V	
	AVss	Analog Vss Supply	Vss – 0.3	_	Vss + 0.3	V	
	IAD	Module Current	_	500	_	μA	VDD = 5V
		(during conversion)	—	250	—	μA	VDD = 2.5V
	IADO	Module Current Off	—		1.0	μA	
AC Tim	ing Parame	eters			1		1
A10	Fthr	Throughput Rate	_	_	200 75	ksps ksps	VDD = 5V, single channel VDD < 3V, single channel
A11	Tad	A/D Clock Period	385	_	20,000	ns	VDD = 5V
	_		1000	_	20,000	ns	VDD = 3V
A12	TRC	A/D Internal RC Oscillator Period	_	500 750	1500 2250	ns	PIC18F parts
			_	10000	20000	ns	AVDD < 3.0V
A13	TCNV	Conversion Time ⁽¹⁾	12	12	12	TAD	
A14	TACQ	Acquisition Time ⁽²⁾	2 ⁽²⁾	_	_	TAD	
A16	Ттс	Conversion Start from External	1/4 Tcy	_	_		
Reference Inputs							
A20	Vref	Reference Voltage for 10-Bit Resolution (VREF+ – VREF-)	1.5 1.8	_	AVDD – AVSS AVDD – AVSS	V V	$VDD \ge 3V$ VDD < 3V
A21	VREFH	Reference Voltage High (AVDD or VREF+)	1.5V	_	AVDD	V	$VDD \ge 3V$
A22	VREFL	Reference Voltage Low (AVss or VREF-)	AVss	_	VREFH – 1.5V	V	
A23	IREF	Reference Current	_	150 μA 75 μA			VDD = 5V VDD = 2.5V
Analog	Input Char	acteristics		·			I
A26	Vain	Input Voltage ⁽³⁾	AVss - 0.3		AVDD + 0.3	V	
A30	Zain	Recommended Impedance of Analog Voltage Source	—	—	2.5	kΩ	
A31	ZCHIN	Analog Channel Input Impedance	_	_	10.0	kΩ	VDD = 3.0V
DC Per	formance						
A41	A41 NR Resolution 10 bits —						
A42	EIL	Integral Nonlinearity	—	—	<±1	LSb	$VDD \ge 3.0V$ VREFH $\ge 3.0V$
A43	EIL	Differential Nonlinearity	—	—	<±1	LSb	$VDD \ge 3.0V$ VREFH $\ge 3.0V$
A45	EOFF	Offset Error	—	±0.5	<±1.5	LSb	$\begin{array}{l} VDD \geq 3.0V \\ VREFH \geq 3.0V \end{array}$
A46	EGA	Gain Error	—	±0.5	<±1.5	LSb	$\begin{array}{l} VDD \geq 3.0V \\ VREFH \geq 3.0V \end{array}$
A47	—	Monotonicity ⁽⁴⁾	guaranteed		—	$VDD \ge 3.0V$ VREFH $\ge 3.0V$	

Note 1: Conversion time does not include acquisition time. See Section 21.0 "10-Bit High-Speed Analog-to-Digital Converter (A/D) Module" for a full discussion of acquisition time requirements.

2: In Sequential modes, TACQ should be 12 TAD or greater.

3: For VDD < 2.7V and temperature below 0°C, VAIN should be limited to range < VDD/2.

4: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

27.0 PACKAGING INFORMATION

27.1 Package Marking Information

28-Lead SPDIP (Skinny PDIP)



28-Lead SOIC



Example PIC18F2331-I/SP@3 1010017

Example



28-Lead QFN



Example



Legend	: XXX Y YY WW NNN (©3) *	Customer-specific information Year code (last digit of calendar year) Year code (last 2 digits of calendar year) Week code (week of January 1 is week '01') Alphanumeric traceability code Pb-free JEDEC designator for Matte Tin (Sn) This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.	
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.		

27.1 Package Marking Information (Continued)

40-Lead PDIP



 \bigcirc



44-Lead TQFP



Example



PIC18F4331-I/P (e3) 1010017 MICROCHIP

44-Lead QFN



Example

