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	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f27j53-i-ss

TABLE 1-3: PIC18F2XJ53 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	28-SPDIP/ SSOP/ SOIC	28-QFN			
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/C3IND/INT0/RP3	21	18			
RB0			I/O	TTL/DIG	Digital I/O.
AN12			I	Analog	Analog Input 12.
C3IND			I	Analog	Comparator 3 Input D.
INT0			I	ST	External Interrupt 0.
RP3			I/O	ST/DIG	Remappable Peripheral Pin 3 input/output.
RB1/AN10/C3INC/RTCC/RP4	22	19			
RB1			I/O	TTL/DIG	Digital I/O.
AN10			I	Analog	Analog Input 10.
C3INC			I	Analog	Comparator 3 input.
RTCC			O	DIG	Asynchronous serial transmit data output.
RP4			I/O	ST/DIG	Remappable Peripheral Pin 4 input/output.
RB2/AN8/C2INC/CTED1/ VMO/REFO/RP5	23	20			
RB2			I/O	TTL/DIG	Digital I/O.
AN8			I	Analog	Analog Input 8.
C2INC			I	Analog	Comparator 2 Input C.
CTED1			I	ST	CTMU Edge 1 input.
VMO			O	DIG	External USB Transceiver D- data output.
REFO			O	DIG	Reference output clock.
RP5			I/O	ST/DIG	Remappable Peripheral Pin 5 input/output.
RB3/AN9/C3INA/CTED2/ VPO/RP6	24	21			
RB3			I/O	TTL/DIG	Digital I/O.
AN9			I	Analog	Analog Input 9.
C3INA			I	Analog	Comparator 3 Input A.
CTED2			I	ST	CTMU edge 2 Input.
VPO			O	DIG	External USB Transceiver D+ data output.
RP6			I	ST/DIG	Remappable Peripheral Pin 6 input/output.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD}); DIG = Digital output; I^2C = Open-Drain, I^2C specific

Note 1: RA7 and RA6 will be disabled if OSC1 and OSC2 are used for the clock function.

2: 5.5V tolerant.

REGISTER 3-1: OSCTUNE: OSCILLATOR TUNING REGISTER (ACCESS F9Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN(1)	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled)
0 = 31 kHz device clock derived directly from INTRC internal oscillator

bit 6 **PLLEN:** Frequency Multiplier Enable bit(1)
1 = 96 MHz PLL is enabled
0 = 96 MHz PLL is disabled

bit 5-0 **TUN<5:0>:** Frequency Tuning bits
011111 = Maximum frequency
011110
•
•
•
000001
000000 = Center frequency; oscillator module is running at the calibrated frequency
111111
•
•
•
100000 = Minimum frequency

Note 1: When the CFGPLLEN Configuration bit is used to enable the PLL, clearing OSCTUNE<6> will not disable the PLL.

3.3 Oscillator Settings for USB

When the PIC18F47J53 family devices are used for USB connectivity, a 6 MHz or 48 MHz clock must be provided to the USB module for operation in either Low-Speed or Full-Speed modes, respectively. This may require some forethought in selecting an oscillator frequency and programming the device.

The full range of possible oscillator configurations compatible with USB operation is shown in Table 3-5.

3.3.1 LOW-SPEED OPERATION

The USB clock for Low-Speed mode is derived from the primary oscillator or from the 96 MHz PLL. In order to operate the USB module in Low-Speed mode, a 6 MHz clock must be provided to the USB module.

See Table 3-4 and Table 3-5 for possible combinations which can be used for low-speed USB operation.

TABLE 3-4: CLOCK FOR LOW-SPEED USB

System Clock	CPDIV<1:0>	Microcontroller Clock	LS48MHZ	USB Clock
48	11	48 MHz	1	48/8 = 6 MHz
48	10	48/2 = 24 MHz	1	48/8 = 6 MHz
48	01	48/3 = 16 MHz	1	48/8 = 6 MHz
48	00	48/6 = 8 MHz	1	48/8 = 6 MHz
24	11	24 MHz	0	24/4 = 6 MHz
24	10	24/2 = 12 MHz	0	24/4 = 6 MHz
24	01	24/3 = 8 MHz	0	24/4 = 6 MHz
24	00	24/6 = 4 MHz	0	24/4 = 6 MHz

4.6.2 I/O PINS DURING DEEP SLEEP

During Deep Sleep, the general purpose I/O pins will retain their previous states.

Pins that are configured as inputs (TRIS bit set) prior to entry into Deep Sleep will remain high-impedance during Deep Sleep.

Pins that are configured as outputs (TRIS bit clear) prior to entry into Deep Sleep will remain as output pins during Deep Sleep. While in this mode, they will drive the output level determined by their corresponding LAT bit at the time of entry into Deep Sleep.

When the device wakes back up, the I/O pin behavior depends on the type of wake up source.

If the device wakes back up by an RTCC alarm, INT0 interrupt, DSWDT or ULPWU event, all I/O pins will continue to maintain their previous states, even after the device has finished the POR sequence and is executing application code again. Pins configured as inputs during Deep Sleep will remain high-impedance, and pins configured as outputs will continue to drive their previous value.

After waking up, the TRIS and LAT registers will be reset, but the I/O pins will still maintain their previous states. If firmware modifies the TRIS and LAT values for the I/O pins, they will not immediately go to the newly configured states. Once the firmware clears the RELEASE bit (DSCONL<0>), the I/O pins will be “released”. This causes the I/O pins to take the states configured by their respective TRIS and LAT bit values.

If the Deep Sleep BOR (DSBOR) circuit is enabled, and V_{DD} drops below the DSBOR and V_{DD} rail POR thresholds, the I/O pins will be immediately released similar to clearing the RELEASE bit. All previous state information will be lost, including the general purpose DSGPR0 and DSGPR1 contents. See **Section 4.6.5 “Deep Sleep Brown-Out Reset (DSBOR)”** for additional details regarding this scenario

If a $\overline{MCLR}$ Reset event occurs during Deep Sleep, the I/O pins will also be released automatically, but in this case, the DSGPR0 and DSGPR1 contents will remain valid.

In all other Deep Sleep wake-up cases, application firmware needs to clear the RELEASE bit in order to reconfigure the I/O pins.

4.6.3 DEEP SLEEP WAKE-UP SOURCES

The device can be awakened from Deep Sleep mode by a $\overline{MCLR}$, POR, RTCC, INT0 I/O pin interrupt, DSWDT or ULPWU event. After waking, the device performs a POR. When the device is released from Reset, code execution will begin at the device’s Reset vector.

The software can determine if the wake-up was caused from an exit from Deep Sleep mode by reading the DS bit (WDTCON<3>). If this bit is set, the POR was caused by a Deep Sleep exit. The DS bit must be manually cleared by the software.

The software can determine the wake event source by reading the DSWAKEH and DSWAKEL registers. When the application firmware is done using the DSWAKEH and DSWAKEL status registers, individual bits do not need to be manually cleared before entering Deep Sleep again. When entering Deep Sleep mode, these registers are automatically cleared.

4.6.3.1 Wake-up Event Considerations

Deep Sleep wake-up events are only monitored while the processor is fully in Deep Sleep mode. If a wake-up event occurs before Deep Sleep mode is entered, the event status will not be reflected in the DSWAKE registers. If the wake-up source asserts prior to entering Deep Sleep, the CPU will either go to the interrupt vector (if the wake source has an interrupt bit and the interrupt is fully enabled) or will abort the Deep Sleep entry sequence by executing past the SLEEP instruction if the interrupt was not enabled. In this case, a wake-up event handler should be placed after the SLEEP instruction to process the event and re-attempt entry into Deep Sleep if desired.

When the device is in Deep Sleep with more than one wake-up source simultaneously enabled, only the first wake-up source to assert will be detected and logged in the DSWAKEH/DSWAKEL status registers.

4.6.4 DEEP SLEEP WATCHDOG TIMER (DSWDT)

Deep Sleep has its own dedicated WDT (DSWDT) with a postscaler for time-outs of 2.1 ms to 25.7 days, configurable through the bits, DSWDTPS<3:0>.

The DSWDT can be clocked from either the INTRC or the T1OSC/T1CKI input. If the T1OSC/T1CKI source will be used with a crystal, the T1OSCEN bit in the T1CON register needs to be set prior to entering Deep Sleep. The reference clock source is configured through the DSWDTOSC bit.

DSWDT is enabled through the DSWDTEN bit. Entering Deep Sleep mode automatically clears the DSWDT. See **Section 28.0 “Special Features of the CPU”** for more information.

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
RPINR22	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR21	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR17	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR16	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR14	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR13	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR12	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR9	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR8	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR7	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR15	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR6	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR4	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR3	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR2	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPINR1	PIC18F2XJ53	PIC18F4XJ53	---1 1111	---1 1111	---u uuuu
RPOR24	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR23	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR22	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR21	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR20	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR19	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR18	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR17	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR13	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR12	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR11	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR10	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR9	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR8	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR7	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu
RPOR6	PIC18F2XJ53	PIC18F4XJ53	---0 0000	---0 0000	---u uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-1 for Reset value for specific condition.

5: Not implemented for PIC18F2XJ53 devices.

6: Not implemented for "LF" devices.

9.0 INTERRUPTS

Devices of the PIC18F47J53 family have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

There are 19 registers, which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3, PIR4, PIR5
- PIE1, PIE2, PIE3, PIE4, PIE5
- IPR1, IPR2, IPR3, IPR4, IPR5

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, interrupt sources have three bits to control their operation. They 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

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, 0008h or 0018h, 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 0008h 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 low-priority 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 (0008h or 0018h). Once in the Interrupt Service Routine (ISR), 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.

12.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

The T0CON register (Register 12-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

Figure 12-1 provides a simplified block diagram of the Timer0 module in 8-bit mode. Figure 12-2 provides a simplified block diagram of the Timer0 module in 16-bit mode.

REGISTER 12-1: T0CON: TIMER0 CONTROL REGISTER (ACCESS FD5h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
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 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS<2:0>**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

13.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Reset on ECCP Special Event Trigger
- Device clock status flag (SOSCRUN)
- Timer with gated control

Figure 13-1 displays a simplified block diagram of the Timer1 module.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 is controlled through the T1CON Control register (Register 13-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

The FOSC clock source (TMR1CS<1:0> = 01) should not be used with the ECCP capture/compare features. If the timer will be used with the capture or compare features, always select one of the other timer clocking options.

REGISTER 13-1: T1CON: TIMER1 CONTROL REGISTER (ACCESS FCDh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	RD16	TMR1ON
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 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
10 = Timer1 clock source is the T1OSC or T1CKI pin
01 = Timer1 clock source is the system clock (FOSC)(1)
00 = Timer1 clock source is the instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Source Select bit
When TMR1CS<1:0> = 10:
1 = Power up the Timer1 crystal driver and supply the Timer1 clock from the crystal output
0 = Timer1 crystal driver is off, Timer1 clock is from the T1CKI input pin(2)
When TMR1CS<1:0> = 0x:
1 = Power up the Timer1 crystal driver
0 = Timer1 crystal driver is off(2)

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit
TMR1CS<1:0> = 10:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS<1:0> = 0x:
This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 0x.

bit 1 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

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

Note 1: The FOSC clock source should not be selected if the timer will be used with the ECCP capture/compare features.
2: The Timer1 oscillator crystal driver is powered whenever T1OSCEN (T1CON<3>) or T3OSCEN (T3CON<3>) = 1. The circuit is enabled by the logical OR of these two bits. When disabled, the inverter and feedback resistor are disabled to eliminate power drain. The TMR1ON and TMR3ON bits do not have to be enabled to power up the crystal driver.

REGISTER 16-1: TxCON: TIMER4/6/8 CONTROL REGISTER (ACCESS F76h, BANKED F1Eh, BANKED F1Bh)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TxOUTPS3	TxOUTPS2	TxOUTPS1	TxOUTPS0	TMRxON	TxCKPS1	TxCKPS0
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 6-3 **TxOUTPS<3:0>:** Timerx Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMRxON**: Timerx On bit
1 = Timerx is on
0 = Timerx is off

bit 1-0 **TxCKPS<1:0>:** Timerx Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

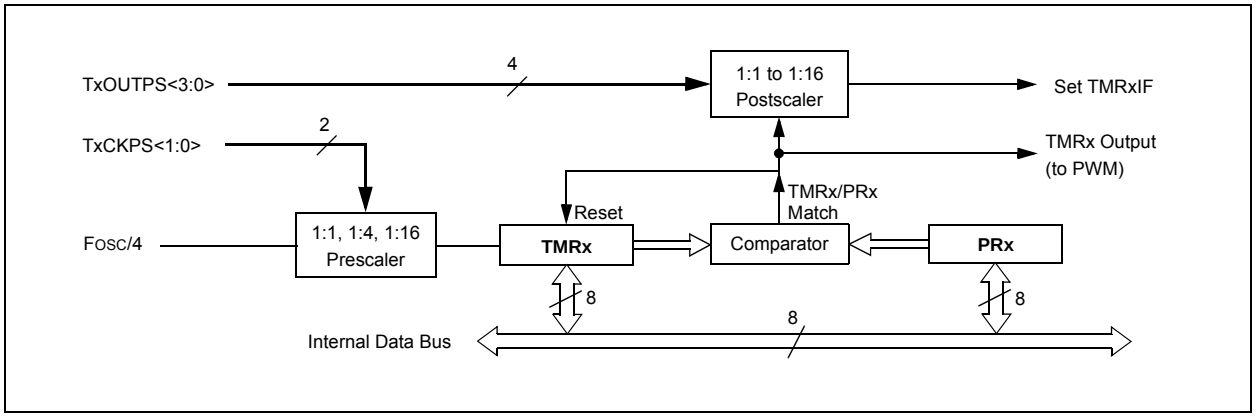
16.2 Timer4/6/8 Interrupt

The Timer4/6/8 modules have 8-bit Period registers, PRx, that are both readable and writable. Timer4/6/8 increment from 00h until they match PR4/6/8 and then reset to 00h on the next increment cycle. The PRx registers are initialized to FFh upon Reset.

16.3 Output of TMRx

The outputs of TMRx (before the postscaler) are used only as a PWM time base for the ECCP modules. They are not used as baud rate clocks for the MSSP modules as is the Timer2 output.

FIGURE 16-1: TIMER4 BLOCK DIAGRAM



17.1 RTCC MODULE REGISTERS

The RTCC module registers are divided into following categories:

RTCC Control Registers

- RTCCFG
- RTCCAL
- PADCFG1
- ALRMCFG
- ALRMRPT

RTCC Value Registers

- RTCVALH and RTCVALL – Can access the following registers
  - YEAR
  - MONTH
  - DAY
  - WEEKDAY
  - HOUR
  - MINUTE
  - SECOND

Alarm Value Registers

- ALRMVALH and ALRMVALL – Can access the following registers:
  - ALRMMNTH
  - ALRMDAY
  - ALRMWD
  - ALRMHR
  - ALRMMIN
  - ALRMSEC

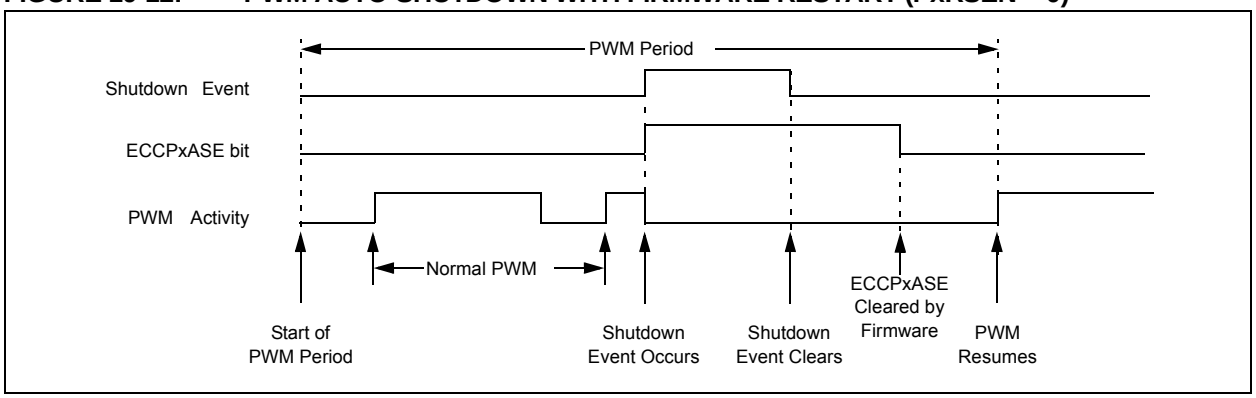
Note: The RTCVALH and RTCVALL registers can be accessed through RTCRPT<1:0>. ALRMVALH and ALRMVALL can be accessed through ALRMPTR<1:0>.

TABLE 18-4: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1/3/5/7

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
PIR4	CCP10IF	CCP9IF	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
PIE4	CCP10IE	CCP9IE	CCP8IE	CCP7IE	CCP6IE	CCP5IE	CCP4IE	CCP3IE
IPR4	CCP10IP	CCP9IP	CCP8IP	CCP7IP	CCP6IP	CCP5IP	CCP4IP	CCP3IP
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0
TRISE	RDPU	REPU	—	—	—	TRISE2	TRISE1	TRISE0
TMR1L	Timer1 Register Low Byte							
TMR1H	Timer1 Register High Byte							
TMR3L	Timer3 Register Low Byte							
TMR3H	Timer3 Register High Byte							
TMR5L	Timer5 Register Low Byte							
TMR5H	Timer5 Register High Byte							
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	RD16	TMR1ON
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	$\overline{T3SYNC}$	RD16	TMR3ON
T5CON	TMR5CS1	TMR5CS0	T5CKPS1	T5CKPS0	T5OSCEN	$\overline{T5SYNC}$	RD16	TMR5ON
CCPR4L	CCPR4L7	CCPR4L6	CCPR4L5	CCPR4L4	CCPR4L3	CCPR4L2	CCPR4L1	CCPR4L0
CCPR4H	CCPR4H7	CCPR4H6	CCPR4H5	CCPR4H4	CCPR4H3	CCPR4H2	CCPR4H1	CCPR4H0
CCPR5L	CCPR5L7	CCPR5L6	CCPR5L5	CCPR5L4	CCPR5L3	CCPR5L2	CCPR5L1	CCPR5L0
CCPR5H	CCPR5H7	CCPR5H6	CCPR5H5	CCPR5H4	CCPR5H3	CCPR5H2	CCPR5H1	CCPR5H0
CCPR6L	CCPR6L7	CCPR6L6	CCPR6L5	CCPR6L4	CCPR6L3	CCPR6L2	CCPR6L1	CCPR6L0
CCPR6H	CCPR6H7	CCPR6H6	CCPR6H5	CCPR6H4	CCPR6H3	CCPR6H2	CCPR6H1	CCPR6H0
CCPR7L	CCPR7L7	CCPR7L6	CCPR7L5	CCPR7L4	CCPR7L3	CCPR7L2	CCPR7L1	CCPR7L0
CCPR7H	CCPR7H7	CCPR7H6	CCPR7H5	CCPR7H4	CCPR7H3	CCPR7H2	CCPR7H1	CCPR7H0
CCPR8L	CCPR8L7	CCPR8L6	CCPR8L5	CCPR8L4	CCPR8L3	CCPR8L2	CCPR8L1	CCPR8L0
CCPR8H	CCPR8H7	CCPR8H6	CCPR8H5	CCPR8H4	CCPR8H3	CCPR8H2	CCPR8H1	CCPR8H0
CCPR9L	CCPR9L7	CCPR9L6	CCPR9L5	CCPR9L4	CCPR9L3	CCPR9L2	CCPR9L1	CCPR9L0
CCPR9H	CCPR9H7	CCPR9H6	CCPR9H5	CCPR9H4	CCPR9H3	CCPR9H2	CCPR9H1	CCPR9H0
CCPR10L	CCPR10L7	CCPR10L6	CCPR10L5	CCPR10L4	CCPR10L3	CCPR10L2	CCPR10L1	CCPR10L0
CCPR10H	CCPR10H7	CCPR10H6	CCPR10H5	CCPR10H4	CCPR10H3	CCPR10H2	CCPR10H1	CCPR10H0
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0
CCP6CON	—	—	DC6B1	DC6B0	CCP6M3	CCP6M2	CCP6M1	CCP6M0
CCP7CON	—	—	DC7B1	DC7B0	CCP7M3	CCP7M2	CCP7M1	CCP7M0
CCP8CON	—	—	DC8B1	DC8B0	CCP8M3	CCP8M2	CCP8M1	CCP8M0
CCP9CON	—	—	DC9B1	DC9B0	CCP9M3	CCP9M2	CCP9M1	CCP9M0
CCP10CON	—	—	DC10B1	DC10B0	CCP10M3	CCP10M2	CCP10M1	CCP10M0
CCPTMRS1	C7TSEL1	C7TSEL0	—	C6TSEL0	—	C5TSEL0	C4TSEL1	C4TSEL0
CCPTMRS2	—	—	—	C10TSEL0	—	C9TSEL0	C8TSEL1	C8TSEL0

Legend: — = unimplemented, read as '0'. Shaded cells are not used by Capture/Compare or Timer1/3/5.

FIGURE 19-12: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PxRSEN = 0)



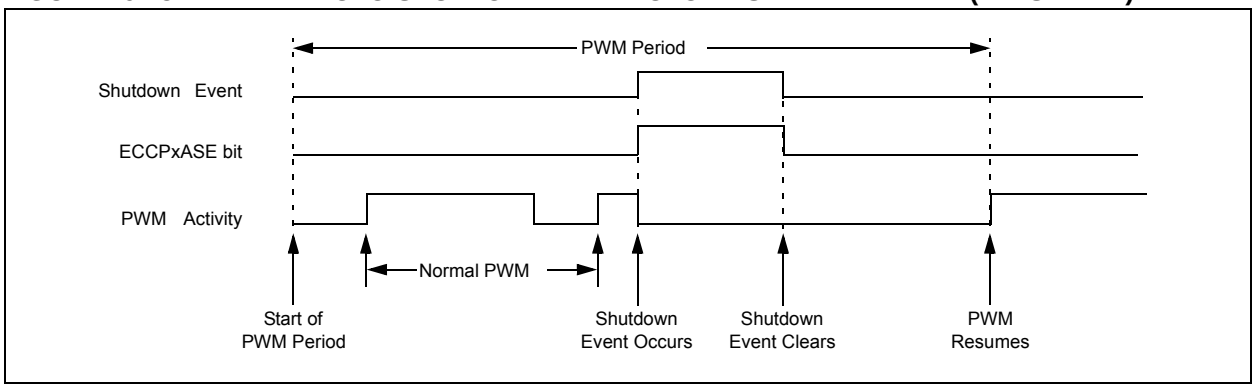
19.4.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PxRSEN bit (ECCPxDEL<7>).

If auto-restart is enabled, the ECCPxASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the ECCPxASE bit will be cleared via hardware and normal operation will resume.

The module will wait until the next PWM period begins, however, before re-enabling the output pin. This behavior allows the auto-shutdown with auto-restart features to be used in applications based on the current mode of PWM control.

FIGURE 19-13: PWM AUTO-SHUTDOWN WITH AUTO-RESTART ENABLED (PxRSEN = 1)



19.4.8 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2 will not increment and the state of the module will not change. If the ECCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from HFINTOSC and the postscaler may not be immediately stable.

In PRI_IDLE mode, the primary clock will continue to clock the ECCPx module without change.

19.4.8.1 Operation with Fail-Safe Clock Monitor (FSCM)

If the Fail-Safe Clock Monitor (FSCM) is enabled, a clock failure will force the device into the power-managed RC_RUN mode and the OSCFIF bit of the PIR2 register will be set. The ECCPx will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

19.4.9 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the ECCP registers to their Reset states.

This forces the ECCP module to reset to a state compatible with previous, non-enhanced CCP modules used on other PIC18 and PIC16 devices.

TABLE 19-4: REGISTERS ASSOCIATED WITH ECCP1/2/3 MODULE AND TIMER1/2/3/4/6/8

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR4	CCP10IF	CCP9IF	CCP8IF	CCP7IF	CCP6IF	CCP5IF	CCP4IF	CCP3IF
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE4	CCP10IE	CCP9IE	CCP8IE	CCP7IE	CCP6IE	CCP5IE	CCP4IE	CCP3IE
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR4	CCP10IP	CCP9IP	CCP8IP	CCP7IP	CCP6IP	CCP5IP	CCP4IP	CCP3IP
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0
TRISE	RDPU	REPU	—	—	—	TRISE2	TRISE1	TRISE0
TMR1H	Timer1 Register High Byte							
TMR1L	Timer1 Register Low Byte							
TMR2	Timer2 Register							
TMR3H	Timer3 Register High Byte							
TMR3L	Timer3 Register Low Byte							
TMR4	Timer4 Register							
TMR6	Timer6 Register							
TMR8	Timer8 Register							
PR2	Timer2 Period Register							
PR4	Timer4 Period Register							
PR6	Timer6 Period Register							
PR8	Timer8 Period Register							
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	RD16	TMR1ON
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	$\overline{T3SYNC}$	RD16	TMR3ON
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
T6CON	—	T6OUTPS3	T6OUTPS2	T6OUTPS1	T6OUTPS0	TMR6ON	T6CKPS1	T6CKPS0

21.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>) are set while the Transmit Shift Register is loaded with data.

Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 21-10 for the timing of the Break character sequence.

21.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

21.2.6 RECEIVING A BREAK CHARACTER

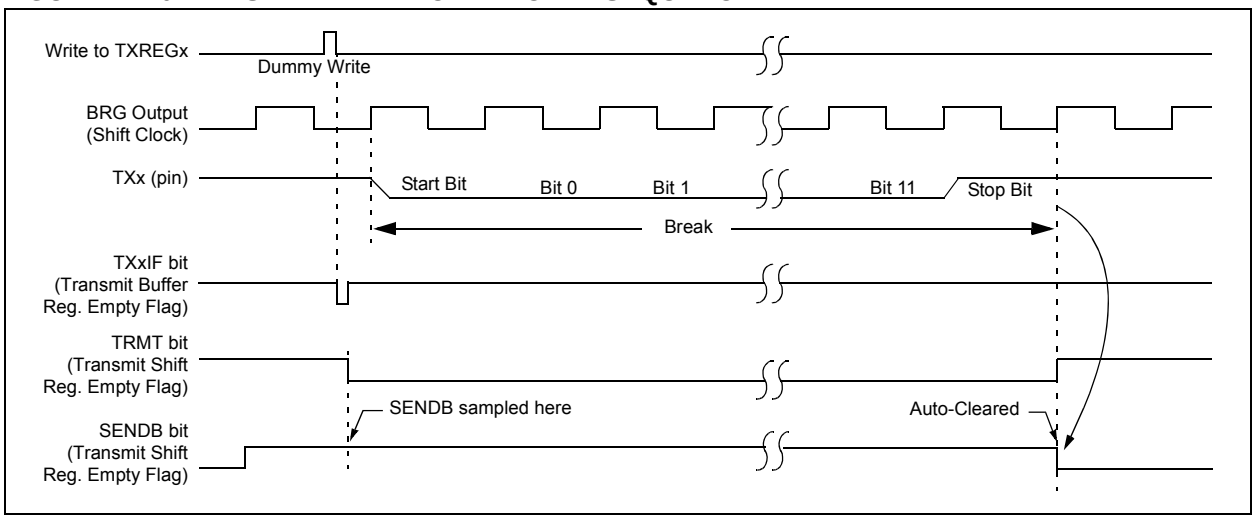
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 of the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 21.2.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.

FIGURE 21-10: SEND BREAK CHARACTER SEQUENCE



22.0 10/12-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module in the PIC18F47J53 family of devices has 10 inputs for the 28-pin devices and 13 inputs for the 44-pin devices. This module allows conversion of an analog input signal to a corresponding 10 or 12-bit digital number.

The module has these registers:

- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Port Configuration Register 0 (ANCON0)
- A/D Port Configuration Register 1 (ANCON1)
- A/D Result Registers (ADRESH and ADRESL)
- A/D Trigger Register (ADCTRIG)
- Configuration Register 3 High (ADCSEL, CONFIG3H<1>)

The ADCON0 register, shown in Register 22-1, controls the operation of the A/D module.

The ADCON1 register, shown in Register 22-1, configures the A/D clock source, programmed acquisition time and justification. The ANCON0 and ANCON1 registers, in Register 22-1 and Register 22-2, configure the functions of the port pins.

The ADCSEL Configuration bit (CONFIG3H<1>) sets the module for 10 or 12-bit conversions. The 10-Bit Conversion mode is useful for applications that favor conversion speed over conversion resolution.

MULLW **Multiply Literal with W**

Syntax: MULLW k

Operands: $0 \le k \le 255$

Operation: (W) x k → PRODH:PRODL

Status Affected: None

Encoding:

0000	1101	kkkk	kkkk

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte.

W is unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write registers PRODH: PRODL

Example: MULLW 0C4h

Before Instruction
W = E2h
PRODH = ?
PRODL = ?
After Instruction
W = E2h
PRODH = ADh
PRODL = 08h

MULWF **Multiply W with f**

Syntax: MULWF f {,a}

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

Operation: (W) x (f) → PRODH:PRODL

Status Affected: None

Encoding:

0000	001a	ffff	ffff

Description: An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

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 29.2.3 "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 registers PRODH: PRODL

Example: MULWF REG, 1

Before Instruction
W = C4h
REG = B5h
PRODH = ?
PRODL = ?
After Instruction
W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

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 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 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' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

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 29.2.3 "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

31.2 DC Characteristics: Power-Down and Supply Current PIC18F47J53 Family (Industrial) (Continued)

PIC18LF47J53 Family		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J53 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param. No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (IDD)(2)						
	PIC18LFXXJ53	1.45	3.0	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 8 MHz, **RC_RUN** mode, Internal RC Oscillator
		1.48	3.0	mA	+25°C		
		1.52	3.0	mA	+85°C		
	PIC18LFXXJ53	2.12	3.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		2.10	3.8	mA	+25°C		
		2.10	3.9	mA	+85°C		
	PIC18FXXJ53	1.65	3.6	mA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		1.68	3.6	mA	+25°C		
		1.71	3.9	mA	+85°C		
	PIC18FXXJ53	2.26	4.3	mA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		2.13	4.3	mA	+25°C		
		2.10	4.6	mA	+85°C		
	PIC18LFXXJ53	1.5	11	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 31 kHz, **RC_IDLE** mode, Internal RC Oscillator, INTSRC = 0
		1.7	11	µA	+25°C		
		3.9	20	µA	+85°C		
	PIC18LFXXJ53	2.0	12	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		2.4	13	µA	+25°C		
		5.4	23	µA	+85°C		
	PIC18FXXJ53	19.0	60	µA	-40°C	VDD = 2.15V, VDDCORE = 10 µF	
		23.2	60	µA	+25°C		
		29.9	75	µA	+85°C		
	PIC18FXXJ53	19.7	65	µA	-40°C	VDD = 3.3V, VDDCORE = 10 µF	
		24.0	65	µA	+25°C		
		31.4	90	µA	+85°C		

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

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

3: Low-power Timer1 with standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: This is the module differential current when the USB module is enabled and clocked at 48 MHz, but with no USB cable attached. When the USB cable is attached or data is being transmitted, the current consumption may be much higher (see **Section 23.6.4 "USB Transceiver Current Consumption"**). During USB Suspend mode (USBEN = 1, SUSPND = 1, bus in Idle state), the USB module current will be dominated by the D+ or D- pull-up resistor. The integrated pull-up resistors use "resistor switching" according to the `resistor_ecn` supplement to the USB 2.0 Specifications, and therefore, may be as low as 900Ω during Idle conditions.

TABLE 31-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS
TABLE 4-2:

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)

Param. No.	Sym.	Characteristics	Min.	Typ.	Max.	Units	Comments
	VRGOUT	Regulator Output Voltage	2.35	2.5	2.7	V	Regulator enabled, VDD = 3.0V
	CEFC	External Filter Capacitor Value(1)	5.4	10	18	µF	ESR < 3Ω recommended ESR < 5Ω required

Note 1: CEFC applies for PIC18**F** devices in the family. For PIC18**LF** devices in the family, there is no specific minimum or maximum capacitance for VDDCORE, although proper supply rail bypassing should still be used.

TABLE 31-5: ULPWU SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D100	IULP	**Ultra Low-Power Wake-up Current**	—	60	—	nA	Net of I/O leakage and current sink at 1.6V on pin, VDD = 3.3V See Application Note *AN879,* "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879)

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 31-6: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.(1)**	**Max.**	**Units**	**Conditions**
	IOUT1	CTMU Current Source, Base Range	—	550	—	nA	CTMUICON<1:0> = 01
	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	µA	CTMUICON<1:0> = 10
	IOUT3	CTMU Current Source, 100x Range	—	55	—	µA	CTMUICON<1:0> = 11

Note 1: Nominal value at center point of current trim range (CTMUICON<7:2> = 000000).

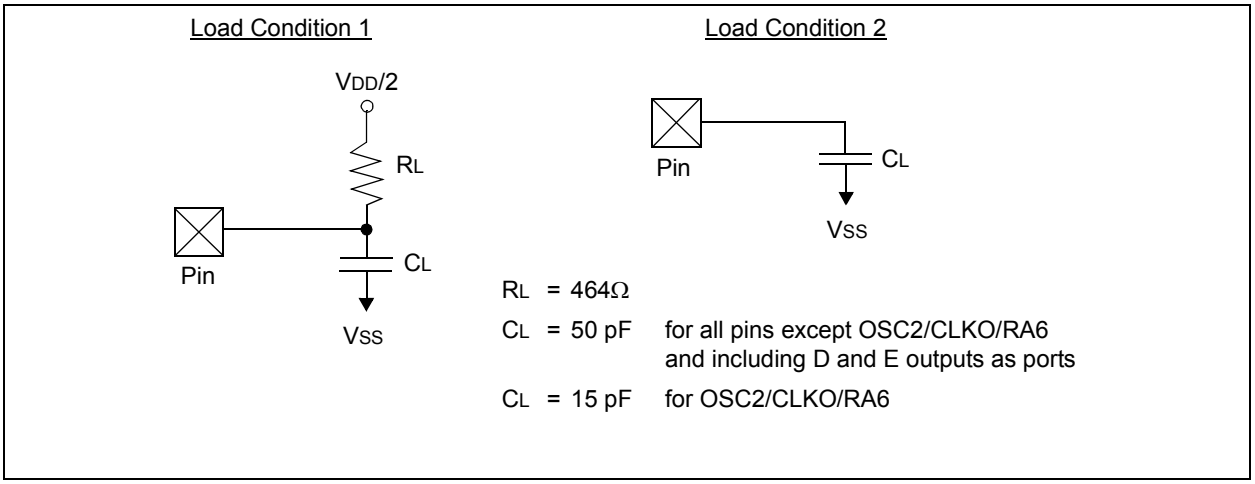
31.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 31-9 apply to all timing specifications unless otherwise noted. Figure 31-4 specifies the load conditions for the timing specifications.

TABLE 31-9: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial Operating voltage VDD range as described in **Section 31.1** and **Section 31.3**.

FIGURE 31-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



31.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 31-5: EXTERNAL CLOCK TIMING

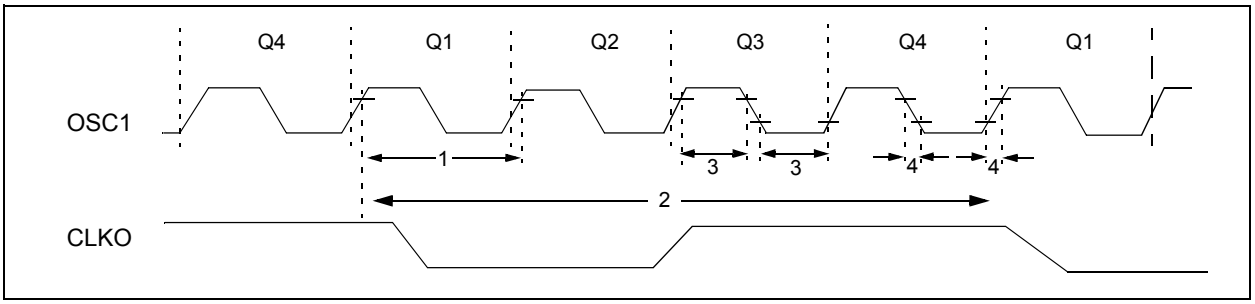
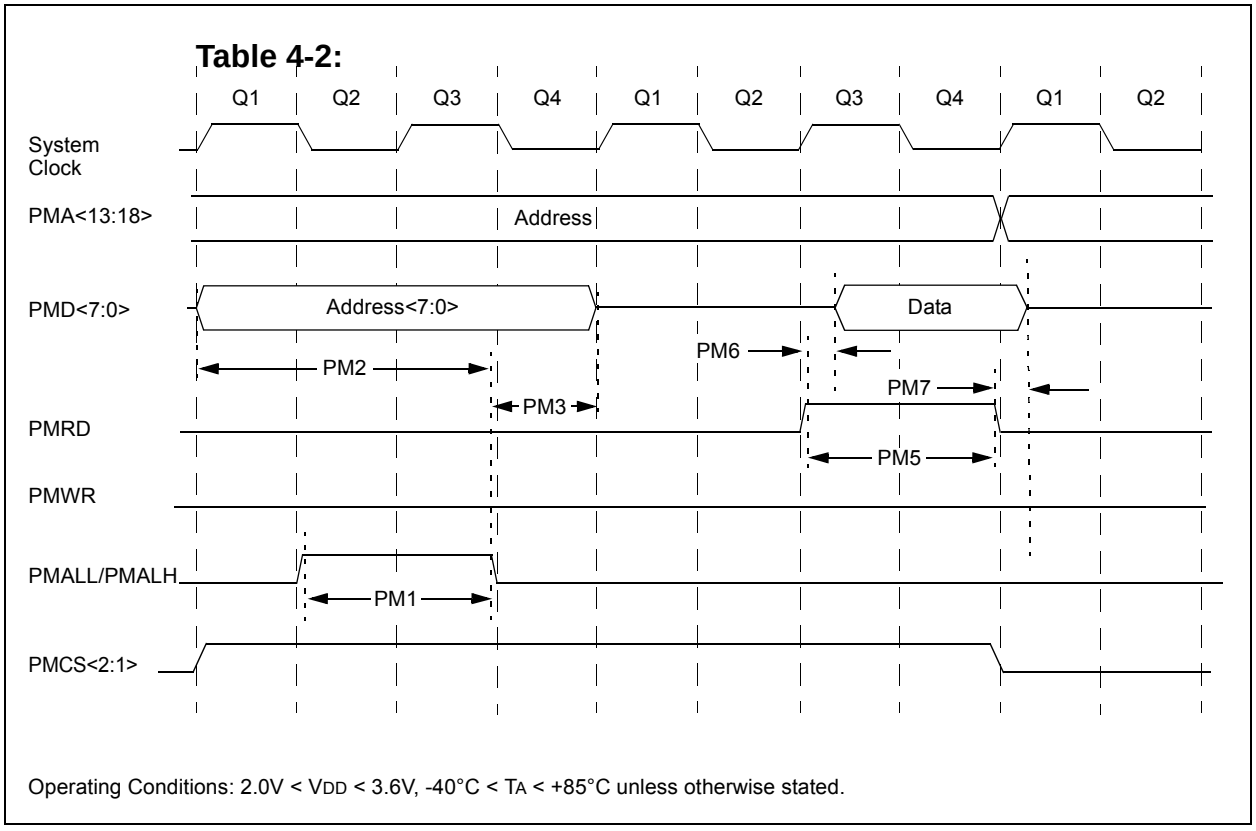


TABLE 31-17: ENHANCED CAPTURE/COMPARE/PWM REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
50	TccL	ECCPx Input Low Time	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
51	TccH	ECCPx Input High Time	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
52	TccP	ECCPx Input Period 3 TCY + 40 N		$\frac{3\ T_{CY} + 40}{N}$	—	ns	N = prescale value (1, 4 or 16)
53	TccR	ECCPx Output Fall Time		—	25	ns	
54	TccF	ECCPx Output Fall Time		—	25	ns	

FIGURE 31-10: PARALLEL MASTER PORT READ TIMING DIAGRAM



Operating Conditions: 2.0V < VDD < 3.6V, -40°C < TA < +85°C unless otherwise stated.