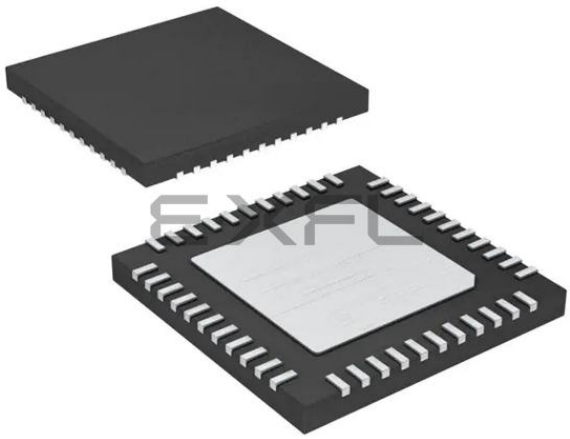


Welcome to E-XFL.COM

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
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 2.75V
Data Converters	A/D 13x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46j13t-i-ml

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 = PLL is enabled
0 = 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.

The use of the flag and control bits in the OSCCON register is discussed in more detail in **Section 4.0 "Low-Power Modes"**.

Note 1: The Timer1 crystal driver is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select the Timer1 clock source will be ignored, unless the CONFIG2L register's SOSCSEL<1:0> bits are set to Digital mode.

2: If Timer1 is driving a crystal, it is recommended that the Timer1 oscillator be operating and stable prior to switching to it as the clock source; otherwise, a very long delay may occur while the Timer1 oscillator starts.

3.3.2 OSCILLATOR TRANSITIONS

PIC18F47J13 Family devices contain circuitry to prevent clock "glitches" when switching between clock sources. A short pause in the device clock occurs during the clock switch. The length of this pause is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Clock transitions are discussed in more detail in **Section 4.1.2 "Entering Power-Managed Modes"**.

REGISTER 3-2: OSCCON: OSCILLATOR CONTROL REGISTER (ACCESS FD3h)

R/W-0	R/W-1	R/W-1	R/W-0	R-1(1)	U-1	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	—	SCS1	SCS0
bit 7							bit 0

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

bit 7 **IDLEN:** Idle Enable bit
1 = Device enters Idle mode on SLEEP instruction
0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
When using INTOSC to drive the 4x PLL, select 8 MHz or 4 MHz only to avoid operating the 4x PLL outside of specification.
111 = 8 MHz (INTOSC drives clock directly)
110 = 4 MHz(2)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (from either INTOSC/256 or INTRC directly)(3)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)
1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running
0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready

bit 2 **Unimplemented:** Read as '1'

bit 1-0 **SCS<1:0>:** System Clock Select bits
11 = Postscaled internal clock (INTRC/INTOSC derived)
10 = Reserved
01 = Timer1 oscillator
00 = Primary clock source (INTOSC postscaler output when FOSC<2:0> = 001 or 000)
00 = Primary clock source (CPU divider output for other values of FOSC<2:0>)

Note 1: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.
2: Default output frequency of INTOSC on Reset (4 MHz).
3: Source selected by the INTSRC bit (OSCTUNE<7>).

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	MCLR Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
TOSU	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu(1)
TOSH	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu(1)
TOSL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu(1)
STKPTR	PIC18F2XJ13	PIC18F4XJ13	00-0 0000	uu-0 0000	uu-u uuuu(1)
PCLATU	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	PIC18F2XJ13	PIC18F4XJ13	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F2XJ13	PIC18F4XJ13	0000 000x	0000 000u	uuuu uuuu(3)
INTCON2	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu(3)
INTCON3	PIC18F2XJ13	PIC18F4XJ13	1100 0000	1100 0000	uuuu uuuu(3)
INDF0	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
POSTINC0	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
POSTDEC0	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
PREINC0	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
PLUSW0	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
FSR0H	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
FSR0L	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
POSTINC1	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
POSTDEC1	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
PREINC1	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
PLUSW1	PIC18F2XJ13	PIC18F4XJ13	N/A	N/A	N/A
FSR1H	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
FSR1L	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- 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 the Reset value for a specific condition.

5: Not implemented on PIC18F2XJ13 devices.

6: Not implemented on "LF" devices.

6.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Specifically, the use of the Access Bank for many of the core PIC18 instructions is different. This is due to the introduction of a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode; inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

6.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under proper conditions, instructions that use the Access Bank, that is, most bit and byte-oriented instructions, can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset or Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- The use of the Access Bank is forced ('a' = 0); and
- The file address argument is less than or equal to 5Fh.

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

6.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte and bit-oriented instructions, or almost one half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte and bit-oriented instructions are not affected if they do not use the Access Bank (Access RAM bit is '1') or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is provided in Figure 6-9.

Those who desire to use byte or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 28.2.1 "Extended Instruction Syntax"**.

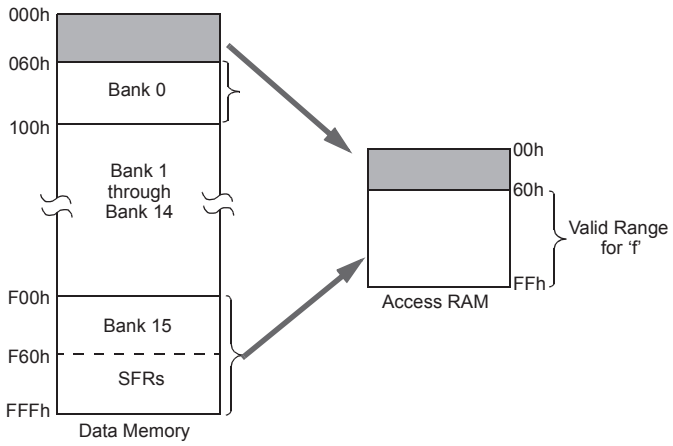
FIGURE 6-9: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

Example Instruction: `ADDWF, f, d, a` (Opcode: `0010 01da ffff ffff`)

When a = 0 and f ≥ 60h:

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and FFFh. This is the same as locations, F60h to FFFh (Bank 15), of data memory.

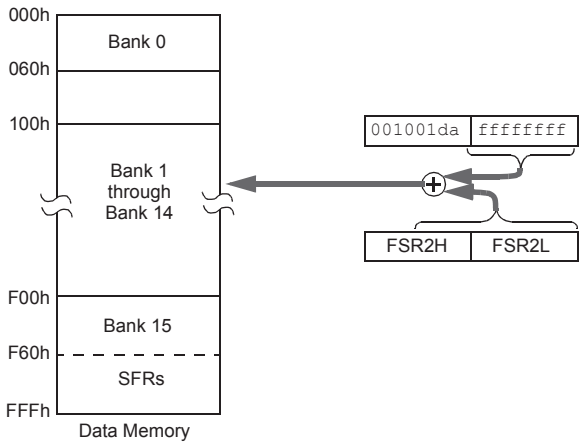
Locations below 060h are not available in this addressing mode.



When a = 0 and f ≤ 5Fh:

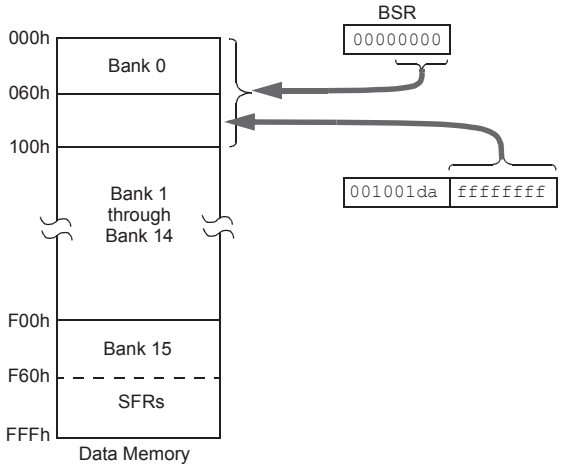
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

Note that in this mode, the correct syntax is:
`ADDWF [k], d`
where 'k' is the same as 'f'.



When a = 1 (all values of f):

The instruction executes in Direct mode (also known as Direct Long mode). 'f' is interpreted as a location in one of the 16 banks of the data memory space. The bank is designated by the Bank Select Register (BSR). The address can be in any implemented bank in the data memory space.



EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY

```
                    MOVLW   CODE_ADDR_UPPER       ; Load TBLPTR with the base address
                    MOVWF   TBLPTRU               ; of the memory block, minus 1
                    MOVLW   CODE_ADDR_HIGH
                    MOVWF   TBLPTRH
                    MOVLW   CODE_ADDR_LOW
                    MOVWF   TBLPTRL
ERASE_BLOCK
                    BSF     EECON1, WREN          ; enable write to memory
                    BSF     EECON1, FREE          ; enable Erase operation
                    BCF     INTCON, GIE           ; disable interrupts
                    MOVLW   0x55
                    MOVWF   EECON2                ; write 55h
                    MOVLW   0xAA
                    MOVWF   EECON2                ; write 0AAh
                    BSF     EECON1, WR            ; start erase (CPU stall)
                    BSF     INTCON, GIE           ; re-enable interrupts
                    MOVLW   D'16'
                    MOVWF   WRITE_COUNTER         ; Need to write 16 blocks of 64 to write
                                                  ; one erase block of 1024
RESTART_BUFFER
                    MOVLW   D'64'
                    MOVWF   COUNTER
                    MOVLW   BUFFER_ADDR_HIGH      ; point to buffer
                    MOVWF   FSR0H
                    MOVLW   BUFFER_ADDR_LOW
                    MOVWF   FSR0L
FILL_BUFFER
                    ...                           ; read the new data from I2C, SPI,
                                                  ; PSP, USART, etc.
WRITE_BUFFER
                    MOVLW   D'64'                 ; number of bytes in holding register
                    MOVWF   COUNTER
WRITE_BYTE_TO_HREGS
                    MOVFF   POSTINC0, WREG        ; get low byte of buffer data
                    MOVWF   TABLAT                ; present data to table latch
                    TBLWT+*                       ; write data, perform a short write
                                                  ; to internal TBLWT holding register.
                    DECFSZ  COUNTER               ; loop until buffers are full
                    BRA     WRITE_BYTE_TO_HREGS
PROGRAM_MEMORY
                    BSF     EECON1, WREN          ; enable write to memory
                    BCF     INTCON, GIE           ; disable interrupts
                    MOVLW   0x55
   Required         MOVWF   EECON2                ; write 55h
   Sequence         MOVLW   0xAA
                    MOVWF   EECON2                ; write 0AAh
                    BSF     EECON1, WR            ; start program (CPU stall)
                    BSF     INTCON, GIE           ; re-enable interrupts
                    BCF     EECON1, WREN          ; disable write to memory

                    DECFSZ  WRITE_COUNTER         ; done with one write cycle
                    BRA     RESTART_BUFFER        ; if not done replacing the erase block
```

REGISTER 17-2: RTCCAL: RTCC CALIBRATION REGISTER (BANKED F3Eh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
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 7-0 **CAL<7:0>:** RTCC Drift Calibration bits

01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every minute
.
.
.
00000001 = Minimum positive adjustment; adds four RTCC clock pulses every minute
00000000 = No adjustment
11111111 = Minimum negative adjustment; subtracts four RTCC clock pulses every minute
.
.
.
10000000 = Maximum negative adjustment; subtracts 512 RTCC clock pulses every minute

REGISTER 17-3: PADCFG1: PAD CONFIGURATION REGISTER 1 (BANKED F3Ch)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	RTSECSEL1(1)	RTSECSEL0(1)	PMPTTL(2)
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 7-3 **Unimplemented:** Read as '0'

bit 2-1 **RTSECSEL<1:0>:** RTCC Seconds Clock Output Select bits(1)

11 = Reserved; do not use
10 = RTCC source clock is selected for the RTCC pin (pin can be INTRC or T1OSC, depending on the RTCOSC (CONFIG3L<1>) setting)
01 = RTCC seconds clock is selected for the RTCC pin
00 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit(2)

1 = PMP module uses TTL input buffers
0 = PMP module uses Schmitt input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RTCCFG<2>) bit must be set.

2: Available only on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13). For 28-pin devices, the bit is U-0.

18.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP4IE bit (PIE4<1>) clear to avoid false interrupts and should clear the flag bit, CCP4IF, following any such change in operating mode.

18.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCP4M<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Doing that also will not clear the prescaler counter – meaning the first capture may be from a non-zero prescaler.

Example 18-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 18-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP4CON       ; Turn CCP module off
MOVLW   NEW_CAPT_PS   ; Load WREG with the
                      ; new prescaler mode
                      ; value and CCP ON
MOVWF   CCP4CON       ; Load CCP4CON with
                      ; this value
```

18.3 Compare Mode

In Compare mode, the 16-bit CCPR4 register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCP4 pin can be:

- Driven high
- Driven low
- Toggled (high-to-low or low-to-high)
- Unchanged (that is, reflecting the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCP4M<3:0>). At the same time, the interrupt flag bit, CCP4IF, is set.

Figure 18-2 gives the Compare mode block diagram

18.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRISx bit.

Note:	Clearing the CCP4CON register will force the RB4 compare output latch (depending on device configuration) to the default low level. This is not the PORTB I/O data latch.

18.3.2 TIMER1/3/5 MODE SELECTION

If the CCP module is using the compare feature in conjunction with any of the Timer1/3/5 timers, the timers must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the compare operation may not work.

Note:	Details of the timer assignments for the CCP modules are given in Table 18-2 and Table 18-3.

18.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCP4M<3:0> = 1010), the CCP4 pin is not affected. Only a CCP interrupt is generated, if enabled, and the CCP4IE bit is set.

18.3.4 SPECIAL EVENT TRIGGER

Both CCP modules are equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCP4M<3:0> = 1011).

For either CCP module, the Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module’s time base. This allows the CCPRx registers to serve as a programmable period register for either timer.

The Special Event Trigger for CCP4 cannot start an A/D conversion.

Note:	The Special Event Trigger of ECCP1 can start an A/D conversion, but the A/D Converter must be enabled. For more information, see **Section 19.0 “Enhanced Capture/Compare/PWM (ECCP) Module”**.

19.4 PWM (Enhanced Mode)

The Enhanced PWM mode can generate a PWM signal on up to four different output pins with up to 10 bits of resolution. It can do this through four different PWM Output modes:

- Single PWM mode
- Half-Bridge PWM mode
- Full-Bridge PWM, Forward mode
- Full-Bridge PWM, Reverse mode

To select an Enhanced PWM mode, the PxM bits of the CCPxCON register must be set appropriately.

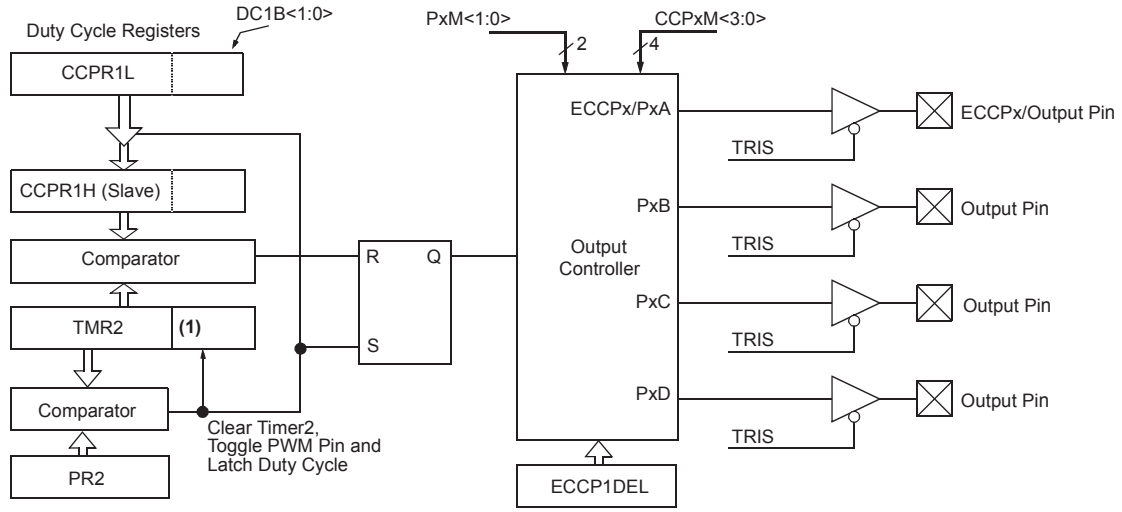
The PWM outputs are multiplexed with I/O pins and are designated: PxA, PxB, PxC and PxD. The polarity of the PWM pins is configurable and is selected by setting the CCPxM bits in the CCPxCON register appropriately.

Table 19-1 provides the pin assignments for each Enhanced PWM mode.

Figure 19-3 provides an example of a simplified block diagram of the Enhanced PWM module.

Note:	To prevent the generation of an incomplete waveform when the PWM is first enabled, the ECCP module waits until the start of a new PWM period before generating a PWM signal.

FIGURE 19-3: SIMPLIFIED BLOCK DIAGRAM OF THE ENHANCED PWM MODE EXAMPLE



Note 1: The 8-bit timer, TMR2 register, is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

Note 1:	The TRIS register value for each PWM output must be configured appropriately.
2:	Any pin not used by an Enhanced PWM mode is available for alternate pin functions.

20.5.4 CLOCK STRETCHING

Both 7-Bit and 10-Bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPxCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCLx pin to be held low at the end of each data receive sequence.

20.5.4.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode, on the falling edge of the ninth clock at the end of the ACK sequence, if the BF bit is set, the CKP bit in the SSPxCON1 register is automatically cleared, forcing the SCLx output to be held low. The CKP bit being cleared to '0' will assert the SCLx line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and read the contents of the SSPxBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 20-15).

Note 1: If the user reads the contents of the SSPxBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

20.5.4.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/W bit cleared to '0'. The release of the clock line occurs upon updating SSPxADD. Clock stretching will occur on each data receive sequence, as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPxADD register before the falling edge of the ninth clock occurs, and if the user has not cleared the BF bit by reading the SSPxBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

20.5.4.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit, after the falling edge of the ninth clock, if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's Interrupt Service Routine (ISR) must set the CKP bit before transmission is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and load the contents of the SSPxBUF before the master device can initiate another transmit sequence (see Figure 20-10).

Note 1: If the user loads the contents of SSPxBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit.

20.5.4.4 Clock Stretching for 10-Bit Slave Transmit Mode

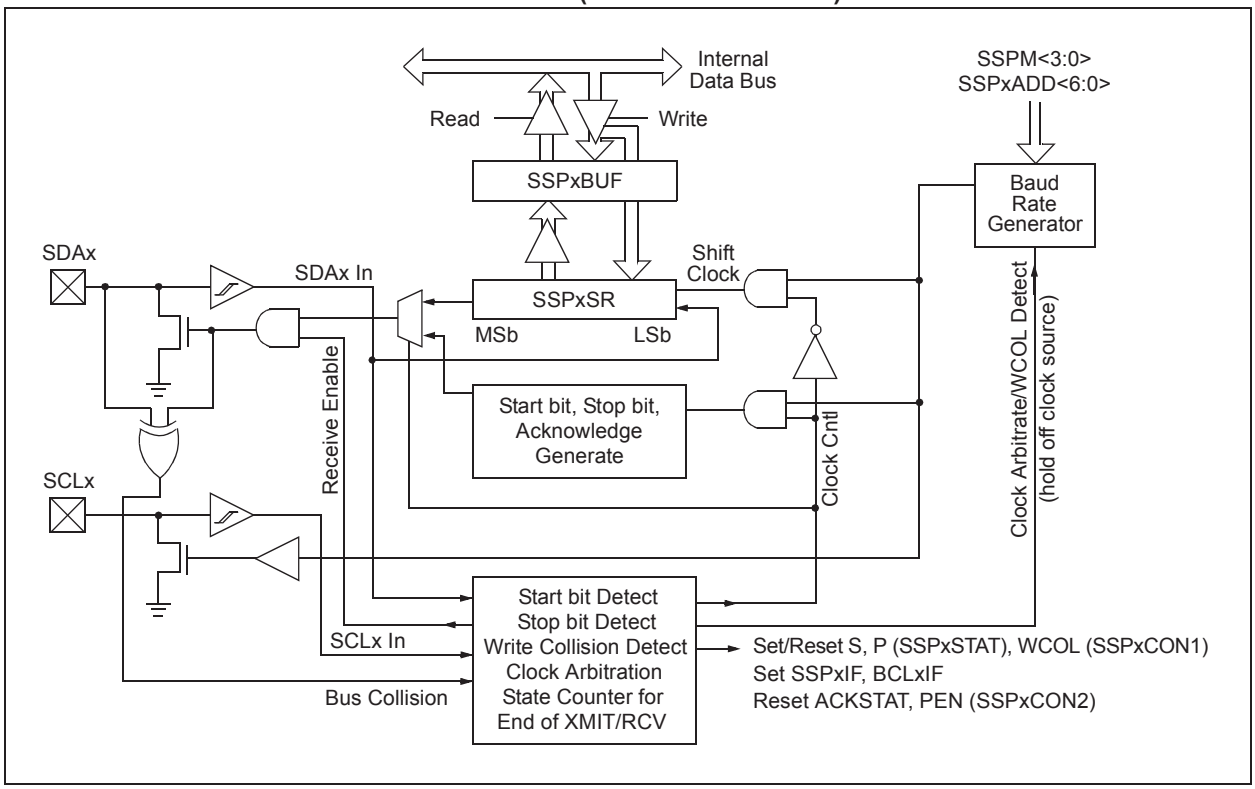
In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-Bit Slave Transmit mode (see Figure 20-13).

Note: The MSSP module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

The following events will cause the MSSP Interrupt Flag bit, SSPxIF, to be set (and MSSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 20-18: MSSPx BLOCK DIAGRAM (I^2C MASTER MODE)



20.5.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDAx or SCLx is sampled low at the beginning of the Start condition (Figure 20-28).

b) SCLx is sampled low before SDAx is asserted low (Figure 20-29).

During a Start condition, both the SDAx and the SCLx pins are monitored.

If the SDAx pin is already low, or the SCLx pin is already low, then all of the following occur:

- The Start condition is aborted
- The BCLxIF flag is set
- The MSSP module is reset to its inactive state (Figure 20-28)

The Start condition begins with the SDAx and SCLx pins deasserted. When the SDAx pin is sampled high, the BRG is loaded from SSPxADD<6:0> and counts down to 0. If the SCLx pin is sampled low while SDAx is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDAx pin is sampled low during this count, the BRG is reset and the SDAx line is asserted early (Figure 20-30). If, however, a '1' is sampled on the SDAx pin, the SDAx pin is asserted low at the end of the BRG count. The BRG is then reloaded and counts down to 0. If the SCLx pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCLx pin is asserted low.

Note: The reason that a bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDAx before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 20-28: BUS COLLISION DURING START CONDITION (SDAx ONLY)

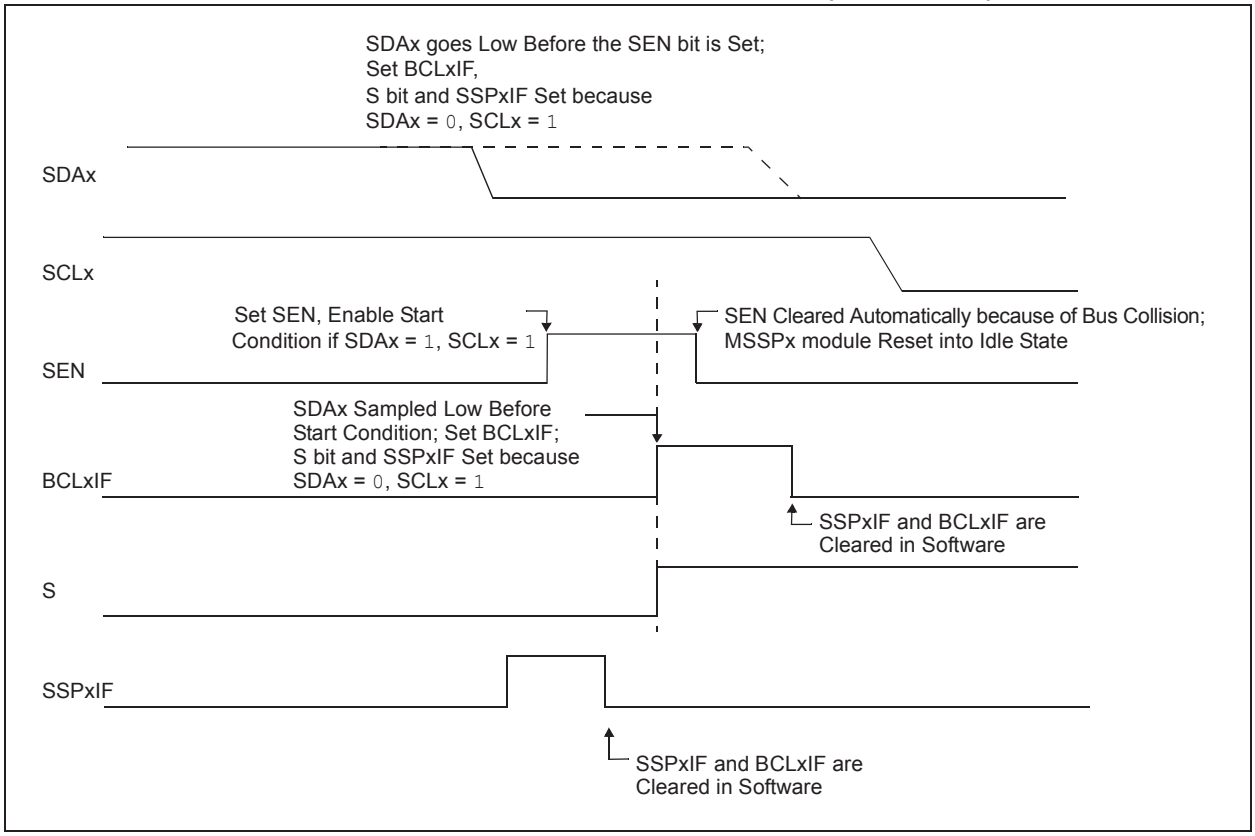


FIGURE 22-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

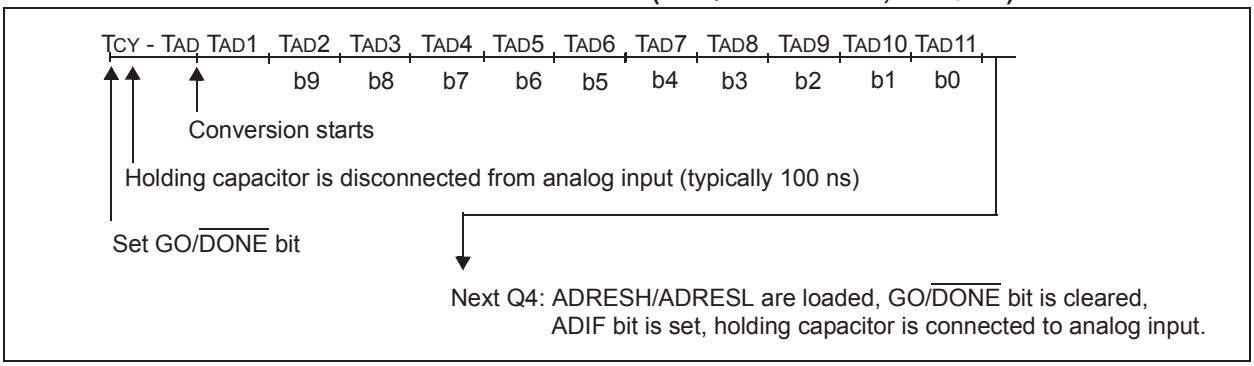
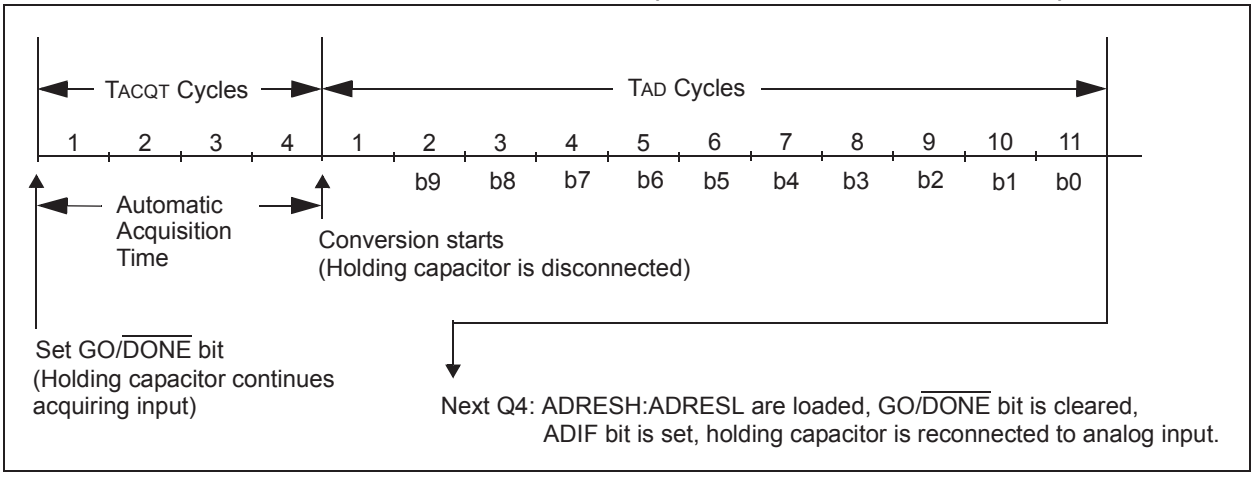


FIGURE 22-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



22.7 A/D Converter Calibration

The A/D Converter in the PIC18F47J13 family of devices includes a self-calibration feature, which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON1<6>). The next time the GO/DONE bit is set, the module will perform an offset calibration and store the result internally. Thus, subsequent offsets will be compensated.

Example 22-1 provides an example of a calibration routine.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset or if there are other major changes in operating conditions.

23.0 COMPARATOR MODULE

The analog comparator module contains three comparators that can be independently configured in a variety of ways. The inputs can be selected from the analog inputs and two internal voltage references. The digital outputs are available at the pin level and can also be read through the control register. Multiple output and interrupt event generation is also available. Figure 23-1 provides a generic single comparator from the module.

Key features of the module are:

- Independent comparator control
- Programmable input configuration
- Output to both pin and register levels
- Programmable output polarity

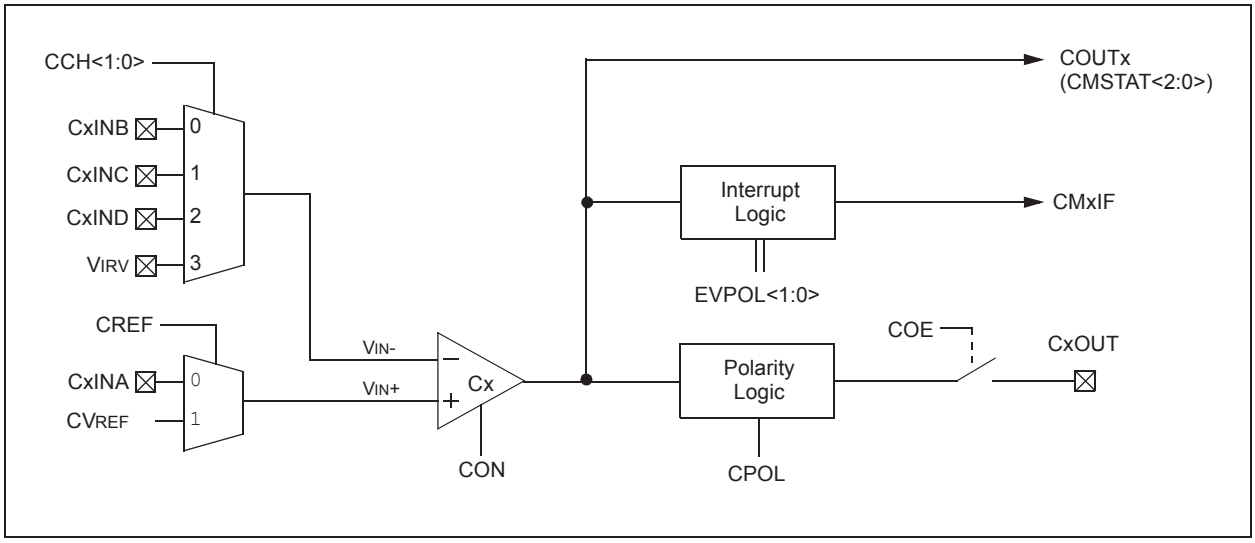
• Independent interrupt generation for each comparator with configurable interrupt-on-change

23.1 Registers

The CMxCON registers (Register 23-1) select the input and output configuration for each comparator, as well as the settings for interrupt generation.

The CMSTAT register (Register 23-2) provides the output results of the comparators. The bits in this register are read-only.

FIGURE 23-1: COMPARATOR SIMPLIFIED BLOCK DIAGRAM



26.9 Registers

There are three control registers for the CTMU:

- CTMUCONH
- CTMUCONL
- CTMUICON

The CTMUCONH and CTMUCONL registers (Register 26-1 and Register 26-2) contain control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUICON register (Register 26-3) has bits for selecting the current source range and current source trim.

REGISTER 26-1: CTMUCONH: CTMU CONTROL REGISTER HIGH (ACCESS FB3h)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 4 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 3 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 2 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 1 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 0 **CTTRIG:** CTMU Special Event Trigger bit
1 = CTMU Special Event Trigger is enabled
0 = CTMU Special Event Trigger is disabled

28.2.2 EXTENDED INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \le k \le 63$
$f \in [0, 1, 2]$

Operation: FSR(f) + k → FSR(f)

Status Affected: None

Encoding:

1110	1000	ffkk	kkkk

Description: The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ADDFSR 2, 0x23`

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDULNK **Add Literal to FSR2 and Return**

Syntax: ADDULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 + k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1000	11kk	kkkk

Description: The 6-bit literal 'k' is added to the contents of FSR2. A `RETURN` is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a `NOP` is performed during the second cycle.

This may be thought of as a special case of the `ADDFSR` instruction, where f = 3 (binary '`11`'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: `ADDULNK 0x23`

Before Instruction
FSR2 = 03FFh
PC = 0100h
After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

30.2 DC Characteristics: Power-Down and Supply Current PIC18F47J13 Family (Industrial) (Continued)

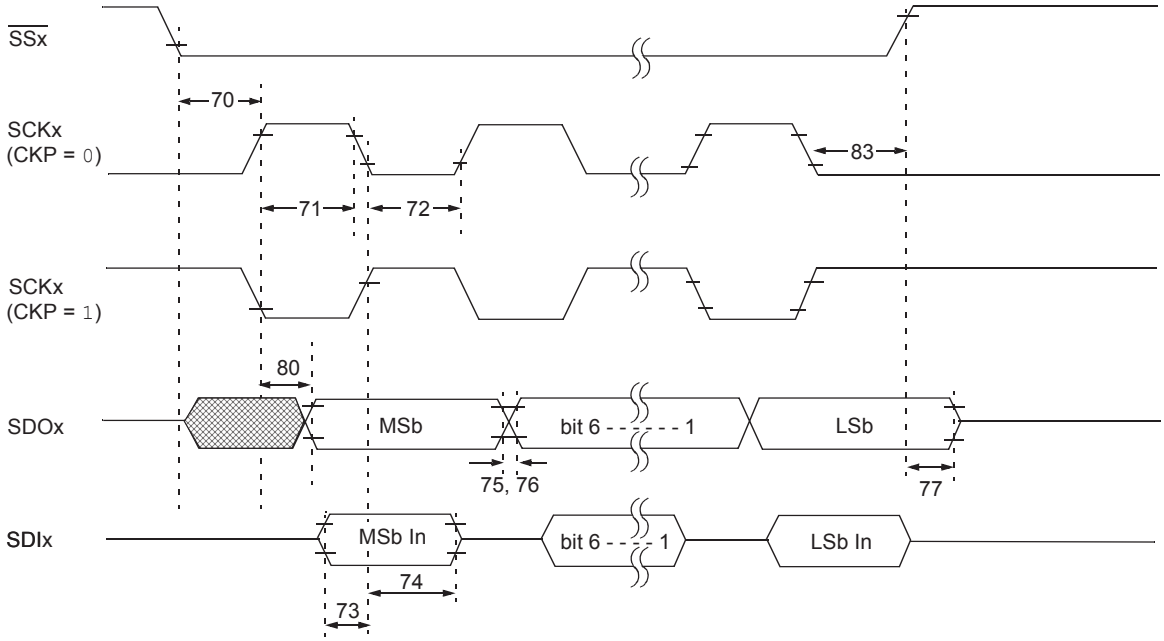
PIC18LF47J13 Family		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J13 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2)						
	PIC18LFXXJ13	5.5	14.2	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 31 kHz **RC_RUN** mode, Internal RC Oscillator, INTSRC = 0
		5.8	14.2	µA	+25°C		
		7.9	19.0	µA	+85°C		
	PIC18LFXXJ13	8.4	16.5	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		8.5	16.5	µA	+25°C		
		11.3	25.0	µA	+85°C		
	PIC18FXXJ13	23.7	60.0	µA	-40°C	VDD = 2.15V VDDCORE = 10 µF Capacitor	
		27.8	60.0	µA	+25°C		
		34.0	70.0	µA	+85°C		
	PIC18FXXJ13	26.1	70.0	µA	-40°C	VDD = 3.3V VDDCORE = 10 µF Capacitor	
		29.6	70.0	µA	+25°C		
		36.2	96.0	µA	+85°C		
	PIC18LFXXJ13	0.87	1.5	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0	FOSC = 4 MHz, **RC_RUN** mode, Internal RC Oscillator
		0.91	1.5	mA	+25°C		
		0.95	1.6	mA	+85°C		
	PIC18LFXXJ13	1.23	2.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		1.24	2.0	mA	+25°C		
		1.25	2.0	mA	+85°C		
	PIC18FXXJ13	0.99	2.4	mA	-40°C	VDD = 2.15V, VDDCORE = 10 µF capacitor	
		1.02	2.4	mA	+25°C		
		1.06	2.6	mA	+85°C		
	PIC18FXXJ13	1.31	2.6	mA	-40°C	VDD = 3.3V, VDDCORE = 10 µF capacitor	
		1.25	2.6	mA	+25°C		
		1.26	2.7	mA	+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 a high-impedance state and tied to VDD or VSS, and all features that add delta current are 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 (such as WDT). 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.

FIGURE 30-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)



Note: Refer to Figure 30-4 for load conditions.

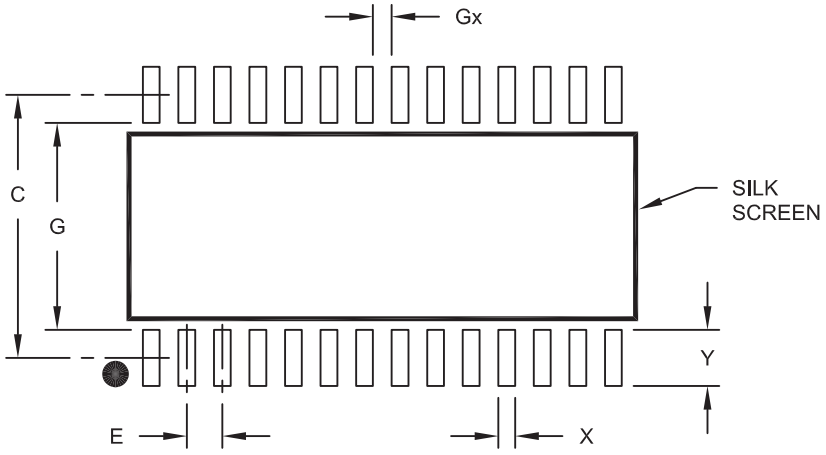
TABLE 30-23: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ to Write to SSPxBUF		3 Tcy	—	ns	
71	TscH	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single byte	40	—	ns	**(Note 1)**
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single byte	40	—	ns	**(Note 1)**
73	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge		25	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge		35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
				100	—	ns	VDD = 2.15V
75	TdoR	SDOx Data Output Rise Time		—	25	ns	PORTB or PORTC
76	TdoF	SDOx Data Output Fall Time		—	25	ns	PORTB or PORTC
77	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance		10	70	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge		—	50	ns	VDD = 3.3V, VDDCORE = 2.5V
				—	100	ns	VDD = 2.15V
83	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ after SCKx Edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter #73A.
2: Only if Parameter #71A and #72A are used.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

APPENDIX B: MIGRATION FROM PIC18F46J11 TO PIC18F47J13

Code for the devices in the PIC18F46J11 family can be migrated to the PIC18F47J13 without many changes. The differences between the two device families are listed in Table B-1.

TABLE B-1: NOTABLE DIFFERENCES BETWEEN PIC18F47J13 AND PIC18F46J11 FAMILIES

Characteristic	PIC18F47J13 Family	18F46J11 Family
Maximum Program Memory	128 Kbytes	64 Kbytes
Oscillator Options	PLL can be enabled at start-up with the Configuration bit option. 96 MHz PLL circuit is available via the Configuration bit setting, allowing 48 MHz operation from INTOSC. Default 4x PLL circuit is still available.	Requires firmware to set the PLLEN bit at run time. 4x PLL circuit only. Maximum operating frequency from INTOSC is 32 MHz.
SOSC Oscillator Options	Low-power oscillator option for SOSC, with run-time switch.	Low-power oscillator option for SOSC, only via the Configuration bit setting.
T1CKI Clock Input	T1CKI can be used as a clock input without enabling the Timer1 oscillator.	No
Timers	8	5
ECCP	3	2
CCP	7	0
SPI FOSC/8 Master Clock Option	Yes	No
Second I^2C Port	Yes, all packages.	Yes, but only on 44-pin devices.
ADC	13 Channel, 10/12-Bit Conversion modes with Special Event Trigger option.	13 Channel, 10-bit only.
Peripheral Module Disable Bits	Yes, allowing further power reduction.	No
Band Gap Voltage Reference Output	Yes, enabled on pin, RA1, by setting the VBGOE bit (WDTCON<4>).	No
$\overline{REPU}$/$\overline{RDPU}$ Pull-Up Enable Bits	Moved to TRISE register (avoids read, modify, write issues).	Pull-up bits configured in PORTE register
Comparators	Three, each with four input pin selections.	Two, each with two input pin selections.
Increased Output Drive Strength	RA0 through RA5, RDx and REx.	No
PWRT Period	F Devices – 500 μs LF Devices – 46 ms	F Devices – 1 ms LF Devices – 64 ms