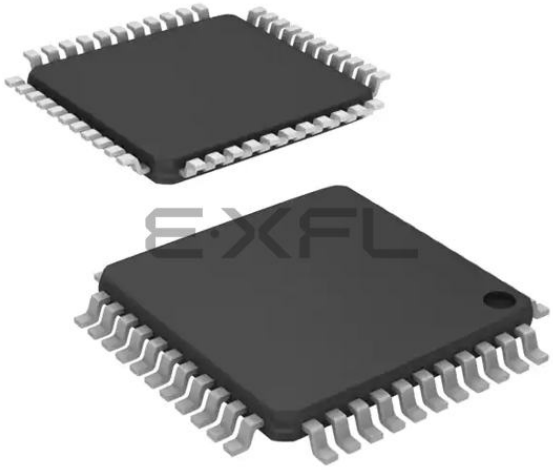


Welcome to E-XFL.COM

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-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46j13-i-pt

3.2.5.1 OSCTUNE Register

The internal oscillator's output has been calibrated at the factory but can be adjusted in the user's application. This is done by writing to the OSCTUNE register (Register 3-1).

When the OSCTUNE register is modified, the INTOSC frequency will begin shifting to the new frequency. The INTOSC clock will typically stabilize within 1 μs. Code execution continues during this shift. There is no indication that the shift has occurred.

The OSCTUNE register also contains the INTSRC bit. The INTSRC bit allows users to select which internal oscillator provides the clock source when the 31 kHz frequency option is selected. This is covered in larger detail in **Section 3.3.1 "Oscillator Control Register"**.

The PLLEN bit, contained in the OSCTUNE register, can be used to enable or disable the internal PLL when running in one of the PLL type oscillator modes (e.g., INTOSCPLL). Oscillator modes that do not contain "PLL" in their name cannot be used with the PLL. In these modes, the PLL is always disabled regardless of the setting of the PLLEN bit.

When configured for one of the PLL enabled modes, setting the PLLEN bit does not immediately switch the device clock to the PLL output. The PLL requires up to electrical parameter, $t_{rc,}$ to start-up and lock, during which time, the device continues to be clocked. Once the PLL output is ready, the microcontroller core will automatically switch to the PLL derived frequency.

3.2.5.2 Internal Oscillator Output Frequency and Drift

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8.0 MHz. However, this frequency may drift as V_{DD} or temperature changes, which can affect the controller operation in a variety of ways.

The low-frequency INTRC oscillator operates independently of the INTOSC source. Any changes in INTOSC across voltage and temperature are not necessarily reflected by changes in INTRC and vice versa.

3.2.5.3 Compensating for INTOSC Drift

It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. This has no effect on the INTRC clock source frequency.

Tuning the INTOSC source requires knowing when to make the adjustment, in which direction it should be made, and in some cases, how large a change is needed. When using the EUSART, for example, an adjustment may be required when it begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high; to adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low; to compensate, increment OSCTUNE to increase the clock frequency.

It is also possible to verify device clock speed against a reference clock. Two timers may be used: one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator. Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

Finally, an ECCP module can use free-running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is greater than the calculated time, the internal oscillator block is running too fast; to compensate, decrement the OSCTUNE register. If the measured time is less than the calculated time, the internal oscillator block is running too slow; to compensate, increment the OSCTUNE register.

© 2010-2017 Microchip Technology Inc.

REGISTER 3-3: OSCCON2: OSCILLATOR CONTROL REGISTER 2 (ACCESS F87h)

U-0	R-0(2)	U-0	R/W-1	R/W-0(2)	R/W-1	U-0	U-0
—	SOSCRUN	—	SOSCDRV	SOSCGO(3)	—	—	—
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 **SOSCRUN:** SOSC Run Status bit
- 1 = System clock comes from secondary SOSC
- 0 = System clock comes from an oscillator other than SOSC

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

bit 4 **SOSCDRV:** SOSC Drive Control bit
- 1 = T1OSC/SOSC oscillator drive circuit is selected by Configuration bits, CONFIG2L<4:3>
- 0 = Low-power T1OSC/SOSC circuit is selected

bit 3 **SOSCGO:** Oscillator Start Control bit(3)
- 1 = Turns on the oscillator, even if no peripherals are requesting it
- 0 = Oscillator is shut off unless peripherals are requesting it

bit 2 **Reserved:** Maintain as '1'

bit 1-0 **Unimplemented:** Read as '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: When the SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.

5.0 RESET

The PIC18F47J13 Family of devices differentiates among various kinds of Reset:

a) Power-on Reset (POR)
b) MCLR Reset during normal operation
c) MCLR Reset during power-managed modes
d) Watchdog Timer (WDT) Reset (during execution)
e) Configuration Mismatch (CM)
f) Brown-out Reset (BOR)
g) RESET Instruction
h) Stack Full Reset
i) Stack Underflow Reset
j) Deep Sleep Reset

This section discusses Resets generated by MCLR, POR and BOR, and covers the operation of the various start-up timers.

For information on WDT Resets, see **Section 27.2 "Watchdog Timer (WDT)"**. For Stack Reset events, see **Section 6.1.4.4 "Stack Full and Underflow Resets"** and for Deep Sleep mode, see **Section 4.6 "Deep Sleep Mode"**.

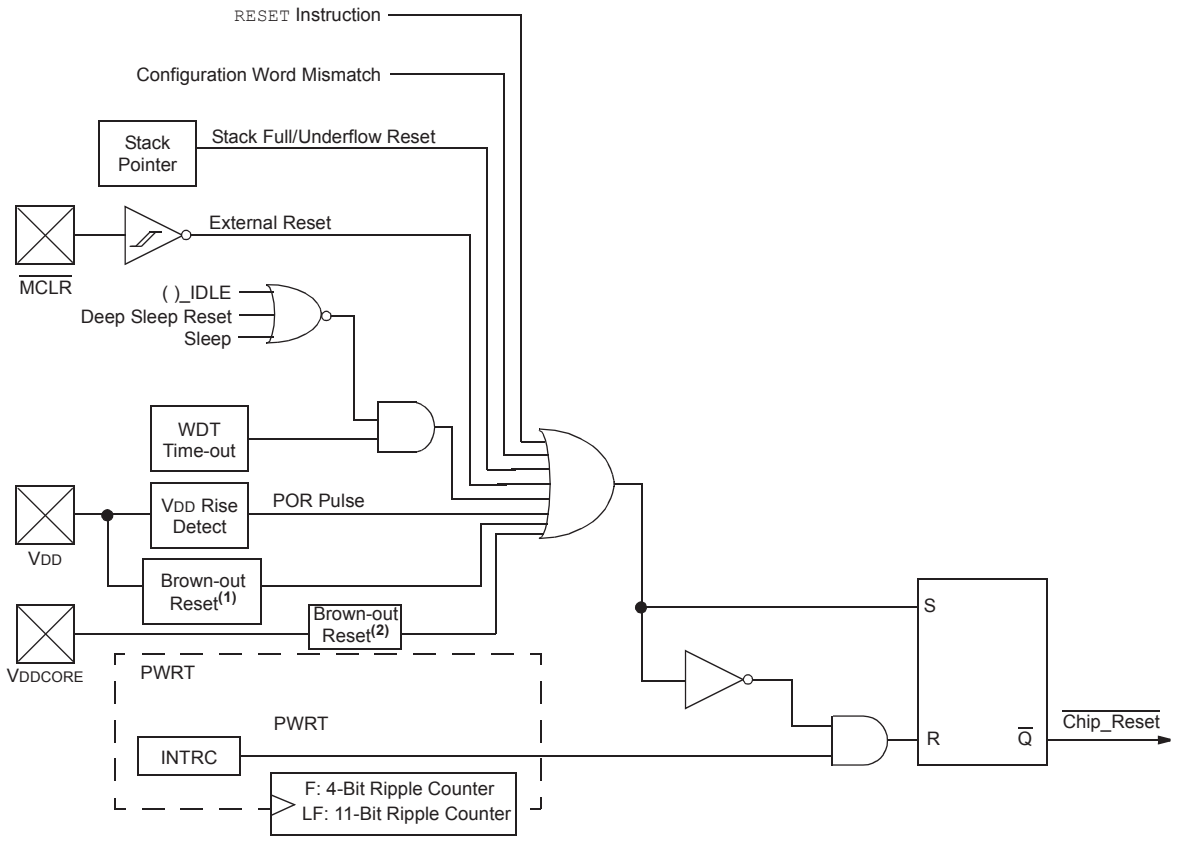
Figure 5-1 provides a simplified block diagram of the on-chip Reset circuit.

5.1 RCON Register

Device Reset events are tracked through the RCON register (Register 5-1). The lower five bits of the register indicate that a specific Reset event has occurred. In most cases, these bits can only be set by the event and must be cleared by the application after the event. The state of these flag bits, taken together, can be read to indicate the type of Reset that just occurred. This is described in more detail in **Section 5.7 "Reset State of Registers"**.

The RCON register also has a control bit for setting interrupt priority (IPEN). Interrupt priority is discussed in **Section 9.0 "Interrupts"**.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: The VDD monitoring BOR circuit can be enabled or disabled on "LF" devices based on the CONFIG3L<DSBOREN> Configuration bit. On "F" devices, the VDD monitoring BOR circuit is only enabled during Deep Sleep mode by CONFIG3L<DSBOREN>.

2: The VDDCORE monitoring BOR circuit is only implemented on "F" devices. It is always used, except while in Deep Sleep mode. The VDDCORE monitoring BOR circuit has a trip point threshold of VBOR (parameter D005).

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
RPOR16	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR15	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR14	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR13	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR12	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR11	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR10	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR9	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR8	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR7	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR6	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR5	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR4	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR3	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR2	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR1	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
RPOR0	PIC18F2XJ13	PIC18F4XJ13	---0 0000	---0 0000	---u uuuu
PPSCON	PIC18F2XJ13	PIC18F4XJ13	---- ---0	---- ---0	---- ---u
PMDIS3	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDIS2	PIC18F2XJ13	PIC18F4XJ13	-0-0 0000	-0-0 0000	-u-u uuuu
PMDIS1	PIC18F2XJ13	PIC18F4XJ13	0000 000-	0000 000-	uuuu uuu-
PMDIS0	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
ADCTRIG	PIC18F2XJ13	PIC18F4XJ13	---- --00	---- --00	---- --uu

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.

TABLE 6-4: REGISTER FILE SUMMARY (PIC18F47J13 FAMILY) (CONTINUED)

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FDBh	PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W								N/A
FDAh	FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2 High Byte				---- 0000
FD9h	FSR2L	Indirect Data Memory Address Pointer 2 Low Byte								xxxx xxxx
FD8h	STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx
FD7h	TMR0H	Timer0 Register High Byte								0000 0000
FD6h	TMR0L	Timer0 Register Low Byte								xxxx xxxx
FD5h	T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111
FD3h	OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	—	SCS1	SCS0	0110 q100
FD2h	CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111
FD1h	CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111
FD0h	RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0-11 11qq
FCFh	TMR1H	Timer1 Register High Byte								xxxx xxxx
FCEh	TMR1L	Timer1 Register Low Bytes								xxxx xxxx
FCDh	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	RD16	TMR1ON	0000 0000
FCCh	TMR2	Timer2 Register								0000 0000
FCBh	PR2	Timer2 Period Register								1111 1111
FCAh	T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000
FC9h	SSP1BUF	MSSP1 Receive Buffer/Transmit Register								xxxx xxxx
FC8h	SSP1ADD	MSSP1 Address Register (I^2C Slave mode). MSSP1 Baud Rate Reload Register (I^2C Master mode).								0000 0000
FC8h	SSP1MSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111
FC7h	SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000
FC6h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000
FC5h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
FC5h	SSP1CON2	GCEN	ACKSTAT	ADMSK5	ADMSK4	ADMSK3	ADMSK2	ADMSK1	SEN	0000 0000
FC4h	ADRESH	A/D Result Register High Byte								xxxx xxxx
FC3h	ADRESL	A/D Result Register Low Byte								xxxx xxxx
FC2h	ADCON0	VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	0000 0000
FC1h	ADCON1	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000
FC0h	WDTCON	REGSLP	LVDSTAT	ULPLVL	VBGOE	DS	ULPEN	ULPSINK	SWDTEN	1qq0 q000
FBFh	PSTR1CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	00-0 0001
FBEh	ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000
FBDh	ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000
FBCh	CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx
FBBh	CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx
FBAh	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000
FB9h	PSTR2CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	00-0 0001
FB8h	ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000
FB7h	ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000
FB6h	CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx
FB5h	CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx
FB4h	CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000
FB3h	CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	0-00 0000
FB2h	CTMUCONL	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000 00xx
FB1h	CTMUICON	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	0000 0000
FB0h	SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								0000 0000
FAFh	RCREG1	EUSART1 Receive Register								0000 0000
FAEh	TXREG1	EUSART1 Transmit Register								0000 0000
FADh	TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010

Note 1: Applicable for 28-pin devices (PIC18F26J13, PIC18F27J13, PIC18LF26J13 and PIC18LF27J13).
2: Applicable for 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).
3: Value on POR, BOR.

© 2010-2017 Microchip Technology Inc.

TABLE 10-5: PORTB I/O SUMMARY

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB0/AN12/ C3IND/INT0/ RP3	RB0	1	I	TTL	PORTB<0> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input is enabled.(1)
		0	O	DIG	LATB<0> data output; not affected by an analog input.
	AN12	1	I	ANA	A/D Input Channel 12.(1)
	C3IND	1	I	ANA	Comparator 3 Input D.
	INT0	1	I	ST	External Interrupt 0 input.
	RP3	1	I	ST	Remappable Peripheral Pin 3 input.
		0	O	DIG	Remappable Peripheral Pin 3 output.
RB1/AN10/ C3INC/PMBE/ RTCC/RP4	RB1	1	I	TTL	PORTB<1> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared. Disabled when an analog input is enabled.(1)
		0	O	DIG	LATB<1> data output; not affected by an analog input.
	AN10	1	I	ANA	A/D Input Channel 10.(1)
	C3INC	1	I	ANA	Comparator 3 Input C.
	PMBE(3)	x	O	DIG	Parallel Master Port byte enable.
	RTCC	0	O	DIG	Asynchronous serial transmit data output (USART module).
	RP4	1	I	ST	Remappable Peripheral Pin 4 input.
		0	O	DIG	Remappable Peripheral Pin 4 output.
RB2/AN8/ C2INC/CTED1/ PMA3/REFO/ RP5	RB2	1	I	TTL	PORTB<2> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared. Disabled when an analog input is enabled.(1)
		0	O	DIG	LATB<2> data output; not affected by an analog input.
	AN8	1	I	ANA	A/D Input Channel 8.(1)
	C2INC	1	I	ANA	Comparator 2 Input C.
	CTED1	1	I	ST	CTMU Edge 1 input.
	PMA3(3)	x	O	DIG	Parallel Master Port address.
	REFO	0	O	DIG	Reference output clock.
	RP5	1	I	ST	Remappable Peripheral Pin 5 input.
		0	O	DIG	Remappable Peripheral Pin 5 output.
RB3/AN9/ C3INA/CTED2/ PMA2/RP6	RB3	0	O	DIG	LATB<3> data output; not affected by analog input.
		1	I	TTL	PORTB<3> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared. Disabled when analog input is enabled.(1)
	AN9	1	I	ANA	A/D Input Channel 9.(1)
	C3INA	1	I	ANA	Comparator 3 Input A.
	CTED2	1	I	ST	CTMU Edge 2 input.
	PMA2(3)	x	O	DIG	Parallel Master Port address.
	RP6	1	I	ST	Remappable Peripheral Pin 6 input.
		0	O	DIG	Remappable Peripheral Pin 6 output.

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

Note 1: Pins are configured as analog inputs by default on POR. Using these pins for digital inputs requires setting the appropriate bits in the ANCON1 register.

2: All other pin functions are disabled when ICSP™ or ICD is enabled.

3: Only on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).

4: Only on 28-pin devices (PIC18F26J13, PIC18F27J13, PIC18LF26J13 and PIC18LF27J13).

REGISTER 11-5: PMEH: PARALLEL PORT ENABLE REGISTER HIGH BYTE (BANKED F57h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
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 **PTEN<15:14>:** PMCS Port Enable bits
1 = PMA<15:14> function as either PMA<15:14> or PMCS
0 = PMA<15:14> function as port I/O

bit 5-0 **PTEN<13:8>:** PMP Address Port Enable bits
1 = PMA<13:8> function as PMP address lines
0 = PMA<13:8> function as port I/O

Note 1: This register is only available on 44-pin devices.

REGISTER 11-6: PMEL: PARALLEL PORT ENABLE REGISTER LOW BYTE (BANKED F56h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
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 7-2 **PTEN<7:2>:** PMP Address Port Enable bits
1 = PMA<7:2> function as PMP address lines
0 = PMA<7:2> function as port I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
1 = PMA<1:0> function as either PMA<1:0> or PMALH and PMALL
0 = PMA<1:0> pads functions as port I/O

Note 1: This register is only available on 44-pin devices.

11.2.4 BUFFERED PARALLEL SLAVE PORT MODE

Buffered Parallel Slave Port mode is functionally identical to the Legacy PSP mode with one exception, the implementation of 4-level read and write buffers. Buffered PSP mode is enabled by setting the INCM bits in the PMMODEH register. If the INCM<1:0> bits are set to '11', the PMP module will act as the buffered PSP.

When the Buffered PSP mode is active, the PMDIN1L, PMDIN1H, PMDIN2L and PMDIN2H registers become the write buffers and the PMDOUT1L, PMDOUT1H, PMDOUT2L and PMDOUT2H registers become the read buffers. Buffers are numbered, 0 through 3, starting with the lower byte of PMDIN1L to PMDIN2H as the read buffers and PMDOUT1L to PMDOUT2H as the write buffers.

11.2.4.1 READ FROM SLAVE PORT

For read operations, the bytes will be sent out sequentially, starting with Buffer 0 (PMDOUT1L<7:0>) and ending with Buffer 3 (PMDOUT2H<7:0>) for every read strobe. The module maintains an internal pointer to keep track of which buffer is to be read. Each buffer has a corresponding read status bit, OBxE, in the PMSTATL register. This bit is cleared when a buffer contains data that has not been written to the bus and is set when data is written to the bus. If the current buffer location being read from is empty, a buffer underflow is generated and the Buffer Overflow Flag bit, OBUF, is set. If all four OBxE status bits are set, then the Output Buffer Empty flag (OBE) will also be set.

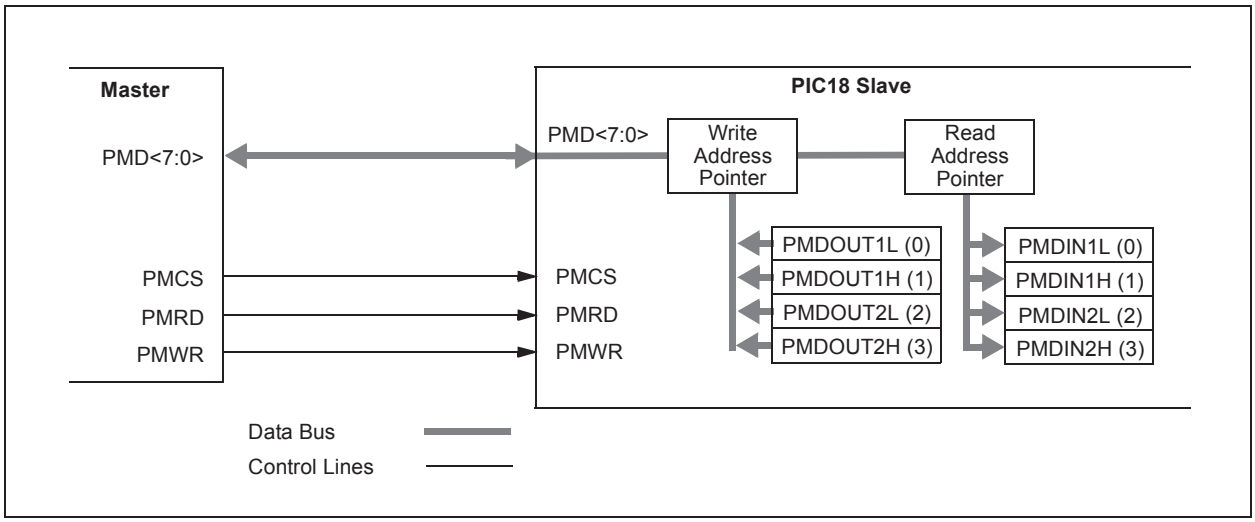
11.2.4.2 WRITE TO SLAVE PORT

For write operations, the data has to be stored sequentially, starting with Buffer 0 (PMDIN1L<7:0>) and ending with Buffer 3 (PMDIN2H<7:0>). As with read operations, the module maintains an internal pointer to the buffer that is to be written next.

The input buffers have their own write status bits: IBxF in the PMSTATH register. The bit is set when the buffer contains unread incoming data and cleared when the data has been read. The flag bit is set on the write strobe. If a write occurs on a buffer when its associated IBxF bit is set, the Buffer Overflow flag, IBOV, is set; any incoming data in the buffer will be lost. If all four IBxF flags are set, the Input Buffer Full Flag (IBF) is set.

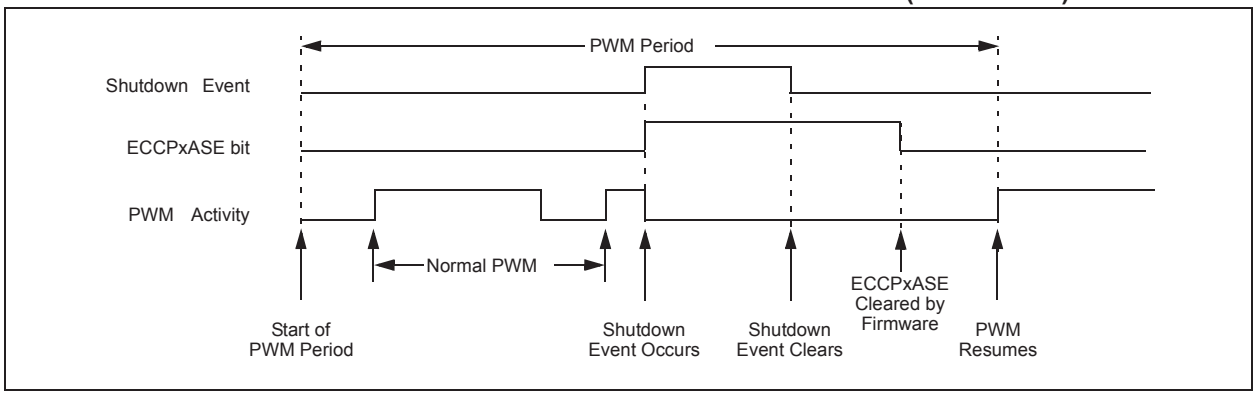
In Buffered Slave mode, the module can be configured to generate an interrupt on every read or write strobe (IRQM<1:0> = 01). It can be configured to generate an interrupt on a read from Read Buffer 3 or a write to Write Buffer 3, which is essentially an interrupt every fourth read or write strobe (RQM<1:0> = 11). When interrupting every fourth byte for input data, all Input Buffer registers should be read to clear the IBxF flags. If these flags are not cleared, then there is a risk of hitting an overflow condition.

FIGURE 11-5: PARALLEL MASTER/SLAVE CONNECTION BUFFERED EXAMPLE



NOTES:

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)

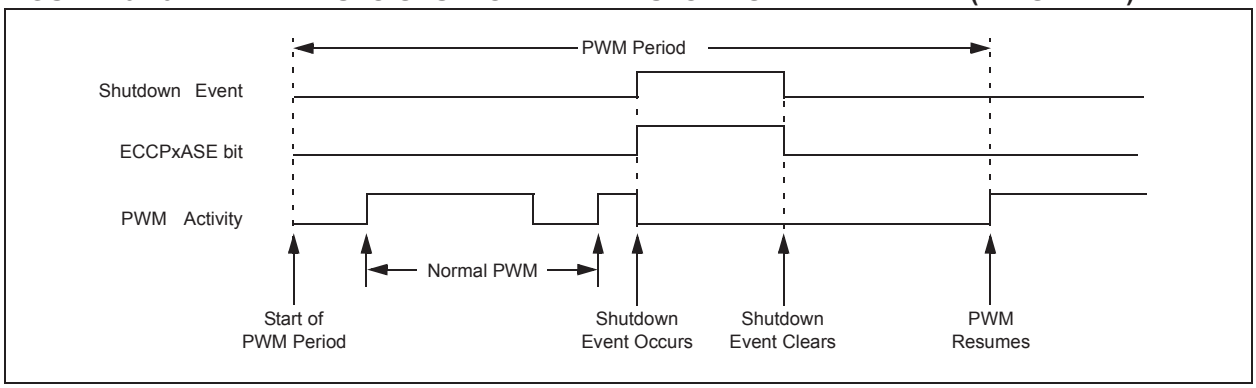


TABLE 19-4: REGISTERS ASSOCIATED WITH ECCP1/2/3 MODULE AND TIMER1/2/3/4/6/8 (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T8CON	—	T8OUTPS3	T8OUTPS2	T8OUTPS1	T8OUTPS0	TMR8ON	T8CKPS1	T8CKPS0
CCPR1H	Capture/Compare/PWM Register 1 High Byte							
CCPR1L	Capture/Compare/PWM Register 1 Low Byte							
CCPR2H	Capture/Compare/PWM Register 2 High Byte							
CCPR2L	Capture/Compare/PWM Register 2 Low Byte							
CCPR3H	Capture/Compare/PWM Register 3 High Byte							
CCPR3L	Capture/Compare/PWM Register 3 Low Byte							
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0

20.5 I²C Mode

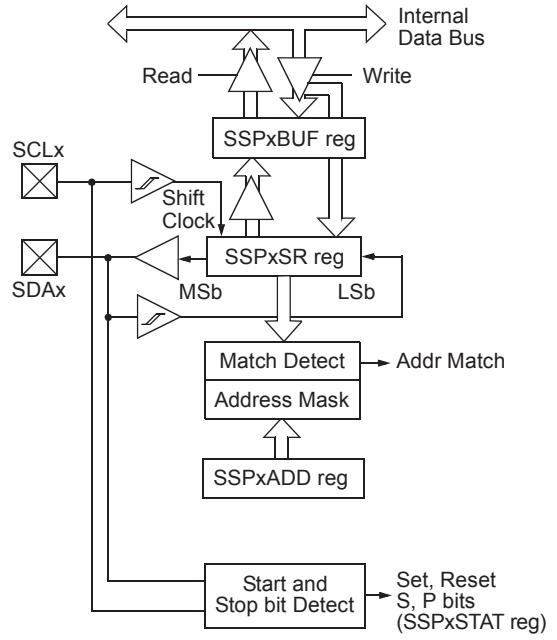
The MSSP module in I²C mode fully implements all master and slave functions (including general call support), and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications and 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial Clock (SCLx) – RC3/SCK1/SCL1/RP14, RD0/PMD0/SCL2 (44-pin devices) or RB4/CCP4/PMA1/KBI0/SCL2/RP7 (28-pin devices)
- Serial Data (SDAx) – RC4/SDI1/SDA1/RP15, RD1/PMD1/SDA2 (44-pin devices) or RB5/CCP5/KBI1/SDA2/RP8 (28-pin devices)

The user must configure these pins as inputs by setting the associated TRISx bits. These pins are up to 5.5V tolerant, allowing direct use in I²C buses operating at voltages higher than V_{DD}.

FIGURE 20-7: MSSPx BLOCK DIAGRAM (I²C MODE)



Note: Only port I/O names are used in this diagram for the sake of brevity. Refer to the text for a full list of multiplexed functions.

20.5.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Control Register 2 (SSPxCON2)
- MSSPx Status Register (SSPxSTAT)
- Serial Receive/Transmit Buffer Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible
- MSSPx Address Register (SSPxADD)
- MSSPx 7-Bit Address Mask Register (SSPxMSK)

SSPxCON1, SSPxCON2 and SSPxSTAT are the control and status registers in I²C mode operation. The SSPxCON1 and SSPxCON2 registers are readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

SSPxSR is the shift register used for shifting data in or out. SSPxBUF is the buffer register to which data bytes are written to or read from.

SSPxADD contains the slave device address when the MSSP is configured in I²C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPxADD act as the Baud Rate Generator (BRG) reload value.

SSPxMSK holds the slave address mask value when the module is configured for 7-Bit Address Masking mode. While it is a separate register, it shares the same SFR address as SSPxADD; it is only accessible when the SSPM<3:0> bits are specifically set to permit access. Additional details are provided in **Section 20.5.3.4 "7-Bit Address Masking Mode"**.

In receive operations, SSPxSR and SSPxBUF together, create a double-buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not double-buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

EXAMPLE 21-1: CALCULATING BAUD RATE ERROR

```
For a device with Fosc of 16 MHz, desired baud rate of 9600, Asynchronous mode, and
8-bit BRG:
Desired Baud Rate = Fosc/(64 ([SPBRGHx:SPBRGx] + 1))
Solving for SPBRGHx:SPBRGx:
   X  =  ((Fosc/Desired Baud Rate)/64) – 1
      =  ((16000000/9600)/64) – 1
      =  [25.042] = 25
Calculated Baud Rate=16000000/(64 (25 + 1))
      =  9615
Error  =  (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
      =  (9615 – 9600)/9600 = 0.16%
```

TABLE 21-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGHx	EUSARTx Baud Rate Generator High Byte							
SPBRGx	EUSARTx Baud Rate Generator Low Byte							

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

REGISTER 22-3: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

U-1	U-1	U-1	U-1	R/WO-1	U-0	R/WO-1	R/WO-1
—	—	—	—	MSSPMSK	—	ADCSEL	IOL1WAY
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Program the corresponding Flash Configuration bit to '1'

bit 3 **MSSPMSK:** MSSP 7-Bit Address Masking Mode Enable bit
1 = 7-Bit Address Masking mode is enabled
0 = 5-Bit Address Masking mode is enabled

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

bit 1 **ADCSEL:** A/D Converter Mode bit
1 = 10-Bit Conversion mode is enabled
0 = 12-Bit Conversion mode is enabled

bit 0 **IOL1WAY:** IOLOCK One Way Set Enable bit
1 = IOLOCK bit (PPSCON<0>) can be set once, provided the unlock sequence has been completed. Once set, the Peripheral Pin Select registers cannot be written to a second time.
0 = IOLOCK bit (PPSCON<0>) can be set and cleared as needed, provided the unlock sequence has been completed

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

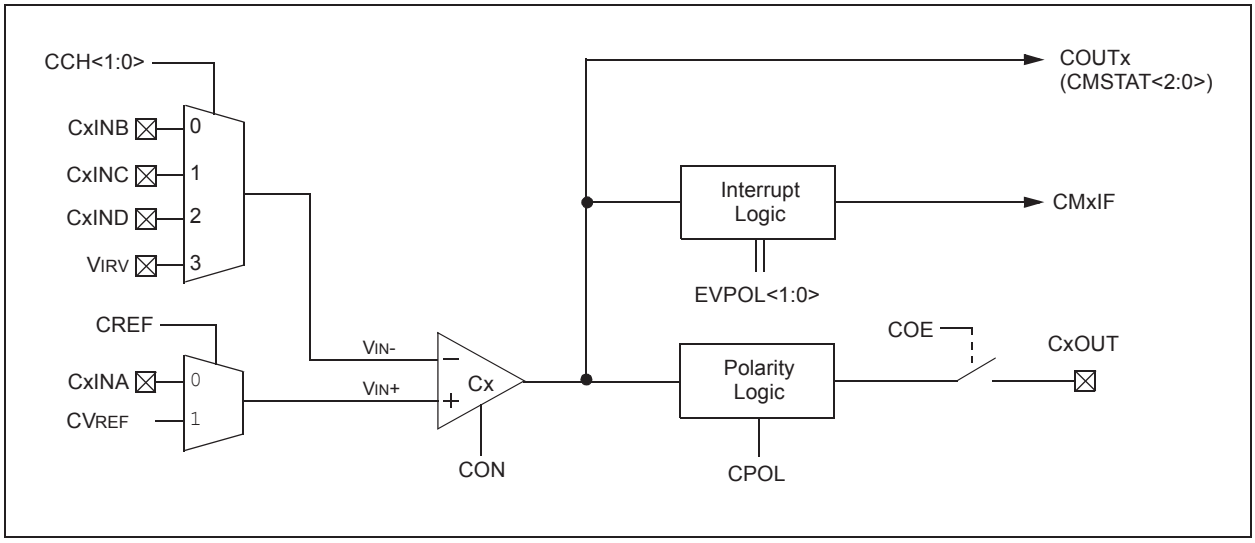
• Independent interrupt generation for each comparator with configurable interrupt-on-change

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



23.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode when enabled. Each operational comparator will consume additional current. To minimize power consumption while in Sleep mode, turn off the comparators (CON = 0) before entering Sleep. If the device wakes up from Sleep, the contents of the CMxCON register are not affected.

23.8 Effects of a Reset

A device Reset forces the CMxCON registers to their Reset state. This forces both comparators and the voltage reference to the OFF state.

TABLE 23-3: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

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
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR5	—	—	CM3IF	TMR8IF	TMR6IF	TMR5IF	TMR5GIF	TMR1GIF
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE5	—	—	CM3IE	TMR8IE	TMR6IE	TMR5IE	TMR5GIE	TMR1GIE
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR5	—	—	CM3IP	TMR8IP	TMR6IP	TMR5IP	TMR5GIP	TMR1GIP
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
CMSTAT	—	—	—	—	—	COUT3	COUT2	COUT1
ANCON0	PCFG7(1)	PCFG6(1)	PCFG5(1)	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
ANCON1	VBGEN	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0

Legend: — = unimplemented, read as '0'. Shaded cells are not related to comparator operation.

Note 1: These bits are not implemented on 28-pin devices.

EXAMPLE 26-3: CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 25                            //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                     //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                        //for unsigned conversion 10 sig bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA

int main(void)
{
    int i;
    int j = 0;                               //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

                                             //assume CTMU and A/D have been setup correctly
                                             //see Example 25-1 for CTMU & A/D setup
    setup();

        CTMUCONHbits.CTMUEN = 1;             // Enable the CTMU
        CTMUCONLbits.EDG1STAT = 0;           // Set Edge status bits to zero
        CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;           //drain charge on the circuit
        DELAY;                              //wait 125us
        CTMUCONHbits.IDISSEN = 0;           //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;          //Begin charging the circuit
                                            //using CTMU current source
        DELAY;                              //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;          //Stop charging circuit

        PIR1bits.ADIF = 0;                  //make sure A/D Int not set
        ADCON0bits.GO=1;                    //and begin A/D conv.
        while(!PIR1bits.ADIF);              //Wait for A/D convert complete


        Vread = ADRES;                      //Get the value from the A/D
        PIR1bits.ADIF = 0;                  //Clear A/D Interrupt Flag
        VTot += Vread;                      //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);            //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                   //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

TABLE 27-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS

Configuration Register (Volatile)	Configuration Register Address	Flash Configuration Byte Address
CONFIG1L	300000h	XXXF8h
CONFIG1H	300001h	XXXF9h
CONFIG2L	300002h	XXXFAh
CONFIG2H	300003h	XXXFBh
CONFIG3L	300004h	XXXFCh
CONFIG3H	300005h	XXXFDh
CONFIG4L	300006h	XXXFEh
CONFIG4H	300007h	XXXFFh

TABLE 27-2: CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprog. Value(1)
300000h	CONFIG1L	$\overline{\text{DEBUG}}$	XINST	STVREN	CFGPLLEN	PLLDIV2	PLLDIV1	PLLDIV0	WDTEN	1111 1111
300001h	CONFIG1H	—(2)	—(2)	—(2)	—(2)	—	CP0	—	—	1111 -1--
300002h	CONFIG2L	IESO	FCMEN	CLKOEC	SOSCSEL1	SOSCSEL0	FOSC2	FOSC1	FOSC0	1111 1111
300003h	CONFIG2H	—(2)	—(2)	—(2)	—(2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	1111 1111
300004h	CONFIG3L	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	1111 1111
300005h	CONFIG3H	—(2)	—(2)	—(2)	—(2)	MSSPMSK	PLLSEL	ADCSEL	IOL1WAY	1111 1111
300006h	CONFIG4L	WPCFG	WPFP6	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0	1111 1111
300007h	CONFIG4H	—(2)	—(2)	—(2)	—(2)	—	—	WPEND	WPDIS	1111 --11
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xx1x xxxx(3)
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0101 10x1(3)

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.

2: The value of these bits in program memory should always be programmed to '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

3: See Register 27-9 and Register 27-10 for DEVID values. These registers are read-only and cannot be programmed by the user.

TABLE 28-2: PIC18F47J13 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb LSb	Status Affected	Notes
BIT-ORIENTED OPERATIONS						
BCF	f, b, a	Bit Clear f	1	1001 bbba ffff ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000 bbba ffff ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011 bbba ffff ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010 bbba ffff ffff	None	3, 4
BTG	f, b, a	Bit Toggle f	1	0111 bbba ffff ffff	None	1, 2
CONTROL OPERATIONS						
BC	n	Branch if Carry	1 (2)	1110 0010 nnnn nnnn	None	
BN	n	Branch if Negative	1 (2)	1110 0110 nnnn nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110 0011 nnnn nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110 0111 nnnn nnnn	None	
BNOV	n	Branch if Not Overflow	1 (2)	1110 0101 nnnn nnnn	None	
BNZ	n	Branch if Not Zero	1 (2)	1110 0001 nnnn nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110 0100 nnnn nnnn	None	
BRA	n	Branch Unconditionally	2	1101 0nnn nnnn nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110 0000 nnnn nnnn	None	
CALL	n, s	Call Subroutine 1st word	2	1110 110s kkkk kkkk	None	
		2nd word		1111 kkkk kkkk kkkk		
CLRWDT	—	Clear Watchdog Timer	1	0000 0000 0000 0100	$\overline{TO}$, $\overline{PD}$	
DAW	—	Decimal Adjust WREG	1	0000 0000 0000 0111	C	
GOTO	n	Go to Address 1st word	2	1110 1111 kkkk kkkk	None	
		2nd word		1111 kkkk kkkk kkkk		
NOP	—	No Operation	1	0000 0000 0000 0000	None	
NOP	—	No Operation	1	1111 xxxx xxxx xxxx	None	4
POP	—	Pop Top of Return Stack (TOS)	1	0000 0000 0000 0110	None	
PUSH	—	Push Top of Return Stack (TOS)	1	0000 0000 0000 0101	None	
RCALL	n	Relative Call	2	1101 1nnn nnnn nnnn	None	
RESET		Software Device Reset	1	0000 0000 1111 1111	All	
RETFIE	s	Return from Interrupt Enable	2	0000 0000 0001 000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with Literal in WREG	2	0000 1100 kkkk kkkk	None	
RETURN	s	Return from Subroutine	2	0000 0000 0001 001s	None	
SLEEP	—	Go into Standby mode	1	0000 0000 0000 0011	$\overline{TO}$, $\overline{PD}$	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as an input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

CALLW **Subroutine Call Using WREG**

Syntax: CALLW

Operands: None

Operation: (PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → PCU

Status Affected: None

Encoding:

0000	0000	0001	0100

Description: First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched.

Unlike CALL, there is no option to update W, STATUS or BSR.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read WREG	Push PC to stack	No operation
No operation	No operation	No operation	No operation

Example: HERE CALLW

Before Instruction
PC = address (HERE)
PCLATH = 10h
PCLATU = 00h
W = 06h
After Instruction
PC = 001006h
TOS = address (HERE + 2)
PCLATH = 10h
PCLATU = 00h
W = 06h

MOVSF **Move Indexed to f**

Syntax: MOVSF [z_s], f_d

Operands: $0 \le z_s \le 127$
$0 \le f_d \le 4095$

Operation: ((FSR2) + z_s) → f_d

Status Affected: None

Encoding:

1st word (source)	1110	1011	0zzz	$zzzz_s$
2nd word (destin.)	1111	ffff	ffff	$ffff_d$

Description: The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s', in the first word, to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).

The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVSF [0x05], REG2

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
REG2 = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
REG2 = 33h