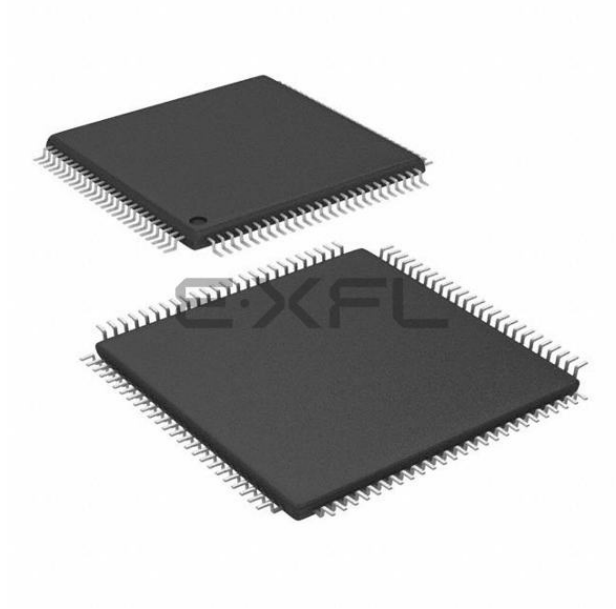


Welcome to E-XFL.COM

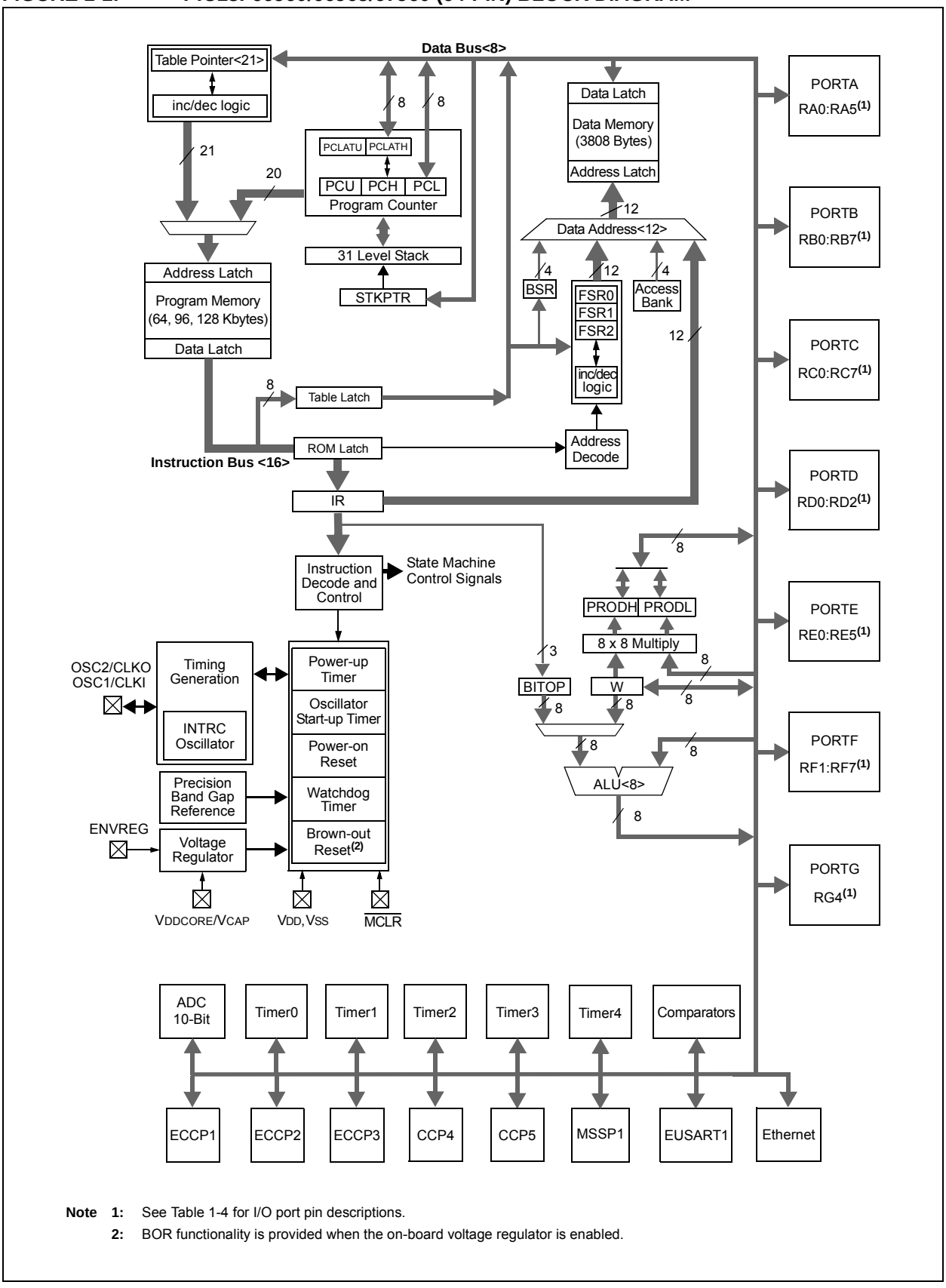
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	41.667MHz
Connectivity	EBI/EMI, Ethernet, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	70
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3808 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f96j60t-i-pf

FIGURE 1-1: PIC18F66J60/66J65/67J60 (64-PIN) BLOCK DIAGRAM



Note 1: See Table 1-4 for I/O port pin descriptions.

2: BOR functionality is provided when the on-board voltage regulator is enabled.

TABLE 1-6: PIC18F96J60/96J65/97J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTF is a bidirectional I/O port.
RF0/AN5	12			
RF0		I/O	ST	Digital I/O.
AN5		I	Analog	Analog Input 5.
RF1/AN6/C2OUT	28			
RF1		I/O	ST	Digital I/O.
AN6		I	Analog	Analog Input 6.
C2OUT		O	—	Comparator 2 output.
RF2/AN7/C1OUT	23			
RF2		I/O	ST	Digital I/O.
AN7		I	Analog	Analog Input 7.
C1OUT		O	—	Comparator 1 output.
RF3/AN8	22			
RF3		I/O	ST	Digital I/O.
AN8		I	Analog	Analog Input 8.
RF4/AN9	21			
RF4		I/O	ST	Digital I/O.
AN9		I	Analog	Analog Input 9.
RF5/AN10/CVREF	20			
RF5		I/O	ST	Digital I/O.
AN10		I	Analog	Analog Input 10.
CVREF		O	—	Comparator reference voltage output.
RF6/AN11	19			
RF6		I/O	ST	Digital I/O.
AN11		I	Analog	Analog Input 11.
RF7/$\overline{SS1}$	18			
RF7		I/O	ST	Digital I/O.
$\overline{SS1}$		I	TTL	SPI slave select input.

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 VDD)

Note 1: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX Configuration bit is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset, RESET Instruction, Stack Resets, CM Reset	Wake-up via WDT or Interrupt
TOSU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---0 uuuu(1)
TOSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu(1)
TOSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu(1)
STKPTR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	00-0 0000	uu-0 0000	uu-u uuuu(1)
PCLATU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000u	uuuu uuuu(3)
INTCON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu(3)
INTCON3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1100 0000	1100 0000	uuuu uuuu(3)
INDF0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR0H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR1H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- 0000	---- 0000	---- uuuu
INDF2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR2H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.
Shaded cells indicate conditions do not apply for the designated device.

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.

TABLE 6-5: REGISTER FILE SUMMARY (PIC18F97J60 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Values on POR, BOR	Details on Page:
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	70, 97
TMR0H	Timer0 Register High Byte								0000 0000	70, 171
TMR0L	Timer0 Register Low Byte								xxxx xxxx	70, 171
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	70, 171
OSCCON	IDLEN	—	—	—	OSTS(3)	—	SCS1	SCS0	0--- q-00	70, 53
ECON1	TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	—	—	0000 00--	70, 227
WDTCON	—	—	—	—	—	—	—	SWDTEN	--- ---0	70, 368
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0-q1 1100	70, 64, 143
TMR1H	Timer1 Register High Byte								xxxx xxxx	70, 175
TMR1L	Timer1 Register Low Byte								xxxx xxxx	70, 175
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	70, 175
TMR2	Timer2 Register								0000 0000	70, 180
PR2	Timer2 Period Register								1111 1111	70, 180
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	70, 180
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								xxxx xxxx	70, 279
SSP1ADD	MSSP1 Address Register (I^2C™ Slave mode), MSSP1 Baud Rate Reload Register (I^2C Master mode)								0000 0000	70, 279
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	70, 270, 280
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	70, 271, 281
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	70, 282
	GCEN	ACKSTAT	ADMSK5(4)	ADMSK4(4)	ADMSK3(4)	ADMSK2(4)	ADMSK1(4)	SEN		
ADRESH	A/D Result Register High Byte								xxxx xxxx	70, 347
ADRESL	A/D Result Register Low Byte								xxxx xxxx	70, 347
ADCON0	ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	0-00 0000	70, 339
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	70, 340
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	70, 341
CCPR1H	Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	70, 193
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	70, 193
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	70, 198
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	70, 193
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	70, 193
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	70, 198
CCPR3H	Capture/Compare/PWM Register 3 High Byte								xxxx xxxx	70, 193
CCPR3L	Capture/Compare/PWM Register 3 Low Byte								xxxx xxxx	70, 193
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	70, 198
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	70, 212
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	70, 355
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	70, 349
TMR3H	Timer3 Register High Byte								xxxx xxxx	70, 183
TMR3L	Timer3 Register Low Byte								xxxx xxxx	70, 183

Legend: x = unknown; u = unchanged; - = unimplemented, read as '0'; q = value depends on condition; r = reserved bit, do not modify. Shaded cells are unimplemented, read as '0'.

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

2: Bit 21 of the PC is only available in Serial Programming modes.

3: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

4: Alternate names and definitions for these bits when the MSSP module is operating in I^2C™ Slave mode.

5: These bits and/or registers are only available in 100-pin devices; otherwise, they are unimplemented and read as '0'. Reset values shown apply only to 100-pin devices.

6: These bits and/or registers are only available in 80-pin and 100-pin devices. In 64-pin devices, they are unimplemented and read as '0'. Reset values are shown for 100-pin devices.

7: In Microcontroller mode, the bits in this register are unwritable and read as '0'.

8: PLLEN is only available when either ECPLL or HSPLL Oscillator mode is selected; otherwise, read as '0'.

9: Implemented in 100-pin devices in Microcontroller mode only.

TABLE 6-5: REGISTER FILE SUMMARY (PIC18F97J60 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Values on POR, BOR	Details on Page:
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0000 0000	71, 183
PSPCON(5)	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	71, 169
SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								0000 0000	71, 320
RCREG1	EUSART1 Receive Register								0000 0000	71, 327
TXREG1	EUSART1 Transmit Register								xxxx xxxx	71, 329
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	71, 320
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	71, 320
EECON2	Program Memory Control Register (not a physical register)								---- ----	71, 106
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	---0 x00-	71, 107
IPR3	SSP2IP(5)	BCL2IP(5)	RC2IP(6)	TX2IP(6)	TMR4IP	CCP5IP	CCP4IP	CCP3IP	1111 1111	71, 142
PIR3	SSP2IF(5)	BCL2IF(5)	RC2IF(6)	TX2IF(6)	TMR4IF	CCP5IF	CCP4IF	CCP3IF	0000 0000	71, 136
PIE3	SSP2IE(5)	BCL2IE(5)	RC2IE(6)	TX2IE(6)	TMR4IE	CCP5IE	CCP4IE	CCP3IE	0000 0000	71, 139
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	—	TMR3IP	CCP2IP	1111 1-11	71, 141
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	—	TMR3IF	CCP2IF	0000 0-00	71, 135
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	—	TMR3IE	CCP2IE	0000 0-00	71, 138
IPR1	PSPIP(9)	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	71, 140
PIR1	PSPIF(9)	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	71, 134
PIE1	PSPIE(9)	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	71, 137
MEMCON(5,7)	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	71, 116
OSCTUNE	PPST1	PLLEN(8)	PPST0	PPRE	—	—	—	—	0000 ----	71, 51
TRISJ(6)	TRISJ7(5)	TRISJ6(5)	TRISJ5(6)	TRISJ4(6)	TRISJ3(5)	TRISJ2(5)	TRISJ1(5)	TRISJ0(5)	1111 1111	71, 167
TRISH(6)	TRISH7(6)	TRISH6(6)	TRISH5(6)	TRISH4(6)	TRISH3(6)	TRISH2(6)	TRISH1(6)	TRISH0(6)	1111 1111	71, 165
TRISG	TRISG7(5)	TRISG6(5)	TRISG5(5)	TRISG4	TRISG3(6)	TRISG2(6)	TRISG1(6)	TRISG0(6)	1111 1111	71, 163
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0(5)	1111 1111	71, 161
TRISE	TRISE7(6)	TRISE6(6)	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111	71, 159
TRISD	TRISD7(5)	TRISD6(5)	TRISD5(5)	TRISD4(5)	TRISD3(5)	TRISD2	TRISD1	TRISD0	1111 1111	71, 156
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	71, 153
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	71, 150
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	71, 147
LATJ(6)	LATJ7(5)	LATJ6(5)	LATJ5(6)	LATJ4(6)	LATJ3(5)	LATJ2(5)	LATJ1(5)	LATJ0(5)	xxxx xxxx	71, 167
LATH(6)	LATH7(6)	LATH6(6)	LATH5(6)	LATH4(6)	LATH3(6)	LATH2(6)	LATH1(6)	LATH0(6)	xxxx xxxx	71, 165
LATG	LATG7(5)	LATG6(5)	LATG5(5)	LATG4	LATG3(6)	LATG2(6)	LATG1(6)	LATG0(6)	xxxx xxxx	72, 163
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0(5)	xxxx xxxx	72, 161
LATE	LATE7(6)	LATE6(6)	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx	72, 159
LATD	LATD7(5)	LATD6(5)	LATD5(5)	LATD4(5)	LATD3(5)	LATD2	LATD1	LATD0	xxxx xxxx	72, 156
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	72, 153
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	72, 150
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	00xx xxxx	72, 147
PORTJ(6)	RJ7(5)	RJ6(5)	RJ5(6)	RJ4(6)	RJ3(5)	RJ2(5)	RJ1(5)	RJ0(5)	xxxx xxxx	72, 167
PORTH(6)	RH7(6)	RH6(6)	RH5(6)	RH4(6)	RH3(6)	RH2(6)	RH1(6)	RH0(6)	0000 xxxx	72, 165
PORTG	RG7(5)	RG6(5)	RG5(5)	RG4	RG3(6)	RG2(6)	RG1(6)	RG0(6)	111x xxxx	72, 163

Legend: x = unknown; u = unchanged; - = unimplemented, read as '0'; q = value depends on condition; r = reserved bit, do not modify. Shaded cells are unimplemented, read as '0'.

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

2: Bit 21 of the PC is only available in Serial Programming modes.

3: Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

4: Alternate names and definitions for these bits when the MSSP module is operating in I^2C™ Slave mode.

5: These bits and/or registers are only available in 100-pin devices; otherwise, they are unimplemented and read as '0'. Reset values shown apply only to 100-pin devices.

6: These bits and/or registers are only available in 80-pin and 100-pin devices. In 64-pin devices, they are unimplemented and read as '0'. Reset values are shown for 100-pin devices.

7: In Microcontroller mode, the bits in this register are unwritable and read as '0'.

8: PLLEN is only available when either ECPLL or HSPLL Oscillator mode is selected; otherwise, read as '0'.

9: Implemented in 100-pin devices in Microcontroller mode only.

REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
OSCFIF	CMIF	ETHIF	r	BCL1IF	—	TMR3IF	CCP2IF
bit 7							bit 0

Legend:	r = Reserved bit		
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 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = System oscillator failed, clock input has changed to INTRC (must be cleared in software)
0 = System clock is operating

bit 6 **CMIF:** Comparator Interrupt Flag bit
1 = Comparator input has changed (must be cleared in software)
0 = Comparator input has not changed

bit 5 **ETHIF:** Ethernet Module Interrupt Flag bit
1 = An Ethernet module interrupt event has occurred; query EIR register to resolve source
0 = No Ethernet interrupt event has occurred

bit 4 **Reserved:** Maintain as '0'

bit 3 **BCL1IF:** Bus Collision Interrupt Flag bit (MSSP1 module)
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

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

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **CCP2IF:** ECCP2 Interrupt Flag bit
Capture mode:
1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred
Compare mode:
1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 register compare match occurred
PWM mode:
Unused in this mode.

REGISTER 10-12: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SSP2IP(1)	BCL2IP(1)	RC2IP(2)	TX2IP(2)	TMR4IP	CCP5IP	CCP4IP	CCP3IP
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 **SSP2IP:** MSSP2 Interrupt Priority bit(1)
1 = High priority
0 = Low priority

bit 6 **BCL2IP:** Bus Collision Interrupt Priority bit (MSSP2 module)(1)
1 = High priority
0 = Low priority

bit 5 **RC2IP:** EUSART2 Receive Interrupt Priority bit(2)
1 = High priority
0 = Low priority

bit 4 **TX2IP:** EUSART2 Transmit Interrupt Priority bit(2)
1 = High priority
0 = Low priority

bit 3 **TMR4IE:** TMR4 to PR4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP5IP:** CCP5 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP4IP:** CCP4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **CCP3IP:** ECCP3 Interrupt Priority bit
1 = High priority
0 = Low priority

Note 1: Implemented in 100-pin devices only.
2: Implemented in 80-pin and 100-pin devices only.

11.11 Parallel Slave Port (PSP)

Note: The Parallel Slave Port is only implemented on 100-pin devices.

PORTD can also function as an 8-bit wide, Parallel Slave Port, or microprocessor port, when control bit, PSPMODE (PSPCON<4>), is set. It is asynchronously readable and writable by the external world through the $\overline{RD}$ control input pin, RE0/AD8/$\overline{RD}$/P2D and $\overline{WR}$ control input pin, RE1/AD9//$\overline{WR}$/P2C.

Note: The Parallel Slave Port is available only in Microcontroller mode.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit, PSPMODE, enables port pin, RE0/AD8/$\overline{RD}$/P2D, to be the $\overline{RD}$ input, RE1/AD9//$\overline{WR}$/P2C to be the $\overline{WR}$ input and RE2/AD10//$\overline{CS}$/P2B to be the $\overline{CS}$ (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set).

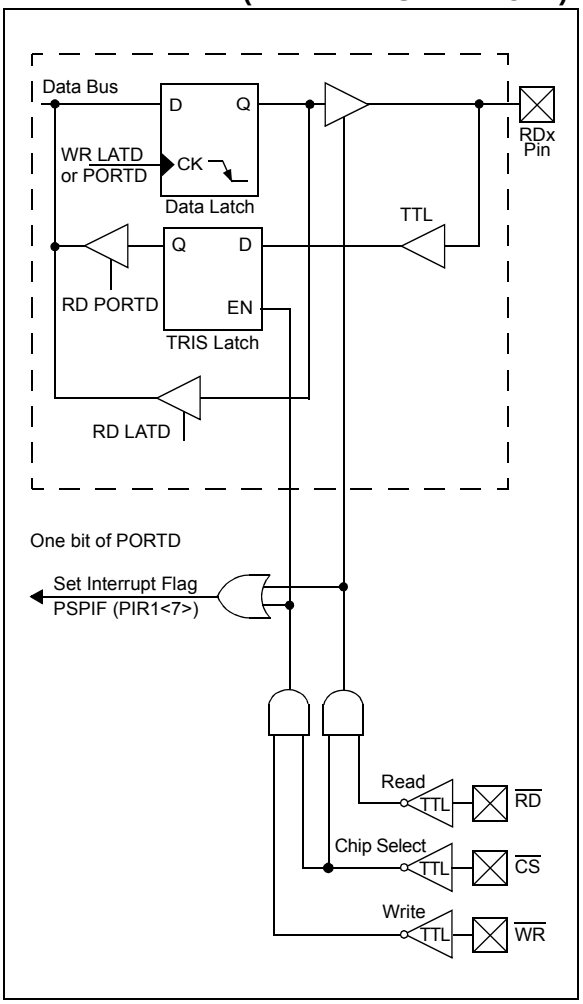
A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits are both set when the write ends.

A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low. The data in PORTD is read out and the OBF bit is set. If the user writes new data to PORTD to set OBF, the data is immediately read out; however, the OBF bit is not set.

When either the $\overline{CS}$ or $\overline{RD}$ lines is detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP. When this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 11-3 and Figure 11-4, respectively.

FIGURE 11-2: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)

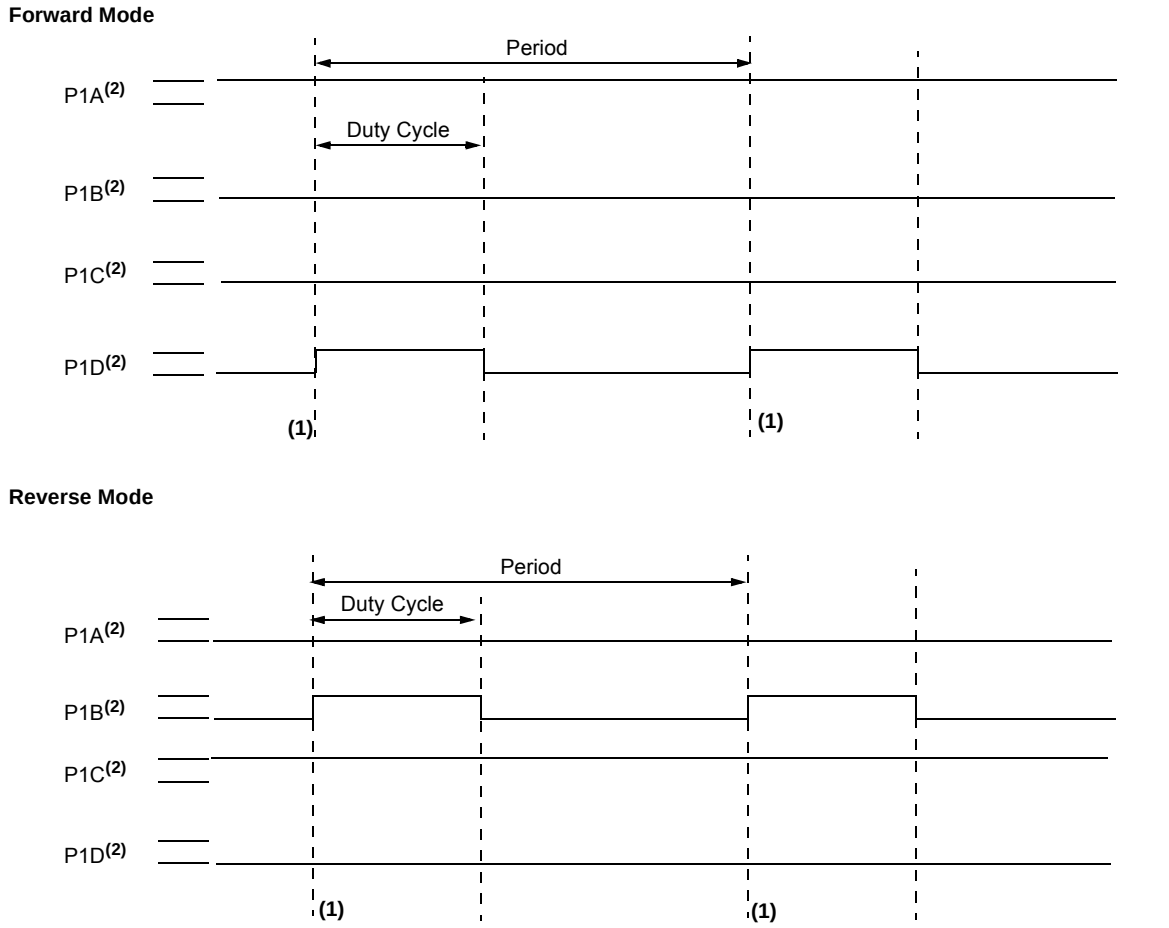


18.4.5 FULL-BRIDGE MODE

In Full-Bridge Output mode, four pins are used as outputs; however, only two outputs are active at a time. In the Forward mode, pin, P1A, is continuously active and pin, P1D, is modulated. In the Reverse mode, pin, P1C, is continuously active and pin, P1B, is modulated. These are illustrated in Figure 18-6.

P1A, P1B, P1C and P1D outputs are multiplexed with the data latches of the port pins listed in Table 18-1 and Table 18-3. The corresponding TRIS bits must be cleared to make the P1A, P1B, P1C and P1D pins outputs.

FIGURE 18-6: FULL-BRIDGE PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.
2: Output signal is shown as active-high.

19.2.4 MAC AND MII REGISTERS

These SFRs are used to control the operations of the MAC, and through the MIIM, the PHY. The MAC and MII registers occupy data addresses, E80h-E85h, E8Ah and EA0h through EB9h.

Although MAC and MII registers appear in the general memory map of the microcontroller, these registers are embedded inside the MAC module. Host interface logic translates the microcontroller data/address bus data to be able to access these registers. The host interface logic imposes restrictions on how firmware is able to access the MAC and MII SFRs. See the following notes.

Note 1: Do not access the MAC and MII SFRs unless the Ethernet module is enabled (ETHEN = 1).

2: Back-to-back accesses of MAC or MII registers are not supported. Between any instruction which addresses a MAC or MII register, at least one NOP or other instruction must be executed.

The three MACON registers control specific MAC operations and packet configuration operations. They are shown in Register 19-4 through Register 19-6.

The MII registers are used to control the MIIM interface and serve as the communication channel with the PHY registers. They are shown in Register 19-7 and Register 19-8.

REGISTER 19-4: MACON1: MAC CONTROL REGISTER 1

U-0	U-0	U-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	r	TXPAUS	RXPAUS	PASSALL	MARXEN
bit 7							bit 0

Legend:	r = Reserved bit		
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 7-5 **Unimplemented:** Read as ‘0’

bit 4 **Reserved:** Do not use

bit 3 **TXPAUS:** Pause Control Frame Transmission Enable bit
1 = Allow the MAC to transmit pause control frames (needed for flow control in full duplex)
0 = Disallow pause frame transmissions

bit 2 **RXPAUS:** Pause Control Frame Reception Enable bit
1 = Inhibit transmissions when pause control frames are received (normal operation)
0 = Ignore pause control frames which are received

bit 1 **PASSALL:** Pass All Received Frames Enable bit
1 = Control frames received by the MAC will be written into the receive buffer if not filtered out
0 = Control frames will be discarded after being processed by the MAC (normal operation)

bit 0 **MARXEN:** MAC Receive Enable bit
1 = Enable packets to be received by the MAC
0 = Disable packet reception

REGISTER 19-20: ERXFCON: ETHERNET RECEIVE FILTER CONTROL REGISTER

R/W-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
UCEN	ANDOR	CRCEN	PMEN	MPEN	HTEN	MCEN	BCEN
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 **UCEN:** Unicast Filter Enable bit
When ANDOR = 1:
1 = Packets not having a destination address matching the local MAC address will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Packets with a destination address matching the local MAC address will be accepted
0 = Filter is disabled

bit 6 **ANDOR:** AND/OR Filter Select bit
1 = AND: Packets will be rejected unless all enabled filters accept the packet
0 = OR: Packets will be accepted unless all enabled filters reject the packet

bit 5 **CRCEN:** Post-Filter CRC Check Enable bit
1 = All packets with an invalid CRC will be discarded
0 = The CRC validity will be ignored

bit 4 **PMEN:** Pattern Match Filter Enable bit
When ANDOR = 1:
1 = Packets must meet the Pattern Match criteria or they will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Packets which meet the Pattern Match criteria will be accepted
0 = Filter is disabled

bit 3 **MPEN:** Magic Packet Filter Enable bit
When ANDOR = 1:
1 = Packets must be Magic Packets for the local MAC address or they will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Magic Packets for the local MAC address will be accepted
0 = Filter is disabled

bit 2 **HTEN:** Hash Table Filter Enable bit
When ANDOR = 1:
1 = Packets must meet the Hash Table criteria or they will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Packets which meet the Hash Table criteria will be accepted
0 = Filter is disabled

bit 1 **MCEN:** Multicast Filter Enable bit
When ANDOR = 1:
1 = The LSb of the first byte of the packet's destination address must be set or it will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Packets which have the LSb of the first byte in the destination address set will be accepted
0 = Filter is disabled

bit 0 **BCEN:** Broadcast Filter Enable bit
When ANDOR = 1:
1 = Packets must have a destination address of FF-FF-FF-FF-FF-FF or they will be discarded
0 = Filter is disabled
When ANDOR = 0:
1 = Packets which have a destination address of FF-FF-FF-FF-FF-FF will be accepted
0 = Filter is disabled

19.8.5 PATTERN MATCH FILTER

The Pattern Match filter selects up to 64 bytes from the incoming packet and calculates an IP checksum of the bytes. The checksum is then compared to the EPMCS registers. The packet meets the Pattern Match filter criteria if the calculated checksum matches the EPMCS registers. The Pattern Match filter may be useful for filtering packets which have expected data inside them.

To use the Pattern Match filter, the application must program the Pattern Match offset (EPMOH:EPMOL), all of the Pattern Match mask bytes (EPMM0:EPMM7) and the Pattern Match Checksum register pair (EPMCSH:EPMCSL). The Pattern Match offset should be loaded with the offset from the beginning of the destination address field to the 64-byte window which will be used for the checksum computation. Within the 64-byte window, each individual byte can be selectively included or excluded from the checksum computation by setting or clearing the respective bit in the Pattern Match mask. If a packet is received which would cause the 64-byte window to extend past the end of the CRC, the filter criteria will immediately not be met, even if the corresponding mask bits are all '0'.

Note:	In all cases, the value of the Pattern Match offset must be even for proper operation. Programming the EMPO register pair with an odd value will cause unpredictable results.

The Pattern Match Checksum registers should be programmed to the checksum which is expected for the selected bytes. The checksum is calculated in the same manner that the DMA module calculates checksums (see **Section 19.9.2 "Checksum Calculations"**). Data bytes which have corresponding mask bits programmed to '0' are completely removed for purposes of calculating the checksum, as opposed to treating the data bytes as zero.

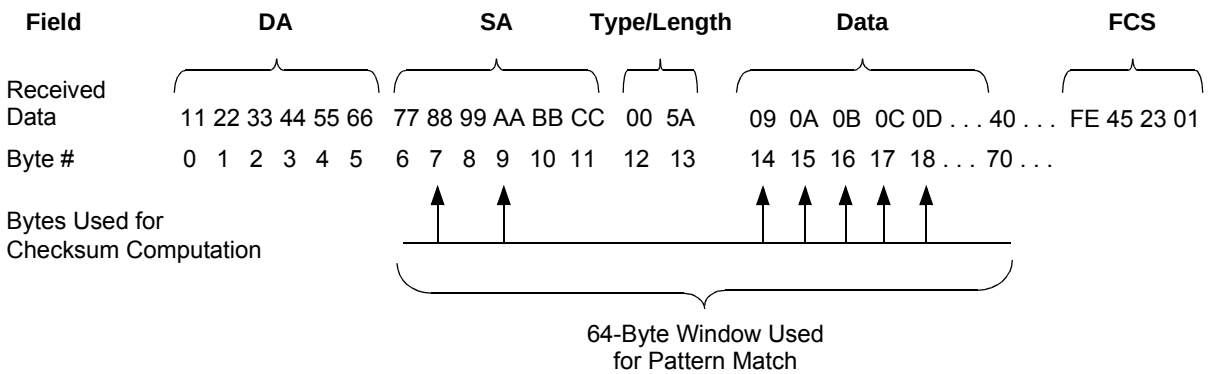
As an example, if the application wished to filter all packets having a particular source MAC address of 00-04-A3-FF-FF-FF, it could program the Pattern Match offset to 0000h and then set bits 6 and 7 of EPMM0 and bits 0, 1, 2 and 3 of EPMM1 (assuming all other mask bits are '0'). The proper checksum to program into the EPMCS registers would be 5BFCh. As an alternative configuration, it could program the offset to 0006h and set bits 0, 1, 2, 3, 4 and 5 of EPMM0. The checksum would still be 5BFCh. However, the second case would be less desirable as packets less than 70 bytes long could never meet the Pattern Match criteria, even if they would generate the proper checksum given the mask configuration.

Another example of a Pattern Match filter is illustrated in Figure 19-15.

FIGURE 19-15: SAMPLE PATTERN MATCH FORMAT

Input Configuration:

EMPOH:EPMOL = 0006h
EPMM7:EPMM0 = 0000000000001F0Ah
EPMCSH:EPMCSL = 563Fh



Values Used for Checksum Computation = {88h, AAh, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 00h}
(00h padding byte added by hardware)

Note: Received data is shown in hexadecimal. Byte numbers are shown in decimal format.

22.1 A/D Acquisition Requirements

For the A/D Converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 22-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor, CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 22-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 22-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 22-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 22-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$
or
$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)$$

EQUATION 22-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$
$$T_{AMP} = 0.2\ \mu s$$
$$T_{COFF} = (\text{Temp} - 25°C)(0.02\ \mu s/°C)$$
$$(85°C - 25°C)(0.02\ \mu s/°C)$$
$$1.2\ \mu s$$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, T_{COFF} = 0 ms.

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)\ \mu s$$
$$-(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega)\ln(0.0004883)\ \mu s$$
$$1.05\ \mu s$$
$$T_{ACQ} = 0.2\ \mu s + 1\ \mu s + 1.2\ \mu s$$
$$2.4\ \mu s$$

25.4 Two-Speed Start-up

The Two-Speed Start-up feature helps to minimize the latency period, from oscillator start-up to code execution, by allowing the microcontroller to use the INTRC oscillator as a clock source until the primary clock source is available. It is enabled by setting the IESO Configuration bit.

Two-Speed Start-up should be enabled only if the primary oscillator mode is HS or HSPLL (Crystal-Based) modes. Since the EC and ECPLL modes do not require an Oscillator Start-up Timer delay, Two-Speed Start-up should be disabled.

When enabled, Resets and wake-ups from Sleep mode cause the device to configure itself to run from the internal oscillator block as the clock source, following the time-out of the Power-up Timer after a Power-on Reset is enabled. This allows almost immediate code execution while the primary oscillator starts and the OST is running. Once the OST times out, the device automatically switches to PRI_RUN mode.

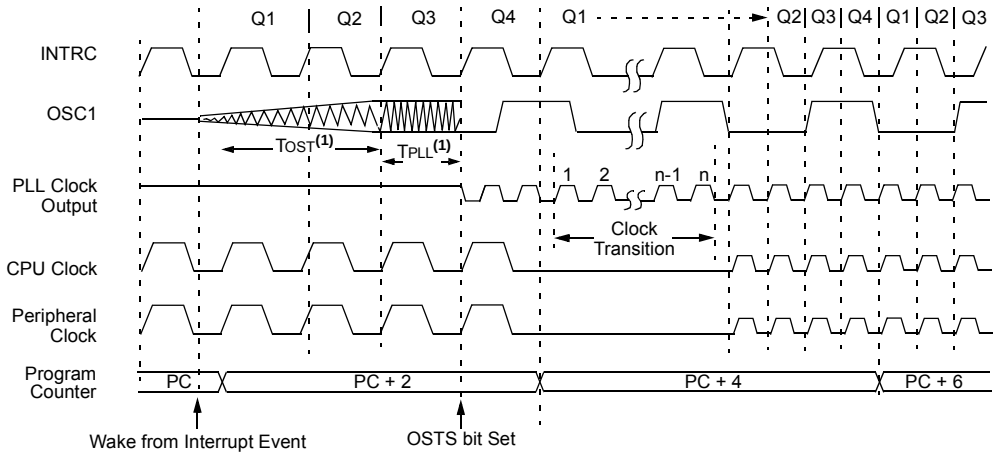
In all other power-managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

25.4.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTRC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-managed modes, including serial SLEEP instructions (refer to **Section 4.1.4 “Multiple Sleep Commands”**). In practice, this means that user code can change the SCS<1:0> bit settings, or issue SLEEP instructions, before the OST times out. This would allow an application to briefly wake-up, perform routine “housekeeping” tasks and return to Sleep before the device starts to operate from the primary oscillator.

User code can also check if the primary clock source is currently providing the device clocking by checking the status of the OSTS bit (OSCCON<3>). If the bit is set, the primary oscillator is providing the clock. Otherwise, the internal oscillator block is providing the clock during wake-up from Reset or Sleep mode.

FIGURE 25-3: TIMING TRANSITION FOR TWO-SPEED START-UP (INTRC TO HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

26.0 INSTRUCTION SET SUMMARY

The PIC18F97J60 family of devices incorporates the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

26.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 26-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 26-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator ‘f’ specifies which file register is to be used by the instruction. The destination designator ‘d’ specifies where the result of the operation is to be placed. If ‘d’ is zero, the result is placed in the WREG register. If ‘d’ is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator ‘f’ represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 26-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 26-2, lists the standard instructions recognized by the Microchip MPASM™ Assembler.

Section 26.1.1 “Standard Instruction Set” provides a description of each instruction.

TBLRD **Table Read**

Syntax: TBLRD (*; *+; *-; +*)

Operands: None

Operation: if TBLRD*,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change
if TBLRD*+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR
if TBLRD*-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT

Status Affected: None

Encoding:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

TBLRD **Table Read (Continued)**

Example 1:
```
TBLRD   *+ ;
```
Before Instruction
TABLAT = 55h
TBLPTR = 00A356h
MEMORY(00A356h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 00A357h

Example 2:
```
TBLRD   +* ;
```
Before Instruction
TABLAT = AAh
TBLPTR = 01A357h
MEMORY(01A357h) = 12h
MEMORY(01A358h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 01A358h

26.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, the PIC18F97J60 family of devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment Indirect and Indexed Addressing operations and the implementation of Indexed Literal Offset Addressing for many of the standard PIC18 instructions.

The additional features of the extended instruction set are enabled by default on unprogrammed devices. Users must properly set or clear the XINST Configuration bit during programming to enable or disable these features.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers, or use them for Indexed Addressing. Two of the instructions, ADDFSR and SUBFSR, each have an additional special instantiation for using FSR2. These versions (ADDULNK and SUBULNK) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize reentrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- Dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- Function Pointer invocation
- Software Stack Pointer manipulation
- Manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 26-3. Detailed descriptions are provided in **Section 26.2.2 "Extended Instruction Set"**. The opcode field descriptions in Table 26-1 (page 376) apply to both the standard and extended PIC18 instruction sets.

Note:	The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

26.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of Indexed Addressing, it is enclosed in square brackets ("[ ]"). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 26.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands"**.

Note:	In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces ("{ }").

TABLE 26-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word				Status Affected
				MSb			LSb	
ADDFSR	f, k	Add Literal to FSR	1	1110	1000	ffkk	kkkk	None
ADDULNK	k	Add Literal to FSR2 and Return	2	1110	1000	11kk	kkkk	None
CALLW		Call Subroutine using WREG	2	0000	0000	0001	0100	None
MOVSF	z_s, f_d	Move z_s (source) to 1st word f_d (destination) 2nd word	2	1110 1111	1011 ffff	0zzz ffff	zzzz ffff	None
MOVSS	z_s, z_d	Move z_s (source) to 1st word z_d (destination) 2nd word	2	1110 1111	1011 xxxx	1zzz xzzz	zzzz zzzz	None
PUSHL	k	Store Literal at FSR2, Decrement FSR2	1	1110	1010	kkkk	kkkk	None
SUBFSR	f, k	Subtract Literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK	k	Subtract Literal from FSR2 and Return	2	1110	1001	11kk	kkkk	None

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 [05h], REG2

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

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

28.3 DC Characteristics: PIC18F97J60 Family (Industrial) (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	VOL	**Output Low Voltage**				
D080		I/O Ports:				
		PORTD, PORTE, PORTJ	—	0.4	V	IOL = 4 mA, VDD = 3.3V, -40°C to +85°C
		PORTA<5:2>, PORTF, PORTG, PORTH	—	0.4	V	IOL = 2 mA, VDD = 3.3V, -40°C to +85°C
		PORTA<1:0>, PORTB, PORTC	—	0.4	V	IOL = 8 mA, VDD = 3.3V, -40°C to +85°C
D083		OSC2/CLKO (EC, ECPLL modes)	—	0.4	V	IOL = 2 mA, VDD = 3.3V, -40°C to +85°C
	VOH	**Output High Voltage(1)**				
D090		I/O Ports:			V	
		PORTD, PORTE, PORTJ	2.4	—	V	IOH = -4 mA, VDD = 3.3V, -40°C to +85°C
		PORTA<5:2>, PORTF, PORTG, PORTH	2.4	—	V	IOH = -2 mA, VDD = 3.3V, -40°C to +85°C
		PORTA<1:0>, PORTB, PORTC	2.4	—	V	IOH = -8 mA, VDD = 3.3V, -40°C to +85°C
D092		OSC2/CLKO (EC, ECPLL modes)	2.4	—	V	IOH = -1.0 mA, VDD = 3.3V, -40°C to +85°C
		Capacitive Loading Specs on Output Pins				
D100	COSC2	OSC2 pin	—	15	pF	In HS mode when external clock is used to drive OSC1
D101	CIO	All I/O pins and OSC2 (in Internal RC mode, EC, ECPLL)	—	50	pF	To meet the AC timing specifications
D102	CB	SCLx, SDAx	—	400	pF	I^2C™ specification

Note 1: Negative current is defined as current sourced by the pin.

APPENDIX A: REVISION HISTORY

Revision A (March 2006)

Original data sheet for the PIC18F97J60 family of devices.

Revision B (October 2006)

First revision. Includes preliminary electrical specifications; revised and updated material on the Ethernet module; updated material on Reset integration; and updates to the device memory map.

Revision C (June 2007)

Corrected Table 10.2: Input Voltage Levels; added content on Ethernet module's reading and writing to the buffer; added new, 100-lead PT 12x12x1 mm TQFP package to "Package Marking Information" and "Package Details" sections; updated other package details drawings; changed Product Identification System examples.

Revision D (January 2008)

Added one line to "Ethernet Features" description. Added land pattern schematics for each package.

Revision E (October 2009)

Updated to remove Preliminary status.

Revision F (April 2011)

Added Brown-out Reset (BOR) specs, added Ethernet RX Auto-Polarity circuit section, added EMI filter section, added **Section 2.0 "Guidelines for Getting Started with PIC18FJ Microcontrollers"**, changed the opcode encoding of the PUSHL instruction to 1110 1010 kkk kkkk and changed the 2 TOSC Maximum Device Frequency in Table 22-1 from 2.68 MHz to the correct value of 2.86 MHz. Updated comparator input offset voltage maximum to the correct value of 25 mV.