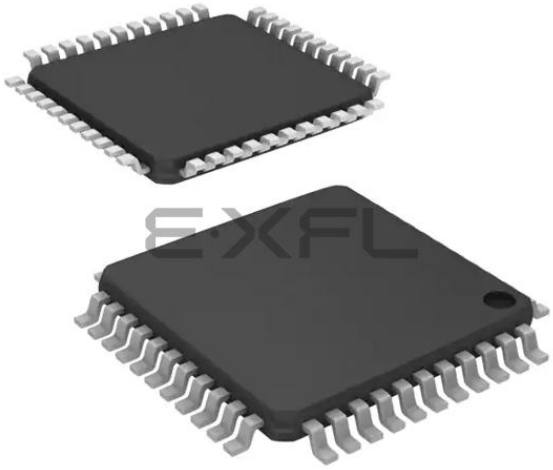


Welcome to E-XFL.COM

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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f44k22-e-pt

TABLE 2: PIC18(L)F2XK22 PIN SUMMARY

28-SSOP, SOIC 28-SPDIP	28-QFN, UQFN	I/O	Analog	Comparator	CTMU	SR Latch	Reference	(E)CCP	EUSART	MSSP	Timers	Interrupts	Pull-up	Basic
2	27	RA0	AN0	C12IN0-										
3	28	RA1	AN1	C12IN1-										
4	1	RA2	AN2	C2IN+			VREF- DACOUT							
5	2	RA3	AN3	C1IN+			VREF+							
6	3	RA4		C1OUT		SRQ		CCP5			T0CKI			
7	4	RA5	AN4	C2OUT		SRNQ	HLVDIN			$\overline{SS}$1				
10	7	RA6												OSC2 CLKO
9	6	RA7												OSC1 CLKI
21	18	RB0	AN12			SRI		CCP4 FLT0		$\overline{SS}$2		INT0	Y	
22	19	RB1	AN10	C12IN3-				P1C		SCK2 SCL2		INT1	Y	
23	20	RB2	AN8		CTED1			P1B		SDI2 SDA2		INT2	Y	
24	21	RB3	AN9	C12IN2-	CTED2			CCP2 P2A(1)		SDO2			Y	
25	22	RB4	AN11					P1D			T5G	IOC	Y	
26	23	RB5	AN13					CCP3 P3A(4) P2B(3)			T1G T3CKI(2)	IOC	Y	
27	24	RB6							TX2/CK2			IOC	Y	PGC
28	25	RB7							RX2/DT2			IOC	Y	PGD
11	8	RC0						P2B(3)			SOSCO T1CKI T3CKI(2) T3G			
12	9	RC1						CCP2 P2A(1)			SOSCI			
13	10	RC2	AN14		CTPLS			CCP1 P1A			T5CKI			
14	11	RC3	AN15							SCK1 SCL1				
15	12	RC4	AN16							SDI1 SDA1				
16	13	RC5	AN17							SDO1				
17	14	RC6	AN18					CCP3 P3A(4)	TX1/CK1					
18	15	RC7	AN19					P3B	RX1/DT1					
1	26	RE3												$\overline{MCLR}$ VPP
8, 19 19	5, 16 16	VSS												VSS
20	17	VDD												VDD

Note 1: CCP2/P2A multiplexed in fuses.
2: T3CKI multiplexed in fuses.
3: P2B multiplexed in fuses.
4: CCP3/P3A multiplexed in fuses.

© 2010-2016 Microchip Technology Inc.

2.9 Effects of Power-Managed Modes on the Various Clock Sources

For more information about the modes discussed in this section see **Section 3.0 "Power-Managed Modes"**. A quick reference list is also available in Table 3-1.

When PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. For all other power-managed modes, the oscillator using the OSC1 pin is disabled. The OSC1 pin (and OSC2 pin, if used by the oscillator) will stop oscillating.

In secondary clock modes (SEC_RUN and SEC_IDLE), the secondary oscillator (SOSC) is operating and providing the device clock. The secondary oscillator may also run in all power-managed modes if required to clock Timer1, Timer3 or Timer5.

In internal oscillator modes (INTOSC_RUN and INTOSC_IDLE), the internal oscillator block provides the device clock source. The 31.25 kHz LFINTOSC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 24.3 "Watchdog Timer (WDT)"**, **Section 2.12 "Two-Speed Clock Start-up Mode"** and **Section 2.13 "Fail-Safe Clock Monitor"** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up). The HFINTOSC and MFINTOSC outputs may be used directly to clock the device or may be divided down by the postscaler. The HFINTOSC and MFINTOSC outputs are disabled when the clock is provided directly from the LFINTOSC output.

When the Sleep mode is selected, all clock sources are stopped. Since all the transistor switching currents have been stopped, Sleep mode achieves the lowest current consumption of the device (only leakage currents).

Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The LFINTOSC is required to support WDT operation. Other features may be operating that do not require a device clock source (i.e., SSP slave, PSP, INTn pins and others). Peripherals that may add significant current consumption are listed in **Section 27.8 "DC Characteristics: Input/Output Characteristics, PIC18(L)F2X/4XK22".**

2.10 Power-up Delays

Power-up delays are controlled by two timers, so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 4.6 "Device Reset Timers"**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up. It is enabled by clearing (= 0) the PWRTEN Configuration bit.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (LP, XT and HS modes). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

When the PLL is enabled with external oscillator modes, the device is kept in Reset for an additional 2 ms, following the OST delay, so the PLL can lock to the incoming clock frequency.

There is a delay of interval T_{CSD}, following POR, while the controller becomes ready to execute instructions. This delay runs concurrently with any other delays. This may be the only delay that occurs when any of the EC, RC or INTIOSC modes are used as the primary clock source.

When the HFINTOSC is selected as the primary clock, the main system clock can be delayed until the HFINTOSC is stable. This is user selectable by the HFOFST bit of the CONFIG3H Configuration register. When the HFOFST bit is cleared, the main system clock is delayed until the HFINTOSC is stable. When the HFOFST bit is set, the main system clock starts immediately.

In either case, the HFIOFS bit of the OSCCON register can be read to determine whether the HFINTOSC is operating and stable.

© 2010-2016 Microchip Technology Inc.

TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FFFh	TOSU	—	—	—	Top-of-Stack, Upper Byte (TOS<20:16>)					---0 0000
FFEh	TOSH	Top-of-Stack, High Byte (TOS<15:8>)								0000 0000
FFDh	TOSL	Top-of-Stack, Low Byte (TOS<7:0>)								0000 0000
FFCh	STKPTR	STKFUL	STKUNF	—	STKPTR<4:0>					00-0 0000
FFBh	PCLATU	—	—	—	Holding Register for PC<20:16>					---0 0000
FFAh	PCLATH	Holding Register for PC<15:8>								0000 0000
FF9h	PCL	Holding Register for PC<7:0>								0000 0000
FF8h	TBLPTRU	—	—	Program Memory Table Pointer Upper Byte(TBLPTR<21:16>)						--00 0000
FF7h	TBLPTRH	Program Memory Table Pointer High Byte(TBLPTR<15:8>)								0000 0000
FF6h	TBLPTRL	Program Memory Table Pointer Low Byte(TBLPTR<7:0>)								0000 0000
FF5h	TABLAT	Program Memory Table Latch								0000 0000
FF4h	PRODH	Product Register, High Byte								xxxx xxxx
FF3h	PRODL	Product Register, Low Byte								xxxx xxxx
FF2h	INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x
FF1h	INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1
FF0h	INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00
FEFh	INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								---- ----
FEEh	POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								---- ----
FEDh	POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								---- ----
FECh	PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								---- ----
FEBh	PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W								---- ----
FEAh	FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0, High Byte				---- 0000
FE9h	FSR0L	Indirect Data Memory Address Pointer 0, Low Byte								xxxx xxxx
FE8h	WREG	Working Register								xxxx xxxx
FE7h	INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)								---- ----
FE6h	POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)								---- ----
FE5h	POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)								---- ----
FE4h	PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)								---- ----
FE3h	PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W								---- ----
FE2h	FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1, High Byte				---- 0000
FE1h	FSR1L	Indirect Data Memory Address Pointer 1, Low Byte								xxxx xxxx
FE0h	BSR	—	—	—	—	Bank Select Register				---- 0000
FDFh	INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)								---- ----
FDEh	POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)								---- ----
FDDh	POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)								---- ----
FDCh	PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)								---- ----
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								---- ----
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	T0PS<2:0>			1111 1111
FD3h	OSCCON	IDLEN	IRCF<2:0>			OSTS	HFIOFS	SCS<1:0>		0011 q000
FD2h	OSCCON2	PLLRDY	SOSCRUN	—	MFIOSEL	SOSCGO	PRISD	MFIOFS	LFIOFS	00-0 01x0

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: PIC18(L)F4XK22 devices only.

2: PIC18(L)F2XK22 devices only.

3: PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.

4: PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FD1h	WDTCON	—	—	—	—	—	—	—	SWDTEN	---- ---0
FD0h	RCON	IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	01-1 1100
FCFh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx
FCEh	TMR1L	Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx
FCDh	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1SOSCEN	$\overline{T1SYNC}$	T1RD16	TMR1ON	0000 0000
FCCh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{DONE}$	T1GVAL	T1GSS<1:0>		0000 xx00
FCBh	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000
FCAh	SSP1MSK	SSP1 MASK Register bits								1111 1111
FC9h	SSP1BUF	SSP1 Receive Buffer/Transmit Register								xxxx xxxx
FC8h	SSP1ADD	SSP1 Address Register in I^2C Slave Mode. SSP1 Baud Rate Reload Register in I^2C Master Mode								0000 0000
FC7h	SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000
FC6h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000
FC5h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
FC4h	ADRESH	A/D Result, High Byte								xxxx xxxx
FC3h	ADRESL	A/D Result, Low Byte								xxxx xxxx
FC2h	ADCON0	—	CHS<4:0>					GO/$\overline{DONE}$	ADON	--00 0000
FC1h	ADCON1	TRIGSEL	—	—	—	PVCFG<1:0>		NVCFG<1:0>		0--- 0000
FC0h	ADCON2	ADFM	—	ACQT<2:0>			ADCS<2:0>			0-00 0000
FBFh	CCPR1H	Capture/Compare/PWM Register 1, High Byte								xxxx xxxx
FBEh	CCPR1L	Capture/Compare/PWM Register 1, Low Byte								xxxx xxxx
FBDh	CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				0000 0000
FBCh	TMR2	Timer2 Register								0000 0000
FBBh	PR2	Timer2 Period Register								1111 1111
FBAh	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000
FB9h	PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	---0 0001
FB8h	BAUDCON1	ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN	0100 0-00
FB7h	PWM1CON	P1RSEN	P1DC<6:0>							0000 0000
FB6h	ECCP1AS	CCP1ASE	CCP1AS<2:0>			PSS1AC<1:0>		PSS1BD<1:0>		0000 0000
FB4h	T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{DONE}$	T3GVAL	T3GSS<1:0>		0000 0x00
FB3h	TMR3H	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								xxxx xxxx
FB2h	TMR3L	Least Significant Byte of the 16-bit TMR3 Register								xxxx xxxx
FB1h	T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSCEN	$\overline{T3SYNC}$	T3RD16	TMR3ON	0000 0000
FB0h	SPBRGH1	EUSART1 Baud Rate Generator, High Byte								0000 0000
FAFh	SPBRG1	EUSART1 Baud Rate Generator, Low Byte								0000 0000
FAEh	RCREG1	EUSART1 Receive Register								0000 0000
FADh	TXREG1	EUSART1 Transmit Register								0000 0000
FACh	TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010
FABh	RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
FAAh	EEADRH(5)	—	—	—	—	—	—	EEADR<9:8>		---- --00
FA9h	EEADR	EEADR<7:0>								0000 0000
FA8h	EEDATA	EEPROM Data Register								0000 0000
FA7h	EECON2	EEPROM Control Register 2 (not a physical register)								---- --00
FA6h	EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000
FA5h	IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	0000 0000
FA4h	PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	0000 0000
FA3h	PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	0000 0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: PIC18(L)F4XK22 devices only.
2: PIC18(L)F2XK22 devices only.
3: PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.
4: PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

© 2010-2016 Microchip Technology Inc.

REGISTER 9-7: PIR4: PERIPHERAL INTERRUPT (FLAG) REGISTER 4

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF
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 **CCP5IF:** CCP5 Interrupt Flag bits

Capture mode:
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare mode:
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM mode:
Unused in PWM mode.

bit 1 **CCP4IF:** CCP4 Interrupt Flag bits

Capture mode:
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare mode:
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM mode:
Unused in PWM mode.

bit 0 **CCP3IF:** ECCP3 Interrupt Flag bits

Capture mode:
1 = A TMR register capture occurred (must be cleared in software)
0 = No TMR register capture occurred

Compare mode:
1 = A TMR register compare match occurred (must be cleared in software)
0 = No TMR register compare match occurred

PWM mode:
Unused in PWM mode.

10.4 PORTC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is multiplexed with several peripheral functions (Table 10-8). The pins have Schmitt Trigger input buffers.

Some of these pin functions can be relocated to alternate pins using the Control fuse bits in CONFIG3H. RC0 is the default pin for T3CKI. Clearing the T3CMX bit moves the pin function to RB5. RC1 is the default pin for the CCP2 peripheral pin. Clearing the CCP2MX bit moves the pin function to the RB3 pin.

Two other pin functions, P2B and CCP3, can be relocated from their default pins to PORTC pins by clearing the control fuses in CONFIG3H. Clearing P2BMX and CCP3MX moves the pin functions to RC0 and RC6**(1)**/RE0**(2)**, respectively.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. The EUSART and MSSP peripherals override the TRIS bit to make a pin an output or an input, depending on the peripheral configuration. Refer to the corresponding peripheral section for additional information.

Note:	On a Power-on Reset, these pins are configured as analog inputs.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

EXAMPLE 10-3: INITIALIZING PORTC

```
MOVLB   0xF     ; Set BSR for banked SFRs
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
MOVLW   30h     ; Value used to
                ; enable digital inputs
MOVWF   ANSELC  ; RC<3:2> dig input enable
                ; No ANSEL bits for RC<1:0>
                ; RC<7:6> dig input enable
```

10.4.1 PORTC OUTPUT PRIORITY

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTC pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

10.5 PORTD Registers

Note: PORTD is only available on 40-pin and 44-pin devices.

PORTD is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., disable the output driver). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

All of the PORTD pins are multiplexed with analog and digital peripheral modules. See Table 10-11.

Note: On a Power-on Reset, these pins are configured as analog inputs.

EXAMPLE 10-4: INITIALIZING PORTD

```
MOVLB   0xF     ; Set BSR for banked SFRs
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
MOVLW   30h     ; Value used to
                ; enable digital inputs
MOVWF   ANSELD  ; RD<3:0> dig input enable
                ; RC<7:6> dig input enable
```

10.5.1 PORTD OUTPUT PRIORITY

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTD pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

REGISTER 10-2: PORTE: PORTE REGISTER

U-0	U-0	U-0	U-0	R/W-u/x	R/W-u/x	R/W-u/x	R/W-u/x
—	—	—	—	RE3(1)	RE2(2), (3)	RE1(2), (3)	RE0(2), (3)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
-n/n = Value at POR and BOR/Value at all other Resets		

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **RE3:** PORTE Input bit value(1)

bit 2-0 **RE<2:0>:** PORTE I/O bit values(2), (3)

Note 1: Port is available as input only when MCLRE = 0.

2: Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

3: Available on PIC18(L)F4XK22 devices.

REGISTER 10-3: ANSELA – PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0
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

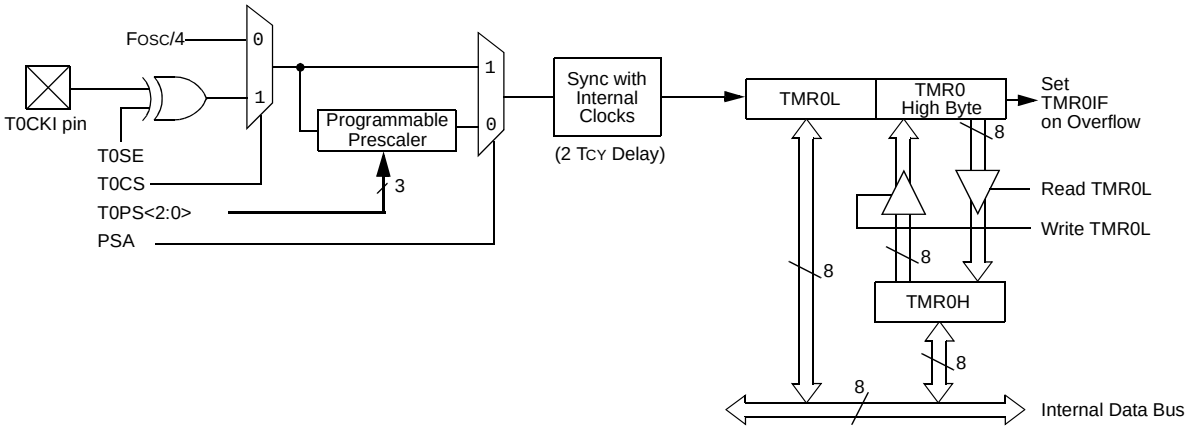
bit 7-6 **Unimplemented:** Read as '0'

bit 5 **ANSA5:** RA5 Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

bit 4 **Unimplemented:** Read as '0'

bit 3-0 **ANSA<3:0>:** RA<3:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

11.4 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable; its value is set by the PSA and T0PS<2:0> bits of the T0CON register which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When the prescaler is assigned, prescale values from 1:2 through 1:256 in integer power-of-2 increments are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

Note:	Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

11.4.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed “on-the-fly” during program execution.

11.5 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit of the INTCON register. Before re-enabling the interrupt, the TMR0IF bit must be cleared by software in the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	110
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS<2:0>			154
TMR0H	Timer0 Register, High Byte								—
TMR0L	Timer0 Register, Low Byte								—
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used by Timer0.

© 2010-2016 Microchip Technology Inc.

15.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPxOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 15-5.

An MSSPx interrupt is generated for each transferred data byte. Flag bit, SSPxIF, must be cleared by software.

When the SEN bit of the SSPxCON2 register is set, SCLx will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register, except sometimes in 10-bit mode. See **Section 15.2.3 "SPI Master Mode"** for more detail.

15.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSPx module configured as an I^2C slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 15-14 and Figure 15-5 are used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I^2C communication.

1. Start bit detected.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDAx low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCLx line.
8. The master clocks out a data byte.
9. Slave drives SDAx low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes Idle.

15.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the 8th falling edge of SCLx. These additional interrupts allow the slave software to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 15-16 displays a module using both address and data holding. Figure 15-17 includes the operation with the SEN bit of the SSPxCON2 register set.

1. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSPxIF is set and CKP cleared after the 8th falling edge of SCLx.
3. Slave clears the SSPxIF.
4. Slave can look at the ACKTIM bit of the SSPxCON3 register to determine if the SSPxIF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPxBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPxIF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSPxIF

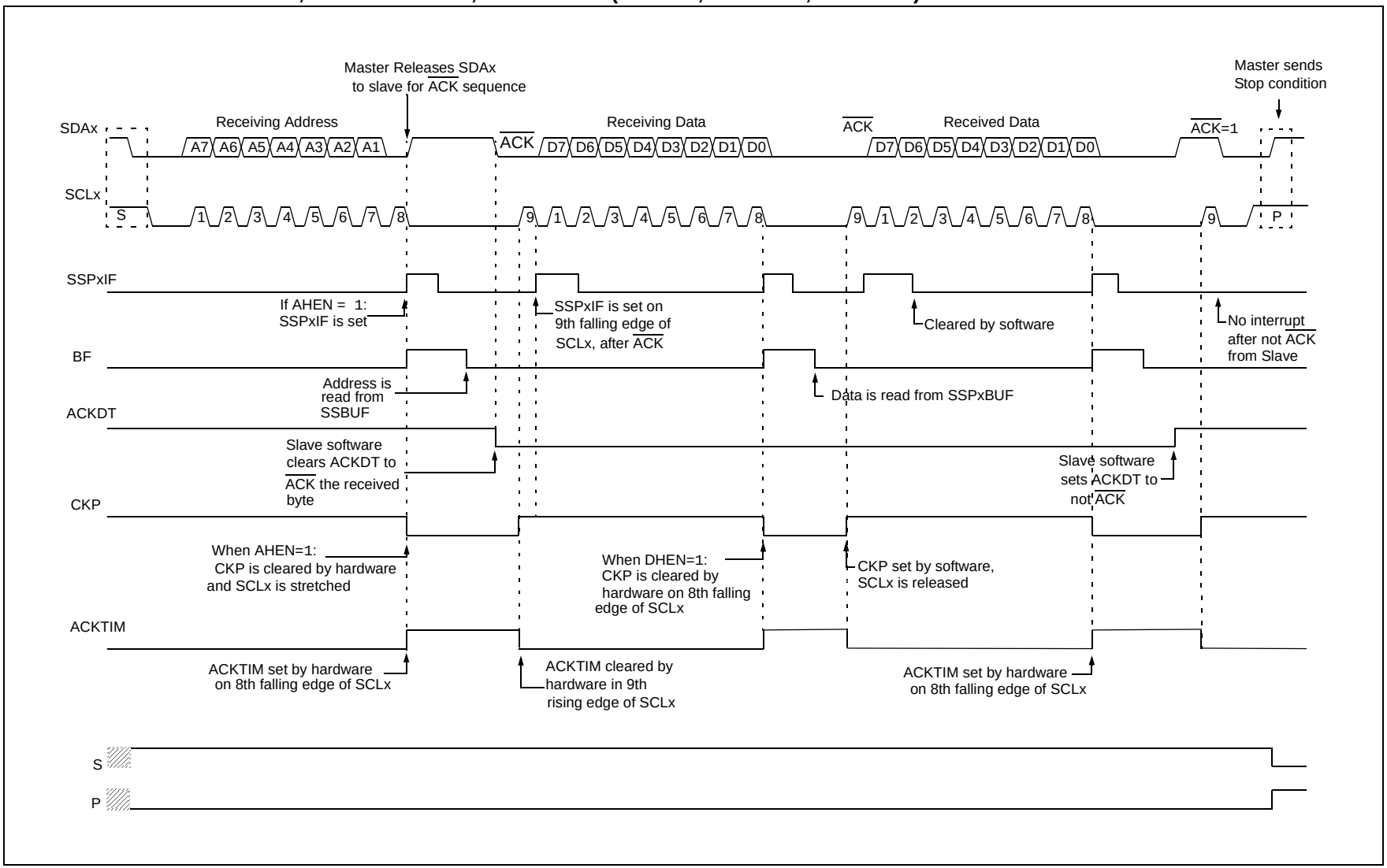
.

Note:	SSPxIF is still set after the 9th falling edge of SCLx even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPxIF not set.

11. SSPxIF set and CKP cleared after 8th falling edge of SCLx for a received data byte.
12. Slave looks at ACKTIM bit of SSPxCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPxBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

© 2010-2016 Microchip Technology Inc.

FIGURE 15-16: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)



© 2010-2016 Microchip Technology Inc.

16.5.1.6 Synchronous Master Reception

Data is received at the RXx/DTx pin. The RXx/DTx pin output driver must be disabled by setting the corresponding TRIS bits when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTAx register) or the Continuous Receive Enable bit (CREN of the RCSTAx register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RXx/DTx pin on the trailing edge of the TXx/CKx clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCxIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREGx. The RCxIF bit remains set as long as there are un-read characters in the receive FIFO.

16.5.1.7 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TXx/CKx line. The TXx/CKx pin output driver must be disabled by setting the associated TRIS bit when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

16.5.1.8 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREGx is read to access the FIFO. When this happens the OERR bit of the RCSTAx register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREGx.

If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

16.5.1.9 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTAx register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTAx register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREGx.

16.5.1.10 Synchronous Master Reception Setup:

1. Initialize the SPBRGHx, SPBRGx register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Set the RXx/DTx and TXx/CKx TRIS controls to ‘1’.
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Disable RXx/DTx and TXx/CKx output drivers by setting the corresponding TRIS bits.
4. Ensure bits CREN and SREN are clear.
5. If using interrupts, set the GIE/GIEH and PEIE/GIEL bits of the INTCON register and set RCxIE.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
9. Read the RCSTAx register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREGx register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

17.1.7 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON2 register controls the output format.

Figure 17-2 shows the two output formats.

FIGURE 17-2: 10-BIT A/D CONVERSION RESULT FORMAT

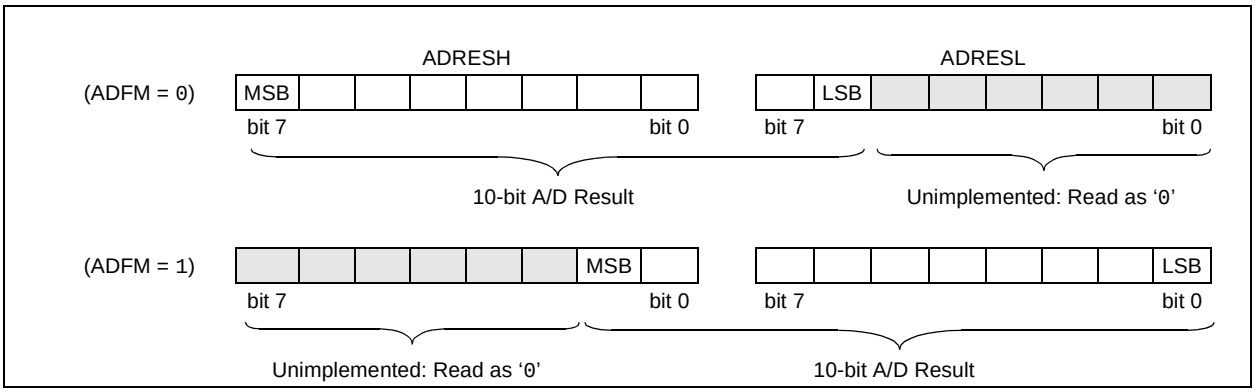


TABLE 17-2: REGISTERS ASSOCIATED WITH A/D OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	295
ADCON1	TRIGSEL	—	—	—	PVCFG<1:0>		NVCFG<1:0>		296
ADCON2	ADFM	—	ACQT<2:0>			ADCS<2:0>			297
ADRESH	A/D Result, High Byte								298
ADRESL	A/D Result, Low Byte								298
ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	149
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	150
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	150
ANSELD(1)	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	150
ANSELE(1)	—	—	—	—	—	ANSE2	ANSE1	ANSE0	151
CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				198
CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	323
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	124
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	120
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	115
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
PMD2	—	—	—	—	CTMUMD	CMP2MD	CMP1MD	ADCMD	54
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
TRISE	WPUE3	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)	151

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used by this module.

Note 1: Available on PIC18(L)F4XK22 devices.

TABLE 17-3: CONFIGURATION REGISTERS ASSOCIATED WITH THE ADC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used by the ADC module.

EXAMPLE 19-2: CURRENT CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 500                           //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA
#define ADSCALE 1023                        //for unsigned conversion 10 sig bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+

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

//assume CTMU and A/D have been set up 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

}
```


© 2010-2016 Microchip Technology Inc.

REGISTER 19-2: CTMUCONL: CTMU CONTROL REGISTER 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
EDG2POL	EDG2SEL<1:0>		EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
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 **EDG2POL:** Edge 2 Polarity Select bit
1 = Edge 2 programmed for a positive edge response
0 = Edge 2 programmed for a negative edge response

bit 6-5 **EDG2SEL<1:0>:** Edge 2 Source Select bits
11 = CTED1 pin
10 = CTED2 pin
01 = ECCP1 Special Event Trigger
00 = ECCP2 Special Event Trigger

bit 4 **EDG1POL:** Edge 1 Polarity Select bit
1 = Edge 1 programmed for a positive edge response
0 = Edge 1 programmed for a negative edge response

bit 3-2 **EDG1SEL<1:0>:** Edge 1 Source Select bits
11 = CTED1 pin
10 = CTED2 pin
01 = ECCP1 Special Event Trigger
00 = ECCP2 Special Event Trigger

bit 1 **EDG2STAT:** Edge 2 Status bit
1 = Edge 2 event has occurred
0 = Edge 2 event has not occurred

bit 0 **EDG1STAT:** Edge 1 Status bit
1 = Edge 1 event has occurred
0 = Edge 1 event has not occurred

© 2010-2016 Microchip Technology Inc.

ANDWF **AND W with f**

Syntax: ANDWF f {,d {,a}}

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:	0001	01da	ffff	ffff

Description: The contents of W are AND'ed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `ANDWF REG, 0, 0`

Before Instruction
W = 17h
REG = C2h
After Instruction
W = 02h
REG = C2h

BC **Branch if Carry**

Syntax: BC n

Operands: $-128 \le n \le 127$

Operation: if CARRY bit is '1'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:	1110	0010	nnnn	nnnn

Description: If the CARRY bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BC 5`

Before Instruction
PC = address (HERE)
After Instruction
If CARRY = 1;
PC = address (HERE + 12)
If CARRY = 0;
PC = address (HERE + 2)

© 2010-2016 Microchip Technology Inc.

.

27.7 DC Characteristics: Secondary Oscillator Supply Current, PIC18(L)F2X/4XK22

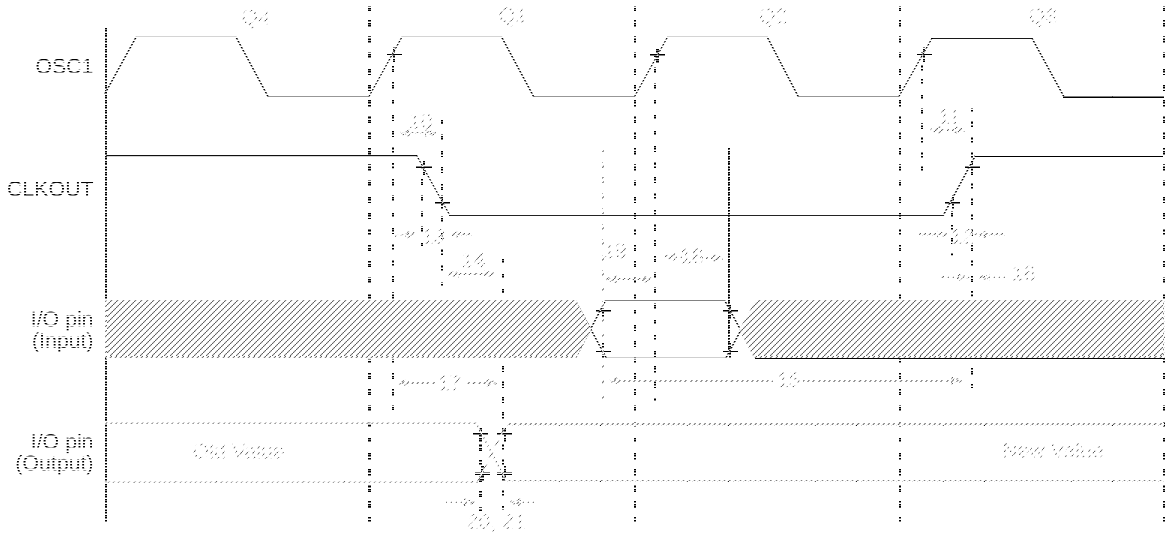
PIC18LF2X/4XK22		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
PIC18F2X/4XK22		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Device Characteristics**	**Typ**	**Max**	**Units**	**Conditions**		
D130	Supply Current (IDD)(1),(2)	3.5	23	µA	-40°C	VDD = 1.8V	Fosc = 32 kHz (**SEC_RUN** mode, SOSC source)
		3.7	25	µA	+25°C		
		3.8	—	µA	+60°C		
		4.0	28	µA	+85°C		
		5.1	30	µA	+125°C		
D131		6.2	26	µA	-40°C	VDD = 3.0V	
		6.4	30	µA	+25°C		
		6.5	—	µA	+60°C		
		6.8	35	µA	+85°C		
		7.8	40	µA	+125°C		
D132		15	35	µA	-40°C	VDD = 2.3V	Fosc = 32 kHz (**SEC_RUN** mode, SOSC source)
		16	35	µA	+25°C		
		17	35	µA	+85°C		
		19	50	µA	+125°C		
D133		18	50	µA	-40°C	VDD = 3.0V	
		19	50	µA	+25°C		
		21	50	µA	+85°C		
		22	60	µA	+125°C		
D134		19	55	µA	-40°C	VDD = 5.0V	
		20	55	µA	+25°C		
		22	55	µA	+85°C		
		23	70	µA	+125°C		

Note 1: 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.
Test condition: All Peripheral Module Control bits in PMD0, PMD1 and PMD2 set to '1'.

2: The test conditions for all IDD measurements in active operation mode are:
All I/O pins set as outputs driven to Vss;
$\overline{MCLR}$ = VDD;
SOSCI / SOSCO = complementary external square wave, from rail-to-rail.

© 2010-2016 Microchip Technology Inc.

FIGURE 27-8: CLKOUT AND I/O TIMING



Note: Refer to Figure 27-6 for load conditions.

TABLE 27-10: CLKOUT AND I/O TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
10	TosH2ckL	OSC1 ↑ to CLKOUT ↓	—	75	200	ns	**(Note 1)**
11	TosH2ckH	OSC1 ↑ to CLKOUT ↑	—	75	200	ns	**(Note 1)**
12	TckR	CLKOUT Rise Time	—	35	100	ns	**(Note 1)**
13	TckF	CLKOUT Fall Time	—	35	100	ns	**(Note 1)**
14	TckL2ioV	CLKOUT ↓ to Port Out Valid	—	—	0.5 TCY + 20	ns	**(Note 1)**
15	TioV2ckH	Port In Valid before CLKOUT ↑	0.25 TCY + 25	—	—	ns	**(Note 1)**
16	TckH2ioI	Port In Hold after CLKOUT ↑	0	—	—	ns	**(Note 1)**
17	TosH2ioV	OSC1 ↑ (Q1 cycle) to Port Out Valid	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	100	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns	
20	TioR	Port Output Rise Time	— —	40 15	72 32	ns ns	VDD = 1.8V VDD = 3.3V - 5.0V
21	TioF	Port Output Fall Time	— —	28 15	55 30	ns ns	VDD = 1.8V VDD = 3.3V - 5.0V
22†	TINP	INTx pin High or Low Time	20	—	—	ns	
23†	TRBP	RB<7:4> Change KBIx High or Low Time	TCY	—	—	ns	

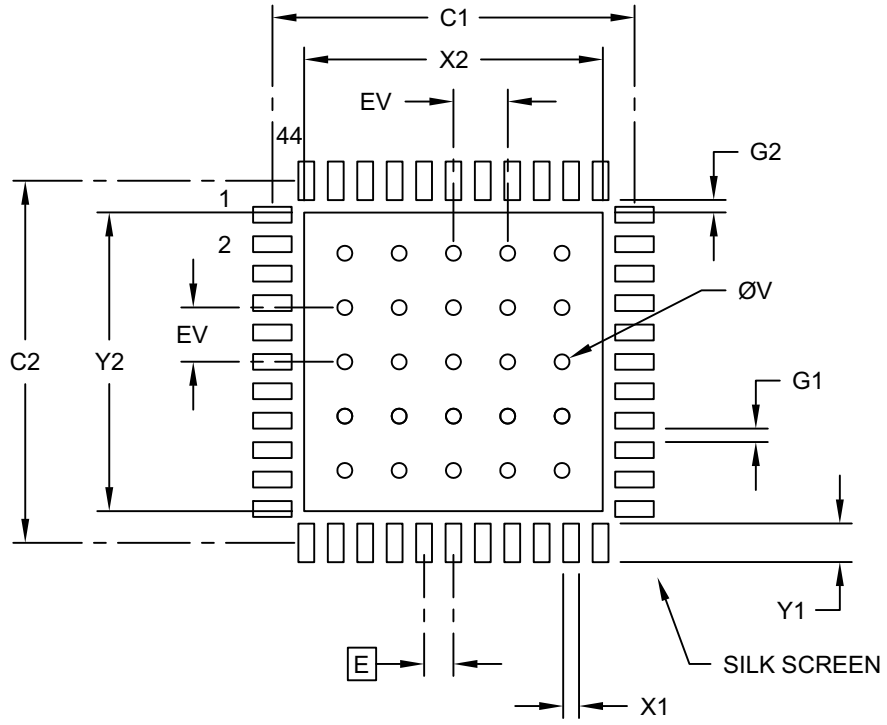
† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC mode, where CLKOUT output is 4 x TOSC.

© 2010-2016 Microchip Technology Inc.

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

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		0.65 BSC	
Optional Center Pad Width	X2			6.60
Optional Center Pad Length	Y2			6.60
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.85
Contact Pad to Contact Pad (X40)	G1	0.30		
Contact Pad to Center Pad (X44)	G2	0.28		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing No. C04-2103C

© 2010-2016 Microchip Technology Inc.