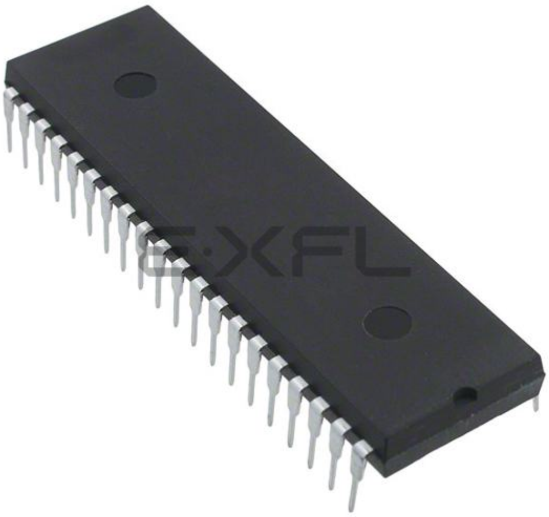
E·XFL

Microchip Technology - PIC18LC452-I/P Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

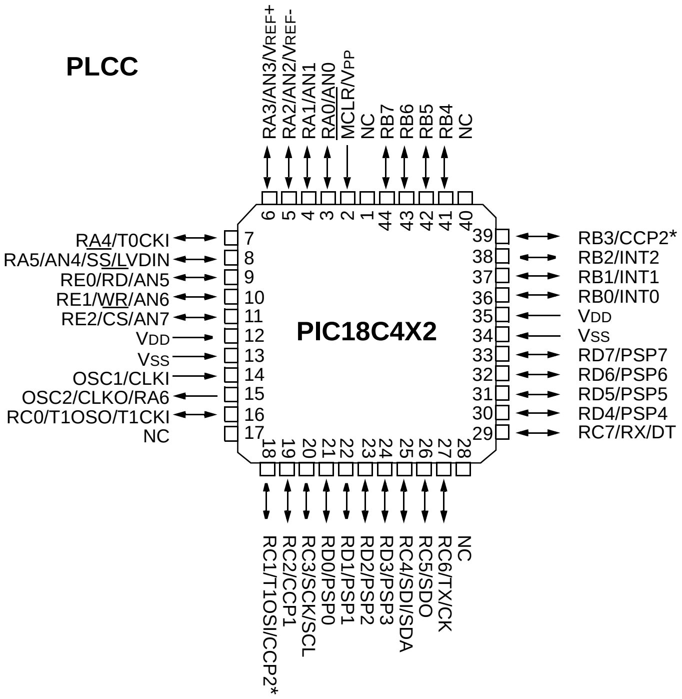
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

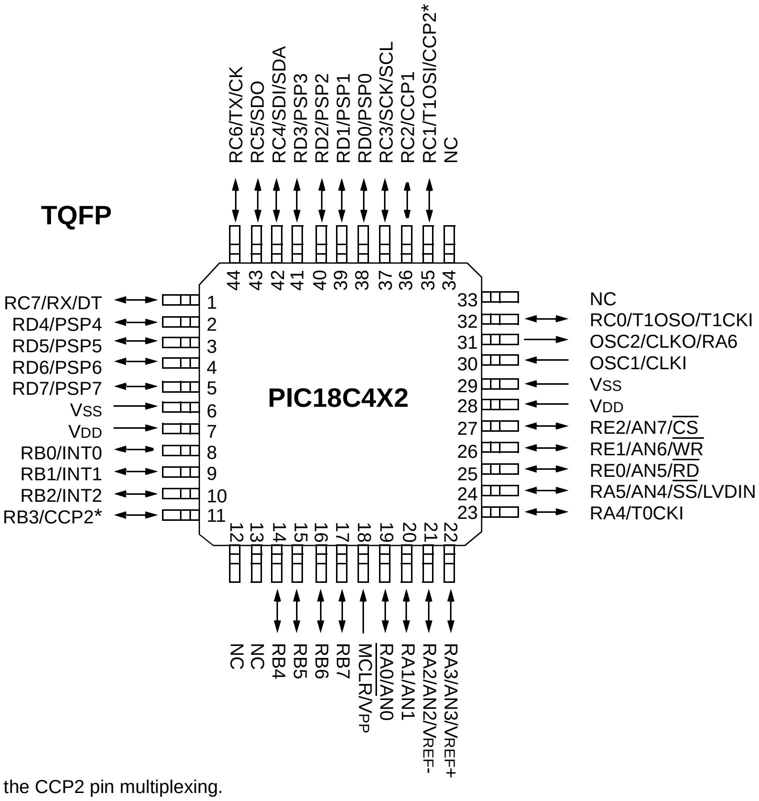
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lc452-i-p

Pin Diagrams

PLCC



TQFP



* RB3 is the alternate pin for the CCP2 pin multiplexing.

Note: Pin compatible with 44-pin PIC16C7X devices.

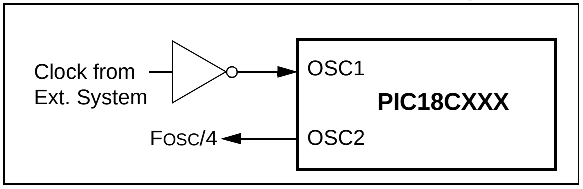
© 1999-2013 Microchip Technology Inc.

2.4 External Clock Input

The EC and ECIO oscillator modes require an external clock source to be connected to the OSC1 pin. The feedback device between OSC1 and OSC2 is turned off in these modes to save current. There is no oscillator start-up time required after a Power-on Reset or after a recovery from SLEEP mode.

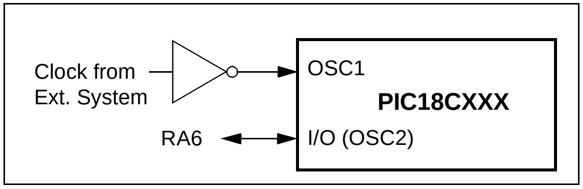
In the EC oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC oscillator mode.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



The ECIO oscillator mode functions like the EC mode, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO oscillator mode.

FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)



2.5 HS/PLL

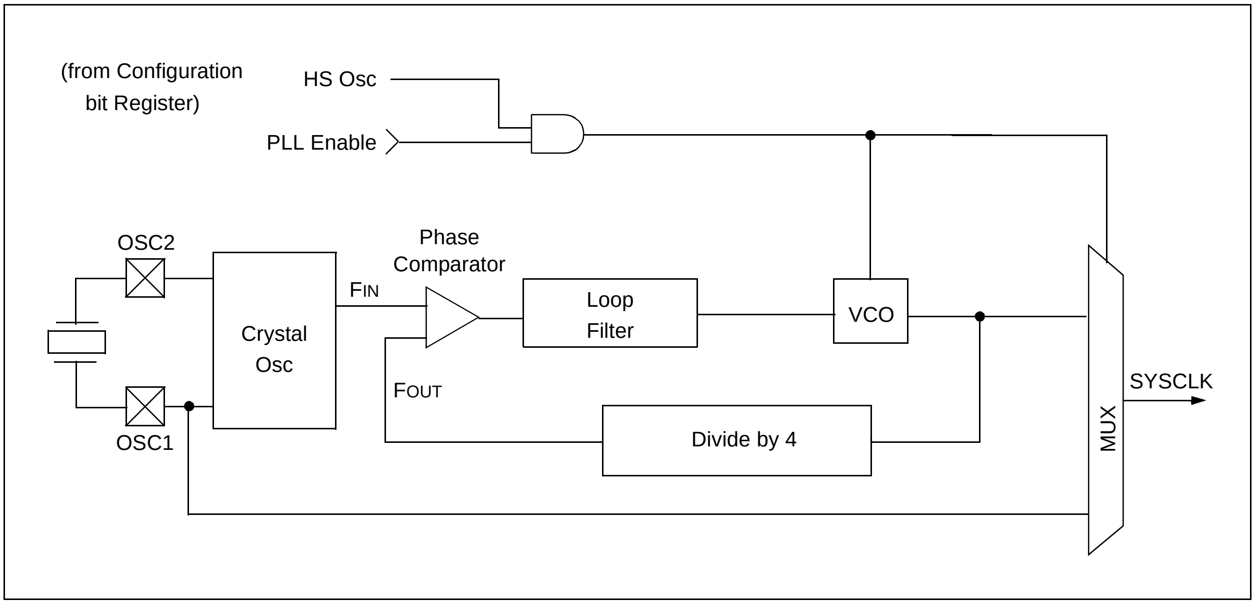
A Phase Locked Loop circuit is provided as a programmable option for users that want to multiply the frequency of the incoming crystal oscillator signal by 4. For an input clock frequency of 10 MHz, the internal clock frequency will be multiplied to 40 MHz. This is useful for customers who are concerned with EMI due to high frequency crystals.

The PLL can only be enabled when the oscillator configuration bits are programmed for HS mode. If they are programmed for any other mode, the PLL is not enabled and the system clock will come directly from OSC1.

The PLL is one of the modes of the FOSC<2:0> configuration bits. The oscillator mode is specified during device programming.

A PLL lock timer is used to ensure that the PLL has locked before device execution starts. The PLL lock timer has a time-out that is called TPLL.

FIGURE 2-6: PLL BLOCK DIAGRAM



REGISTER 4-1: STKPTR REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

bit 7[1] **STKFUL**: Stack Full Flag bit
`1` = Stack became full or overflowed
`0` = Stack has not become full or overflowed

bit 6[1] **STKUNF**: Stack Underflow Flag bit
`1` = Stack underflow occurred
`0` = Stack underflow did not occur

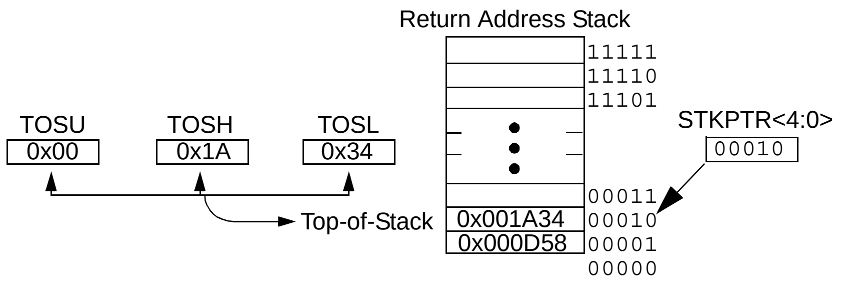
bit 5 **Unimplemented**: Read as '0'

bit 4-0 **SP4:SP0**: Stack Pointer Location bits

Note 1: Bit 7 and bit 6 can only be cleared in user software or by a POR.

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

FIGURE 4-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



4.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable option. To push the current PC value onto the stack, a `PUSH` instruction can be executed. This will increment the stack pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place a return address on the stack.

The ability to pull the TOS value off of the stack and replace it with the value that was previously pushed onto the stack, without disturbing normal execution, is achieved by using the `POP` instruction. The `POP` instruction discards the current TOS by decrementing the stack pointer. The previous value pushed onto the stack then becomes the TOS value.

4.2.4 STACK FULL/UNDERFLOW RESETS

These resets are enabled by programming the STVREN configuration bit. When the STVREN bit is disabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device RESET. When the STVREN bit is enabled, a full or underflow will set the appropriate STKFUL or STKUNF bit and then cause a device RESET. The STKFUL or STKUNF bits are only cleared by the user software or a POR Reset.

TABLE 4-2: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
WDTCON	—	—	—	—	—	—	—	SWDTE	---- ---0	183
RCON	IPEN	LWRT	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0q-1 11qq	53, 56, 74
TMR1H	Timer1 Register High Byte								xxxx xxxx	97
TMR1L	Timer1 Register Low Byte								xxxx xxxx	97
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0-00 0000	97
TMR2	Timer2 Register								0000 0000	101
PR2	Timer2 Period Register								1111 1111	102
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	101
SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	121
SSPADD	SSP Address Register in I^{2}C Slave Mode. SSP Baud Rate Reload Register in I^{2}C Master Mode.								0000 0000	128
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	116
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	118
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	120
ADRESH	A/D Result Register High Byte								xxxx xxxx	171,172
ADRESL	A/D Result Register Low Byte								xxxx xxxx	171,172
ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	165
ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	166
CCPR1H	Capture/Compare/PWM Register1 High Byte								xxxx xxxx	111, 113
CCPR1L	Capture/Compare/PWM Register1 Low Byte								xxxx xxxx	111, 113
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	107
CCPR2H	Capture/Compare/PWM Register2 High Byte								xxxx xxxx	111, 113
CCPR2L	Capture/Compare/PWM Register2 Low Byte								xxxx xxxx	111, 113
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	107
TMR3H	Timer3 Register High Byte								xxxx xxxx	103
TMR3L	Timer3 Register Low Byte								xxxx xxxx	103
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	0000 0000	103
SPBRG	USART1 Baud Rate Generator								0000 0000	151
RCREG	USART1 Receive Register								0000 0000	158, 161, 163
TXREG	USART1 Transmit Register								0000 0000	156, 159, 162
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	149
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	150

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

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO oscillator mode only, and read '0' in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

© 1999-2013 Microchip Technology Inc.

EXAMPLE 6-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
   MOVF     ARG1L, W
   MULWF    ARG2L        ; ARG1L * ARG2L ->
                         ; PRODH:PRODL
   MOVFF    PRODH, RES1 ;
   MOVFF    PRODL, RES0 ;
;
   MOVF     ARG1H, W
   MULWF    ARG2H        ; ARG1H * ARG2H ->
                         ; PRODH:PRODL
   MOVFF    PRODH, RES3 ;
   MOVFF    PRODL, RES2 ;
;
   MOVF     ARG1L, W
   MULWF    ARG2H        ; ARG1L * ARG2H ->
                         ; PRODH:PRODL
   MOVF     PRODL, W     ;
   ADDWF    RES1,  F     ; Add cross
   MOVF     PRODH, W     ; products
   ADDWFC   RES2,  F     ;
   CLRF     WREG,  F     ;
   ADDWFC   RES3,  F     ;
;
   MOVF     ARG1H, W     ;
   MULWF    ARG2L        ; ARG1H * ARG2L ->
                         ; PRODH:PRODL
   MOVF     PRODL, W     ;
   ADDWF    RES1,  F     ; Add cross
   MOVF     PRODH, W     ; products
   ADDWFC   RES2,  F     ;
   CLRF     WREG,  F     ;
   ADDWFC   RES3,  F     ;
```

Example 6-4 shows the sequence to do a 16 x 16 signed multiply. Equation 6-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pairs' Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 6-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned}
&\mathrm{RES3{:}RES0}\\
&\quad = \mathrm{ARG1H{:}ARG1L} \bullet \mathrm{ARG2H{:}ARG2L}\\
&\quad = (\mathrm{ARG1H} \bullet \mathrm{ARG2H} \bullet 2^{16}) +\\
&\qquad (\mathrm{ARG1H} \bullet \mathrm{ARG2L} \bullet 2^{8}) +\\
&\qquad (\mathrm{ARG1L} \bullet \mathrm{ARG2H} \bullet 2^{8}) +\\
&\qquad (\mathrm{ARG1L} \bullet \mathrm{ARG2L}) +\\
&\qquad (-1 \bullet \mathrm{ARG2H{<}7{>}} \bullet \mathrm{ARG1H{:}ARG1L} \bullet 2^{16}) +\\
&\qquad (-1 \bullet \mathrm{ARG1H{<}7{>}} \bullet \mathrm{ARG2H{:}ARG2L} \bullet 2^{16})
\end{aligned}$$

EXAMPLE 6-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVF     ARG1L, W
   MULWF    ARG2L        ; ARG1L * ARG2L ->
                         ; PRODH:PRODL
   MOVFF    PRODH, RES1 ;
   MOVFF    PRODL, RES0 ;
;
   MOVF     ARG1H, W
   MULWF    ARG2H        ; ARG1H * ARG2H ->
                         ; PRODH:PRODL
   MOVFF    PRODH, RES3 ;
   MOVFF    PRODL, RES2 ;
;
   MOVF     ARG1L, W
   MULWF    ARG2H        ; ARG1L * ARG2H ->
                         ; PRODH:PRODL
   MOVF     PRODL, W     ;
   ADDWF    RES1,  F     ; Add cross
   MOVF     PRODH, W     ; products
   ADDWFC   RES2,  F     ;
   CLRF     WREG,  F     ;
   ADDWFC   RES3,  F     ;
;
   MOVF     ARG1H, W     ;
   MULWF    ARG2L        ; ARG1H * ARG2L ->
                         ; PRODH:PRODL
   MOVF     PRODL, W     ;
   ADDWF    RES1,  F     ; Add cross
   MOVF     PRODH, W     ; products
   ADDWFC   RES2, F      ;
   CLRF     WREG, F      ;
   ADDWFC   RES3, F      ;
;
   BTFSS    ARG2H, 7     ; ARG2H:ARG2L neg?
   BRA      SIGN_ARG1    ; no, check ARG1
   MOVF     ARG1L, W     ;
   SUBWF    RES2         ;
   MOVF     ARG1H, W     ;
   SUBWFB   RES3
;
SIGN_ARG1
   BTFSS    ARG1H, 7     ; ARG1H:ARG1L neg?
   BRA      CONT_CODE    ; no, done
   MOVF     ARG2L, W     ;
   SUBWF    RES2         ;
   MOVF     ARG2H, W     ;
   SUBWFB   RES3
;
CONT_CODE
     :
```

© 1999-2013 Microchip Technology Inc.

REGISTER 7-3: INTCON3 REGISTER

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

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

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred
(must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred
(must be cleared in software)
0 = The INT1 external interrupt did not occur

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR reset	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

© 1999-2013 Microchip Technology Inc.

REGISTER 7-7: PERIPHERAL INTERRUPT ENABLE REGISTER 2 (PIE2)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BCLIE	LVDIE	TMR3IE	CCP2IE
bit 7							bit 0

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

bit 3 **BCLIE**: Bus Collision Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **LVDIE**: Low Voltage Detect Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR3IE**: TMR3 Overflow Interrupt Enable bit
1 = Enables the TMR3 overflow interrupt
0 = Disables the TMR3 overflow interrupt

bit 0 **CCP2IE**: CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

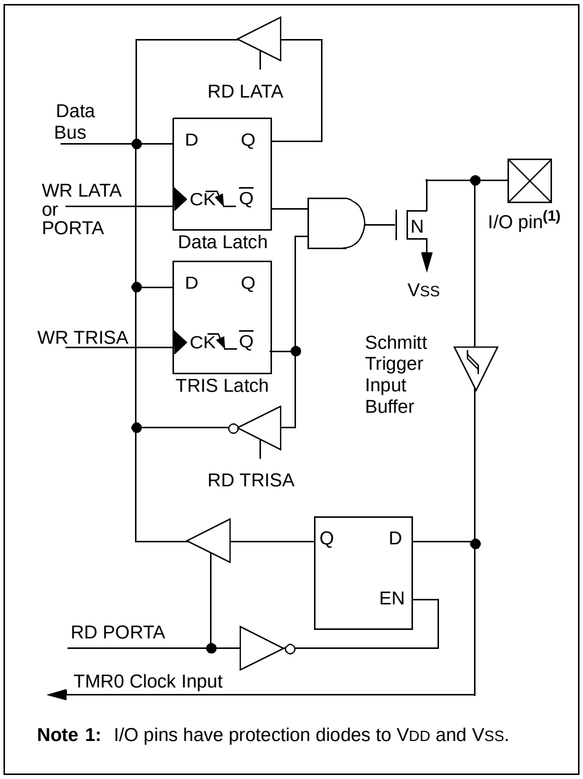
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

© 1999-2013 Microchip Technology Inc.

NOTES:

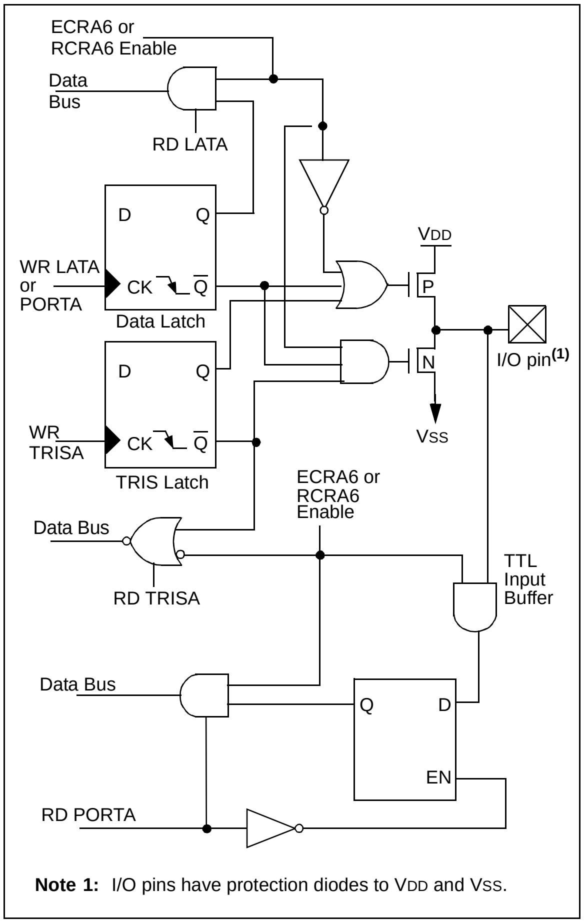
© 1999-2013 Microchip Technology Inc.

FIGURE 8-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 8-3: BLOCK DIAGRAM OF RA6



Note 1: I/O pins have protection diodes to VDD and VSS.

© 1999-2013 Microchip Technology Inc.

15.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

In order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter:

- bit SPEN (RCSTA<7>) must be set (= 1), and
- bits TRISC<7:6> must be cleared (= 0).

Register 15-1 shows the Transmit Status and Control Register (TXSTA) and Register 15-2 shows the Receive Status and Control Register (RCSTA).

REGISTER 15-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don’t care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of transmit data. Can be Address/Data bit or a parity bit.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR reset	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

© 1999-2013 Microchip Technology Inc.

15.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 15-1 shows the formula for computation of the baud rate for different USART modes, which only apply in Master mode (internal clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRG register can be calculated using the formula in Table 15-1. From this, the error in baud rate can be determined.

Example 15-1 shows the calculation of the baud rate error for the following conditions:

- FOSC = 16 MHz
- Desired Baud Rate = 9600
- BRGH = 0
- SYNC = 0

It may be advantageous to use the high baud rate (BRGH = 1), even for slower baud clocks. This is because the FOSC/(16(X + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

15.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

EXAMPLE 15-1: CALCULATING BAUD RATE ERROR

$$\begin{aligned}
\text{Desired Baud Rate} &= \text{FOSC} / \ (64\ (X + 1)) \\
&\text{Solving for X:} \\
X &= (\ (\text{FOSC} / \ \text{Desired Baud rate}) / 64\ )\ - 1 \\
X &= ((16000000 / 9600) / 64) - 1 \\
X &= [25.042] = 25 \\
\text{Calculated Baud Rate} &= 16000000 / (64\ (25 + 1)) \\
&= 9615 \\
\text{Error} &= \frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}} \\
&= (9615 - 9600) / 9600 \\
&= 0.16\%
\end{aligned}$$

TABLE 15-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0 1	(Asynchronous) Baud Rate = FOSC/(64(X+1)) (Synchronous) Baud Rate = FOSC/(4(X+1))	Baud Rate = FOSC/(16(X+1)) NA

Legend: X = value in SPBRG (0 to 255)

TABLE 15-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other RESETS
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

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

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (VDD and VSS) or the voltage level on the RA3/AN3/VREF+ pin and RA2/AN2/VREF-.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off and any conversion is aborted.

Each port pin associated with the A/D converter can be configured as an analog input (RA3 can also be a voltage reference) or as a digital I/O.

The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH/ADRESL registers, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 16-1.

FIGURE 16-1: A/D BLOCK DIAGRAM

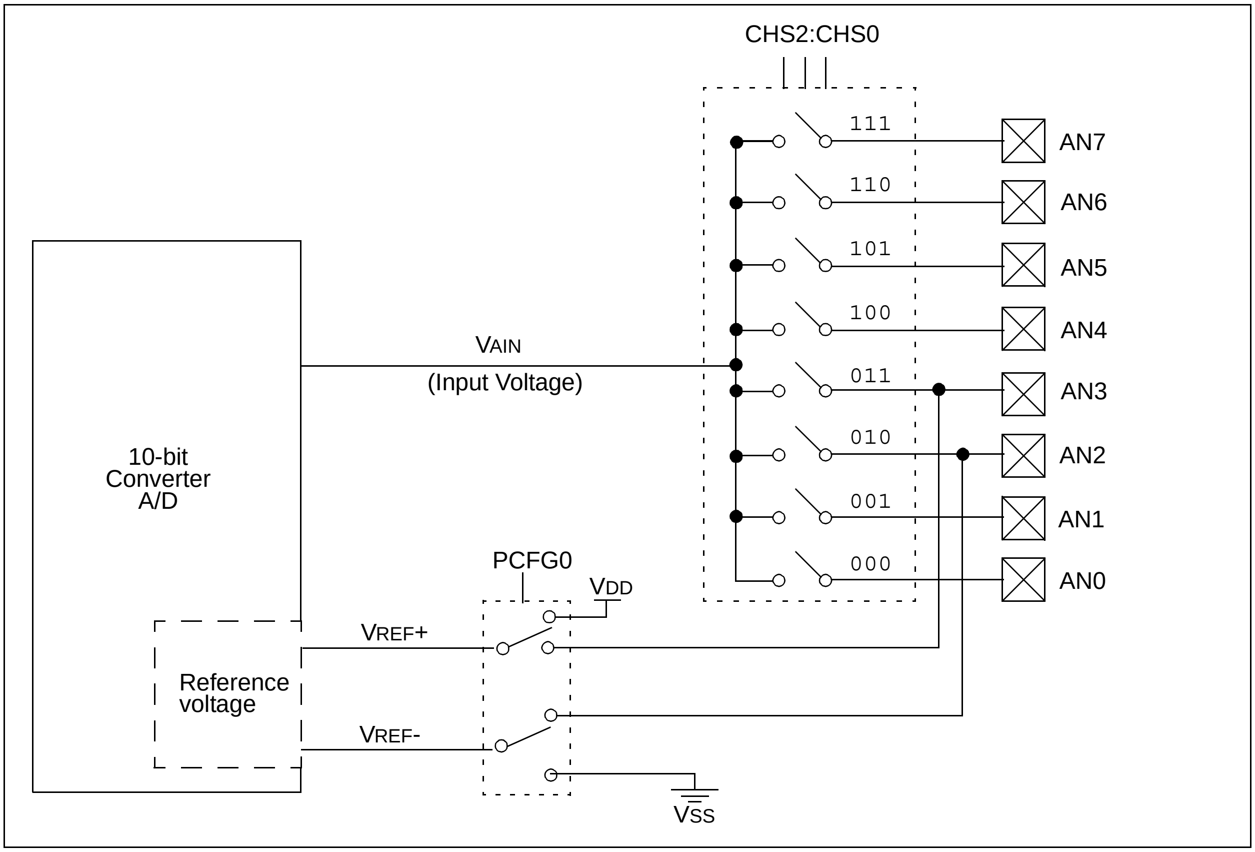


TABLE 19-2: PIC18CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-bit Instruction Word	Status Affected	Notes
				MSb LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d, a	Add WREG and f	1	`0010 01da ffff ffff`	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	`0010 00da ffff ffff`	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	`0001 01da ffff ffff`	Z, N	1,2
CLRF	f, a	Clear f	1	`0110 101a ffff ffff`	Z	2
COMF	f, d, a	Complement f	1	`0001 11da ffff ffff`	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	`0110 001a ffff ffff`	None	4
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	`0110 010a ffff ffff`	None	4
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	`0110 000a ffff ffff`	None	1, 2
DECF	f, d, a	Decrement f	1	`0000 01da ffff ffff`	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	`0010 11da ffff ffff`	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	`0100 11da ffff ffff`	None	1, 2
INCF	f, d, a	Increment f	1	`0010 10da ffff ffff`	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	`0011 11da ffff ffff`	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	`0100 10da ffff ffff`	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	`0001 00da ffff ffff`	Z, N	1, 2
MOVF	f, d, a	Move f	1	`0101 00da ffff ffff`	Z, N	1
MOVFF	f_s, f_d	Move f_s (source) to 1st word	2	`1100 ffff ffff ffff`	None	
		f_d (destination)2nd word		`1111 ffff ffff ffff`		
MOVWF	f, a	Move WREG to f	1	`0110 111a ffff ffff`	None	
MULWF	f, a	Multiply WREG with f	1	`0000 001a ffff ffff`	None	
NEGF	f, a	Negate f	1	`0110 110a ffff ffff`	C, DC, Z, OV, N	1, 2
RLCF	f, d, a	Rotate Left f through Carry	1	`0011 01da ffff ffff`	C, Z, N	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	`0100 01da ffff ffff`	Z, N	1, 2
RRCF	f, d, a	Rotate Right f through Carry	1	`0011 00da ffff ffff`	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	`0100 00da ffff ffff`	Z, N	
SETF	f, a	Set f	1	`0110 100a ffff ffff`	None	
SUBFWB	f, d, a	Subtract f from WREG with borrow	1	`0101 01da ffff ffff`	C, DC, Z, OV, N	1, 2
SUBWF	f, d, a	Subtract WREG from f	1	`0101 11da ffff ffff`	C, DC, Z, OV, N	
SUBWFB	f, d, a	Subtract WREG from f with borrow	1	`0101 10da ffff ffff`	C, DC, Z, OV, N	1, 2
SWAPF	f, d, a	Swap nibbles in f	1	`0011 10da ffff ffff`	None	4
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	`0110 011a ffff ffff`	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	`0001 10da ffff ffff`	Z, N	
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b, a	Bit Clear f	1	`1001 bbba ffff ffff`	None	1, 2
BSF	f, b, a	Bit Set f	1	`1000 bbba ffff ffff`	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	`1011 bbba ffff ffff`	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	`1010 bbba ffff ffff`	None	3, 4
BTG	f, d, a	Bit Toggle f	1	`0111 bbba ffff ffff`	None	1, 2

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

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

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

4: Some instructions are 2 word instructions. The second word of these instructions will be executed as a `NOP`, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

MOVFF Move f to f

Syntax:	[*label*] MOVFF f_s,f_d
Operands:	$0 \le f_s \le 4095$ $0 \le f_d \le 4095$
Operation:	$(f_s) \rightarrow f_d$
Status Affected:	None

Encoding:

1st word (source)	1100	ffff	ffff	$\mathtt{ffff}_s$
2nd word (destin.)	1111	ffff	ffff	$\mathtt{ffff}_d$

Description: The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096 byte data space (000h to FFFh), and location of destination 'f_d' can also be anywhere from 000h to FFFh.
Either source or destination can be WREG (a useful special situation).
`MOVFF` is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).

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

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: `MOVFF   REG1, REG2`

Before Instruction

```
REG1        =   0x33
REG2        =   0x11
```

After Instruction

```
REG1        =   0x33,
REG2        =   0x33
```

MOVLB Move literal to low nibble in BSR

Syntax:	[*label*] MOVLB k
Operands:	$0 \le k \le 255$
Operation:	k → BSR
Status Affected:	None

Encoding:

0000	0001	kkkk	kkkk

Description: The 8-bit literal 'k' is loaded into the Bank Select Register (BSR).

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVLB   5`

Before Instruction

```
BSR register    =   0x02
```

After Instruction

```
BSR register    =   0x05
```

© 1999-2013 Microchip Technology Inc.

MULLW Multiply Literal with WREG

Syntax: [*label*] MULLW k

Operands: 0 ≤ k ≤ 255

Operation: (WREG) x k → PRODH:PRODL

Status Affected: None

Encoding:

`0000`	`1101`	`kkkk`	`kkkk`

Description: An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte.
WREG is unchanged.
None of the status flags are affected.
Note that neither overflow, nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MULLW 0xC4`

Before Instruction

```
WREG      =   0xE2
PRODH     =   ?
PRODL     =   ?
```

After Instruction

```
WREG      =   0xE2
PRODH     =   0xAD
PRODL     =   0x08
```

MULWF Multiply WREG with f

Syntax: [*label*] MULWF f [,a]

Operands: 0 ≤ f ≤ 255
a ∈ [0,1]

Operation: (WREG) x (f) → PRODH:PRODL

Status Affected: None

Encoding:

`0000`	`001a`	`ffff`	`ffff`

Description: An unsigned multiplication is carried out between the contents of WREG and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte.
Both WREG and 'f' are unchanged.
None of the status flags are affected.
Note that neither overflow, nor carry is possible in this operation. A zero result is possible but not detected. If ‘a’ is 0, the Access Bank will be selected, overriding the BSR value. If ‘a’= 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write registers PRODH: PRODL

Example: `MULWF REG, 1`

Before Instruction

```
WREG      =   0xC4
REG       =   0xB5
PRODH     =   ?
PRODL     =   ?
```

After Instruction

```
WREG      =   0xC4
REG       =   0xB5
PRODH     =   0x8A
PRODL     =   0x94
```

21.2 DC Characteristics: PIC18CXX2 (Industrial, Extended) PIC18LCXX2 (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for industrial -40°C ≤ T_A ≤ +125°C for extended			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	V_{IL}	**Input Low Voltage**				
		I/O ports:				
D030		with TTL buffer	Vss	$0.15V_{DD}$	V	V_{DD} < 4.5V
D030A			—	0.8	V	4.5V ≤ V_{DD} ≤ 5.5V
D031		with Schmitt Trigger buffer RC3 and RC4	Vss Vss	$0.2V_{DD}$ $0.3V_{DD}$	V V	
D032		$\overline{MCLR}$	V_{SS}	$0.2V_{DD}$	V	
D032A		OSC1 (in XT, HS and LP modes) and T1OSI	V_{SS}	$0.3V_{DD}$	V	
D033		OSC1 (in RC and EC mode)[1]	V_{SS}	$0.2V_{DD}$	V	
	V_{IH}	**Input High Voltage**				
		I/O ports:				
D040		with TTL buffer	$0.25V_{DD}$ + 0.8V	V_{DD}	V	V_{DD} < 4.5V
D040A			2.0	V_{DD}	V	4.5V ≤ V_{DD} ≤ 5.5V
D041		with Schmitt Trigger buffer RC3 and RC4	$0.8V_{DD}$ $0.7V_{DD}$	V_{DD} V_{DD}	V V	
D042		$\overline{MCLR}$, OSC1 (EC mode)	$0.8V_{DD}$	V_{DD}	V	
D042A		OSC1 (in XT, HS and LP modes) and T1OSI	$0.7V_{DD}$	V_{DD}	V	
D043		OSC1 (RC mode)[1]	$0.9V_{DD}$	V_{DD}	V	
	I_{IL}	**Input Leakage Current[2,3]**				
D060		I/O ports	—	±1	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at hi-impedance
D061		$\overline{MCLR}$	—	±5	μA	Vss ≤ V_{PIN} ≤ V_{DD}
D063		OSC1	—	±5	μA	Vss ≤ V_{PIN} ≤ V_{DD}
	I_{PU}	**Weak Pull-up Current**				
D070	I_{PURB}	PORTB weak pull-up current	50	400	μA	V_{DD} = 5V, V_{PIN} = V_{SS}

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC MCU be driven with an external clock while in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

© 1999-2013 Microchip Technology Inc.

TABLE 21-21: A/D CONVERTER CHARACTERISTICS: PIC18CXX2 (INDUSTRIAL, EXTENDED) PIC18LCXX2 (INDUSTRIAL)

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
A01	NR	Resolution		— —	— —	10 10	bit bit	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A03	EIL	Integral linearity error		— —	— —	<±1 <±2	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A04	EDL	Differential linearity error		— —	— —	<±1 <±2	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A05	EFS	Full scale error		— —	— —	<±1 <±1	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A06	EOFF	Offset error		— —	— —	<±1 <±1	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A10	—	Monotonicity		guaranteed[3]			—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage (VREFH - VREFL)		0V	—	—	V	
A20A				3V	—	—	V	For 10-bit resolution
A21	VREFH	Reference voltage High		AVSS	—	AVDD + 0.3V	V	
A22	VREFL	Reference voltage Low		AVSS - 0.3V	—	AVDD	V	
A25	VAIN	Analog input voltage		AVSS - 0.3V	—	VREF + 0.3V	V	
A30	ZAIN	Recommended impedance of analog voltage source		—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	PIC18**C**XXX	—	180	—	µA	Average current consumption when A/D is on **(Note 1)**.
			PIC18**LC**XXX	—	90	—	µA	
A50	IREF	VREF input current **(Note 2)**		10	—	1000	µA	During VAIN acquisition. Based on differential of VHOLD to VAIN. To charge CHOLD, see Section 16.0.
				—	—	10	µA	During A/D conversion cycle.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.
VREF current is from RA2/AN2/VREF- and RA3/AN3/VREF+ pins or AVDD and AVSS pins, whichever is selected as reference input.

2: VSS ≤ VAIN ≤ VREF

3: The A/D conversion result never decreases with an increase in the Input Voltage, and has no missing codes.

FIGURE 22-9: TYPICAL AND MAXIMUM IDD vs. VDD (TIMER1 AS MAIN OSCILLATOR, 32.768 kHz, C = 47 pF)

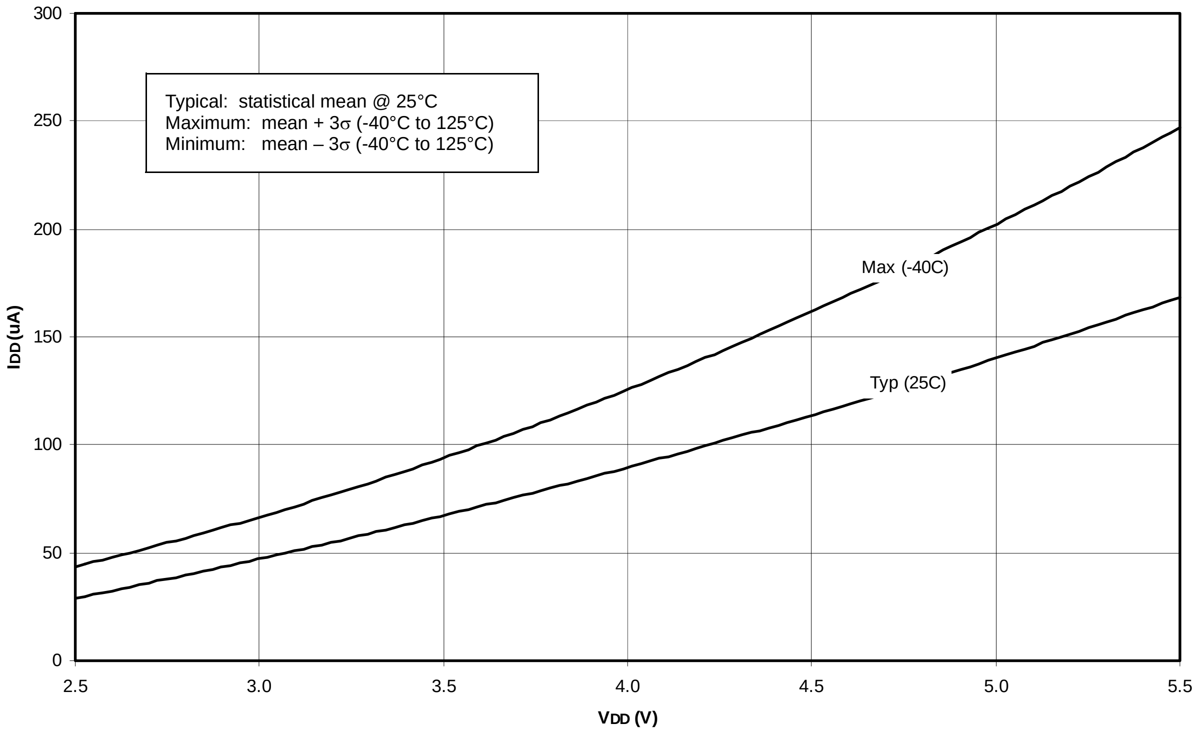
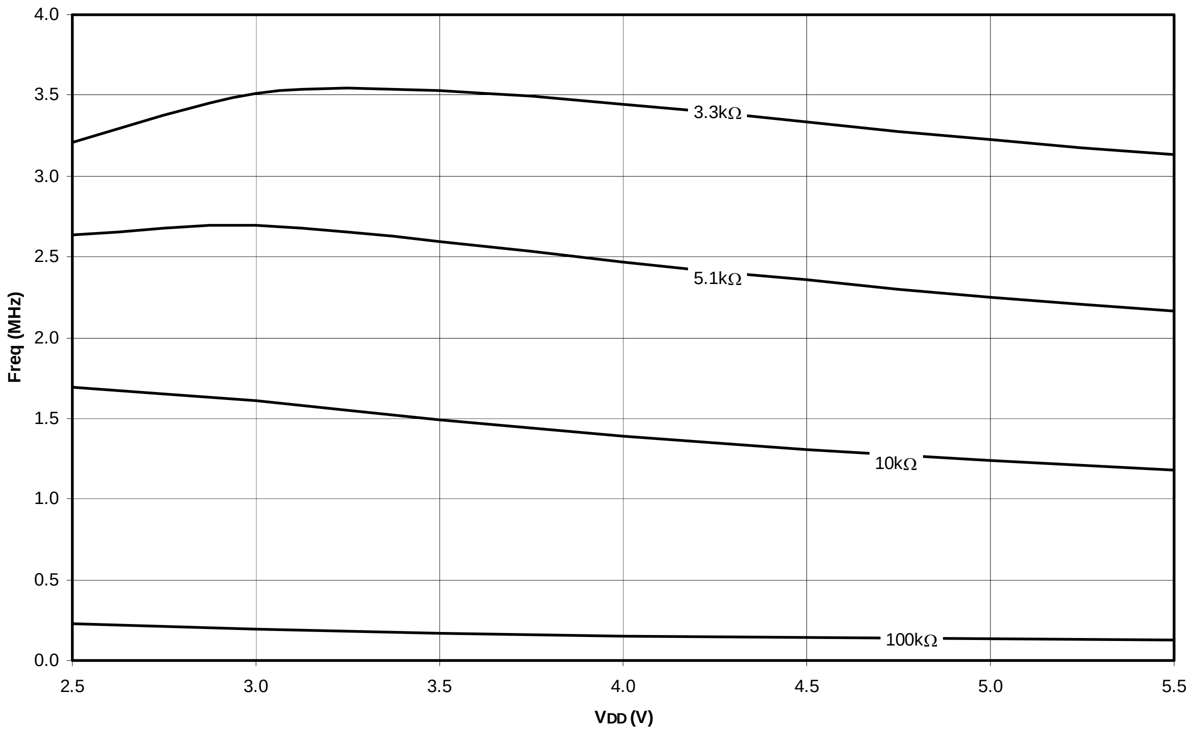


FIGURE 22-10: AVERAGE FOSC vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, 25°C)



© 1999-2013 Microchip Technology Inc.

FIGURE 22-19: ΔILVD vs. VDD OVER TEMPERATURE (LVD ENABLED, VLVD = 4.5V - 4.78V)

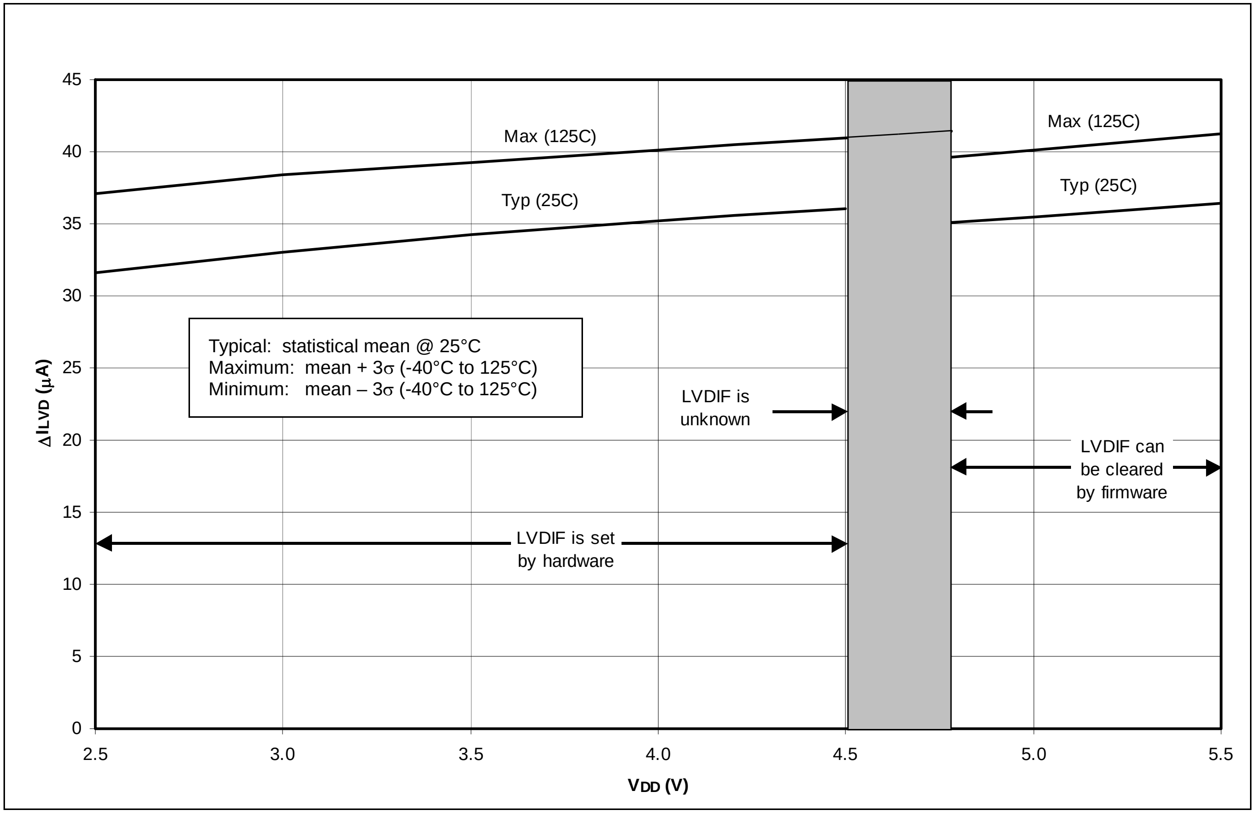
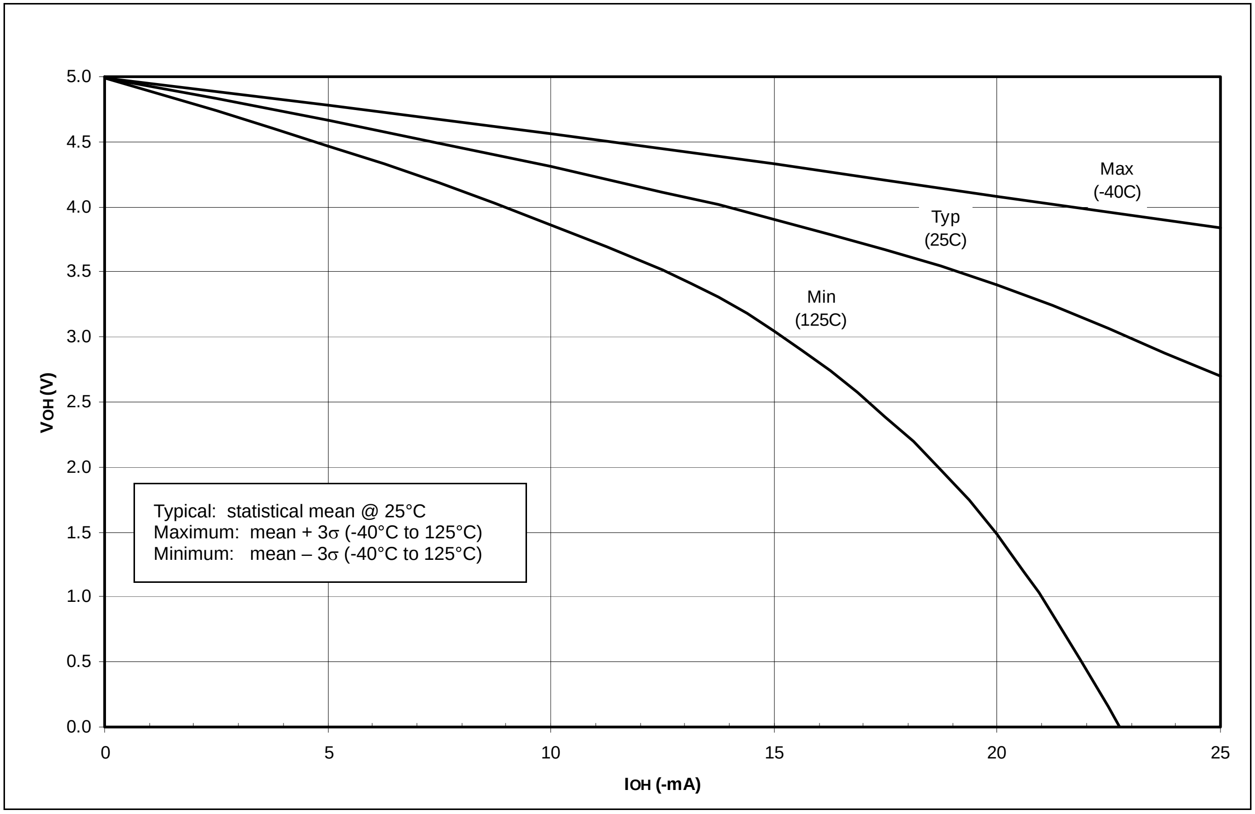


FIGURE 22-20: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO +125°C)

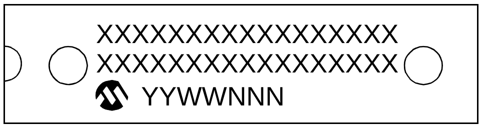


© 1999-2013 Microchip Technology Inc.

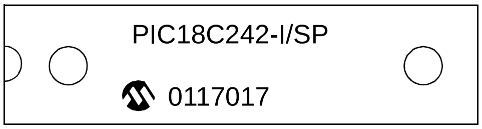
23.0 PACKAGING INFORMATION

23.1 Package Marking Information

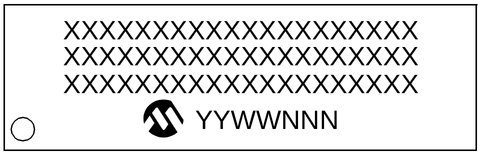
28-Lead PDIP (Skinny DIP)



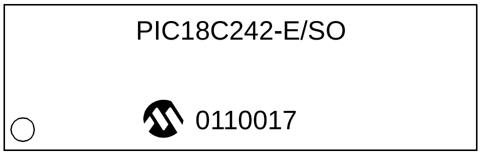
Example



28-Lead SOIC



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.