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	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256gb206t-i-pt

Pin Diagram – Top View (121-Pin BGA)[1]

	1	2	3	4	5	6	7	8	9	10	11
A	RE4	RE3	RG13	RE0	RG0	RF1	ENVREG	N/C	RD12	RD2	RD1
B	N/C	RG15	RE2	RE1	RA7	RF0	V$_{CAP}$	RD5	RD3	V$_{SS}$	RC14
C	RE6	V$_{DD}$	RG12	RG14	RA6	N/C	RD7	RD4	V$_{DD}$	RC13	RD11
D	RC1	RE7	RE5	V$_{SS}$	V$_{SS}$	N/C	RD6	RD13	RD0	n/c	RD10
E	RC4	RC3	RG6	RC2	V$_{DD}$	RG1	N/C	RA15	RD8	RD9	RA14
F	$\overline{MCLR}$	RG8	RG9	RG7	V$_{SS}$	n/c	N/C	V$_{DD}$	OSCI/RC12	V$_{SS}$	OSCO/RC15
G	RE8	RE9	RA0	N/C	V$_{DD}$	V$_{SS}$	V$_{SS}$	N/C	RA5	RA3	RA4
H	PGEC3/RB5	PGED3/RB4	V$_{SS}$	V$_{DD}$	N/C	V$_{DD}$	n/c	V$_{BUS}$/RF7	V$_{USB}$	D+/RG2	RA2
J	RB3	RB2	PGED2/RB7	AV$_{DD}$	RB11	RA1	RB12	N/C	N/C	RF8	D-/RG3
K	PGEC1/RB1	PGED1/RB0	RA10	RB8	N/C	RF12	RB14	V$_{DD}$	RD15	USBID/RF3	RF2
L	PGEC2/RB6	RA9	AV$_{SS}$	RB9	RB10	RF13	RB13	RB15	RD14	RF4	RF5

Note 1: See Table 3 for complete functional pinout descriptions.

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.
Shaded pins indicate pins tolerant to up to +5.5V.

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 8.0 "Oscillator Configuration"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

• AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"
• AN849, "Basic PICmicro® Oscillator Design"
• AN943, "Practical PICmicro® Oscillator Analysis and Design"
• AN949, "Making Your Oscillator Work"

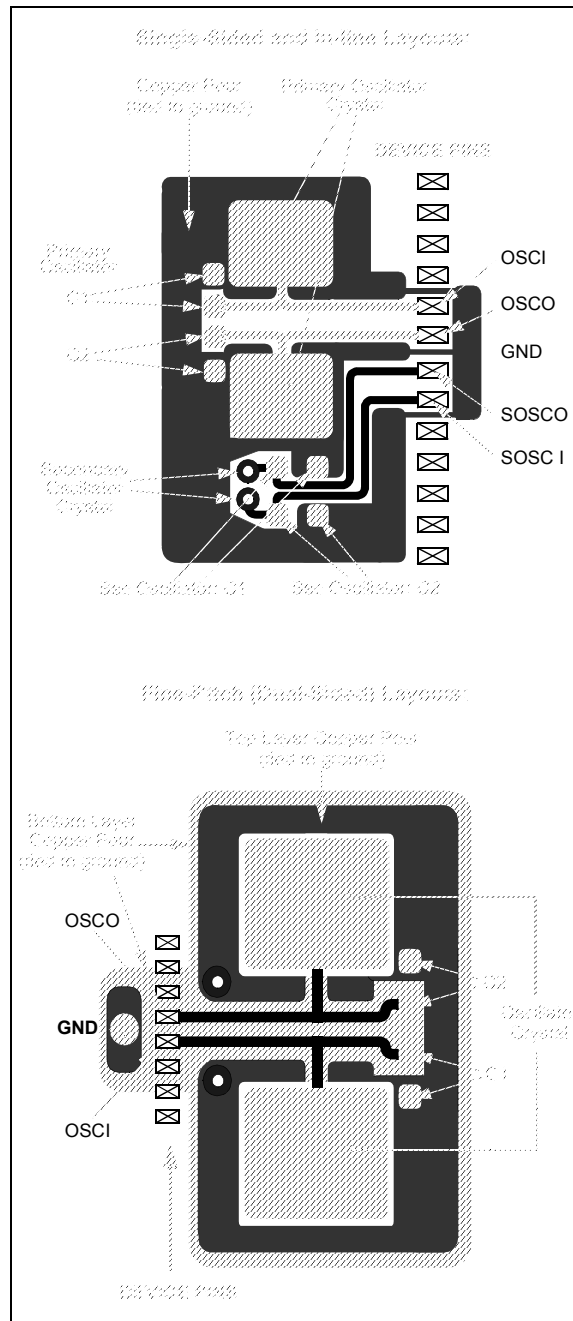**FIGURE 2-4:** **SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**

TABLE 4-8: INPUT CAPTURE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144	Input Capture 1 Buffer Register																0000
IC1TMR	0146	Input Capture 1 Timer Value Register																xxxx
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	Input Capture 2 Buffer Register																0000
IC2TMR	014E	Input Capture 2 Timer Value Register																xxxx
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	Input Capture 3 Buffer Register																0000
IC3TMR	0156	Input Capture 3 Timer Value Register																xxxx
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC4BUF	015C	Input Capture 4 Buffer Register																0000
IC4TMR	015E	Input Capture 4 Timer Value Register																xxxx
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5CON2	0162	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC5BUF	0164	Input Capture 5 Buffer Register																0000
IC5TMR	0166	Input Capture 5 Timer Value Register																xxxx
IC6CON1	0168	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC6CON2	016A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC6BUF	016C	Input Capture 6 Buffer Register																0000
IC6TMR	016E	Input Capture 6 Timer Value Register																xxxx
IC7CON1	0170	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC7CON2	0172	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC7BUF	0174	Input Capture 7 Buffer Register																0000
IC7TMR	0176	Input Capture 7 Timer Value Register																xxxx
IC8CON1	0178	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC8CON2	017A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC8BUF	017C	Input Capture 8 Buffer Register																0000
IC8TMR	017E	Input Capture 8 Timer Value Register																xxxx
IC9CON1	0180	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC9CON2	0182	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC9BUF	0184	Input Capture 9 Buffer Register																0000
IC9TMR	0186	Input Capture 9 Timer Value Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

PIC24FJ256GB210 FAMILY

PIC24FJ256GB210 FAMILY

TABLE 4-21: ADC REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	032E	—	—	—	—	CSSL27	CSSL26	CSSL25	CSSL24	CSSL23[1]	CSSL22[1]	CSSL21[1]	CSSL20[1]	CSSL19[1]	CSSL18[1]	CSSL17[1]	CSSL16[1]	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

Legend: — = unimplemented, read as '0', r = Reserved, maintain as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'

TABLE 4-22: CTMU REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase blocks of eight rows (512 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

When data is written to program memory using `TBLWT` instructions, the data is not written directly to memory. Instead, data written using table writes is stored in holding latches until the programming sequence is executed.

Any number of `TBLWT` instructions can be executed and a write will be successfully performed. However, 64 `TBLWT` instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused address should be programmed with FFFFFFh. This is because the holding latches reset to an unknown state, so if the addresses are left in the Reset state, they may overwrite the locations on rows which were not rewritten.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

> **Note:** Writing to a location multiple times without erasing is *not* recommended.

All of the table write operations are single-word writes (2 instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 JTAG Operation

The PIC24F family supports JTAG boundary scan. Boundary scan can improve the manufacturing process by verifying pin to PCB connectivity.

5.4 Enhanced In-Circuit Serial Programming

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

5.5 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.6 "Programming Operations"** for further details.

5.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

6.1 Special Function Register Reset States

Most of the Special Function Registers (SFRs) associated with the PIC24F CPU and peripherals are reset to a particular value at a device Reset. The SFRs are grouped by their peripheral or CPU function and their Reset values are specified in each section of this manual.

The Reset value for each SFR does not depend on the type of Reset, with the exception of four registers. The Reset value for the Reset Control register, RCON, will depend on the type of device Reset. The Reset value for the Oscillator Control register, OSCCON, will depend on the type of Reset and the programmed values of the FNOSC bits in Flash Configuration Word 2 (CW2) (see Table 6-2). The RCFGCAL and NVMCON registers are only affected by a POR.

6.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 6-3. Note that the system Reset signal, $\overline{SYSRST}$, is released after the POR delay time expires.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable $\overline{SYSRST}$ delay times.

The Fail-Safe Clock Monitor (FSCM) delay determines the time at which the FSCM begins to monitor the system clock source after the $\overline{SYSRST}$ signal is released.

6.3 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen, as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to **Section 8.0 "Oscillator Configuration"** for further details.

TABLE 6-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

Reset Type	Clock Source Determinant
POR	FNOSC Configuration bits (CW2<10:8>)
BOR	
$\overline{MCLR}$	COSC Control bits (OSCCON<14:12>)
WDTO	
SWR	

PIC24FJ256GB210 FAMILY

REGISTER 7-14:   IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IE[1]	INT3IE[1]	—	—	MI2C2IE	SI2C2IE	—
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 15   **Unimplemented:** Read as '0'

bit 14   **RTCIE:** Real-Time Clock/Calendar Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13-7   **Unimplemented:** Read as '0'

bit 6   **INT4IE:** External Interrupt 4 Enable bit[1]
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5   **INT3IE:** External Interrupt 3 Enable bit[1]
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4-3   **Unimplemented:** Read as '0'

bit 2   **MI2C2IE:** Master I2C2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1   **SI2C2IE:** Slave I2C2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

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

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPx or RPIx pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

10.4.3.2   Output Mapping

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains two 6-bit fields, with each field being associated with one RPn pin (see Register 10-29 through Register 10-44). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-4).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '000000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-4:   SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Output Function Number[1]	Function	Output Name
0	NULL[2]	Null
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	$\overline{\text{U1RTS}}$[3]	UART1 Request To Send
5	U2TX	UART2 Transmit
6	$\overline{\text{U2RTS}}$[3]	UART2 Request To Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
18	OC1	Output Compare 1
19	OC2	Output Compare 2
20	OC3	Output Compare 3
21	OC4	Output Compare 4
22	OC5	Output Compare 5
23	OC6	Output Compare 6
24	OC7	Output Compare 7
25	OC8	Output Compare 8
28	U3TX	UART3 Transmit
29	$\overline{\text{U3RTS}}$[3]	UART3 Request To Send
30	U4TX	UART4 Transmit
31	$\overline{\text{U4RTS}}$[3]	UART4 Request To Send
32	SDO3	SPI3 Data Output
33	SCK3OUT	SPI3 Clock Output
34	SS3OUT	SPI3 Slave Select Output
35	OC9	Output Compare 9
36	C3OUT	Comparator 3 Output
37-63	(unused)	NC

Note 1: Setting the RPORx register with the listed value assigns that output function to the associated RPn pin.

   2: The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

   3: IrDA® BCLK functionality uses this output.

REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER[1]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
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 15 **TON:** Timer1 On bit
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer1 Gated Time Accumulation Enable bit
<u>When TCS = 1:</u>
This bit is ignored.
<u>When TCS = 0:</u>
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

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

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit
<u>When TCS = 1:</u>
1 = Synchronize external clock input
0 = Do not synchronize external clock input
<u>When TCS = 0:</u>
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit
1 = External clock from T1CK pin (on the rising edge)
0 = Internal clock (F$_{OSC}$/2)

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

Note 1: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

REGISTER 16-1:   I2CxCON: I2Cx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

Legend:		HC = Hardware Clearable 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 15   **I2CEN:** I2Cx Enable bit

  1 = Enables the I2Cx module and configures the SDAx and SCLx pins as serial port pins
  0 = Disables the I2Cx module; all I^{2}C™ pins are controlled by port functions

bit 14   **Unimplemented:** Read as '0'

bit 13   **I2CSIDL:** Stop in Idle Mode bit

  1 = Discontinues module operation when device enters an Idle mode
  0 = Continues module operation in Idle mode

bit 12   **SCLREL:** SCLx Release Control bit (when operating as I^{2}C slave)

  1 = Releases SCLx clock
  0 = Holds SCLx clock low (clock stretch)
  If STREN = 1:
  Bit is R/W (i.e., software may write '0' to initiate stretch and write '1' to release clock). Hardware is clear
  at the beginning of slave transmission. Hardware is clear at the end of slave reception.
  If STREN = 0:
  Bit is R/S (i.e., software may only write '1' to release clock). Hardware is clear at the beginning of slave
  transmission.

bit 11   **IPMIEN:** Intelligent Platform Management Interface (IPMI) Enable bit

  1 = IPMI Support mode is enabled; all addresses are Acknowledged
  0 = IPMI mode is disabled

bit 10   **A10M:** 10-Bit Slave Addressing bit

  1 = I2CxADD is a 10-bit slave address
  0 = I2CxADD is a 7-bit slave address

bit 9   **DISSLW:** Disable Slew Rate Control bit

  1 = Slew rate control is disabled
  0 = Slew rate control is enabled

bit 8   **SMEN:** SMBus Input Levels bit

  1 = Enables I/O pin thresholds compliant with SMBus specifications
  0 = Disables the SMBus input thresholds

bit 7   **GCEN:** General Call Enable bit (when operating as I^{2}C slave)

  1 = Enables interrupt when a general call address is received in the I2CxRSR (module is enabled for
      reception)
  0 = General call address disabled

bit 6   **STREN:** SCLx Clock Stretch Enable bit (when operating as I^{2}C slave)

  Used in conjunction with the SCLREL bit.
  1 = Enables software or receive clock stretching
  0 = Disables software or receive clock stretching

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF
bit 7							bit 0

Legend:		C = Clearable bit	HS = Hardware Settable 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	
HSC = Hardware Settable/Clearable bit				

bit 15 **ACKSTAT:** Acknowledge Status bit

1 = NACK was detected last
0 = ACK was detected last
Hardware is set or clear at the end of Acknowledge.

bit 14 **TRSTAT:** Transmit Status bit
(When operating as I^{2}C™ master. Applicable to master transmit operation.)

1 = Master transmit is in progress (8 bits + ACK)
0 = Master transmit is not in progress
Hardware is set at the beginning of master transmission; hardware is clear at the end of slave Acknowledge.

bit 13-11 **Unimplemented:** Read as '0'

bit 10 **BCL:** Master Bus Collision Detect bit

1 = A bus collision has been detected during a master operation
0 = No collision
Hardware is set at the detection of a bus collision.

bit 9 **GCSTAT:** General Call Status bit

1 = General call address was received
0 = General call address was not received
Hardware is set when the address matches the general call address; hardware is clear at Stop detection.

bit 8 **ADD10:** 10-Bit Address Status bit

1 = 10-bit address was matched
0 = 10-bit address was not matched
Hardware is set at the match of the 2nd byte of the matched 10-bit address; hardware is clear at Stop detection.

bit 7 **IWCOL:** Write Collision Detect bit

1 = An attempt to write to the I2CxTRN register failed because the I^{2}C module is busy
0 = No collision
Hardware is set at an occurrence of write to I2CxTRN while busy (cleared by software).

bit 6 **I2COV:** Receive Overflow Flag bit

1 = A byte was received while the I2CxRCV register is still holding the previous byte
0 = No overflow
Hardware is set at an attempt to transfer I2CxRSR to I2CxRCV (cleared by software).

bit 5 **D/$\overline{A}$:** Data/Address bit (when operating as I^{2}C slave)

1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was a device address
Hardware is clear at the device address match. Hardware is set after a transmission finishes or by reception of a slave byte.

17.2 Transmitting in 8-Bit Data Mode

1. Set up the UART:
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt two cycles after being set).
4. Write a data byte to the lower byte of UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = 0 and then the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bit, UTXISELx.

17.3 Transmitting in 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bit, UTXISELx.

17.4 Break and Sync Transmit Sequence

The following sequence will send a message frame header, made up of a Break, followed by an auto-baud sync byte.

1. Configure the UART for the desired mode.
2. Set UTXEN and UTXBRK to set up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG; this loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

17.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. A receive interrupt will be generated when one or more data characters have been received as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

17.6 Operation of $\overline{UxCTS}$ and $\overline{UxRTS}$ Control Pins

UARTx Clear to Send ($\overline{UxCTS}$) and Request to Send ($\overline{UxRTS}$) are the two hardware controlled pins that are associated with the UART module. These two pins allow the UART to operate in Simplex and Flow Control mode. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

17.7 Infrared Support

The UART module provides two types of infrared UART support: one is the IrDA clock output to support an external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder. Note that because the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '0'.

17.7.1 IrDA CLOCK OUTPUT FOR EXTERNAL IrDA SUPPORT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the $\overline{UxRTS}$ pin) can be configured to generate the 16x baud clock. With UEN<1:0> = 11, the BCLKx pin will output the 16x baud clock if the UART module is enabled. It can be used to support the IrDA codec chip.

17.7.2 BUILT-IN IrDA ENCODER AND DECODER

The UART has full implementation of the IrDA encoder and decoder as part of the UART module. The built-in IrDA encoder and decoder functionality is enabled using the IREN bit (UxMODE<12>). When enabled (IREN = 1), the receive pin (UxRX) acts as the input from the infrared receiver. The transmit pin (UxTX) acts as the output to the infrared transmitter.

REGISTER 18-1: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER PROTOTYPE, USB MODE (BD0STAT THROUGH BD63STAT)

R/W-x	R/W-x	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
UOWN	DTS	PID3	PID2	PID1	PID0	BC9	BC8
bit 15							bit 8

R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
bit 7							bit 0

Legend:		HSC = Hardware Settable/Clearable 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 15 **UOWN:** USB Own bit

1 = The USB module owns the BD and its corresponding buffer; the CPU must not modify the BD or the buffer

bit 14 **DTS:** Data Toggle Packet bit

1 = Data 1 packet
0 = Data 0 packet

bit 13-10 **PID<3:0>:** Packet Identifier bits (written by the USB module)

<u>In Device mode:</u>
Represents the PID of the received token during the last transfer.
<u>In Host mode:</u>
Represents the last returned PID or the transfer status indicator.

bit 9-0 **BC<9:0>:** Byte Count bits

This represents the number of bytes to be transmitted or the maximum number of bytes to be received during a transfer. Upon completion, the byte count is updated by the USB module with the actual number of bytes transmitted or received.

REGISTER 18-8:   U1CON: USB CONTROL REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-x, HSC	R-x, HSC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
JSTATE	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN
bit 7							bit 0

Legend:			
	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8   **Unimplemented:** Read as '0'

bit 7   **JSTATE:** Live Differential Receiver J-State Flag bit

1 = J-state (differential '0' in low speed, differential '1' in full speed) is detected on the USB
0 = No J-state is detected

bit 6   **SE0:** Live Single-Ended Zero Flag bit

1 = Single-ended zero is active on the USB bus
0 = No single-ended zero is detected

bit 5   **TOKBUSY:** Token Busy Status bit

1 = Token is being executed by the USB module in On-The-Go state
0 = No token is being executed

bit 4   **USBRST:** Module Reset bit

1 = USB Reset has been generated for software Reset; application must set this bit for 50 ms, then clear it
0 = USB Reset is terminated

bit 3   **HOSTEN:** Host Mode Enable bit

1 = USB host capability is enabled; pull-downs on D+ and D- are activated in hardware
0 = USB host capability is disabled

bit 2   **RESUME:** Resume Signaling Enable bit

1 = Resume signaling is activated; software must set bit for 10 ms and then clear to enable remote wake-up
0 = Resume signaling is disabled

bit 1   **PPBRST:** Ping-Pong Buffers Reset bit

1 = Reset all Ping-Pong Buffer Pointers to the even BD banks
0 = Ping-Pong Buffer Pointers are not reset

bit 0   **SOFEN:** Start-of-Frame Enable bit

1 = Start-of-Frame token is sent every one 1 ms
0 = Start-of-Frame token is disabled

REGISTER 18-19: U1EIR: USB ERROR INTERRUPT STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
						EOFEF	
bit 7							bit 0

Legend:		U = Unimplemented bit, read as '0'	
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **BTSEF:** Bit Stuff Error Flag bit

1 = Bit stuff error has been detected
0 = No bit stuff error has been detected

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

bit 5 **DMAEF:** DMA Error Flag bit

1 = A USB DMA error condition is detected; the data size indicated by the BD byte count field is less than the number of received bytes, the received data is truncated
0 = No DMA error

bit 4 **BTOEF:** Bus Turnaround Time-out Error Flag bit

1 = Bus turnaround time-out has occurred
0 = No bus turnaround time-out

bit 3 **DFN8EF:** Data Field Size Error Flag bit

1 = Data field was not an integral number of bytes
0 = Data field was an integral number of bytes

bit 2 **CRC16EF:** CRC16 Failure Flag bit

1 = CRC16 failed
0 = CRC16 passed

bit 1 For Device mode:
CRC5EF: CRC5 Host Error Flag bit

1 = Token packet is rejected due to CRC5 error
0 = Token packet is accepted (no CRC5 error)

For Host mode:
EOFEF: End-of-Frame Error Flag bit

1 = End-of-Frame error has occurred
0 = End-of-Frame interrupt is disabled

bit 0 **PIDEF:** PID Check Failure Flag bit

1 = PID check failed
0 = PID check passed

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

19.0 ENHANCED PARALLEL MASTER PORT (EPMP)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730). The information in this data sheet supersedes the information in the FRM.

The Enhanced Parallel Master Port (EPMP) module provides a parallel 4-bit (Master mode only), 8-bit (Master and Slave modes) or 16-bit (Master mode only) data bus interface to communicate with off-chip modules, such as memories, FIFOs, LCD controllers and other microcontrollers. This module can serve as either the master or the slave on the communication bus. For EPMP Master modes, all external addresses are mapped into the internal Extended Data Space (EDS). This is done by allocating a region of the EDS for each chip select, and then assigning each chip select to a particular external resource, such as a memory or external controller. This region should not be assigned to another device resource, such as RAM or SFRs. To perform a write or read on an external resource, the CPU should simply perform a write or read within the address range assigned for EPMP.

Key features of the EPMP module are:

- Extended Data Space (EDS) Interface allows Direct Access from the CPU
- Up to 23 Programmable Address Lines
- Up to 2 Chip Select Lines
- Up to 2 Acknowledgement Lines (one per chip select)
- 4-Bit, 8-Bit or 16-Bit Wide Data Bus
- Programmable Strobe Options (per chip select)
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Programmable Address/Data Multiplexing
- Programmable Address Wait States
- Programmable Data Wait States (per chip select)
- Programmable Polarity on Control Signals (per chip select)
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support
  - Address Support
  - 4-Byte Deep Auto-Incrementing Buffer

19.1 $\overline{\text{ALTPMP}}$ Setting

Many of the lower order EPMP address pins are shared with ADC inputs. This is an untenable situation for users that need both the ADC channels and the EPMP bus. If the user does not need to use all the address bits, then by clearing the $\overline{\text{ALTPMP}}$ (CW3<12>) Configuration bit, the lower order address bits can be mapped to higher address pins, which frees the ADC channels.

Note:	The alternate PMP pin selection is not available in 64-pin devices (PIC24FJXXXGB206) and so the Configuration bit, ALTPMP, is also not available.

TABLE 19-1: ALTERNATE EPMP PINS[1]

Pin	$\overline{\text{ALTPMP}}$ = 0	$\overline{\text{ALTPMP}}$ = 1
RA14	PMCS2	PMA22
RC4	PMA22	PMCS2
RF12	PMA5	PMA18
RG6	PMA18	PMA5
RG7	PMA20	PMA4
RA3	PMA4	PMA20
RG8	PMA21	PMA3
RA4	PMA3	PMA21

Note 1: The alternate EPMP pins are valid only for 100-pin devices (PIC24FJXXXGB210).

REGISTER 21-2: CRCCON2: CRC CONTROL 2 REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
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 15-13   **Unimplemented:** Read as '0'

bit 12-8   **DWIDTH<4:0>:** Data Word Width Configuration bits
Configures the width of the data word (data word width – 1).

bit 7-5   **Unimplemented:** Read as '0'

bit 4-0   **PLEN<4:0>:** Polynomial Length Configuration bits
Configures the length of the polynomial (polynomial length – 1).

REGISTER 21-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
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 15-1   **X<15:1>:** XOR of Polynomial Term x^n Enable bits

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

23.0 TRIPLE COMPARATOR MODULE

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the associated "*PIC24F Family Reference Manual*".

The triple comparator module provides three dual input comparators. The inputs to the comparator can be configured to use any one of five external analog inputs (CxINA, CxINB, CxINC, CxIND and VREF+) and a voltage reference input from one of the internal band gap references or the comparator voltage reference generator (VBG, VBG/2, VBG/6 and CVREF).

The comparator outputs may be directly connected to the CxOUT pins. When the respective COE equals '1', the I/O pad logic makes the unsynchronized output of the comparator available on the pin.

A simplified block diagram of the module in shown in Figure 23-1. Diagrams of the possible individual comparator configurations are shown in Figure 23-2.

Each comparator has its own control register, CMxCON (Register 23-1), for enabling and configuring its operation. The output and event status of all three comparators is provided in the CMSTAT register (Register 23-2).

FIGURE 23-1: TRIPLE COMPARATOR MODULE BLOCK DIAGRAM

Note 1: Refer Register 24-1 for bit details.
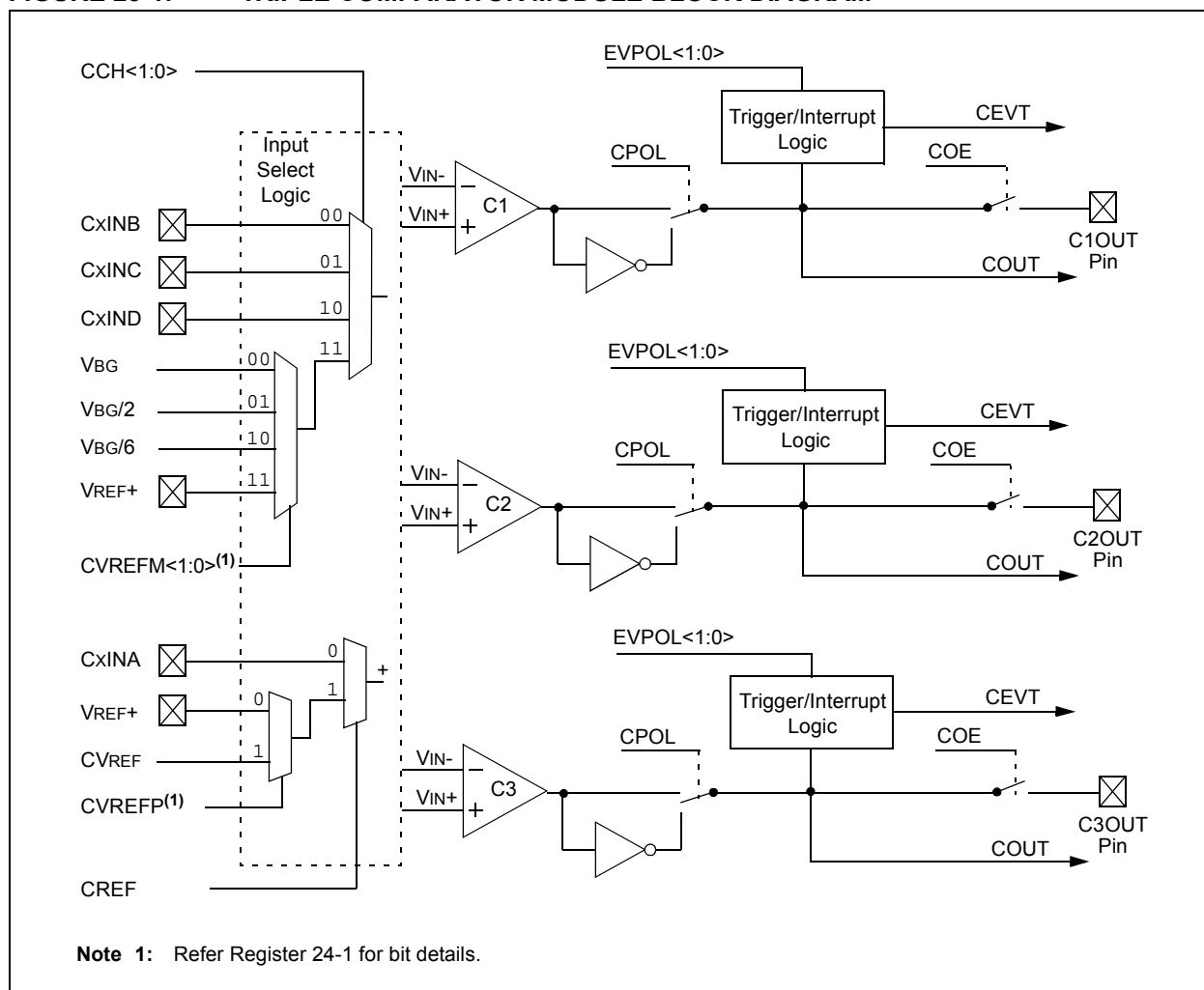
FIGURE 29-4: CLKO AND I/O TIMING CHARACTERISTICS

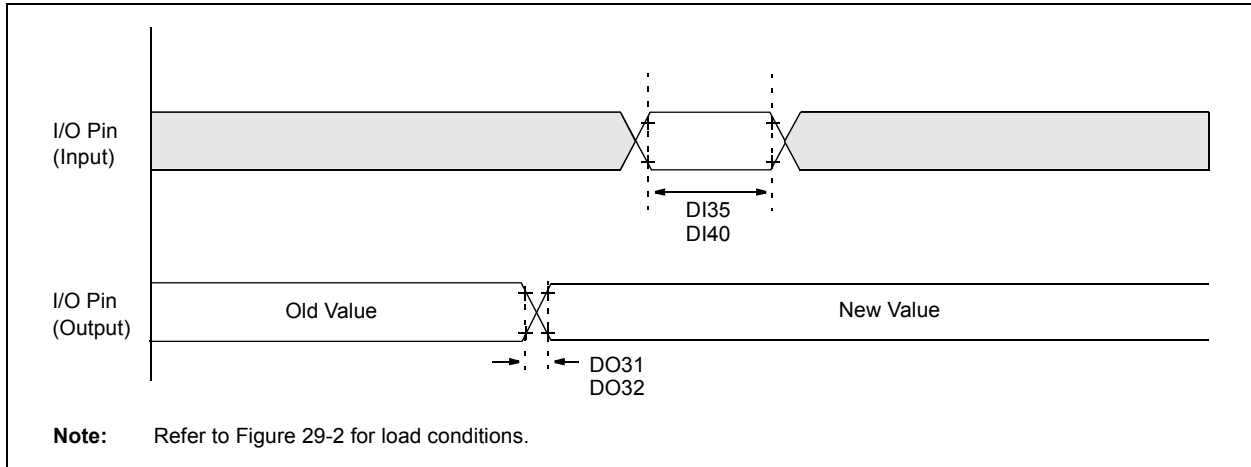Note: Refer to Figure 29-2 for load conditions.

TABLE 29-18: CLKO AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
DO31	T_{IOR}	Port Output Rise Time	—	10	25	ns	
DO32	T_{IOF}	Port Output Fall Time	—	10	25	ns	
DI35	T_{INP}	INTx Pin High or Low Time (input)	20	—	—	ns	
DI40	T_{RBP}	CNx High or Low Time (input)	2	—	—	T_{CY}	

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

NOTES: