Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

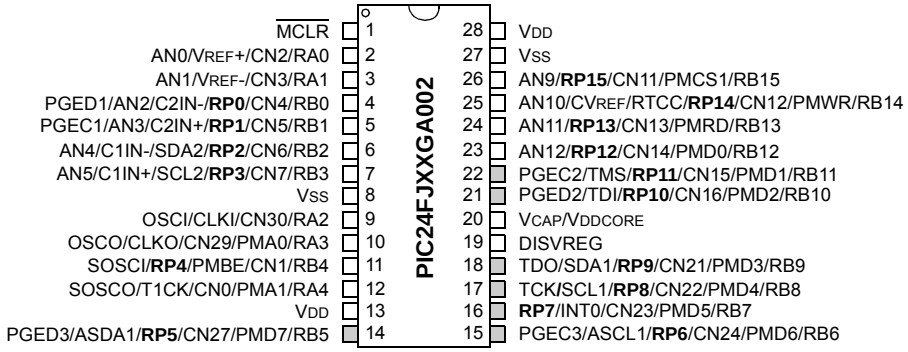
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

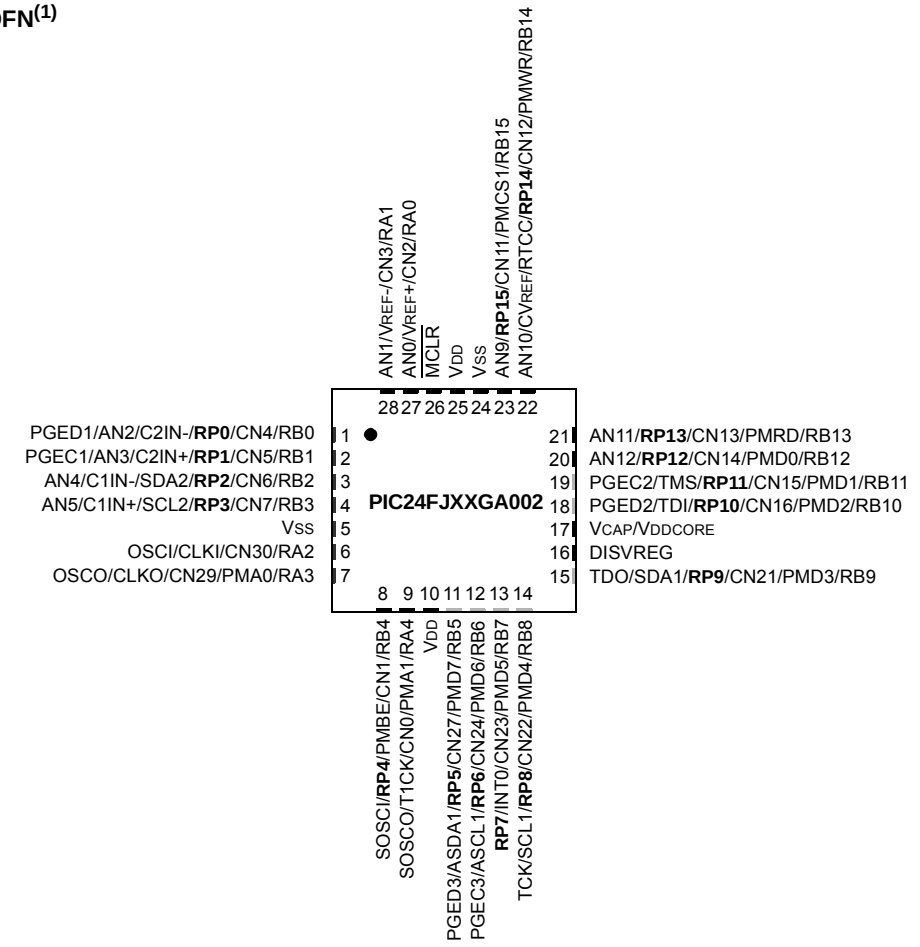
Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	21
Program Memory Size	48KB (16K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj48ga002t-i-ss

Pin Diagrams

28-Pin SPDIP, SSOP, SOIC



28-Pin QFN[(1)]



Legend: **RPn** represents remappable peripheral pins. Gray shading indicates 5.5V tolerant input pins.

Note 1: Back pad on QFN devices should be connected to Vss.

TABLE 1-2: PIC24FJ64GA004 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	28-Pin SPDIP/ SSOP/SOIC	28-Pin QFN	44-Pin QFN/TQFP			
OSCI	9	6	30	I	ANA	Main Oscillator Input Connection.
OSCO	10	7	31	O	ANA	Main Oscillator Output Connection.
PGEC1	5	2	22	I/O	ST	In-Circuit Debugger/Emulator and ICSP™ Programming Clock.
PGEC2	22	19	9	I/O	ST	
PGEC3	14	12	42	I/O	ST	
PGED1	4	1	21	I/O	ST	In-Circuit Debugger/Emulator and ICSP Programming Data.
PGED2	21	18	8	I/O	ST	
PGED3	15	11	41	I/O	ST	
PMA0	10	7	3	I/O	ST/TTL	Parallel Master Port Address Bit 0 Input (Buffered Slave modes) and Output (Master modes).
PMA1	12	9	2	I/O	ST/TTL	Parallel Master Port Address Bit 1 Input (Buffered Slave modes) and Output (Master modes).
PMA2	—	—	27	O	—	Parallel Master Port Address (Demultiplexed Master modes).
PMA3	—	—	38	O	—	
PMA4	—	—	37	O	—	
PMA5	—	—	4	O	—	
PMA6	—	—	5	O	—	
PMA7	—	—	13	O	—	
PMA8	—	—	32	O	—	
PMA9	—	—	35	O	—	
PMA10	—	—	12	O	—	
PMA11	—	—	—	O	—	
PMA12	—	—	—	O	—	
PMA13	—	—	—	O	—	
PMBE	11	8	36	O	—	Parallel Master Port Byte Enable Strobe.
PMCS1	26	23	15	O	—	Parallel Master Port Chip Select 1 Strobe/Address Bit 14.
PMD0	23	20	10	I/O	ST/TTL	Parallel Master Port Data (Demultiplexed Master mode) or Address/Data (Multiplexed Master modes).
PMD1	22	19	9	I/O	ST/TTL	
PMD2	21	18	8	I/O	ST/TTL	
PMD3	18	15	1	I/O	ST/TTL	
PMD4	17	14	44	I/O	ST/TTL	
PMD5	16	13	43	I/O	ST/TTL	
PMD6	15	12	42	I/O	ST/TTL	
PMD7	14	11	41	I/O	ST/TTL	
PMRD	24	21	11	O	—	Parallel Master Port Read Strobe.
PMWR	25	22	14	O	—	Parallel Master Port Write Strobe.

Legend: TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output I^2C™ = I^2C/SMBus input buffer

Note 1: Alternative multiplexing when the I2C1SEL Configuration bit is cleared.

2.7 Configuration of Analog and Digital Pins During ICSP Operations

If an ICSP compliant emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as “digital” pins. Depending on the particular device, this is done by setting all bits in the ADnPCFG register(s), or clearing all bit in the ANSx registers.

All PIC24F devices will have either one or more ADnPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to **Section 21.0 “10-Bit High-Speed A/D Converter”** for more specific information.

The bits in these registers that correspond to the A/D pins that initialized the emulator must not be changed by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the A/D module, as follows:

- For devices with an ADnPCFG register, clear the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.
- For devices with ANSx registers, set the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.

When a Microchip debugger/emulator is used as a programmer, the user application firmware must correctly configure the ADnPCFG or ANSx registers. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic ‘0’, which may affect user application functionality.

2.8 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to V_{SS} on unused pins and drive the output to logic low.

TABLE 4-12: PORTA 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
TRISA	02C0	—	—	—	—	—	TRISA10(1)	TRISA9(1)	TRISA8(1)	TRISA7(1)	—	—	TRISA4	TRISA3(2)	TRISA2(3)	TRISA1	TRISA0	079F
PORTA	02C2	—	—	—	—	—	RA10(1)	RA9(1)	RA8(1)	RA7(1)	—	—	RA4	RA3(2)	RA2(3)	RA1	RA0	0000
LATA	02C4	—	—	—	—	—	LATA10(1)	LATA9(1)	LATA8(1)	LATA7(1)	—	—	LATA4	LATA3(2)	LATA2(3)	LATA1	LATA0	0000
ODCA	02C6	—	—	—	—	—	ODA10(1)	ODA9(1)	ODA8(1)	ODA7(1)	—	—	ODA4	ODA3(2)	ODA2(3)	ODA1	ODA0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: These bits are not available on 28-pin devices; read as '0'.
2: These bits are only available when the primary oscillator is disabled (POSCMD<1:0> = 00); otherwise, read as '0'.
3: These bits are only available when the primary oscillator is disabled or EC mode is selected (POSCMD<1:0> = 00 or 11) and CLKO is disabled (OSCIOFNC = 0); otherwise, read as '0'.

TABLE 4-13: PORTB 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
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

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

TABLE 4-14: PORTC 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
TRISC(1)	02D0	—	—	—	—	—	—	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	03FF
PORTC(1)	02D2	—	—	—	—	—	—	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	0000
LATC(1)	02D4	—	—	—	—	—	—	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	0000
ODCC(1)	02D6	—	—	—	—	—	—	ODC9	OSC8	ODC7	ODC6	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.Bits are not available on 28-pin devices; read as '0'.

TABLE 4-15: PAD CONFIGURATION 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
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL	0000

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

5.0 FLASH PROGRAM MEMORY

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"*, **"Program Memory"** (DS39715).

The PIC24FJ64GA004 family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 2.25V.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24FJ64GA004 family device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (which are named PGCx and PGDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

RTSP is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 64 instructions (192 bytes) at a time and erase program memory in blocks of 512 instructions (1536 bytes) at a time.

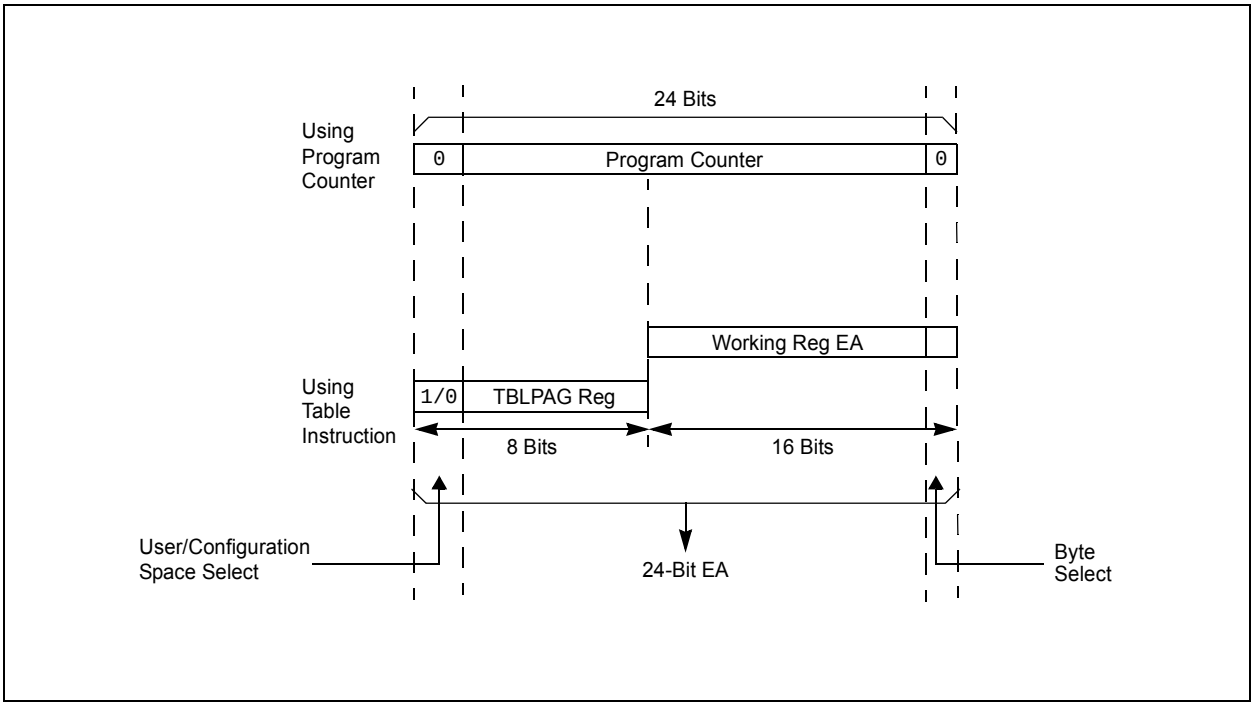
5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the table read and table write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 5-1.

The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.5.2 PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

If a Flash location has been erased, it can be programmed using table write instructions to write an instruction word (24-bit) into the write latch. The TBLPAG register is loaded with the 8 Most Significant Bytes of the Flash address. The TBLWTL and TBLWTH instructions write the desired data into the write latches and specify the lower 16 bits of the program memory address to write to. To configure the NVMCON register for a word write, set the NVMOPx bits (NVMCON<3:0>) to '0011'. The write is performed by executing the unlock sequence and setting the WR bit (see Example 5-4).

EXAMPLE 5-4: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

```
; Setup a pointer to data Program Memory
    MOV     #tblpage(PROG_ADDR), W0       ;
    MOV     W0, TBLPAG                    ;Initialize PM Page Boundary SFR
    MOV     #tbloffset(PROG_ADDR), W0     ;Initialize a register with program memory address

    MOV     #LOW_WORD_N, W2               ;
    MOV     #HIGH_BYTE_N, W3              ;
    TBLWTL  W2, [W0]                      ; Write PM low word into program latch
    TBLWTH  W3, [W0++]                    ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
    MOV     #0x4003, W0                   ;
    MOV     W0, NVMCON                    ; Set NVMOP bits to 0011

    DISI    #5                            ; Disable interrupts while the KEY sequence is written
    MOV     #0x55, W0                     ; Write the key sequence
    MOV     W0, NVMKEY
    MOV     #0xAA, W0
    MOV     W0, NVMKEY
    BSET    NVMCON, #WR                   ; Start the write cycle
    NOP                                   ; 2 NOPs required after setting WR
    NOP                                   ;
```

7.3 Interrupt Control and Status Registers

The PIC24FJ64GA004 family of devices implements a total of 29 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS4
- IEC0 through IEC4
- IPC0 through IPC12, IPC15, IPC16 and IPC18
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the Alternate Interrupt Vector Table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit which is set by the respective peripherals, or external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the Interrupt Priority Level (IPL) for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The INTTREG register contains the associated interrupt vector number and the new CPU Interrupt Priority Level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new Interrupt Priority Level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence that they are listed in Table 7-2. For example, the INT0 (External Interrupt 0) is shown as having a vector number and a natural order priority of 0. Thus, the INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS Register (SR) contains the IPL<2:0> bits (SR<7:5>). These indicate the current CPU Interrupt Priority Level. The user may change the current CPU priority level by writing to the IPLx bits.

The CORCON register contains the IPL3 bit, which together with IPL<2:0>, also indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

All Interrupt registers are described in Register 7-1 through Register 7-31, in the following pages.

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
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 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Uses Alternate Interrupt Vector Table
0 = Uses standard (default) Interrupt Vector Table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-3 **Unimplemented:** Read as ‘0’

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

U-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	U-0
—	—	PMPIF	—	—	—	OC5IF	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
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 15-14 **Unimplemented:** Read as ‘0’

bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12-10 **Unimplemented:** Read as ‘0’

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **Unimplemented:** Read as ‘0’

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-2 **Unimplemented:** Read as ‘0’

bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SPF2IF:** SPI2 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-8: IFS3: INTERRUPT FLAG STATUS REGISTER 3

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

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	MI2C2IF	SI2C2IF	—
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 **RTCIF:** Real-Time Clock/Calendar Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-3 **Unimplemented:** Read as '0'

bit 2 **MI2C2IF:** Master I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **SI2C2IF:** Slave I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 7-23: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
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 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **SPI2IP<2:0>:** SPI2 Event Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SPF2IP<2:0>:** SPI2 Fault Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

10.0 I/O PORTS

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"*, **"I/O Ports with Peripheral Pin Select (PPS)"** (DS39711).

All of the device pins (except VDD, VSS, MCLR and OSCI/CLKI) are shared between the peripherals and the Parallel I/O (PIO) ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

10.1 Parallel I/O (PIO) Ports

A Parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 10-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

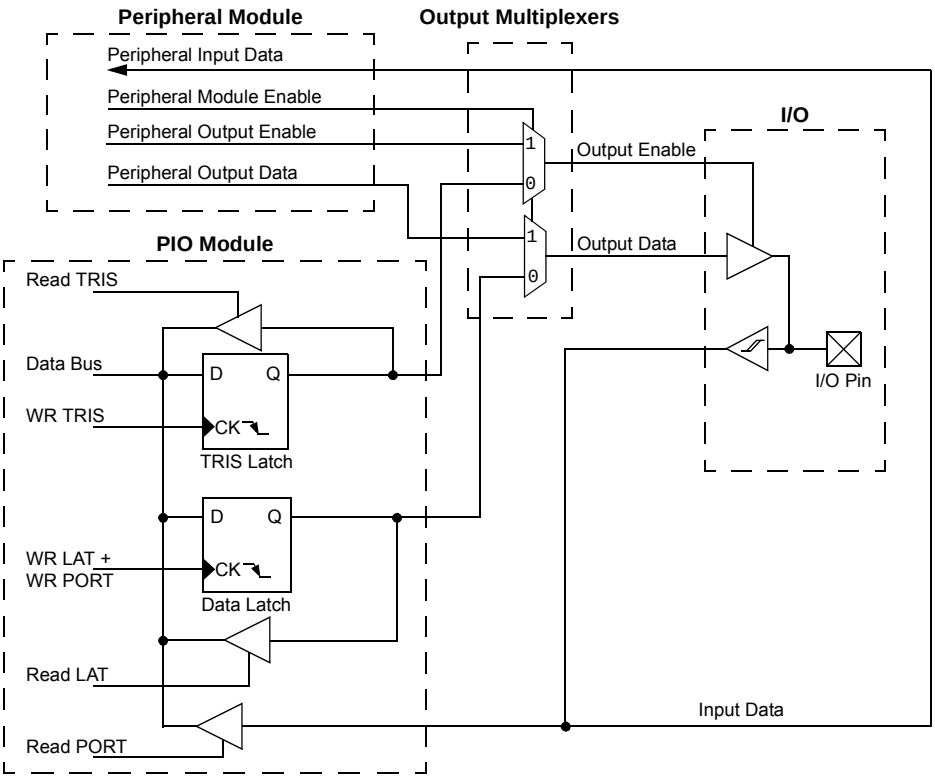
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Output Latch register (LATx), read the latch. Writes to the latch, write the latch. Reads from the port (PORTx), read the port pins, while writes to the port pins, write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless, regarded as a dedicated port because there is no other competing source of outputs.

FIGURE 10-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



10.4.4.3 Configuration Bit Pin Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY (CW2<4>) Configuration bit blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure will not execute and the Peripheral Pin Select Control registers cannot be written to. The only way to clear the bit and re-enable peripheral remapping is to perform a device Reset.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access (with the proper use of the unlock sequence) to the Peripheral Pin Select registers.

10.4.5 CONSIDERATIONS FOR PERIPHERAL PIN SELECTION

The ability to control Peripheral Pin Selection introduces several considerations into application design that could be overlooked. This is particularly true for several common peripherals that are available only as remappable peripherals.

The main consideration is that the Peripheral Pin Selects are not available on default pins in the device's default (Reset) state. Since all RPINRx registers reset to '11111' and all RPORx registers reset to '00000', all Peripheral Pin Select inputs are tied to RP31 and all Peripheral Pin Select outputs are disconnected.

Note:	In tying Peripheral Pin Select inputs to RP31, RP31 does not have to exist on a device for the registers to be reset to it.

This situation requires the user to initialize the device with the proper peripheral configuration before any other application code is executed. Since the IOLOCK bit resets in the unlocked state, it is not necessary to execute the unlock sequence after the device has come out of Reset. For application safety, however, it is best to set IOLOCK and lock the configuration after writing to the control registers.

Because the unlock sequence is timing critical, it must be executed as an assembly language routine in the same manner as changes to the oscillator configuration. If the bulk of the application is written in C or another high-level language, the unlock sequence should be performed by writing in-line assembly.

Choosing the configuration requires the review of all Peripheral Pin Selects and their pin assignments, especially those that will not be used in the application. In all cases, unused pin-selectable peripherals should be disabled completely. Unused peripherals should have their inputs assigned to an unused RPn pin function. I/O pins with unused RPn functions should be configured with the null peripheral output.

The assignment of a peripheral to a particular pin does not automatically perform any other configuration of the pin's I/O circuitry. In theory, this means adding a pin-selectable output to a pin may mean inadvertently driving an existing peripheral input when the output is driven. Users must be familiar with the behavior of other fixed peripherals that share a remappable pin and know when to enable or disable them. To be safe, fixed digital peripherals that share the same pin should be disabled when not in use.

Along these lines, configuring a remappable pin for a specific peripheral does not automatically turn that feature on. The peripheral must be specifically configured for operation and enabled, as if it were tied to a fixed pin. Where this happens in the application code (immediately following device Reset and peripheral configuration or inside the main application routine) depends on the peripheral and its use in the application.

A final consideration is that Peripheral Pin Select functions neither override analog inputs, nor reconfigure pins with analog functions for digital I/O. If a pin is configured as an analog input on device Reset, it must be explicitly reconfigured as a digital I/O when used with a Peripheral Pin Select.

Example 10-2 shows a configuration for bidirectional communication with flow control using UART1. The following input and output functions are used:

- Input Functions: U1RX, $\overline{\text{U1CTS}}$
- Output Functions: U1TX, $\overline{\text{U1RTS}}$

EXAMPLE 10-2: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS

```
// Unlock Registers
__builtin_write_OSCCONL(OSCCON & 0xBF);

// Configure Input Functions (Table 10-2))
// Assign U1RX To Pin RP0
RPINR18bits.U1RXR = 0;
// Assign U1CTS To Pin RP1
RPINR18bits.U1CTSR = 1;
// Configure Output Functions (Table 10-3)
// Assign U1TX To Pin RP2
RPOR1bits.RP2R = 3;
// Assign U1RTS To Pin RP3
RPOR1bits.RP3R = 4;

// Lock Registers
__builtin_write_OSCCONL(OSCCON | 0x40);
```

NOTES:

FIGURE 15-5: SPIx MASTER, FRAME MASTER CONNECTION DIAGRAM

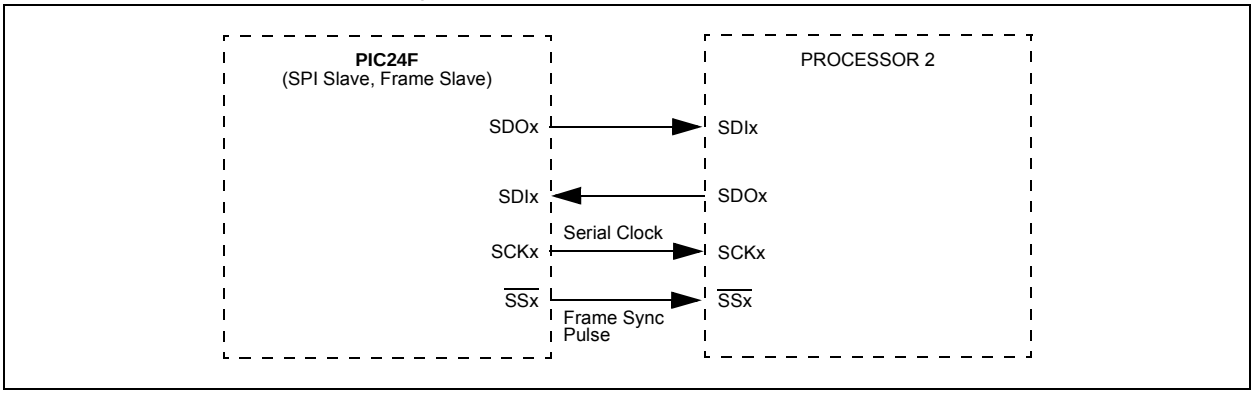


FIGURE 15-6: SPIx MASTER, FRAME SLAVE CONNECTION DIAGRAM

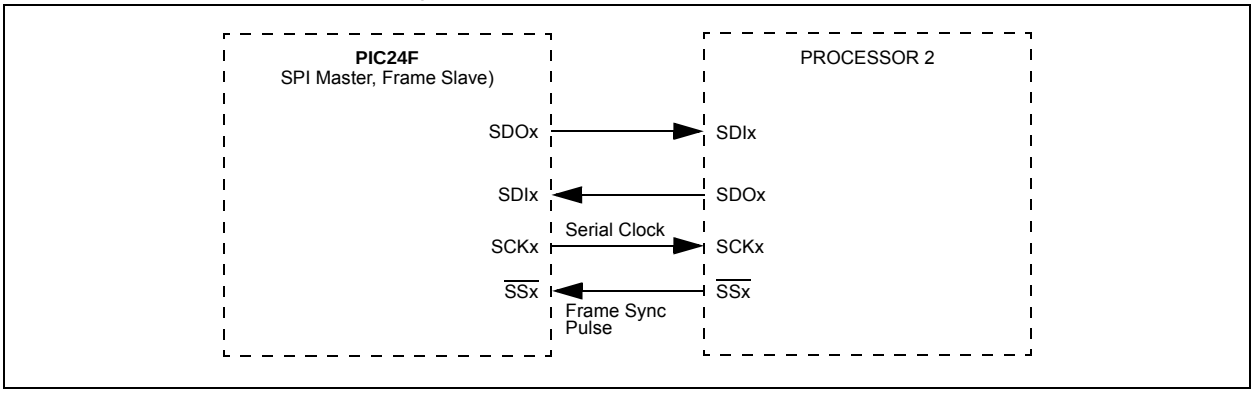


FIGURE 15-7: SPIx SLAVE, FRAME MASTER CONNECTION DIAGRAM

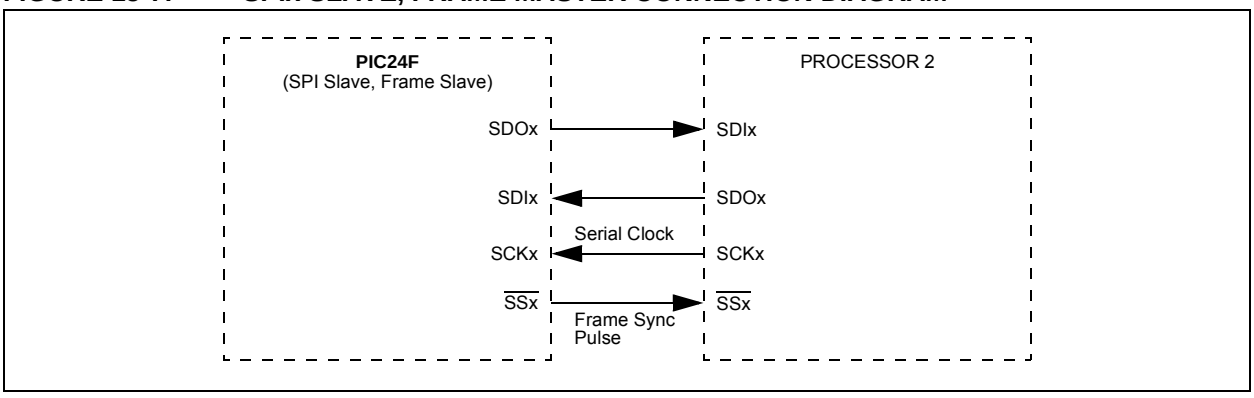
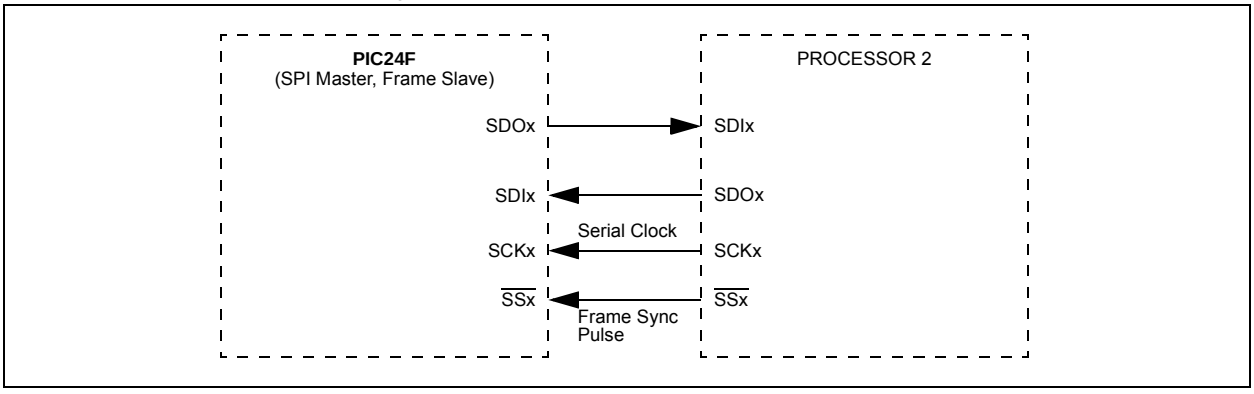


FIGURE 15-8: SPIx SLAVE, FRAME SLAVE CONNECTION DIAGRAM



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(1)	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)
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^2C™ 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 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 an 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 matched 10-bit address. Hardware is clear at Stop detection.

bit 7 **IWCOL:** I2Cx Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I^2C module is busy
0 = No collision
Hardware is set at the occurrence of a write to I2CxTRN while busy (cleared by software).

bit 6 **I2COV:** I2Cx 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).

Note 1: In both Master and Slave modes, the ACKSTAT bit is only updated when transmitting data resulting in the reception of an ACK or NACK from another device. Do not check the state of ACKSTAT when receiving data, either as a slave or a master. Reading ACKSTAT after receiving address or data bytes returns an invalid result.

19.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

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"*, **"Real-Time Clock and Calendar (RTCC)"** (DS39696).

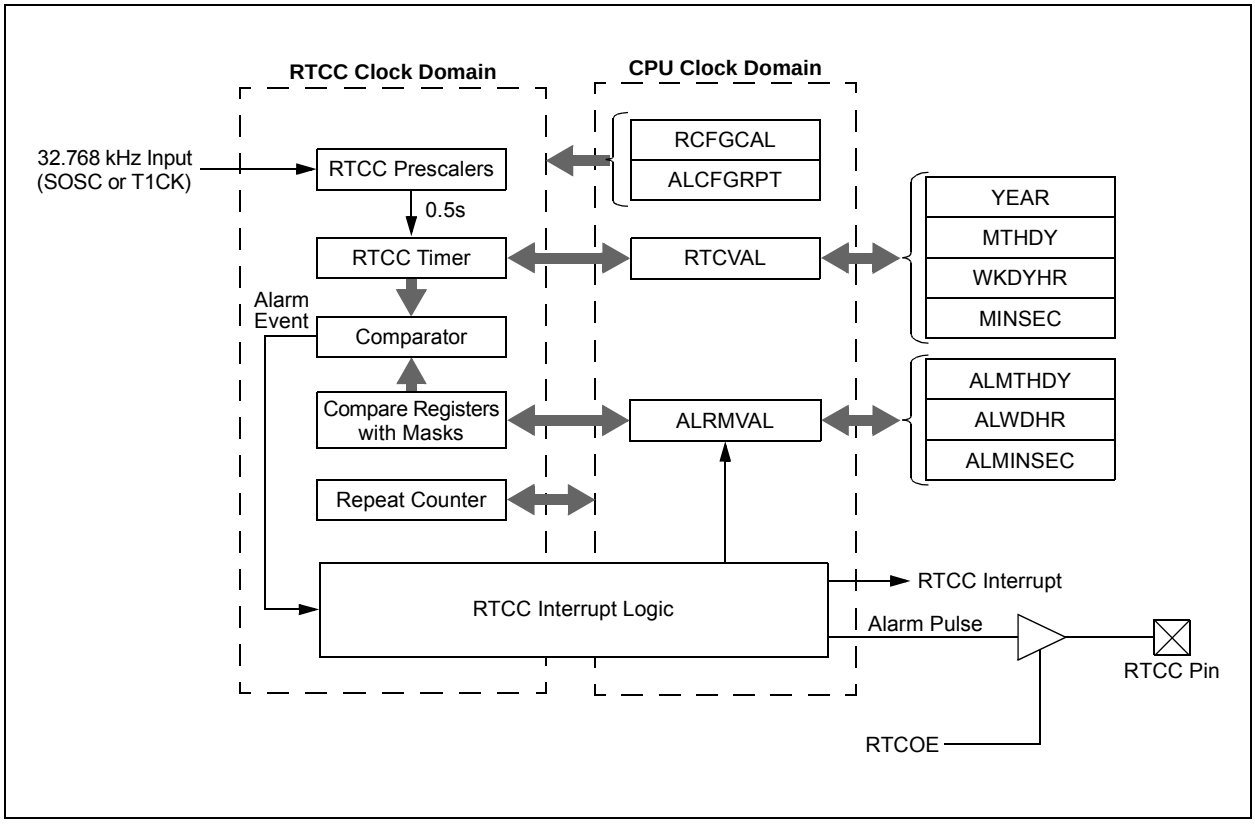
The Real-Time Clock and Calendar (RTCC) provides on-chip, hardware-based clock and calendar functionality with little or no CPU overhead. It is intended for applications where accurate time must be maintained for extended periods, with minimal CPU activity and with limited power resources, such as battery-powered applications.

Key features include:

- Time data in hours, minutes and seconds, with a granularity of one-half second
- 24-hour format (military time) display option
- Calendar data as date, month and year
- Automatic, hardware-based day of week and leap year calculations for dates from 2000 through 2099
- Time and calendar data in BCD format for compact firmware
- Highly configurable alarm function
- External output pin with selectable alarm signal or seconds "tick" signal output
- Time base input from Secondary Oscillator (SOSC) or the T1CK digital clock input (32.768 kHz)
- User calibration feature with auto-adjust

A simplified block diagram of the module is shown in Figure 19-1.The SOSC and RTCC will both remain running while the device is held in Reset with $\overline{\text{MCLR}}$, and will continue running after $\overline{\text{MCLR}}$ is released.

FIGURE 19-1: RTCC BLOCK DIAGRAM



REGISTER 20-2: CRCXOR: CRC XOR POLYNOMIAL REGISTER

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’

REGISTER 24-4: DEVREV: DEVICE REVISION REGISTER

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23							bit 16

U	U	U	U	U	U	U	R
—	—	—	—	—	—	—	MAJRV2
bit 15							bit 8

R	R	U	U	U	R	R	R
MAJRV1	MAJRV0	—	—	—	DOT2	DOT1	DOT0
bit 7							bit 0

Legend: R = Read-only bit	U = Unimplemented bit

bit 23-9 **Unimplemented:** Read as '0'

bit 8-6 **MAJRV<2:0>:** Major Revision Identifier bits

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **DOT<2:0>:** Minor Revision Identifier bits

NOTES: