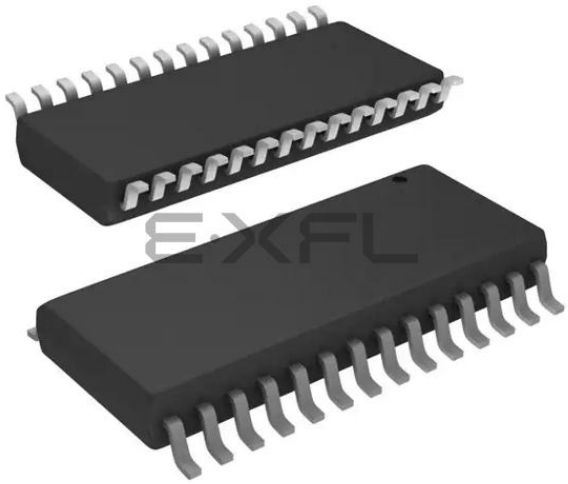


Welcome to E-XFL.COM

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.

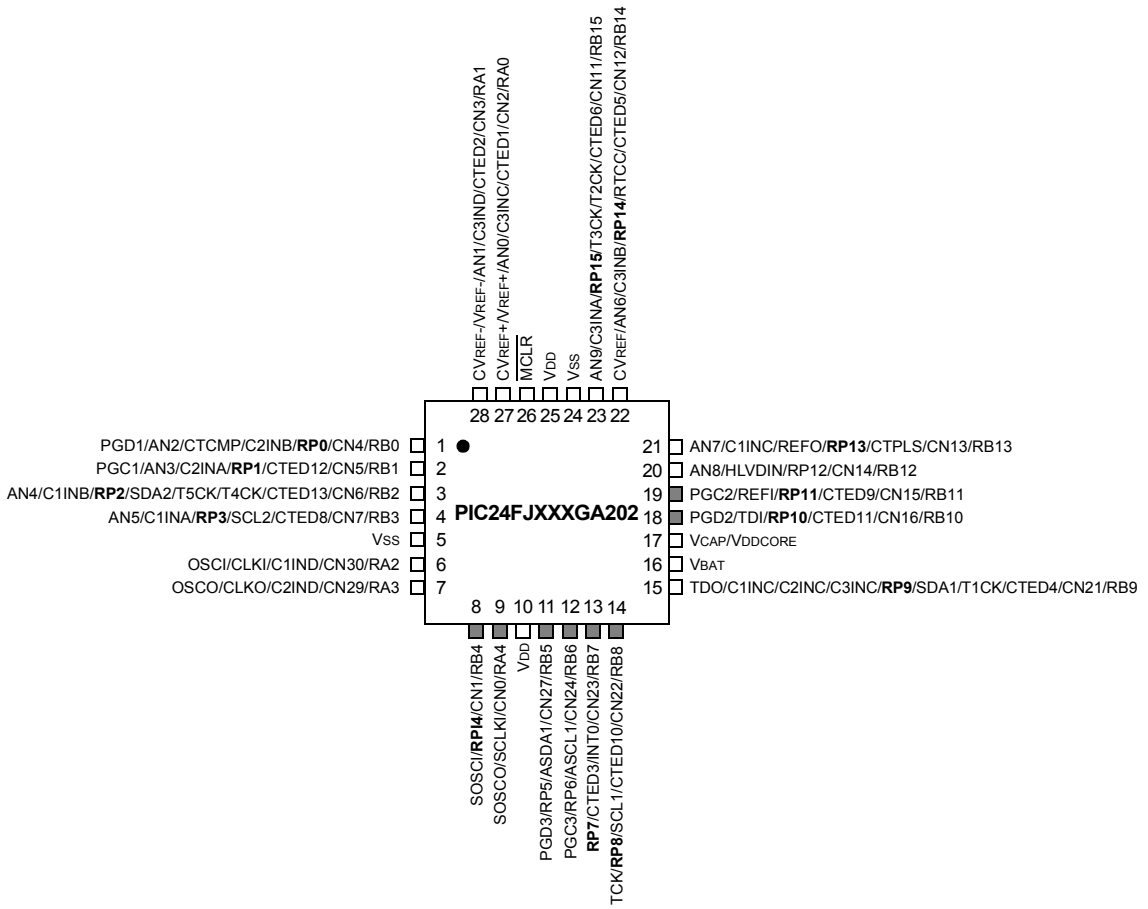
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SmartCard, SPI, UART/USART
Peripherals	AES, Brown-out Detect/Reset, DMA, I²S, HLVD, POR, PWM, WDT
Number of I/O	21
Program Memory Size	128KB (43K 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/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj128ga202t-i-so

Pin Diagrams (Continued)

28-Pin QFN-S(1,2)

5V tolerant



Legend: **RPn** represents remappable peripheral pins.

Note 1: The back pad on QFN devices should be connected to VSS.

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24FJ64GA202
- PIC24FJ128GA202
- PIC24FJ64GA204
- PIC24FJ128GA204

The PIC24FJ128GA204 family expands the capabilities of the PIC24F family by adding a complete selection of Cryptographic Engines, ISO 7816 support and I^2S support to its existing features. This combination, along with its ultra low-power features and Direct Memory Access (DMA) for peripherals, make this family the new standard for mixed-signal PIC® microcontrollers in one economical and power-saving package.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® Digital Signal Controllers (DSCs). The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 32 Kbytes (data)
- A 16-element Working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32 by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as 'C'
- Operational performance up to 16 MIPS

1.1.2 XLP POWER-SAVING TECHNOLOGY

The PIC24FJ128GA204 family of devices introduces a greatly expanded range of power-saving operating modes for the ultimate in power conservation. The new modes include:

- Retention Sleep, with essential circuits being powered from a separate low-voltage regulator
- Deep Sleep without RTCC, for the lowest possible power consumption under software control
- V_{BAT} mode (with or without RTCC), to continue limited operation from a backup battery when V_{DD} is removed

Many of these new low-power modes also support the continuous operation of the low-power, on-chip Real-Time Clock/Calendar (RTCC), making it possible for an application to keep time while the device is otherwise asleep.

Aside from these new features, PIC24FJ128GA204 family devices also include all of the legacy power-saving features of previous PIC24F microcontrollers, such as:

- On-the-Fly Clock Switching, allowing the selection of a lower power clock during run time
- Doze Mode Operation, for maintaining peripheral clock speed while slowing the CPU clock
- Instruction-Based Power-Saving Modes, for quick invocation of Idle and the many Sleep modes

1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC24FJ128GA204 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes
- Two External Clock modes
- A Phase-Locked Loop (PLL) frequency multiplier, which allows clock speeds of up to 32 MHz
- A Fast Internal Oscillator (FRC) – nominal 8 MHz output with multiple frequency divider options and automatic frequency self-calibration during run time
- A separate, Low-Power Internal RC Oscillator (LPRC) – 31 kHz nominal, for low-power, timing-insensitive applications.

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor (FSCM). This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

1.1.4 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve. This extends the ability of applications to grow from the relatively simple, to the powerful and complex, yet still selecting a Microchip device.

TABLE 4-11: SPI1 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
SPI1CON1L	0300	SPIEN	—	SPISIDL	DISSDO	MODE32	MODE16	SMP	CKE	SSEN	CKP	MSTEN	DISSDI	DISSCK	MCLKEN	SPIFE	ENHBUF	0000
SPI1CON1H	0302	AUDEN	SPISGNEXT	IGNROV	IGNTUR	AUDMONO	URDTEN	AUDMOD1	AUDMOD0	FRMEN	FRMSYNC	FRMPOL	MSSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0	0000
SPI1CON2L	0304	—	—	—	—	—	—	—	—	—	—	—	WLENGTH<4:0>					0000
SPI1STATL	0308	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR	SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF	0028
SPI1STATH	030A	—	—	RXELM5	RXELM4	RXELM3	RXELM2	RXELM1	RXELM0	—	—	TXELM5	TXELM4	TXELM3	TXELM2	TXELM1	TXELM0	0000
SPI1BUFL	030C	SPI1BUFL<15:0>																0000
SPI1BUFH	030E	SPI1BUFH<31:16>																0000
SPI1BRGL	0310	—	—	—	SPI1BRG<12:0>													0000
SPI1IMSKL	0314	—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	0000
SPI1IMSKH	0316	RXWIEN	—	RXMSK5	RXMSK4	RXMSK3	RXMSK2	RXMSK1	RXMSK0	TXWIEN	—	TXMSK5	TXMSK4	TXMSK3	TXMSK2	TXMSK1	TXMSK0	0000
SPI1URDTL	0318	SPI1URDTL<15:0>																0000
SPI1URDTH	031A	SPI1URDTH<31:16>																0000

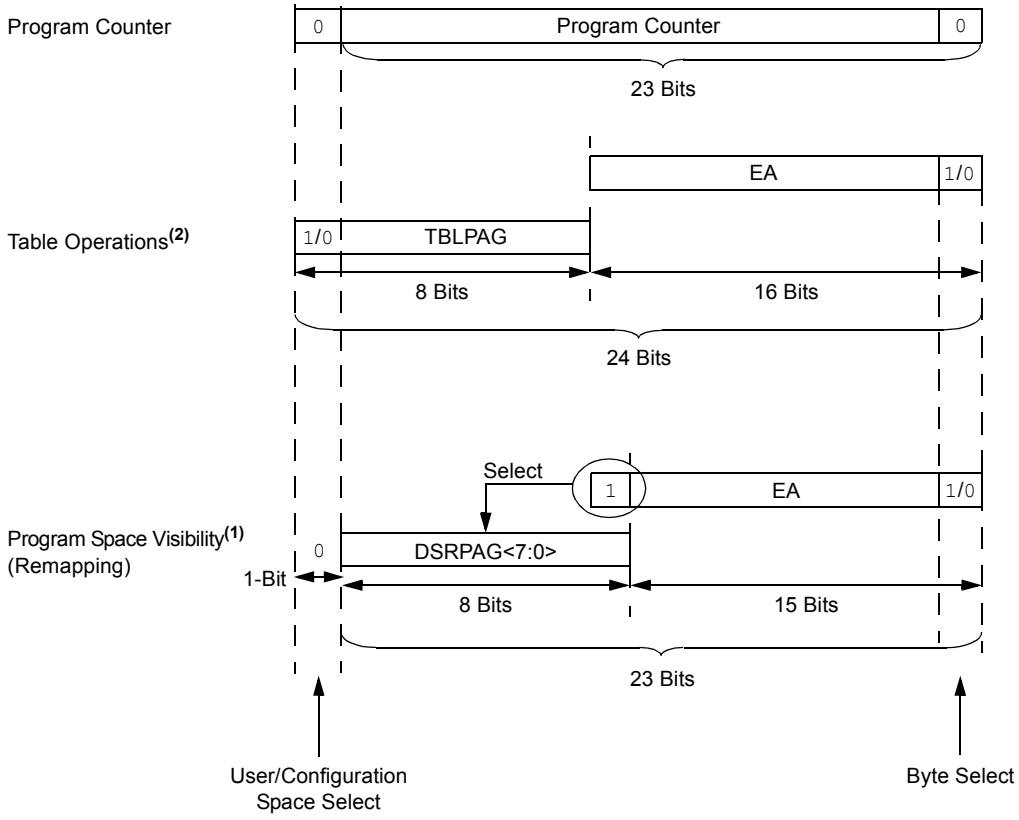
Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-12: SPI2 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
SPI2CON1L	031C	SPIEN	—	SPISIDL	DISSDO	MODE32	MODE16	SMP	CKE	SSEN	CKP	MSTEN	DISSDI	DISSCK	MCLKEN	SPIFE	ENHBUF	0000
SPI2CON1H	031E	AUDEN	SPISGNEXT	IGNROV	IGNTUR	AUDMONO	URDTEN	AUDMOD1	AUDMOD0	FRMEN	FRMSYNC	FRMPOL	MSSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0	0000
SPI2CON2L	0320	—	—	—	—	—	—	—	—	—	—	—	WLENGTH<4:0>					0000
SPI2STATL	0324	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR	SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF	0028
SPI2STATH	0326	—	—	RXELM5	RXELM4	RXELM3	RXELM2	RXELM1	RXELM0	—	—	TXELM5	TXELM4	TXELM3	TXELM2	TXELM1	TXELM0	0000
SPI2BUFL	0328	SPI2BUFL<15:0>																0000
SPI2BUFH	032A	SPI2BUFH<31:16>																0000
SPI2BRGL	032C	—	—	—	SPI2BRG<12:0>													0000
SPI2IMSKL	0330	—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	0000
SPI2IMSKH	0332	RXWIEN	—	RXMSK5	RXMSK4	RXMSK3	RXMSK2	RXMSK1	RXMSK0	TXWIEN	—	TXMSK5	TXMSK4	TXMSK3	TXMSK2	TXMSK1	TXMSK0	0000
SPI2URDTL	0334	SPI2URDTL<15:0>																0000
SPI2URDTH	0336	SPI2URDTH<31:16>																0000

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

FIGURE 4-8: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note 1: DSRPAG<8> acts as word select. DSRPAG<9> should always be '1' to map program memory to data memory.

2: The instructions, TBLRDH/TBLWTH/TBLRDL/TBLWTL, decide if the higher or lower word of program memory is accessed. TBLRDH/TBLWTH instructions access the higher word and TBLRDL/TBLWTL instructions access the lower word. Table Read operations are permitted in the configuration memory space.

REGISTER 7-1: RCON: RESET CONTROL REGISTER (CONTINUED)

bit 5 **SWDTEN:** Software Enable/Disable of WDT bit(4)
1 = WDT is enabled
0 = WDT is disabled

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit(1)
1 = WDT time-out has occurred
0 = WDT time-out has not occurred

bit 3 **SLEEP:** Wake from Sleep Flag bit(1)
1 = Device has been in Sleep mode
0 = Device has not been in Sleep mode

bit 2 **IDLE:** Wake from Idle Flag bit(1)
1 = Device has been in Idle mode
0 = Device has not been in Idle mode

bit 1 **BOR:** Brown-out Reset Flag bit(1)
1 = A Brown-out Reset has occurred (also set after a Power-on Reset)
0 = A Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit(1)
1 = A Power-on Reset has occurred
0 = A Power-on Reset has not occurred

Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

2: If the $\overline{\text{LPCFG}}$ Configuration bit is '1' (unprogrammed), the retention regulator is disabled and the RETEN bit has no effect.

3: Re-enabling the regulator after it enters Standby mode will add a delay, T_{VREG}, when waking up from Sleep. Applications that do not use the voltage regulator should set this bit to prevent this delay from occurring.

4: If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled**,** regardless of the SWDTEN bit setting.

8.3 Interrupt Control and Status Registers

The PIC24FJ128GA204 family of devices implements a total of 43 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS7
- IEC0 through IEC7
- IPC0 through IPC16, IPC18 through IPC22, IPC26 and IPC29
- 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 (AIVT).

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 an 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<7:0>) and the Interrupt Priority 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 order of their vector numbers, as shown in Table 8-2. 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 can change the current CPU priority level by writing to the IPLx bits.

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

The interrupt controller has the Interrupt Controller Test register, INTTREG, which displays the status of the interrupt controller. When an interrupt request occurs, its associated vector number and the new Interrupt Priority Level are latched into INTTREG. This information can be used to determine a specific interrupt source if a generic ISR is used for multiple vectors (such as when ISR remapping is used in bootloader applications) or to check if another interrupt is pending while in an ISR.

All Interrupt registers are described in Register 8-1 through Register 8-45 in the succeeding pages.

REGISTER 8-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SPI3TXIF	SPI3IF	U4TXIF	U4RXIF
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIF	—	I2C2BCIF	I2C1BCIF	U3TXIF	U3RXIF	U3ERIF	—
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 15-12 **Unimplemented:** Read as '0'

bit 11 **SPI3TXIF:** SPI3 Transmit Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPI3IF:** SPI3 General Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **U4TXIF:** UART4 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **U4RXIF:** UART4 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **U4ERIF:** UART4 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

bit 5 **I2C2BCIF:** I2C2 Bus Collision Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **I2C1BCIF:** I2C1 Bus Collision Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **U3TXIF:** UART3 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **U3RXIF:** UART3 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U3ERIF:** UART3 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 8-24: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA1IP2	DMA1IP1	DMA1IP0
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
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
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 15-11 **Unimplemented:** Read as '0'

bit 10-8 **DMA1IP<2:0>:** DMA Channel 1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

bit 6-4 **AD1IP<2:0>:** A/D 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 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 8-40: IPC20: INTERRUPT PRIORITY CONTROL REGISTER 20

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—
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 14-12 **U3TXIP<2:0>:** UART3 Transmitter Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

bit 10-8 **U3RXIP<2:0>:** UART3 Receiver Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

bit 6-4 **U3ERIP<2:0>:** UART3 Error Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

11.3 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the PIC24FJ128GA204 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 82 external inputs that may be selected (enabled) for generating an interrupt request on a Change-of-State.

Registers, CNEN1 through CNEN3, contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin has both a weak pull-up and a weak pull-down connected to it. The pull-ups act as a current source that is connected to the pin, while the pull-downs act as a current sink that is connected to the pin. These eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups and pull-downs are separately enabled using the CNPU1 through CNPU3 registers (for pull-ups), and the CNPD1 through CNPD3 registers (for pull-downs). Each CN pin has individual control bits for its pull-up and pull-down. Setting a control bit enables the weak pull-up or pull-down for the corresponding pin.

When the internal pull-up is selected, the pin pulls up to V_{DD} – 1.1V (typical). When the internal pull-down is selected, the pin pulls down to V_{SS}.

Note:	Pull-ups on Input Change Notification pins should always be disabled whenever the port pin is configured as a digital output.

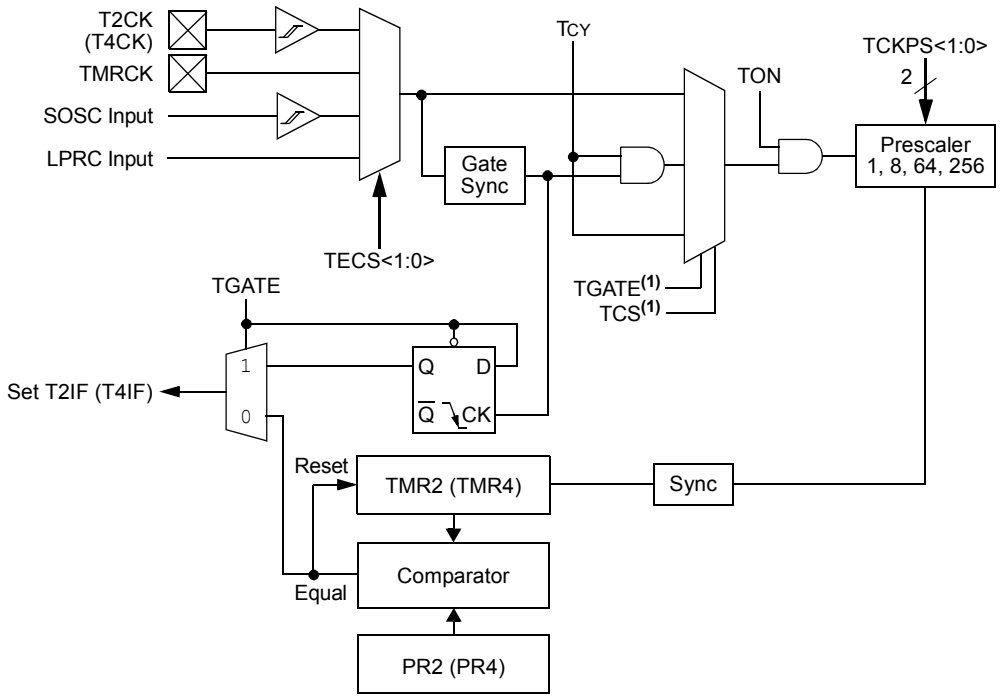
EXAMPLE 11-1: PORT READ/WRITE IN ASSEMBLY

```
MOV   0xFF00, W0   ; Configure PORTB<15:8> as inputs
MOV   W0, TRISB    ; and PORTB<7:0> as outputs
NOP                ; Delay 1 cycle
BTSS  PORTB, #13   ; Next Instruction
```

EXAMPLE 11-2: PORT READ/WRITE IN 'C'

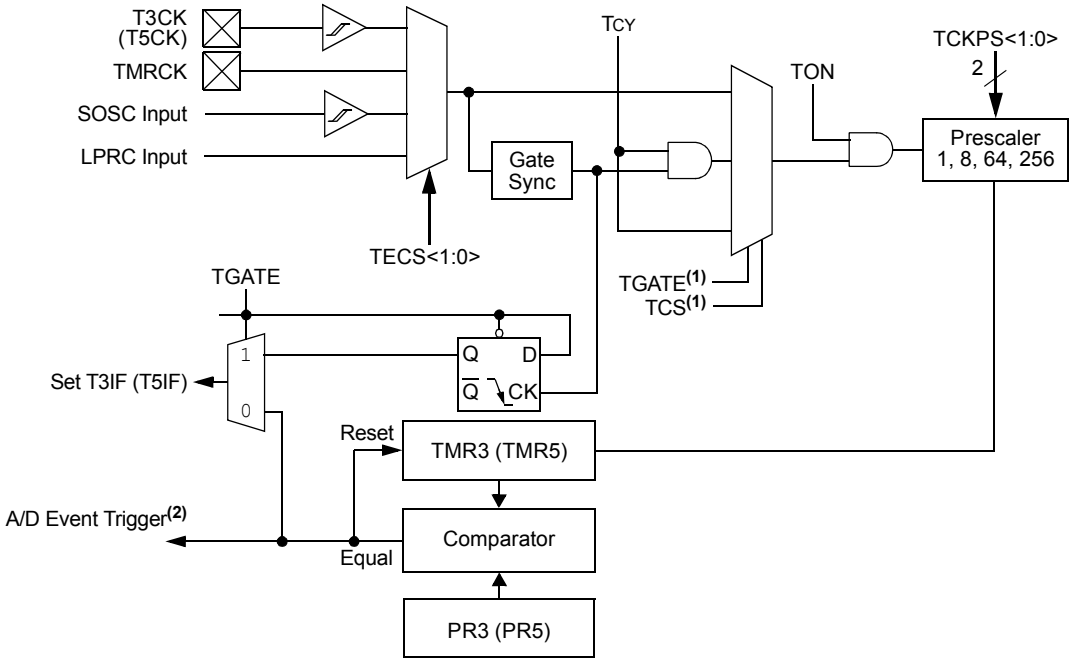
```
TRISB = 0xFF00;              // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
Nop();                       // Delay 1 cycle
If (PORTBbits.RB13){ };      // Next Instruction
```

FIGURE 13-2: TIMER2 AND TIMER4 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn/RPIn pin before use. See **Section 11.4 "Peripheral Pin Select (PPS)"** for more information.

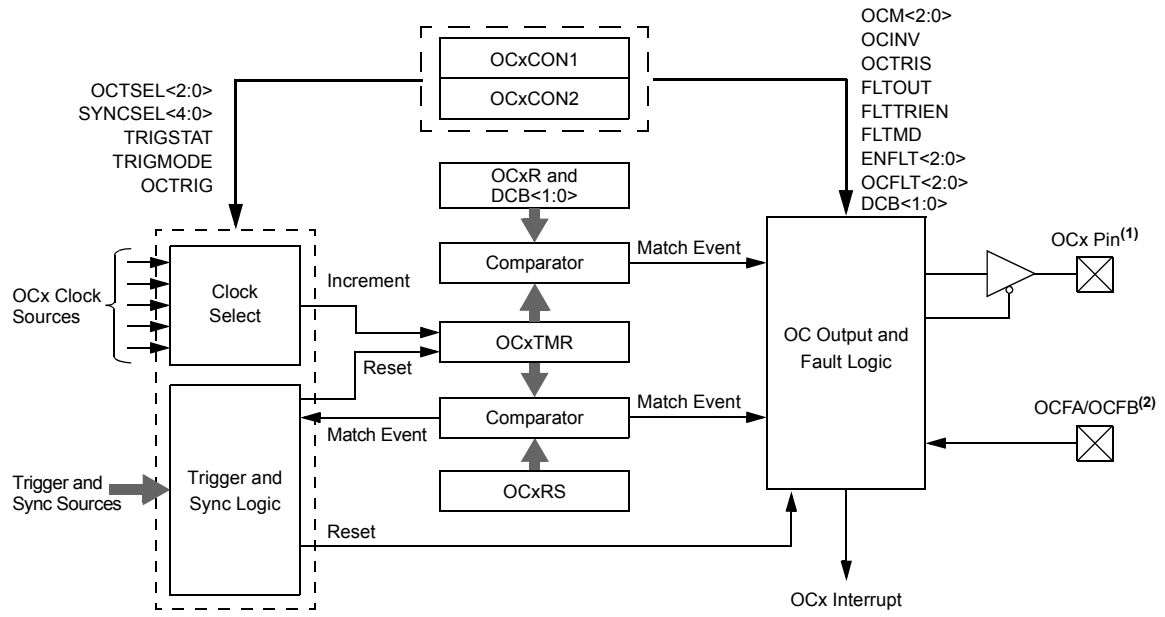
FIGURE 13-3: TIMER3 AND TIMER5 (16-BIT ASYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn/RPIn pin before use. See **Section 11.4 "Peripheral Pin Select (PPS)"** for more information.

2: The A/D Event Trigger is available only on Timer3.

FIGURE 15-1: OUTPUT COMPARE x BLOCK DIAGRAM (16-BIT MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)".**

2: The OCFA/OCFB Fault inputs must be assigned to an available RPn/RPIn pin before use. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)"**.

15.2 Compare Operations

In Compare mode (Figure 15-1), the output compare module can be configured for single-shot or continuous pulse generation. It can also repeatedly toggle an output pin on each timer event.

To set up the module for compare operations:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the required values for the OCxR and (for Double Compare modes) OCxRS Duty Cycle registers:
   a) Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
   b) Calculate the time to the rising edge of the output pulse relative to the timer start value (0000h).
   c) Calculate the time to the falling edge of the pulse based on the desired pulse width and the time to the rising edge of the pulse.
3. Write the rising edge value to OCxR and the falling edge value to OCxRS.
4. Set the Timer Period register, PRy, to a value equal to or greater than the value in OCxRS.
5. Set the OCM<2:0> bits for the appropriate compare operation ('0xx').
6. For Trigger mode operations, set OCTRIG to enable Trigger mode. Set or clear TRIGMODE to configure trigger operation and TRIGSTAT to select a hardware or software trigger. For Synchronous mode, clear OCTRIG.
7. Set the SYNCSEL<4:0> bits to configure the trigger or synchronization source. If free-running timer operation is required, set the SYNCSELx bits to '00000' (no sync/trigger source).
8. Select the time base source with the OCTSEL<2:0> bits. If necessary, set the TON bit for the selected timer, which enables the compare time base to count. Synchronous mode operation starts as soon as the time base is enabled; Trigger mode operation starts after a trigger source event occurs.

REGISTER 16-4: SPIxSTATL: SPIx STATUS REGISTER LOW

U-0	U-0	U-0	R/C-0, HS	R-0, HSC	U-0	U-0	R-0, HSC
—	—	—	FRMERR	SPIBUSY	—	—	SPITUR(1)
bit 15							bit 8

R-0, HSC	R/C-0, HS	R-1, HSC	U-0	R-1, HSC	U-0	R-0, HSC	R-0, HSC
SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF
bit 7							bit 0

Legend:	C = Clearable bit	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	HS = Hardware Settable bit

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **FRMERR**: SPIx Frame Error Status bit
1 = Frame error is detected
0 = No frame error is detected

bit 11 **SPIBUSY:** SPIx Activity Status bit
1 = Module is currently busy with some transactions
0 = No ongoing transactions (at time of read)

bit 10-9 **Unimplemented:** Read as '0'

bit 8 **SPITUR:** SPIx Transmit Underrun Status bit(1)
1 = Transmit buffer has encountered a Transmit Underrun (TUR) condition
0 = Transmit buffer does not have a Transmit Underrun condition

bit 7 **SRMT:** SPIx Shift Register Empty Status bit
1 = No current or pending transactions (i.e., neither SPIxTXB or SPIxTXSR contains data to transmit)
0 = Current or pending transactions

bit 6 **SPIROV:** SPIx Receive Overflow Status bit
1 = A new byte/half-word/word has been completely received when the SPIxRXB was full
0 = No overflow

bit 5 **SPIRBE:** SPIx RX Buffer Empty Status bit
1 = RX buffer is empty
0 = RX buffer is not empty
Standard Buffer Mode:
Automatically set in hardware when SPIxBUF is read from, reading SPIxRXB. Automatically cleared in hardware when SPIx transfers data from SPIxRXSR to SPIxRXB.
Enhanced Buffer Mode:
Indicates RXELM<5:0> = 6'b000000.

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

bit 3 **SPITBE:** SPIx Transmit Buffer Empty Status bit
1 = SPIxTXB is empty
0 = SPIxTXB is not empty
Standard Buffer Mode:
Automatically set in hardware when SPIx transfers data from SPIxTXB to SPIxTXSR. Automatically cleared in hardware when SPIxBUF is written, loading SPIxTXB.
Enhanced Buffer Mode:
Indicates TXELM<5:0> = 6'b000000.

Note 1: SPITUR is cleared when SPIEN = 0. When IGNTUR = 1, SPITUR provides dynamic status of the Transmit Underrun condition, but does not stop RX/TX operation and does not need to be cleared by software.

REGISTER 16-6: SPIxIMSKL: SPIx INTERRUPT MASK REGISTER LOW

U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN
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 **FRMERREN:** Enable Interrupt Events via FRMERR bit
1 = Frame error generates an interrupt event
0 = Frame error does not generate an interrupt event

bit 11 **BUSYEN:** Enable Interrupt Events via SPIBUSY bit
1 = SPIBUSY generates an interrupt event
0 = SPIBUSY does not generate an interrupt event

bit 10-9 **Unimplemented:** Read as '0'

bit 8 **SPITUREN:** Enable Interrupt Events via SPITUR bit
1 = Transmit Underrun (TUR) generates an interrupt event
0 = Transmit Underrun does not generate an interrupt event

bit 7 **SRMTEN:** Enable Interrupt Events via SRMT bit
1 = Shift Register Empty (SRMT) generates an interrupt events
0 = Shift Register Empty does not generate an interrupt events

bit 6 **SPIROVEN:** Enable Interrupt Events via SPIROV bit
1 = SPIx Receive Overflow generates an interrupt event
0 = SPIx Receive Overflow does not generate an interrupt event

bit 5 **SPIRBEN:** Enable Interrupt Events via SPIRBE bit
1 = SPIx RX buffer empty generates an interrupt event
0 = SPIx RX buffer empty does not generate an interrupt event

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

bit 3 **SPITBEN:** Enable Interrupt Events via SPITBE bit
1 = SPIx transmit buffer empty generates an interrupt event
0 = SPIx transmit buffer empty does not generate an interrupt event

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

bit 1 **SPITBFEN:** Enable Interrupt Events via SPITBF bit
1 = SPIx transmit buffer full generates an interrupt event
0 = SPIx transmit buffer full does not generate an interrupt event

bit 0 **SPIRBFEN:** Enable Interrupt Events via SPIRBF bit
1 = SPIx receive buffer full generates an interrupt event
0 = SPIx receive buffer full does not generate an interrupt event

REGISTER 18-6: UxSCINT: UARTx SMART CARD INTERRUPT REGISTER(1)

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	RXRPTIF(2)	TXRPTIF(2)	—	—	WTCIF	GTCIF
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	PARIE(2)	RXRPTIE(2)	TXRPTIE(2)	—	—	WTCIE	GTCIE
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 **RXRPTIF:** Receive Repeat Interrupt Flag bit(2)
1 = Parity error has persisted after the same character has been received five times (four retransmits)
0 = Flag is cleared

bit 12 **TXRPTIF:** Transmit Repeat Interrupt Flag bit(2)
1 = Line error has been detected after the last retransmit per TXRPT<1:0> (see Register 18-5)
0 = Flag is cleared

bit 11-10 **Unimplemented:** Read as '0'

bit 9 **WTCIF:** Waiting Time Counter Interrupt Flag bit
1 = Waiting Time Counter has reached 0
0 = Waiting Time Counter has not reached 0

bit 8 **GTCIF:** Guard Time Counter Interrupt Flag bit
1 = Guard Time Counter has reached 0
0 = Guard Time Counter has not reached 0

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

bit 6 **PARIE:** Parity Interrupt Enable bit(2)
1 = An interrupt is invoked when a character is received with a parity error; see the PERR bit (UxSTA<3>) in Register 18-2 for the interrupt flag
0 = Interrupt is disabled

bit 5 **RXRPTIE:** Receive Repeat Interrupt Enable bit(2)
1 = An interrupt is invoked when a parity error has persisted after the same character has been received five times (four retransmits)
0 = Interrupt is disabled

bit 4 **TXRPTIE:** Transmit Repeat Interrupt Enable bit(2)
1 = An interrupt is invoked when a line error is detected after the last retransmit per the TXRPT<1:0> bits has been completed (see Register 18-5)
0 = Interrupt is disabled

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **WTCIE:** Waiting Time Counter Interrupt Enable bit
1 = Waiting Time Counter interrupt is enabled
0 = Waiting Time Counter interrupt is disabled

bit 0 **GTCIE:** Guard Time Counter Interrupt Enable bit
1 = Guard Time Counter interrupt is enabled
0 = Guard Time Counter interrupt is disabled

Note 1: This register is only available for UART1 and UART2.
2: This bit is applicable to T = 0 only, see the PTRCL bit (UxSCCON<1>).

REGISTER 22-1: CRYCONL: CRYPTOGRAPHIC CONTROL LOW REGISTER

R/W-0	U-0	R/W-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	U-0	R/W-0, HC(1)
CRYON	—	CRYSIDL(3)	ROLLIE	DONEIE	FREEIE	—	CRYGO
bit 15							bit 8

R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)
OPMOD3(2)	OPMOD2(2)	OPMOD1(2)	OPMOD0(2)	CPHRSEL(2)	CPHRMOD2(2)	CPHRMOD1(2)	CPHRMOD0(2)
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 **CRYON:** Cryptographic Enable bit
1 = Module is enabled
0 = Module is disabled

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

bit 13 **CRYSIDL:** Cryptographic Stop in Idle Control bit(3)
1 = Stops module operation in Idle mode
0 = Continues module operation in Idle mode

bit 12 **ROLLIE:** CRYTXTB Rollover Interrupt Enable bit(1)
1 = Generates an interrupt event when the counter portion of CRYTXTB rolls over to '0'
0 = Does not generate an interrupt event when the counter portion of CRYTXTB rolls over to '0'

bit 11 **DONEIE:** Operation Done Interrupt Enable bit(1)
1 = Generates an interrupt event when the current cryptographic operation completes
0 = Does not generate an interrupt event when the current cryptographic operation completes; software must poll the CRYGO or CRYBSY bit to determine when current cryptographic operation is complete

bit 10 **FREEIE:** Input Text Interrupt Enable bit(1)
1 = Generates an interrupt event when the input text (plaintext or ciphertext) is consumed during the current cryptographic operation
0 = Does not generate an interrupt event when the input text is consumed

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

bit 8 **CRYGO:** Cryptographic Engine Start bit(1)
1 = Starts the operation specified by OPMOD<3:0> (cleared automatically when operation is done)
0 = Stops the current operation (when cleared by software); also indicates the current operation has completed (when cleared by hardware)

Note 1: These bits are reset on system Resets or whenever the CRYMD bit is set.
2: Writes to these bit fields are locked out whenever an operation is in progress (CRYGO bit is set).
3: If the device enters Idle mode when CRYSIDL = 1, the module will stop its current operation. Entering into Idle mode while an OTP write operation is in process can result in irreversible corruption of the OTP.

REGISTER 24-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
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	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
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 **ADRC:** A/D Conversion Clock Source bit
1 = RC clock
0 = Clock derived from system clock

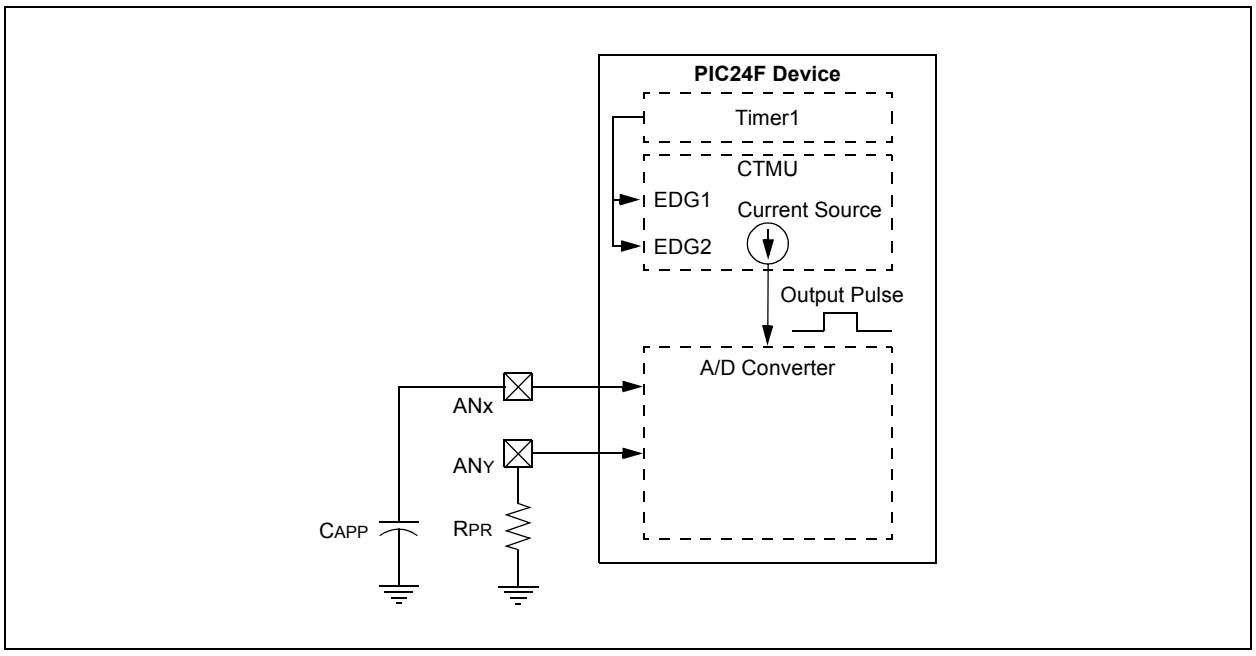
bit 14 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 13 **PUMPEN:** Charge Pump Enable bit
1 = Charge pump for switches is enabled
0 = Charge pump for switches is disabled

bit 12-8 **SAMC<4:0>:** Auto-Sample Time Select bits
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111 = 256 • TCY = TAD
•
•
•
00000001 = 2 • TCY = TAD
00000000 = TCY = TAD

FIGURE 27-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



29.0 SPECIAL FEATURES

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 following sections of the *"dsPIC33/PIC24 Family Reference Manual"*. The information in this data sheet supersedes the information in the FRMs.

- **"Watchdog Timer (WDT)"** (DS39697)
- **"High-Level Device Integration"** (DS39719)
- **"Programming and Diagnostics"** (DS39716)

PIC24FJ128GA204 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation (ICE)

29.1 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped starting at program memory location, F80000h. A detailed explanation of the various bit functions is provided in Register 29-1 through Register 29-6.

Note that address, F80000h, is beyond the user program memory space. In fact, it belongs to the configuration memory space (800000h-FFFFFFh), which can only be accessed using Table Reads and Table Writes.

29.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ128GA204 FAMILY DEVICES

In PIC24FJ128GA204 family devices, the Configuration bytes are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. Configuration data is stored in the four words at the top of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 29-1. These are packed representations of the actual device Configuration bits, whose actual locations are distributed among several locations in configuration space. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration registers during device Resets.

Note: Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The upper byte of all Flash Configuration Words in program memory should always be '0000 0000'. This makes them appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '0's to these locations has no effect on device operation.

Note: Performing a page erase operation on the last page of program memory clears the Flash Configuration Words, enabling code protection as a result. Therefore, users should avoid performing page erase operations on the last page of program memory.

TABLE 29-1: FLASH CONFIGURATION WORD LOCATIONS FOR THE PIC24FJ128GA204 FAMILY

Device	Configuration Word Addresses			
	1	2	3	4
PIC24FJ64GA2XX	ABFEh	ABFCh	ABFAh	ABF8h
PIC24FJ128GA2XX	157FEh	157FCh	157FAh	157F8h

FIGURE 32-3: EXTERNAL CLOCK TIMING

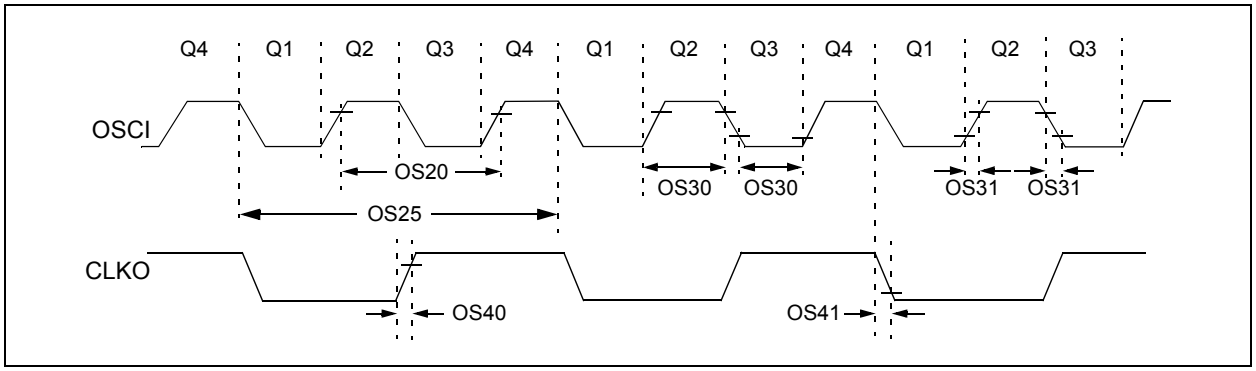


TABLE 32-19: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
OS10	FOSC	External CLKI Frequency (External clocks allowed only in EC mode)	DC 4	— —	32 48	MHz MHz	EC ECPLL **(Note 2)**
		Oscillator Frequency	3.5 4 10 12 31	— — — — —	10 8 32 32 33	MHz MHz MHz MHz kHz	XT XTPLL HS HSPLL SOSC
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	See Parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time(3)	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time(4)	—	6	10	ns	
OS41	TckF	CLKO Fall Time(4)	—	6	10	ns	

Note 1: Data in the "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Represents input to the system clock prescaler. PLL dividers and postscalers must still be configured so that the system clock frequency does not exceed the maximum frequency shown in Figure 32-1.

3: Instruction cycle period (TCY) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type, under standard operating conditions, with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min" values with an external clock applied to the OSCI/CLKI pin. When an external clock input is used, the "Max" cycle time limit is "DC" (no clock) for all devices.

4: Measurements are taken in EC mode. The CLKO signal is measured on the OSCO pin. CLKO is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).