



Welcome to [E-XFL.COM](https://www.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	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga306-i-mr

PIC24FJ128GA310 FAMILY

High-Performance CPU:

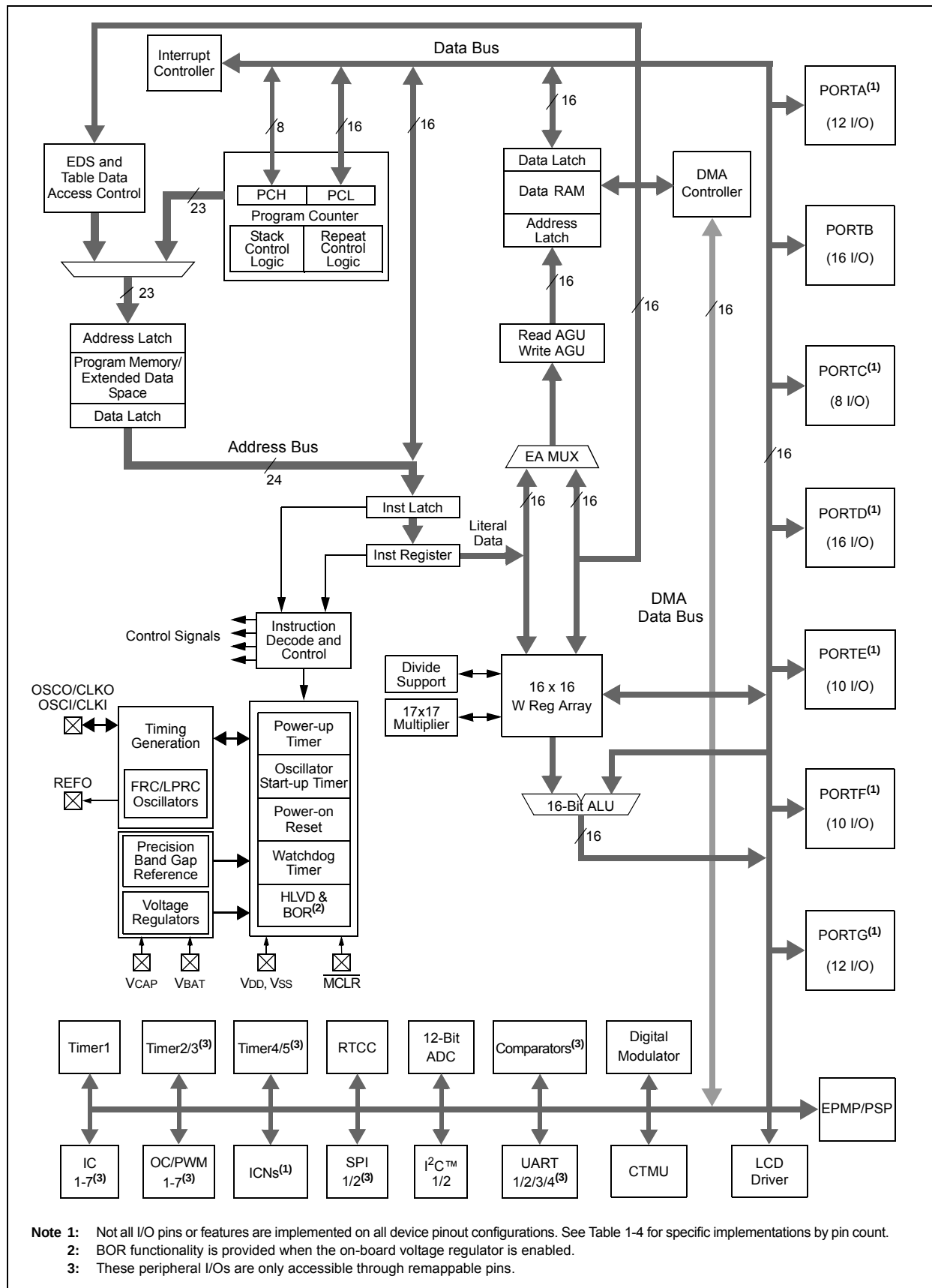
- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator:
 - 4x PLL option
 - Multiple clock divide options
 - Fast start-up
- 17-Bit x 17-Bit Single-Cycle Hardware Fractional/Integer Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- Two On-Chip Voltage Regulators (1.8V and 1.2V) for Regular and Extreme Low-Power Operation
- 20,000 Erase/Write Cycle Endurance Flash Program Memory, Typical
- Flash Data Retention: 20 Years Minimum
- Self-Programmable under Software Control
- Programmable Reference Clock Output
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Emulation (ICE) via 2 Pins
- JTAG Boundary Scan Support
- Fail-Safe Clock Monitor Operation:
 - Detects clock failure and switches to on-chip, low-power RC oscillator
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with Operation Below V_{BOR}
- High/Low-Voltage Detect (HLVD)
- Flexible Watchdog Timer (WDT) with its Own RC Oscillator for Reliable Operation
- Standard and Ultra Low-Power Watchdog Timers (ULPW) for Reliable Operation in Standard and Deep Sleep modes

PIC24FJ128GA310 FAMILY

FIGURE 1-1: PIC24FJ128GA310 FAMILY GENERAL BLOCK DIAGRAM



PIC24FJ128GA310 FAMILY

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 ADC 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 ADxPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to **Section 11.2 “Configuring Analog Port Pins (ANSx)”** for more specific information.

The bits in these registers that correspond to the ADC 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 ADC pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the ADC module, as follows:

- For devices with an ADxPCFG 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 ADxPCFG or ANSx registers. Automatic initialization of these registers is only done during debugger operation. Failure to correctly configure the register(s) will result in all ADC 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 k Ω to 10 k Ω resistor to Vss on unused pins and drive the output to logic low.

PIC24FJ128GA310 FAMILY

TABLE 8-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
DMA Channel 0	4	00001Ch	00011Ch	IFS0<4>	IEC0<4>	IPC1<2:0>
DMA Channel 1	14	000030h	000130h	IFS0<14>	IEC0<14>	IPC3<10:8>
DMA Channel 2	24	000044h	000144h	IFS1<8>	IEC1<8>	IPC6<2:0>
DMA Channel 3	36	00005Ch	00015Ch	IFS2<4>	IEC2<4>	IPC9<2:0>
DMA Channel 4	46	000070h	000170h	IFS2<14>	IEC2<14>	IPC11<10:8>
DMA Channel 5	61	00008Eh	00018Eh	IFS3<13>	IEC3<13>	IPC15<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Capture 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
Input Capture 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
JTAG	117	0000FEh	0001FEh	IFS7<5>	IEC7<5>	IPC29<6:4>
Input Change Notification (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LCD Controller	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
High/Low-Voltage Detect (HLVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Output Compare 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
Output Compare 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
Enhanced Parallel Master Port (EPMP)	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock and Calendar (RTCC)	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>

PIC24FJ128GA310 FAMILY

REGISTER 8-17: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	HLVDIE
bit 15						bit 8	

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
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 **CTMUIE:** CTMU Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 12-9 **Unimplemented:** Read as '0'
- bit 8 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 7-4 **Unimplemented:** Read as '0'
- bit 3 **CRCIE:** CRC Generator Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 2 **U2ERIE:** UART2 Error Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 1 **U1ERIE:** UART1 Error Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 0 **Unimplemented:** Read as '0'

PIC24FJ128GA310 FAMILY

REGISTER 8-18: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U4TXIE	U4RXIE
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIE	—	—	—	U3TXIE	U3RXIE	U3ERIE	—
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 15-10 **Unimplemented:** Read as '0'
- bit 9 **U4TXIE:** UART4 Transmitter Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 8 **U4RXIE:** UART4 Receiver Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 7 **U4ERIE:** UART4 Error Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 6-4 **Unimplemented:** Read as '0'
- bit 3 **U3TXIE:** UART3 Transmitter Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 2 **U3RXIE:** UART3 Receiver Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 1 **U3ERIE:** UART3 Error Interrupt Enable bit
 - 1 = Interrupt request is enabled
 - 0 = Interrupt request is not enabled
- bit 0 **Unimplemented:** Read as '0'

PIC24FJ128GA310 FAMILY

9.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers:

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The CLKDIV register (Register 9-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The OSCTUN register (Register 9-3) allows the user to fine tune the FRC oscillator over a range of approximately $\pm 1.5\%$. Each bit increment or decrement changes the factory calibrated frequency of the FRC oscillator by a fixed amount.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0	R-0	R-0	U-0	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0	R/W-0	R-0 ⁽³⁾	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK ⁽²⁾	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

Legend:	CO = Clearable Only bit	SO = Settable Only 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 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits

- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Reserved
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits⁽¹⁾

- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Reserved
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

Note 1: Reset values for these bits are determined by the FNOSC_x Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1' once the IOLOCK bit is set, it cannot be cleared.

3: This bit also resets to '0' during any valid clock switch or whenever a Non-PLL Clock mode is selected.

PIC24FJ128GA310 FAMILY

REGISTER 9-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<5:0> ⁽¹⁾					
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 15-6

Unimplemented: Read as '0'

bit 5-0

TUN<5:0>: FRC Oscillator Tuning bits⁽¹⁾

011111 = Maximum frequency deviation

011110 =

•

•

•

000001 =

000000 = Center frequency, oscillator is running at factory calibrated frequency

111111 =

•

•

•

100001 =

100000 = Minimum frequency deviation

Note 1: Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

9.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note: The Primary Oscillator mode has three different submodes (XT, HS and EC) which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

9.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSMx Configuration bits in CW2 must be programmed to '00'. (Refer to **Section 29.1 "Configuration Bits"** for further details.) If the FCKSMx Configuration bits are unprogrammed ('1x'), the clock switching function and Fail-Safe Clock Monitor function are disabled. This is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled. It is held at '0' at all times.

PIC24FJ128GA310 FAMILY

11.0 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 “**I/O Ports with Peripheral Pin Select (PPS)**” (DS39711) in the “*dsPIC33/PIC24 Family Reference Manual*”. The information in this data sheet supersedes the information in the FRM.

All of the device pins (except VDD, VSS, MCLR and OSC1/CLKI) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger (ST) inputs for improved noise immunity.

11.1 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 11-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

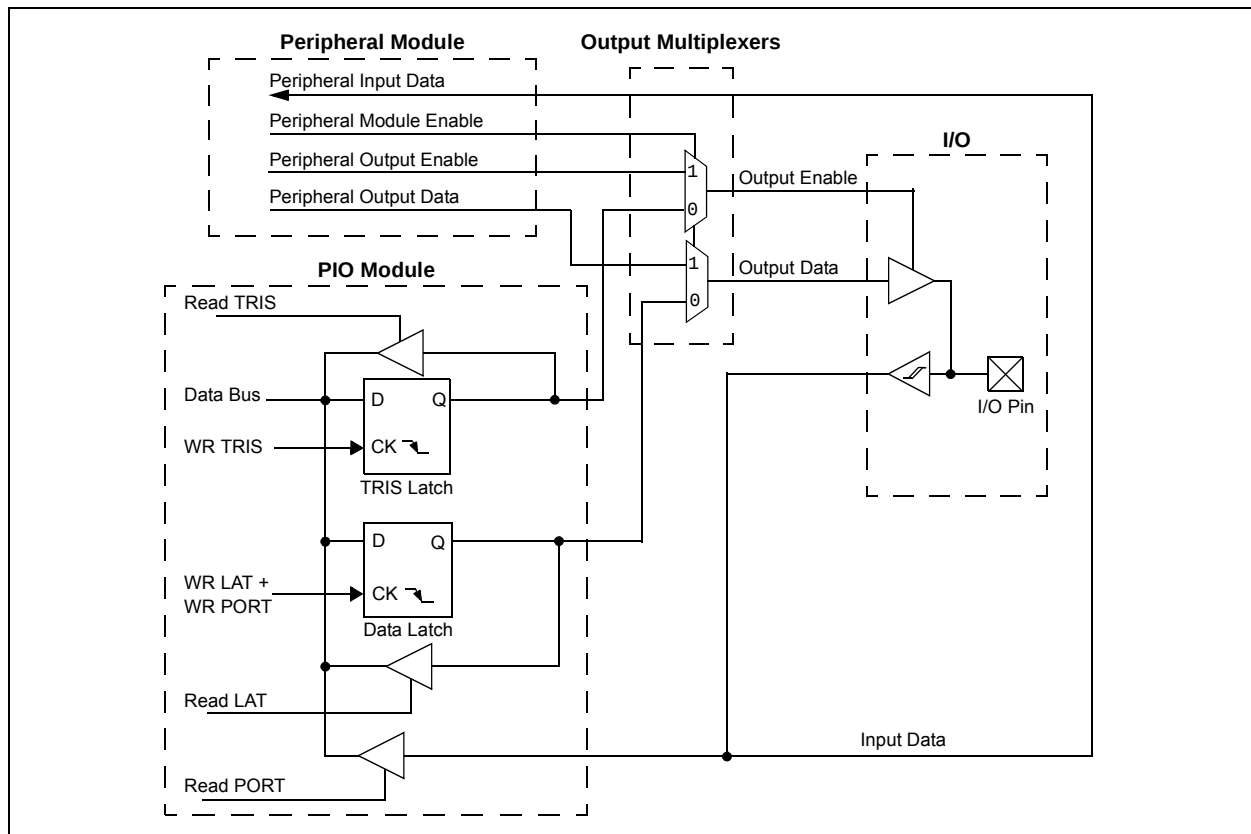
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively 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/Os and one register associated with their operation as analog inputs. 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; 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 regarded as a dedicated port because there is no other competing source of inputs. RC13 and RC14 can be input ports only; they cannot be configured as outputs.

FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



PIC24FJ128GA310 FAMILY

15.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS and OCxR registers. The OCxRS and OCxR registers can be written to at any time, but the duty cycle value is not latched until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation.

Some important boundary parameters of the PWM duty cycle include:

- If OCxR, OCxRS, and PRy are all loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- If OCxRS is greater than PRy, the pin will remain high (100% duty cycle).

See Example 15-1 for PWM mode timing details. Table 15-1 and Table 15-2 show example PWM frequencies and resolutions for a device operating at 4 MIPS and 10 MIPS, respectively.

EQUATION 15-2: CALCULATION FOR MAXIMUM PWM RESOLUTION⁽¹⁾

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10} \left(\frac{F_{CY}}{F_{PWM} \cdot (\text{Timer Prescale Value})} \right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

EXAMPLE 15-1: PWM PERIOD AND DUTY CYCLE CALCULATIONS⁽¹⁾

1. Find the Timer Period register value for a desired PWM frequency of 52.08 kHz, where $F_{OSC} = 8$ MHz with PLL (32 MHz device clock rate) and a Timer2 prescaler setting of 1:1.

$$T_{CY} = 2 \cdot T_{OSC} = 62.5 \text{ ns}$$

$$\text{PWM Period} = 1/\text{PWM Frequency} = 1/52.08 \text{ kHz} = 19.2 \text{ ms}$$

$$\text{PWM Period} = (PR2 + 1) \cdot T_{CY} \cdot (\text{Timer2 Prescale Value})$$

$$19.2 \text{ ms} = (PR2 + 1) \cdot 62.5 \text{ ns} \cdot 1$$

$$PR2 = 306$$

2. Find the maximum resolution of the duty cycle that can be used with a 52.08 kHz frequency and a 32 MHz device clock rate:

$$\text{PWM Resolution} = \log_{10}(F_{CY}/F_{PWM})/\log_{10}(2) \text{ bits}$$

$$= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}(2)) \text{ bits}$$

$$= 8.3 \text{ bits}$$

Note 1: Based on $T_{CY} = 2 \cdot T_{OSC}$; Doze mode and PLL are disabled.

TABLE 15-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS ($F_{CY} = 4$ MHz)⁽¹⁾

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

TABLE 15-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS ($F_{CY} = 16$ MHz)⁽¹⁾

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

PIC24FJ128GA310 FAMILY

REGISTER 15-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1 ⁽³⁾	DCB0 ⁽³⁾	OC32
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

Legend:	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

- bit 15 **FLTMD:** Fault Mode Select bit
1 = Fault mode is maintained until the Fault source is removed and the corresponding OCFLT0 bit is cleared in software
0 = Fault mode is maintained until the Fault source is removed and a new PWM period starts
- bit 14 **FLTOUT:** Fault Out bit
1 = PWM output is driven high on a Fault
0 = PWM output is driven low on a Fault
- bit 13 **FLTTRIEN:** Fault Output State Select bit
1 = Pin is forced to an output on a Fault condition
0 = Pin I/O condition is unaffected by a Fault
- bit 12 **OCINV:** OCMP Invert bit
1 = OCx output is inverted
0 = OCx output is not inverted
- bit 11 **Unimplemented:** Read as '0'
- bit 10-9 **DCB<1:0>:** PWM Duty Cycle Least Significant bits⁽³⁾
11 = Delay OCx falling edge by $\frac{3}{4}$ of the instruction cycle
10 = Delay OCx falling edge by $\frac{1}{2}$ of the instruction cycle
01 = Delay OCx falling edge by $\frac{1}{4}$ of the instruction cycle
00 = OCx falling edge occurs at the start of the instruction cycle
- bit 8 **OC32:** Cascade Two OC Modules Enable bit (32-bit operation)
1 = Cascade module operation is enabled
0 = Cascade module operation is disabled
- bit 7 **OCTRIG:** OCx Trigger/Sync Select bit
1 = Trigger OCx from the source designated by the SYNCSELx bits
0 = Synchronize OCx with the source designated by the SYNCSELx bits
- bit 6 **TRIGSTAT:** Timer Trigger Status bit
1 = Timer source has been triggered and is running
0 = Timer source has not been triggered and is being held clear
- bit 5 **OCTRIIS:** OCx Output Pin Direction Select bit
1 = OCx pin is tri-stated
0 = Output Compare Peripheral x is connected to an OCx pin

- Note 1:** Never use an OCx module as its own trigger source, either by selecting this mode or another equivalent SYNCSELx setting.
- 2:** Use these inputs as trigger sources only and never as sync sources.
- 3:** The DCB<1:0> bits are double-buffered in PWM modes only (OCM<2:0> (OCxCON1<2:0>) = 111, 110).

PIC24FJ128GA310 FAMILY

REGISTER 17-2: 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/A	P	S	R/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 cleared at the end of Acknowledge.
- bit 14 **TRSTAT:** Transmit Status bit
(when operating as I²C™ 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 I²C 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/A:** Data/Address bit (when operating as I²C 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.

18.2 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 the 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. Alternatively, 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.

18.3 Transmitting in 9-Bit Data Mode

1. Set up the UART (as described in **Section 18.2 “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.

18.4 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.

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

1. Set up the UART (as described in **Section 18.2 “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.

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

UARTx Clear-to-Send ($\overline{\text{UxCTS}}$) and Request-to-Send ($\overline{\text{UxRTS}}$) 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.

18.7 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'.

18.7.1 IrDA CLOCK OUTPUT FOR EXTERNAL IrDA SUPPORT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the 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.

18.7.2 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.

PIC24FJ128GA310 FAMILY

REGISTER 19-2: MDSRC: MODULATOR SOURCE CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
SODIS ⁽¹⁾	—	—	—	MS3 ⁽²⁾	MS2 ⁽²⁾	MS1 ⁽²⁾	MS0 ⁽²⁾
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 15-8 **Unimplemented:** Read as '0'

bit 7 **SODIS:** Modulation Source Output Disable bit⁽¹⁾

1 = Output signal driving the peripheral output pin (selected by MDMS<3:0>) is disabled

0 = Output signal driving the peripheral output pin (selected by MDMS<3:0>) is enabled

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **MS<3:0>** Modulation Source Selection bits⁽²⁾

1111 = Unimplemented

1110 = Output Compare/PWM Module 7 output

1101 = Output Compare/PWM Module 6 output

1100 = Output Compare/PWM Module 5 output

1011 = Output Compare/PWM Module 4 output

1010 = Output Compare/PWM Module 3 output

1001 = Output Compare/PWM Module 2 output

1000 = Output Compare/PWM Module 1 output

0111 = UART4 TX output

0110 = UART3 TX output

0101 = UART2 TX output

0100 = UART1 TX output

0011 = SPI2 module output (SDO2)

0010 = SPI1 module output (SDO1)

0001 = Input on MDMIN pin

0000 = Manual modulation using MDBIT (MDCON<0>)

Note 1: This bit is only affected by a POR.

2: These bits are not affected by a POR.

PIC24FJ128GA310 FAMILY

REGISTER 22-1: RCFGCAL: RTCC CALIBRATION/CONFIGURATION REGISTER⁽¹⁾ (CONTINUED)

bit 7-0 **CAL<7:0>**: RTC Drift Calibration bits

01111111 = Maximum positive adjustment; adds 127 RTC clock pulses every 15 seconds

•

•

•

01111111 = Minimum positive adjustment; adds 1 RTC clock pulse every 15 seconds

00000000 = No adjustment

11111111 = Minimum negative adjustment; subtracts 1 RTC clock pulse every 15 seconds

•

•

•

10000000 = Maximum negative adjustment; subtracts 128 RTC clock pulses every 15 seconds

- Note 1:** The RCFGCAL register is only affected by a POR.
- 2:** A write to the RTCEN bit is only allowed when RTCWREN = 1.
- 3:** This bit is read-only; it is cleared to '0' on a write to the lower half of the MINSEC register.

PIC24FJ128GA310 FAMILY

REGISTER 24-10: AD1CSSH: ADC1 INPUT SCAN SELECT REGISTER (HIGH WORD)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CSS<30:24>						
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
CSS<23:16>							
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 14-0 **CSS<30:16>:** ADC Input Scan Selection bits
 1 = Includes corresponding channel for input scan
 0 = Skips channel for input scan

REGISTER 24-11: AD1CSSL: ADC1 INPUT SCAN SELECT REGISTER (LOW WORD)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS<15:8>							
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
CSS<7:0>							
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 15-0 **CSS<15:0>:** ADC Input Scan Selection bits
 1 = Includes corresponding channel for input scan
 0 = Skips channel for input scan

PIC24FJ128GA310 FAMILY

26.0 COMPARATOR VOLTAGE REFERENCE

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 “**Dual Comparator Module**” (DS39710) in the “*dsPIC33/PIC24 Family Reference Manual*”. The information in this data sheet supersedes the information in the FRM.

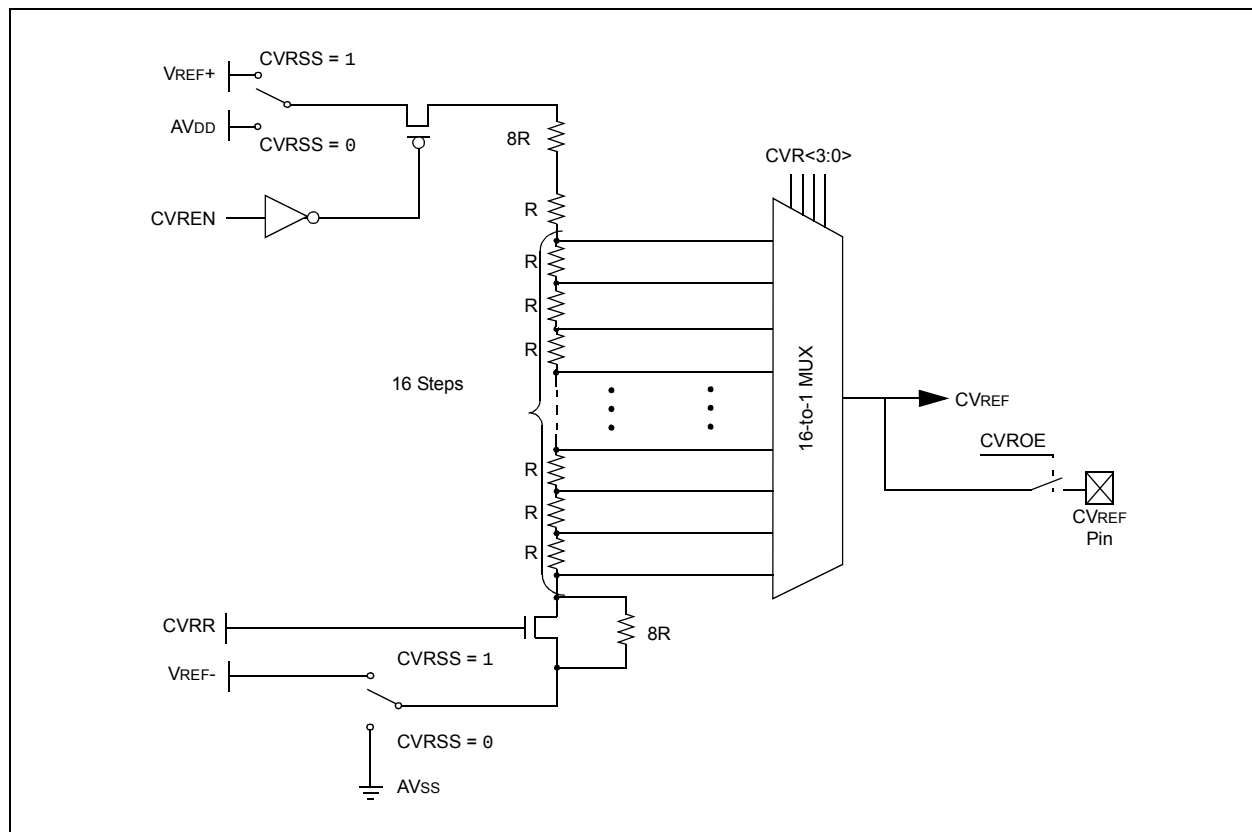
26.1 Configuring the Comparator Voltage Reference

The voltage reference module is controlled through the CVRCON register (Register 26-1). The comparator voltage reference provides two ranges of output voltage, each with 16 distinct levels. The range to be used is selected by the CVRR bit (CVRCON<5>). The primary difference between the ranges is the size of the steps selected by the CVREF Selection bits (CVR<3:0>), with one range offering finer resolution.

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF-. The voltage source is selected by the CVRSS bit (CVRCON<4>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output.

FIGURE 26-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



31.0 INSTRUCTION SET SUMMARY

Note: This chapter is a brief summary of the PIC24F Instruction Set Architecture (ISA) and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction. The instruction set is highly orthogonal and is grouped into four basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 31-1 shows the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 31-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register, 'Wb', without any address modifier
- The second source operand, which is typically a register, 'Ws', with or without an address modifier
- The destination of the result, which is typically a register, 'Wd', with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value, 'f'
- The destination, which could either be the file register, 'f', or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register, 'Wb')

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register, 'Wb', without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register, 'Wd', with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the Table Read and Table Write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSBs are '0's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter (PC) is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all Table Reads and Table Writes, and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

PIC24FJ128GA310 FAMILY

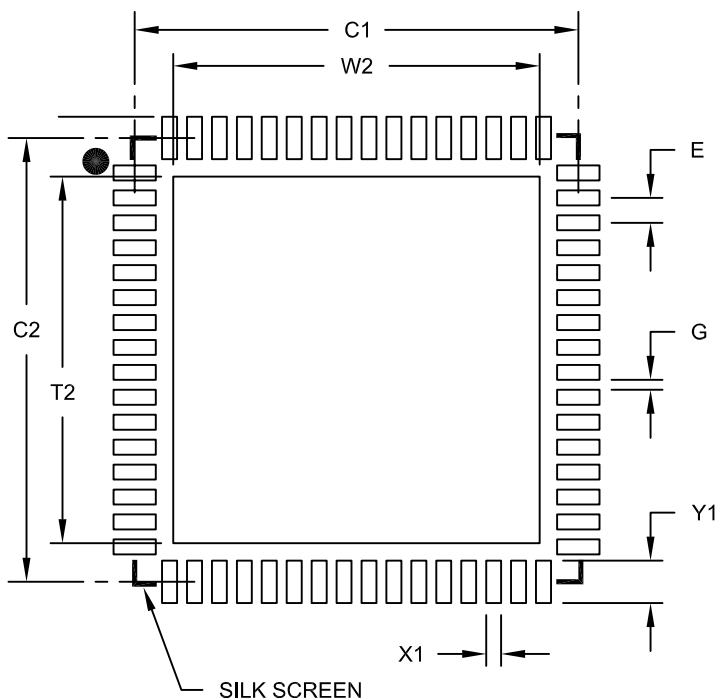
TABLE 31-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
PWRSV	PWRSV #lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
RCALL	RCALL Expr	Relative Call	1	2	None
	RCALL Wn	Computed Call	1	2	None
REPEAT	REPEAT #lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
	REPEAT Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
RESET	RESET	Software Device Reset	1	1	None
RETFIE	RETFIE	Return from Interrupt	1	3 (2)	None
RETLW	RETLW #lit10, Wn	Return with Literal in Wn	1	3 (2)	None
RETURN	RETURN	Return from Subroutine	1	3 (2)	None
RLC	RLC f	f = Rotate Left through Carry f	1	1	C, N, Z
	RLC f, WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
	RLC Ws, Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
RLNC	RLNC f	f = Rotate Left (No Carry) f	1	1	N, Z
	RLNC f, WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
	RLNC Ws, Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC	RRC f	f = Rotate Right through Carry f	1	1	C, N, Z
	RRC f, WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
	RRC Ws, Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC	RRNC f	f = Rotate Right (No Carry) f	1	1	N, Z
	RRNC f, WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
	RRNC Ws, Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
SE	SE Ws, Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM	SETM f	f = FFFFh	1	1	None
	SETM WREG	WREG = FFFFh	1	1	None
	SETM Ws	Ws = FFFFh	1	1	None
SL	SL f	f = Left Shift f	1	1	C, N, OV, Z
	SL f, WREG	WREG = Left Shift f	1	1	C, N, OV, Z
	SL Ws, Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
	SL Wb, Wns, Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
	SL Wb, #lit5, Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
SUB	SUB f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB f, WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB #lit10, Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB Wb, Ws, Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB Wb, #lit5, Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB f	f = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB f, WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB #lit10, Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB Wb, Ws, Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB Wb, #lit5, Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SUBR	SUBR f	f = WREG – f	1	1	C, DC, N, OV, Z
	SUBR f, WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
	SUBR Wb, Ws, Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR Wb, #lit5, Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR f	f = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR f, WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb, Ws, Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb, #lit5, Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SWAP	SWAP.b Wn	Wn = Nibble Swap Wn	1	1	None
	SWAP Wn	Wn = Byte Swap Wn	1	1	None

PIC24FJ128GA310 FAMILY

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]
With 0.40 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	W2			7.35
Optional Center Pad Length	T2			7.35
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2149A