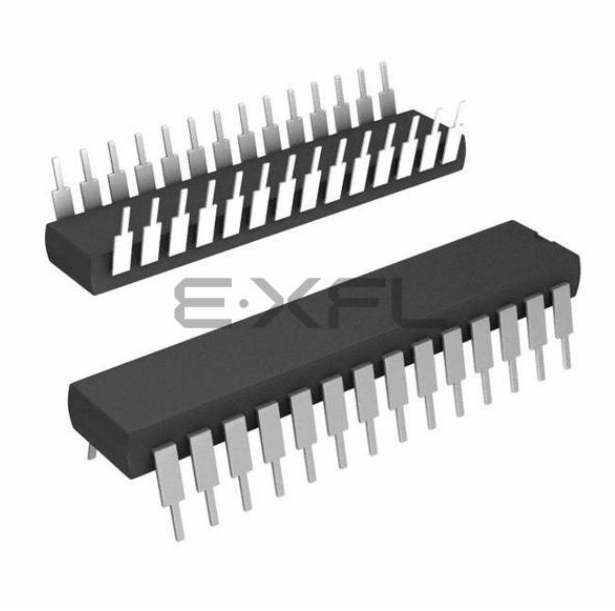


Welcome to E-XFL.COM

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	64KB (64K x 8)
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	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga202-i-sp

TABLE 4-5: INTERRUPT CONTROLLER REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	HLVDIP<2:0>			0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP<2:0>			—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	—	—	—	—	I2C2BCIP2	I2C2BCIP1	I2C2BCIP0	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0	4044
IPC22	00D0	—	SPI3TXIP2	SPI3TXIP1	SPI3TXIP0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC26	00D8	—	—	—	—	—	FSTIP<2:0>			—	—	—	—	—	—	—	—	0400
IPC29	00DE	—	—	—	—	—	—	—	—	—	JTAGIP<2:0>			—	—	—	—	0040
INTTREG	00E0	CPUIRQ	r	VHOLD	—	ILR3	ILR2	ILR1	ILR0	VECNUM7	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

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

TABLE 4-9: I²C™ 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
I2C1RCV	02DA	—	—	—	—	—	—	—	—	I2C1 Receive Register								0000
I2C1TRN	02DC	—	—	—	—	—	—	—	—	I2C1 Transmit Register								00FF
I2C1BRG	02DE	—	—	—	—	Baud Rate Generator Register												0000
I2C1CONL	02E0	I2CEN	—	I2CSIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1CONH	02E2	—	—	—	—	—	—	—	—	—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
I2C1STAT	02E4	ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C1ADD	02E6	—	—	—	—	—	—	I2C1 Address Register										0000
I2C1MSK	02E8	—	—	—	—	—	—	I2C1 Address Mask Register										0000
I2C2RCV	02EA	—	—	—	—	—	—	—	—	I2C2 Receive Register								0000
I2C2TRN	02EC	—	—	—	—	—	—	—	—	I2C2 Transmit Register								00FF
I2C2BRG	02EE	—	—	—	—	Baud Rate Generator Register												0000
I2C2CONL	02F0	I2CEN	—	I2CSIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2CONH	02F2	—	—	—	—	—	—	—	—	—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
I2C2STAT	02F4	ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C2ADD	02F6	—	—	—	—	—	—	I2C2 Address Register										0000
I2C2MSK	02F8	—	—	—	—	—	—	I2C2 Address Mask Register										0000

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

TABLE 4-18: A/D CONVERTER 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
ADC1BUF0	0200	A/D Data Buffer 0/Threshold for Channel 0																xxxx
ADC1BUF1	0202	A/D Data Buffer 1/Threshold for Channel 1																xxxx
ADC1BUF2	0204	A/D Data Buffer 2/Threshold for Channel 2																xxxx
ADC1BUF3	0206	A/D Data Buffer 3/Threshold for Channel 3																xxxx
ADC1BUF4	0208	A/D Data Buffer 4/Threshold for Channel 4																xxxx
ADC1BUF5	020A	A/D Data Buffer 5/Threshold for Channel 5																xxxx
ADC1BUF6	020C	A/D Data Buffer 6/Threshold for Channel 6																xxxx
ADC1BUF7	020E	A/D Data Buffer 7/Threshold for Channel 7																xxxx
ADC1BUF8	0210	A/D Data Buffer 8/Threshold for Channel 8/Threshold for Channel 0 in Windowed Compare mode																xxxx
ADC1BUF9	0212	A/D Data Buffer 9/Threshold for Channel 9/Threshold for Channel 1 in Windowed Compare mode																xxxx
ADC1BUF10	0214	A/D Data Buffer 10/Threshold for Channel 10/Threshold for Channel 2 in Windowed Compare mode(1)																xxxx
ADC1BUF11	0216	A/D Data Buffer 11/Threshold for Channel 11/Threshold for Channel 3 in Windowed Compare mode(1)																xxxx
ADC1BUF12	0218	A/D Data Buffer 12/Threshold for Channel 12/Threshold for Channel 4 in Windowed Compare mode(1)																xxxx
ADC1BUF13	021A	A/D Data Buffer 13																xxxx
ADC1BUF14	021C	A/D Data Buffer 14																xxxx
ADC1BUF15	021E	A/D Data Buffer 15																xxxx
AD1CON1	0220	ADON	—	ADSIDL	DMABM	DMAEN	MODE12	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE	0000
AD1CON2	0222	PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0224	ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0228	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	022A	CSS<31:27>					—	—	—	—	—	—	—	—	—	—	—	0000
AD1CSSL	022C	—	CSS<14:0>(1)															0000
AD1CON4	022E	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>			0000
AD1CON5	0230	ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0	—	—	—	—	WM1	WM0	CM1	CM0	0000
AD1CHITL	0234	—	—	—	CHH<12:0>(1)													0000
AD1CTMENL	0238	—	—	—	CTMEN<12:0>(1)													0000
AD1DMBUF	023A	A/D Conversion Data Buffer (Extended Buffer mode)																xxxx

Legend: — = unimplemented, read as '0'; x = unknown value on Reset. Reset values are shown in hexadecimal.

Note 1: The CSS<12:10>, CHH<12:10> and CTMEN<12:10> bits are unimplemented in 28-pin devices, read as '0'.

5.0 DIRECT MEMORY ACCESS CONTROLLER (DMA)

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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Direct Memory Access Controller (DMA)"** (DS39742). The information in this data sheet supersedes the information in the FRM.

The Direct Memory Access (DMA) Controller is designed to service high data throughput peripherals operating on the SFR bus, allowing them to access data memory directly and alleviating the need for CPU intensive management. By allowing these data-intensive peripherals to share their own data path, the main data bus is also deloaded, resulting in additional power savings.

The DMA Controller functions both as a peripheral and a direct extension of the CPU. It is located on the microcontroller data bus between the CPU and DMA-enabled peripherals, with direct access to SRAM. This partitions the SFR bus into two buses, allowing the DMA Controller access to the DMA capable peripherals located on the new DMA SFR bus. The controller serves as a master device on the DMA SFR bus, controlling data flow from DMA capable peripherals.

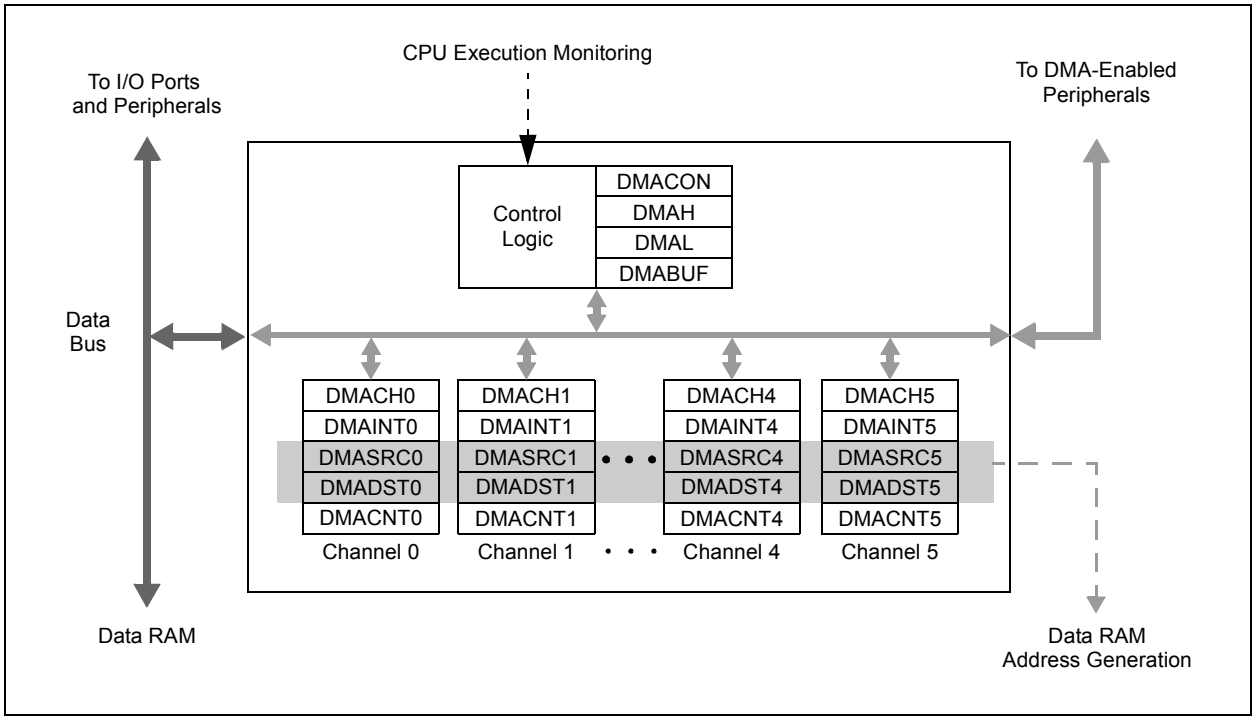
The controller also monitors CPU instruction processing directly, allowing it to be aware of when the CPU requires access to peripherals on the DMA bus and automatically relinquishing control to the CPU as needed. This increases the effective bandwidth for handling data without DMA operations causing a processor stall. This makes the controller essentially transparent to the user.

The DMA Controller has these features:

- Six multiple independent and independently programmable channels
- Concurrent operation with the CPU (no DMA caused Wait states)
- DMA bus arbitration
- Five Programmable Address modes
- Four Programmable Transfer modes
- Four Flexible Internal Data Transfer modes
- Byte or word support for data transfer
- 16-Bit Source and Destination Address register for each channel, dynamically updated and reloadable
- 16-Bit Transaction Count register, dynamically updated and reloadable
- Upper and Lower Address Limit registers
- Counter half-full level interrupt
- Software triggered transfer
- Null Write mode for symmetric buffer operations

A simplified block diagram of the DMA Controller is shown in Figure 5-1.

FIGURE 5-1: DMA FUNCTIONAL BLOCK DIAGRAM



6.0 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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Program Memory"** (DS39715). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ128GA204 family of devices contains internal Flash program memory for storing and executing application code. The program memory is readable, writable and erasable. The Flash memory can be programmed in four ways:

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

ICSP allows a PIC24FJ128GA204 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 (named PGECx and PGEDx, respectively), and three other lines for power (V_{DD}), ground (V_{SS}) 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.

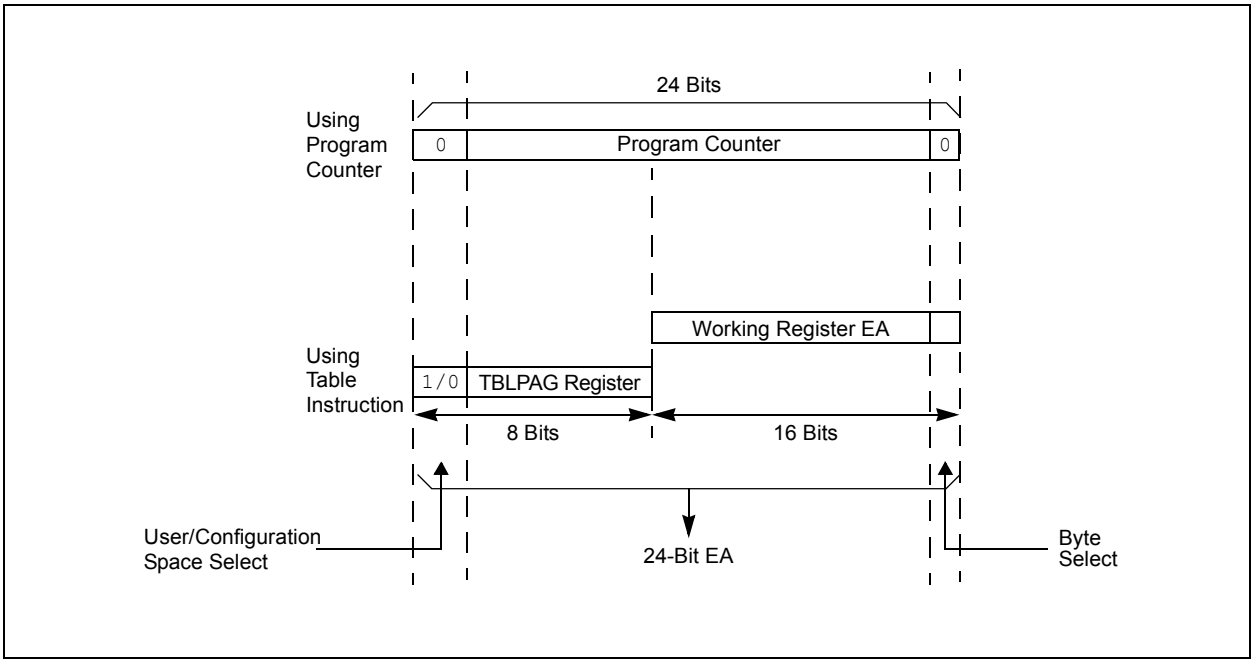
6.1 Table Instructions and Flash Programming

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 6-1.

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 6-1: ADDRESSING FOR TABLE REGISTERS



8.0 INTERRUPT CONTROLLER

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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Interrupts"** (DS70000600). The information in this data sheet supersedes the information in the FRM.

The PIC24F interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the PIC24F CPU. It has the following features:

- Up to 8 processor exceptions and software traps
- Seven user-selectable priority levels
- Interrupt Vector Table (IVT) with up to 118 vectors
- Unique vector for each interrupt or exception source
- Fixed priority within a specified user priority level
- Alternate Interrupt Vector Table (AIVT) for debug support
- Fixed interrupt entry and return latencies

8.1 Interrupt Vector Table

The Interrupt Vector Table (IVT) is shown in Figure 8-1. The IVT resides in program memory, starting at location, 000004h. The IVT contains 126 vectors, consisting of 8 non-maskable trap vectors, plus up to 118 sources of interrupt. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

Interrupt vectors are prioritized in terms of their natural priority; this is linked to their position in the vector table. All other things being equal, lower addresses have a higher natural priority. For example, the interrupt associated with Vector 0 will take priority over interrupts at any other vector address.

PIC24FJ128GA204 family devices implement non-maskable traps and unique interrupts. These are summarized in Table 8-1 and Table 8-2.

8.1.1 ALTERNATE INTERRUPT VECTOR TABLE

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 8-1. The ALTIVT (INTCON2<15>) control bit provides access to the AIVT. If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors.

The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not needed, the AIVT should be programmed with the same addresses used in the IVT.

8.2 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. The PIC24F devices clear their registers in response to a Reset, which forces the PC to zero. The microcontroller then begins program execution at location, 000000h. The user programs a `GOTO` instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

Note: Any unimplemented or unused vector locations in the IVT and AIVT should be programmed with the address of a default interrupt handler routine that contains a `RESET` instruction.

A recommended code sequence for a clock switch includes the following:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8> in two back-to-back instructions.
3. Write the new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0> in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock- sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of the failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 9-1.

EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV         #OSCCONH, w1
MOV         #0x78, w2
MOV         #0x9A, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Set new oscillator selection
MOV.b       WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV         #OSCCONL, w1
MOV         #0x46, w2
MOV         #0x57, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Start oscillator switch operation
BSET        OSCCON,#0
```

9.5 FRC Self-Tuning

PIC24FJ128GA204 family devices include an automatic mechanism to calibrate the FRC during run time. This system uses clock recovery from a source of known accuracy to maintain the FRC within a very narrow margin of its nominal 8 MHz frequency. This allows for a frequency accuracy that exceeds 0.25%, which is well within the requirements.

The self-tune system is controlled by the bits in the upper half of the OSCTUN register. Setting the STEN bit (OSCTUN<15>) enables the system, causing it to recover a calibration clock from a source selected by the STSRC bit (OSCTUN<12>). When STSRC = 0, the system uses the crystal controlled SOSC for its calibration source. Regardless of the source, the system uses the TUN<5:0> bits (OSCTUN<5:0>) to change the FRC's frequency. Frequency monitoring and adjustment is dynamic, occurring continuously during run time. While the system is active, the TUNx bits cannot be written to by software.

Note:	If the SOSC is to be used as the clock recovery source (STSRC = 0), the SOSC must always be enabled.

The self-tune system can generate a hardware interrupt, FSTIF. The interrupt can result from a drift of the FRC from the reference by greater than 0.2% in either direction or whenever the frequency deviation is beyond the ability of the TUNx bits to correct (i.e., greater than 1.5%). The STLOCK and STOR status bits (OSCTUN<11,9>) are used to indicate these conditions.

The STLPOL and STORPOL bits (OSCTUN<10,8>) configure the FSTIF interrupt to occur in the presence or the absence of the conditions. It is the user's responsibility to monitor both the STLOCK and STOR bits to determine the exact cause of the interrupt.

Note:	The STLPOL and STORPOL bits should be ignored when the self-tune system is disabled (STEN = 0).

TABLE 10-2: EXITING POWER-SAVING MODES

Mode	Exit Conditions								Code Execution Resumes
	Interrupts		Resets			RTCC Alarm	WDT	VDD Restore(2)	
	All	INT0	All	POR	MCLR				
Idle	Y	Y	Y	Y	Y	Y	Y	N/A	Next instruction
Sleep (all modes)	Y	Y	Y	Y	Y	Y	Y	N/A	
Deep Sleep	N	Y	N	Y	Y	Y	Y(1)	N/A	Reset vector
VBAT	N	N	N	N	N	N	N	Y	Reset vector

Note 1: Deep Sleep WDT.

2: A POR or POR like Reset results whenever VDD is removed and restored in any mode except for Retention Deep Sleep mode.

10.1.1 INSTRUCTION-BASED POWER-SAVING MODES

Three of the power-saving modes are entered through the execution of the PWRSAV instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. Deep Sleep mode stops clock operation, code execution, and all peripherals, except RTCC and DSWDT. It also freezes I/O states and removes power to Flash memory, and may remove power to SRAM.

The assembly syntax of the PWRSAV instruction is shown in Example 10-1. Sleep and Idle modes are entered directly with a single assembler command. Deep Sleep requires an additional sequence to unlock and enable the entry into Deep Sleep, which is described in **Section 10.4.1 "Entering Deep Sleep Mode"**.

Note: SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

The features enabled with the low-voltage/retention regulator results in some changes to the way that Sleep and Deep Sleep modes behave. See **Section 10.3 "Sleep Mode"** and **Section 10.4 "Deep Sleep Mode"** for additional information.

10.1.1.1 Interrupts Coincident with Power Save Instructions

Any interrupt that coincides with the execution of a PWRSAV instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

For Deep Sleep mode, interrupts that coincide with the execution of the PWRSAV instruction may be lost. If the low-voltage/retention regulator is not enabled, the microcontroller resets on leaving Deep Sleep and the interrupt will be lost.

Interrupts that occur during the Deep Sleep unlock sequence will interrupt the mandatory five-instruction cycle sequence timing and cause a failure to enter Deep Sleep. For this reason, it is recommended to disable all interrupts during the Deep Sleep unlock sequence.

EXAMPLE 10-1: PWRSAV INSTRUCTION SYNTAX

```
// Syntax to enter Sleep mode:
PWRSAV     #SLEEP_MODE        ; Put the device into SLEEP mode
//
//Synatx to enter Idle mode:
PWRSAV     #IDLE_MODE         ; Put the device into IDLE mode
//
// Syntax to enter Deep Sleep mode:
// First use the unlock sequence to set the DSEN bit (see Example 10-2)
BSET       DSCON, #DSEN       ; Enable Deep Sleep
BSET       DSCON, #DSEN       ; Enable Deep Sleep(repeat the command)
PWRSAV     #SLEEP_MODE        ; Put the device into Deep SLEEP mode
```

NOTES:

REGISTER 16-5: SPIxSTATH: SPIx STATUS REGISTER HIGH

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	RXELM5(3)	RXELM4(2)	RXELM3(1)	RXELM2	RXELM1	RXELM0
bit 15							bit 8

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	TXELM5(3)	TXELM4(2)	TXELM3(1)	TXELM2	TXELM1	TXELM0
bit 7							bit 0

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

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RXELM<5:0>:** Receive Buffer Element Count bits (valid in Enhanced Buffer mode)(1,2,3)

bit 7-6 **Unimplemented:** Read as '0'

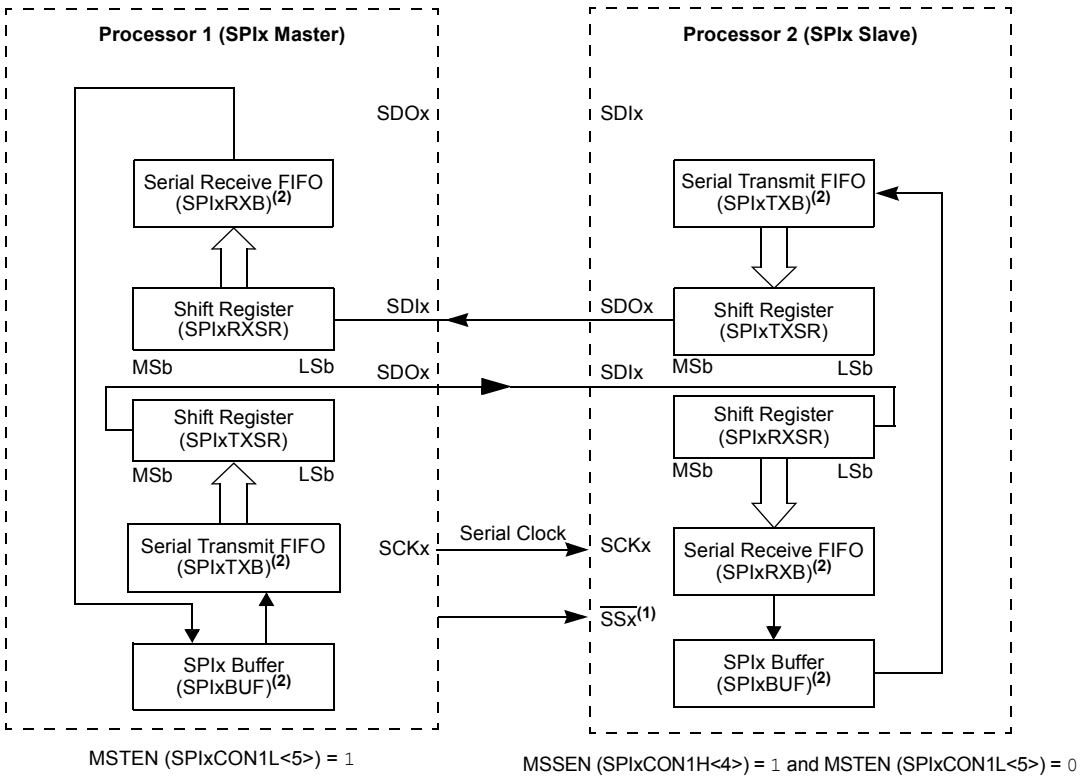
bit 5-0 **TXELM<5:0>:** Transmit Buffer Element Count bits (valid in Enhanced Buffer mode)(1,2,3)

Note 1: RXELM3 and TXELM3 bits are only present when FIFODEPTH = 8 or higher.

2: RXELM4 and TXELM4 bits are only present when FIFODEPTH = 16 or higher.

3: RXELM5 and TXELM5 bits are only present when FIFODEPTH = 32.

FIGURE 16-4: SPIx MASTER/SLAVE CONNECTION (ENHANCED BUFFER MODES)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.

2: User must write transmit data to read the received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory-mapped to SPIxBUF.

FIGURE 16-5: SPIx MASTER, FRAME MASTER CONNECTION DIAGRAM

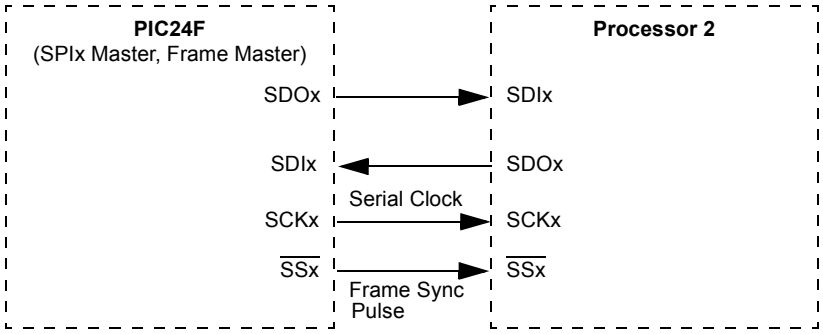
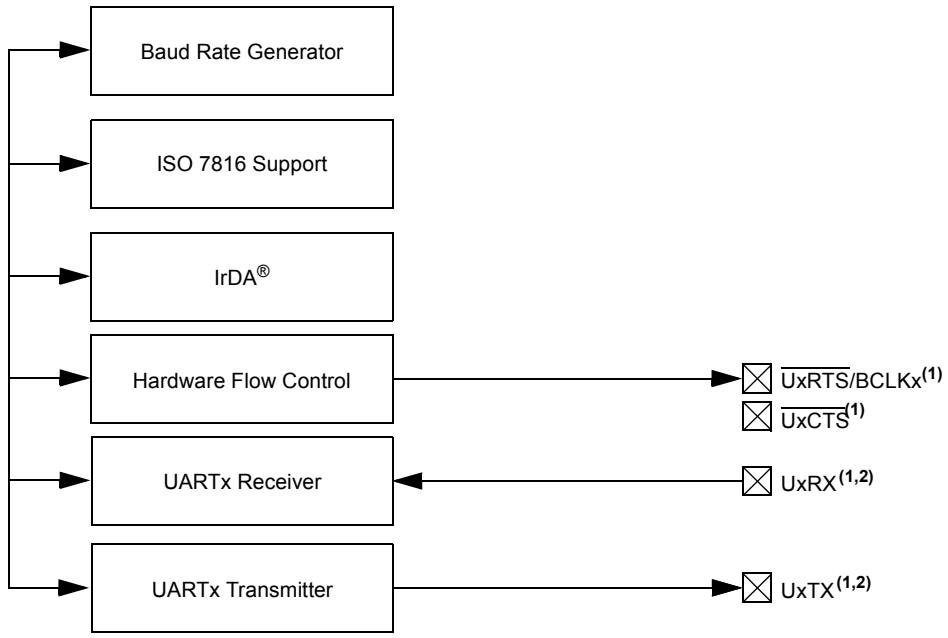


FIGURE 18-1: UARTx SIMPLIFIED BLOCK DIAGRAM



Note 1: The UARTx inputs and outputs must all be assigned to available RPn/RPIn pins before use. See **Section 11.4 "Peripheral Pin Select (PPS)"** for more information.

2: The UxTX and UxRX pins need to be shorted to be used for the Smart Card interface; this should be taken care of by the user.

19.0 DATA SIGNAL MODULATOR (DSM)

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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Data Signal Modulator (DSM)"** (DS39744). The information in this data sheet supersedes the information in the FRM.

The Data Signal Modulator (DSM) allows the user to mix a digital data stream (the "modulator signal") with a carrier signal to produce a modulated output. Both the carrier and the modulator signals are supplied to the DSM module, either internally from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical AND operation of both the carrier and modulator signals, and then it is provided to the MDOUT pin. Using this method, the DSM can generate the following types of key modulation schemes:

- Frequency Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Figure 19-1 shows a simplified block diagram of the Data Signal Modulator peripheral.

FIGURE 19-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR

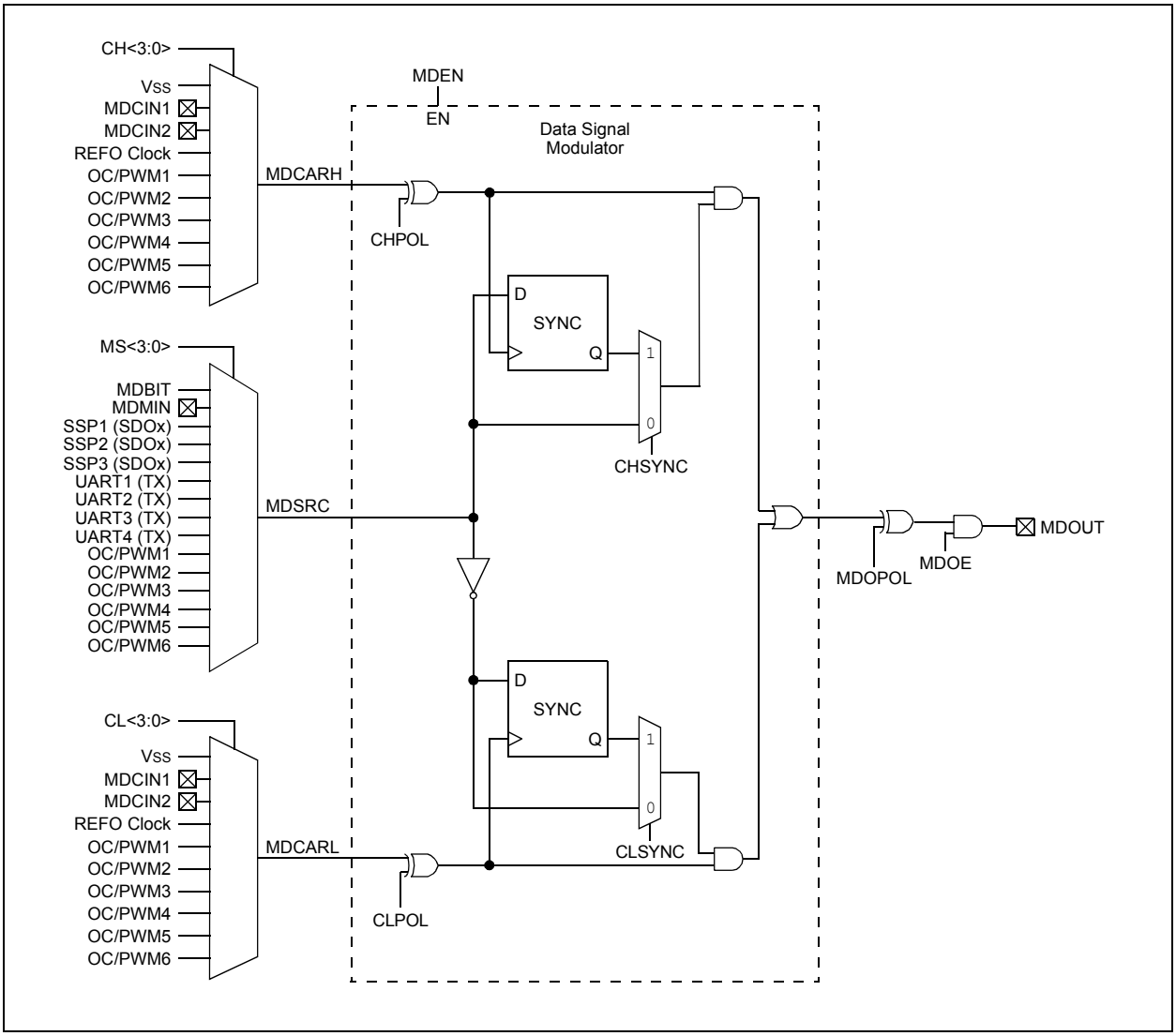


TABLE 20-1: MEMORY ADDRESSABLE IN DIFFERENT MODES

Data Port Size	PMA<9:8>	PMA<7:0>	PMD<7:4>	PMD<3:0>	Accessible memory
Demultiplexed Address (ADRMUX<1:0> = 00)					
8-Bit (PTSZ<1:0> = 00)	Addr<9:8>	Addr<7:0>	Data		1K
4-Bit (PTSZ<1:0> = 01)	Addr<9:8>	Addr<7:0>	—	Data	1K
1 Address Phase (ADRMUX<1:0> = 01)					
8-Bit (PTSZ<1:0> = 00)	—	PMALL	Addr<7:0> Data		1K
4-Bit (PTSZ<1:0> = 01)	Addr<9:8>	PMALL	Addr<7:4>	Addr<3:0>	1K
			—	Data (1)	
2 Address Phases (ADRMUX<1:0> = 10)					
8-Bit (PTSZ<1:0> = 00)	—	PMALL	Addr<7:0>		64K
		PMALH	Addr<15:8>		
		—	Data		
4-Bit (PTSZ<1:0> = 01)	Addr<9:8>	PMALL	Addr<3:0>		1K
		PMALH	Addr<7:4>		
		—	Data		
3 Address Phases (ADRMUX<1:0> = 11)					
8-Bit (PTSZ<1:0> = 00)	—	PMALL	Addr<7:0>		2 Mbytes
		PMALH	Addr<15:8>		
		PMALU	Addr<22:16>		
		—	Data		
4-Bit (PTSZ<1:0> = 01)	Addr<13:12>	PMALL	Addr<3:0>		16K
		PMALH	Addr<7:4>		
		PMALU	Addr<11:8>		
		—	Data		

REGISTER 20-1: PMCON1: EPMP CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0
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 **PMPEN:** EPMP Enable bit
1 = EPMP is enabled
0 = EPMP is disabled

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

bit 13 **PSIDL:** EPMP Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-11 **ADRMUX<1:0>:** Address/Data Multiplexing Selection bits
11 = Lower address bits are multiplexed with data bits using 3 address phases
10 = Lower address bits are multiplexed with data bits using 2 address phases
01 = Lower address bits are multiplexed with data bits using 1 address phase
00 = Address and data appear on separate pins

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

bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master mode
10 = Enhanced PSP: Pins used are PMRD, PMWR, PMCS, PMD<7:0> and PMA<1:0>
01 = Buffered PSP: Pins used are PMRD, PMWR, PMCS and PMD<7:0>
00 = Legacy Parallel Slave Port: Pins used are PMRD, PMWR, PMCS and PMD<7:0>

bit 7-6 **CSF<1:0>:** Chip Select Function bits
11 = Reserved
10 = PMA<14> is used for Chip Select 1
01 = Reserved
00 = PMCS1 is used for Chip Select 1

bit 5 **ALP:** Address Latch Polarity bit
1 = Active-high (PMALL, PMALH and PMALU)
0 = Active-low ($\overline{\text{PMALL}}$, $\overline{\text{PMALH}}$ and $\overline{\text{PMALU}}$)

bit 4 **ALMODE:** Address Latch Strobe Mode bit
1 = Enables "smart" address strobes (each address phase is only present if the current access would cause a different address in the latch than the previous address)
0 = Disables "smart" address strobes

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

bit 2 **BUSKEEP:** Bus Keeper bit
1 = Data bus keeps its last value when not actively being driven
0 = Data bus is in a high-impedance state when not actively being driven

bit 1-0 **IRQM<1:0>:** Interrupt Request Mode bits
11 = Interrupt is generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode), or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = Reserved
01 = Interrupt is generated at the end of a read/write cycle
00 = No interrupt is generated

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.

23.0 32-BIT PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

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 *"dsPIC33/PIC24 Family Reference Manual"*, **"32-Bit Programmable Cyclic Redundancy Check (CRC)"** (DS30009729). The information in this data sheet supersedes the information in the FRM.

The 32-bit programmable CRC generator provides a hardware implemented method of quickly generating checksums for various networking and security applications. It offers the following features:

- User-programmable CRC polynomial equation, up to 32 bits
- Programmable shift direction (little or big-endian)
- Independent data and polynomial lengths
- Configurable interrupt output
- Data FIFO

Figure 23-1 displays a simplified block diagram of the CRC generator. A simple version of the CRC shift engine is displayed in Figure 23-2.

FIGURE 23-1: CRC MODULE BLOCK DIAGRAM

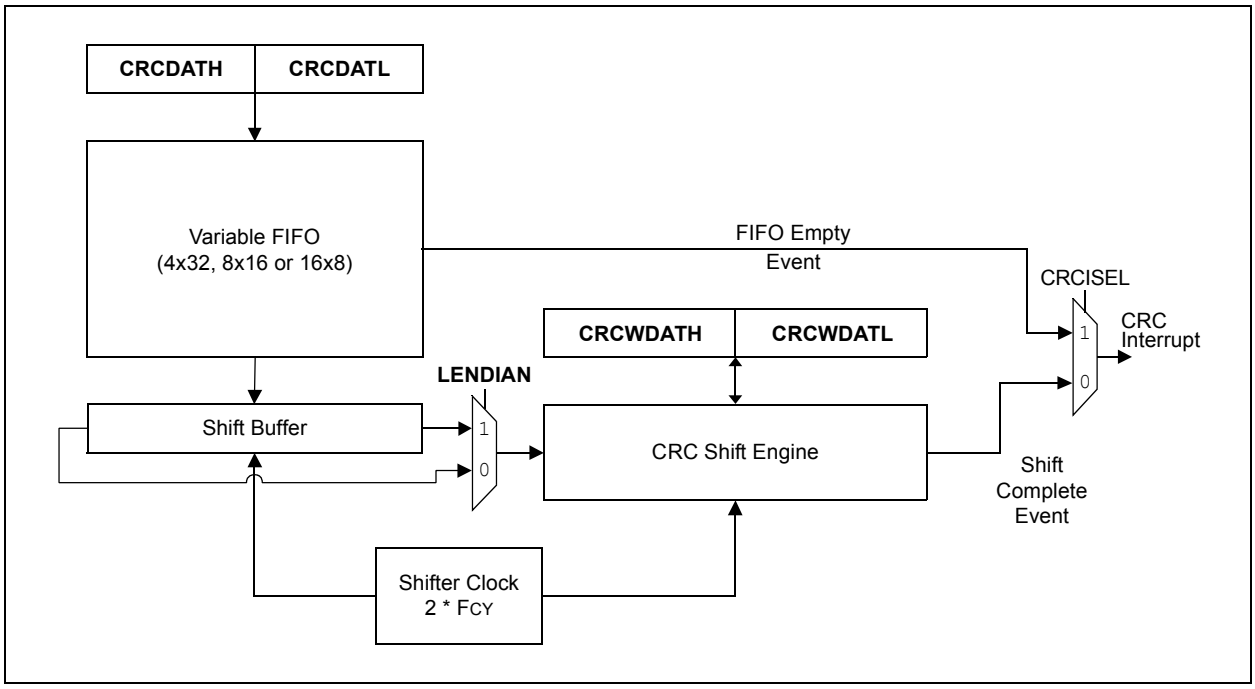
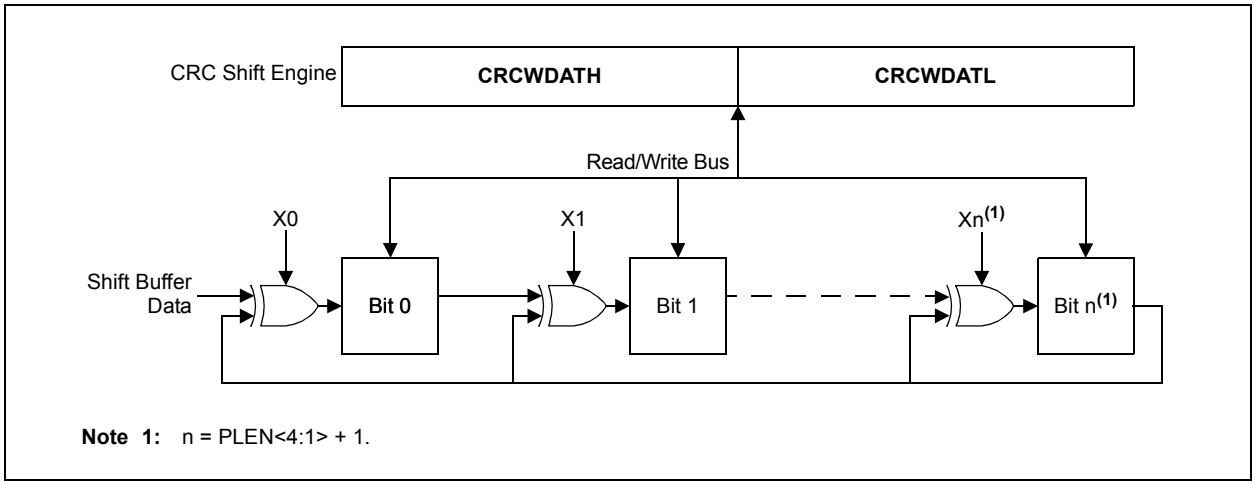


FIGURE 23-2: CRC SHIFT ENGINE DETAIL



Note 1: n = PLEN<4:1> + 1.

27.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's Internal Capacitor (CAD) and a precision resistor for current calibration. Figure 27-2 displays the external connections used for time measurements, and how the CTMU and A/D modules are related in this application. This example also shows both edge events coming from the external CTEDx pins, but other configurations using internal edge sources are possible.

27.3 Pulse Generation and Delay

The CTMU module can also generate an output pulse with edges that are not synchronous with the device's system clock. More specifically, it can generate a pulse with a programmable delay from an edge event input to the module.

When the module is configured for pulse generation delay by setting the TGEN bit (CTMUCON1<12>), the internal current source is connected to the B input of Comparator 2. A Capacitor (CDELAY) is connected to the Comparator 2 pin, C2INB, and the Comparator Voltage Reference, CVREF, is connected to C2INA. CVREF is then configured for a specific trip point. The module begins to charge CDELAY when an edge event is detected. When CDELAY charges above the CVREF trip point, a pulse is output on CTPLS. The length of the pulse delay is determined by the value of CDELAY and the CVREF trip point.

Figure 27-3 illustrates the external connections for pulse generation, as well as the relationship of the different analog modules required. While CTED1 is shown as the input pulse source, other options are available. A detailed discussion on pulse generation with the CTMU module is provided in the *"dsPIC33/PIC24 Family Reference Manual"*.

FIGURE 27-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT

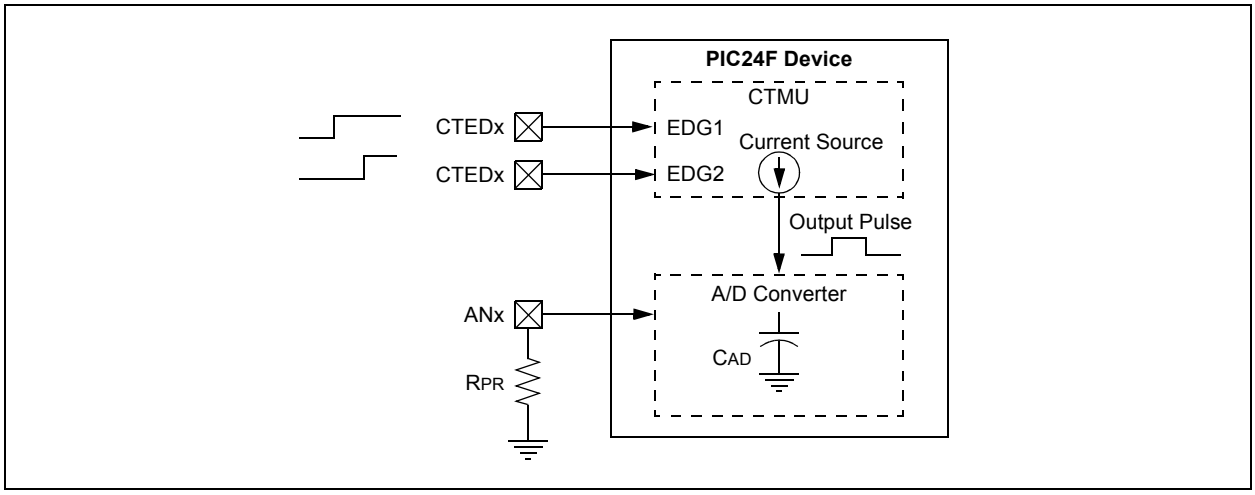


FIGURE 27-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION

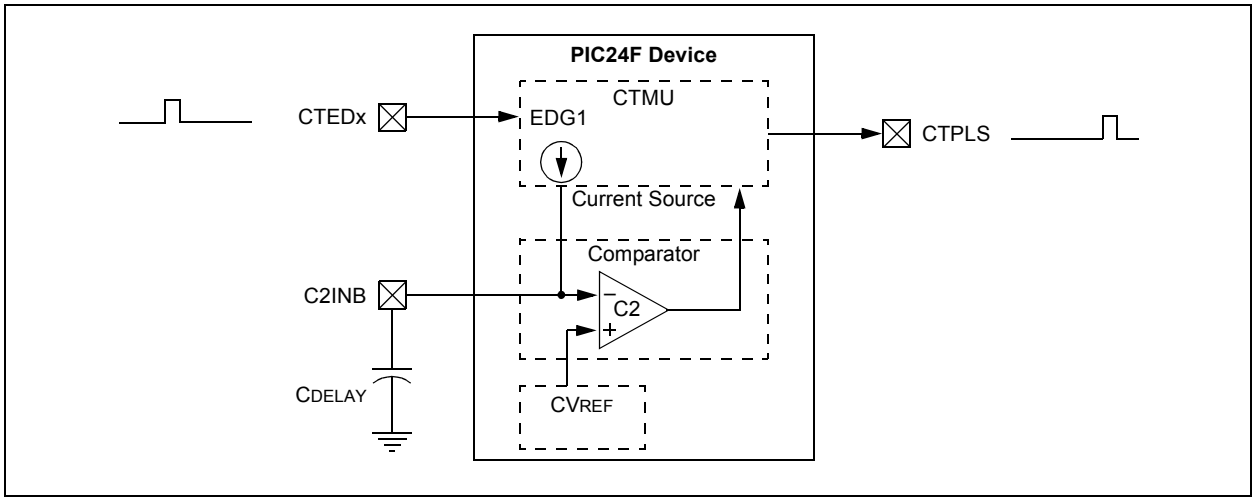


TABLE 32-9: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC 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**
	VOL	**Output Low Voltage**					
DO10		I/O Ports	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2V
DO16		OSCO/CLKO	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2V
	VOH	**Output High Voltage**					
DO20		I/O Ports	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA, VDD = 2V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2V
DO26		OSCO/CLKO	2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.4	—	—	V	IOH = -1.0 mA, VDD = 2V

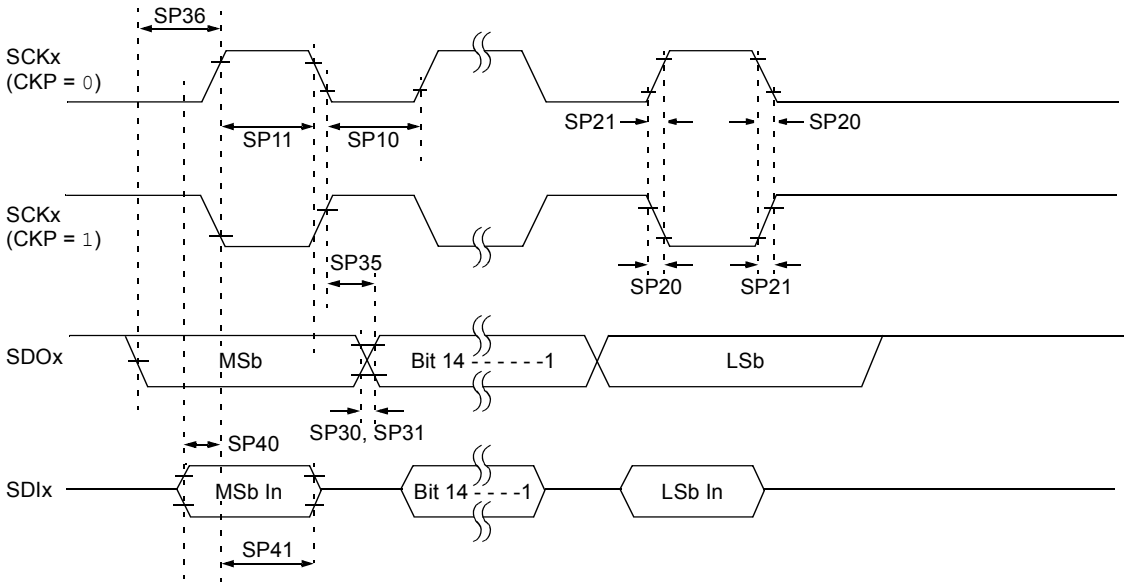
Note 1: Data in the "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 32-10: DC CHARACTERISTICS: PROGRAM MEMORY

DC 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**
		Program Flash Memory					
D130	EP	Cell Endurance	20000	—	—	E/W	-40°C to +125°C
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum Operating Voltage
D132B		VDD for Self-Timed Write	VMIN	—	3.6	V	VMIN = Minimum Operating Voltage
D133A	TIW	Self-Timed Word Write Cycle Time	—	20	—	µs	
		Self-Timed Row Write Cycle Time	—	1.5	—	ms	
D133B	TIE	Self-Timed Page Erase Time	20	—	40	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	If no other specifications are violated
D135	IDDP	Supply Current during Programming	—	5	—	mA	
D136	VOTP	OTP Programming	3.1	—	3.6	V	
D137	TOTP	OTP Memory Write/Bit	—	500	—	µs	

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

FIGURE 32-14: SPIx MODULE MASTER MODE (CKE = 1) TIMING CHARACTERISTICS



Note: Refer to Figure 32-2 for load conditions.

TABLE 32-36: SPIx MODULE MASTER MODE (CKE = 1) 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(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
SP10	TscL	SCKx Output Low Time(3)	TCY/2	—	—	ns	
SP11	TscH	SCKx Output High Time(3)	TCY/2	—	—	ns	
SP20	TscF	SCKx Output Fall Time(4)	—	—	—	ns	See Parameter DO32
SP21	TscR	SCKx Output Rise Time(4)	—	—	—	ns	See Parameter DO31
SP30	TdoF	SDOx Data Output Fall Time(4)	—	—	—	ns	See Parameter DO32
SP31	TdoR	SDOx Data Output Rise Time(4)	—	—	—	ns	See Parameter DO31
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge	—	6	20	ns	
SP36	TdoV2sc, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	23	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	30	—	—	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

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

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.