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	70 MIPs
Connectivity	I²C, IrDA, LINbus, QEI, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, WDT
Number of I/O	21
Program Memory Size	32KB (10.7K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 16
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 6x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN-S (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24ep32mc202t-i-mm

2.7 Oscillator Value Conditions on Device Start-up

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to 3 MHz < F_{IN} < 5.5 MHz to comply with device PLL start-up conditions. This means that if the external oscillator frequency is outside this range, the application must start-up in the FRC mode first. The default PLL settings after a POR with an oscillator frequency outside this range will violate the device operating speed.

Once the device powers up, the application firmware can initialize the PLL SFRs, CLKDIV and PLLFBD, to a suitable value, and then perform a clock switch to the Oscillator + PLL clock source. Note that clock switching must be enabled in the device Configuration Word.

2.8 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state.

Alternatively, connect a 1k to 10k resistor between V_{SS} and unused pins, and drive the output to logic low.

2.9 Application Examples

- Induction heating
- Uninterruptable Power Supplies (UPS)
- DC/AC inverters
- Compressor motor control
- Washing machine 3-phase motor control
- BLDC motor control
- Automotive HVAC, cooling fans, fuel pumps
- Stepper motor control
- Audio and fluid sensor monitoring
- Camera lens focus and stability control
- Speech (playback, hands-free kits, answering machines, VoIP)
- Consumer audio
- Industrial and building control (security systems and access control)
- Barcode reading
- Networking: LAN switches, gateways
- Data storage device management
- Smart cards and smart card readers

Examples of typical application connections are shown in Figure 2-4 through Figure 2-8.

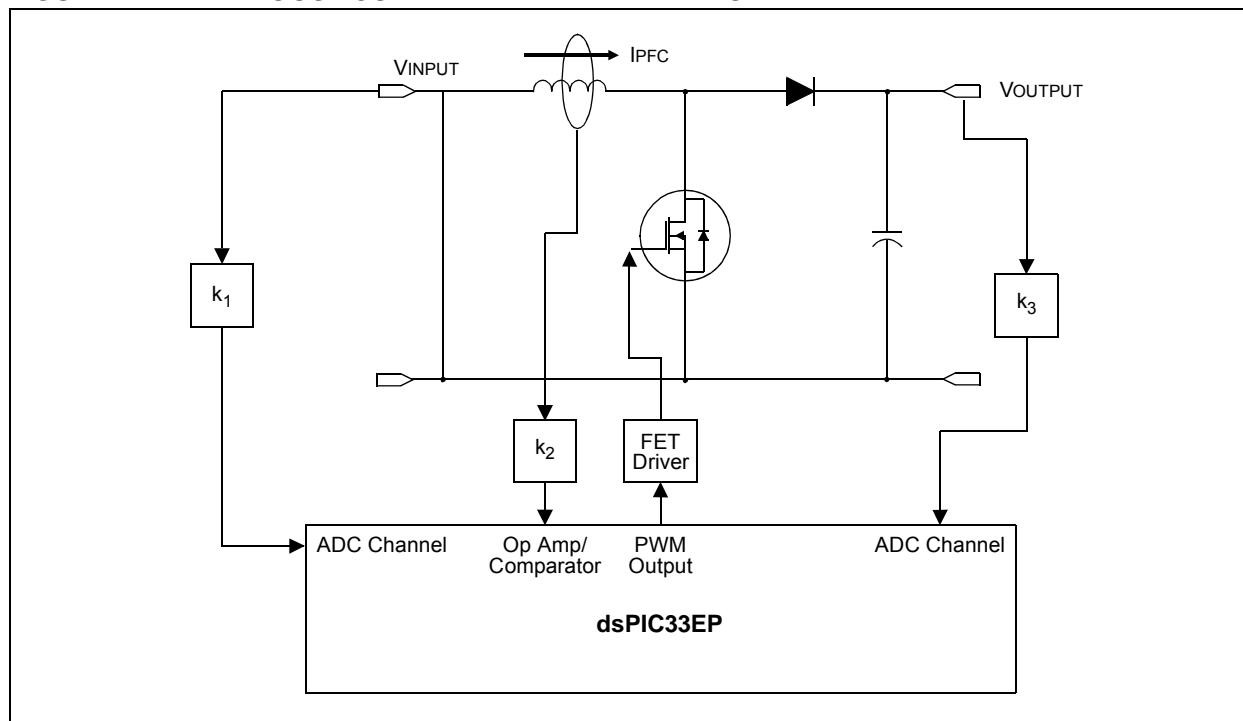### FIGURE 2-4: BOOST CONVERTER IMPLEMENTATION

TABLE 4-17: I2C1 AND I2C2 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	0200	—	—	—	—	—	—	—	—	I2C1 Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	I2C1 Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	Baud Rate Generator									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	—	I2C1 Address Register									0000
I2C1MSK	020C	—	—	—	—	—	—	—	I2C1 Address Mask									0000
I2C2RCV	0210	—	—	—	—	—	—	—	—	I2C2 Receive Register								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	I2C2 Transmit Register								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	Baud Rate Generator									0000
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	—	I2C2 Address Register									0000
I2C2MSK	021C	—	—	—	—	—	—	—	I2C2 Address Mask									0000

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

TABLE 4-18: UART1 AND UART2 REGISTER MAP

SFR 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
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN<1:0>		WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL<1:0>		STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	UART1 Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART1 Receive Register									0000
U1BRG	0228	Baud Rate Generator Prescaler																0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN<1:0>		WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL<1:0>		STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	UART2 Transmit Register									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	UART2 Receive Register									0000
U2BRG	0238	Baud Rate Generator Prescaler																0000

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

TABLE 4-52: PORTG REGISTER MAP FOR PIC24EPXXXGP/MC206 AND dsPIC33EPXXXGP/MC206/506 DEVICES ONLY

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
TRISG	0E60	—	—	—	—	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	—	—	—	—	03C0
PORTG	0E62	—	—	—	—	—	—	RG9	RG8	RG7	RG6	—	—	—	—	—	—	xxxx
LATG	0E64	—	—	—	—	—	—	LATG9	LATG8	LATG7	LATG6	—	—	—	—	—	—	xxxx
ODCG	0E66	—	—	—	—	—	—	ODCG9	ODCG8	ODCG7	ODCG6	—	—	—	—	—	—	0000
CNENG	0E68	—	—	—	—	—	—	CNIEG9	CNIEG8	CNIEG7	CNIEG6	—	—	—	—	—	—	0000
CNPUG	0E6A	—	—	—	—	—	—	CNPUG9	CNPUG8	CNPUG7	CNPUG6	—	—	—	—	—	—	0000
CNPDG	0E6C	—	—	—	—	—	—	CNPDG9	CNPDG8	CNPDG7	CNPDG6	—	—	—	—	—	—	0000

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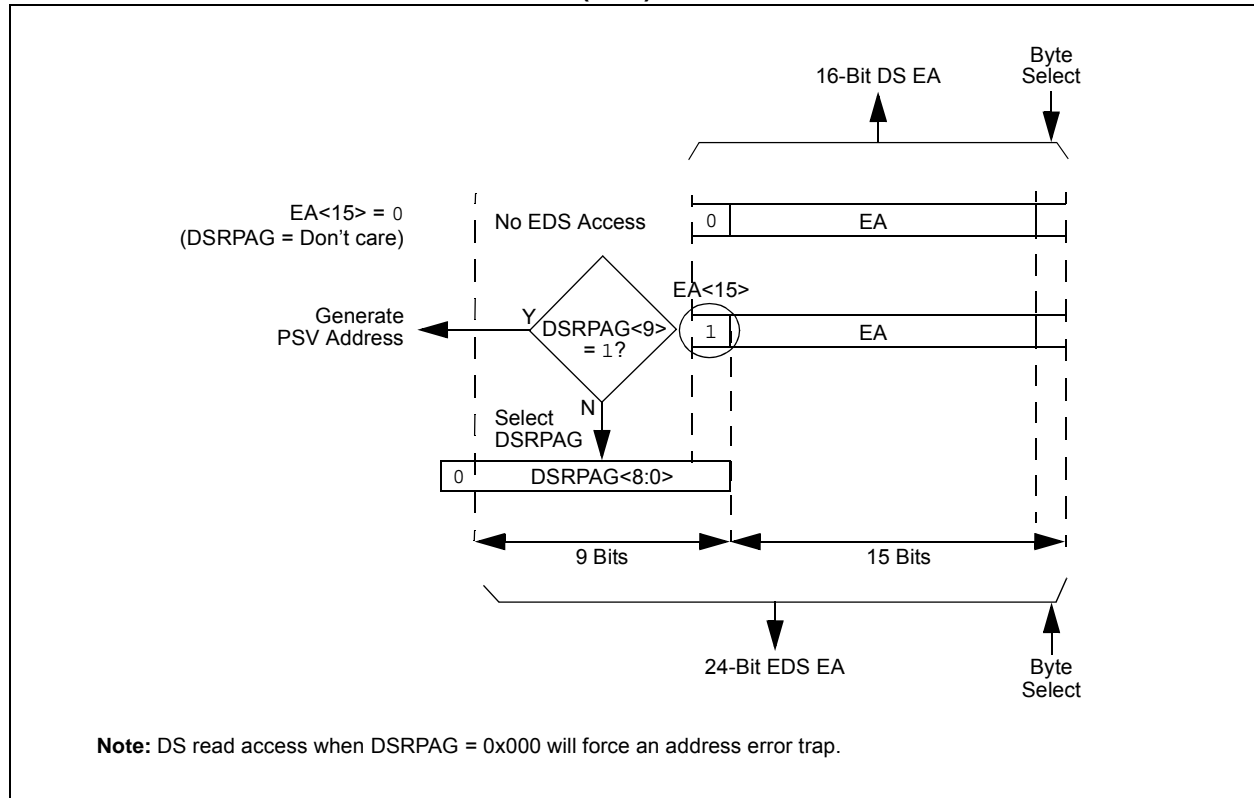
4.4.1   PAGED MEMORY SCHEME

The dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/ 50X and PIC24EPXXXGP/MC20X architecture extends the available Data Space through a paging scheme, which allows the available Data Space to be accessed using `MOV` instructions in a linear fashion for pre-modified and post-modified Effective Addresses (EA). The upper half of the base Data Space address is used in conjunction with the Data Space Page registers, the 10-bit Read Page register (DSRPAG) or the 9-bit Write Page register (DSWPAG), to form an Extended Data Space (EDS) address or Program Space Visibility (PSV) address. The Data Space Page registers are located in the SFR space.

Construction of the EDS address is shown in Example 4-1. When DSRPAG<9> = `0` and the base address bit, EA<15> = `1`, the DSRPAG<8:0> bits are concatenated onto EA<14:0> to form the 24-bit EDS read address. Similarly, when base address bit, EA<15> = `1`, DSWPAG<8:0> are concatenated onto EA<14:0> to form the 24-bit EDS write address.

EXAMPLE 4-1:   EXTENDED DATA SPACE (EDS) READ ADDRESS GENERATION

Note: DS read access when DSRPAG = 0x000 will force an address error trap.

EXAMPLE 4-3:   PAGED DATA MEMORY SPACE

| Local Data Space | EDS (DSRPAG<9:0>/DSWPAG<8:0>) | Program Space (Instruction & Data) | Table Address Space (TBLPAG<7:0>) |

DS_Addr<14:0>

0x0000
Page 0
Reserved
(Will produce an
address error trap)
0x7FFF

0x0000
EDS Page 0x001
(DSRPAG = 0x001)
(DSWPAG = 0x001)
0x7FFF

DS_Addr<15:0>

0x0000
SFR Registers
0x0FFF
0x1000

Up to 8-Kbyte
RAM**(1)**

0x2FFF
0x3000
0x7FFF
0x8000

32-Kbyte
EDS Window

0xFFFF

0x0000
EDS Page 0x1FF
(DSRPAG = 0x1FF)
(DSWPAG = 0x1FF)
0x7FFF

0x0000
EDS Page 0x200
(DSRPAG = 0x200)
No writes allowed
0x7FFF

0x0000
EDS Page 0x2FF
(DSRPAG = 0x2FF)
No writes allowed
0x7FFF

0x0000
EDS Page 0x300
(DSRPAG = 0x300)
No writes allowed
0x7FFF

0x0000
EDS Page 0x3FF
(DSRPAG = 0x3FF)
No writes allowed
0x7FFF

PSV
Program
Memory
(lsw)

PSV
Program
Memory
(MSB)

Program Memory
(lsw – <15:0>)
0x00_0000

0x7F_FFFF

Program Memory
(MSB – <23:16>)
0x00_0000

0x7F_FFFF

DS_Addr<15:0>

0x0000
(TBLPAG = 0x00)
lsw Using
TBLRDL/TBLWTL
MSB Using
TBLRDH/TBLWTH
0xFFFF

0x0000
(TBLPAG = 0x7F)
lsw Using
TBLRDL/TBLWTL
MSB Using
TBLRDH/TBLWTH
0xFFFF

Note   1:   For 64K Flash devices. RAM size and end location is dependent on device; see **Section 4.2 "Data Address Space"** for more information.

4.5 Instruction Addressing Modes

The addressing modes shown in Table 4-63 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the MAC class of instructions differ from those in the other instruction types.

4.5.1 FILE REGISTER INSTRUCTIONS

Most file register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (Near Data Space). Most file register instructions employ a working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same file register or WREG (with the exception of the MUL instruction), which writes the result to a register or register pair. The MOV instruction allows additional flexibility and can access the entire Data Space.

4.5.2 MCU INSTRUCTIONS

The three-operand MCU instructions are of the form:

Operand 3 = Operand 1 <function> Operand 2

where Operand 1 is always a working register (that is, the addressing mode can only be Register Direct), which is referred to as Wb. Operand 2 can be a W register fetched from data memory or a 5-bit literal. The result location can either be a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-Modified
- Register Indirect Pre-Modified
- 5-Bit or 10-Bit Literal

Note:	Not all instructions support all the addressing modes given above. Individual instructions can support different subsets of these addressing modes.

TABLE 4-63:   FUNDAMENTAL ADDRESSING MODES SUPPORTED

Addressing Mode	Description
File Register Direct	The address of the file register is specified explicitly.
Register Direct	The contents of a register are accessed directly.
Register Indirect	The contents of Wn form the Effective Address (EA).
Register Indirect Post-Modified	The contents of Wn form the EA. Wn is post-modified (incremented or decremented) by a constant value.
Register Indirect Pre-Modified	Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA.
Register Indirect with Register Offset (Register Indexed)	The sum of Wn and Wb forms the EA.
Register Indirect with Literal Offset	The sum of Wn and a literal forms the EA.

4.6.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the Effective Address (EA) calculation associated with any W register. Address boundaries check for addresses equal to:

• The upper boundary addresses for incrementing buffers
• The lower boundary addresses for decrementing buffers

It is important to realize that the address boundaries check for addresses less than, or greater than, the upper (for incrementing buffers) and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes can, therefore, jump beyond boundaries and still be adjusted correctly.

> **Note:** The modulo corrected Effective Address is written back to the register only when Pre-Modify or Post-Modify Addressing mode is used to compute the Effective Address. When an address offset (such as [W7 + W2]) is used, Modulo Addressing correction is performed but the contents of the register remain unchanged.

4.7 Bit-Reversed Addressing (dsPIC33EPXXXMC20X/50X and dsPIC33EPXXXGP50X Devices Only)

Bit-Reversed Addressing mode is intended to simplify data reordering for radix-2 FFT algorithms. It is supported by the X AGU for data writes only.

The modifier, which can be a constant value or register contents, is regarded as having its bit order reversed. The address source and destination are kept in normal order. Thus, the only operand requiring reversal is the modifier.

4.7.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

Bit-Reversed Addressing mode is enabled when all these conditions are met:

• BWMx bits (W register selection) in the MODCON register are any value other than '1111' (the stack cannot be accessed using Bit-Reversed Addressing)
• The BREN bit is set in the XBREV register
• The addressing mode used is Register Indirect with Pre-Increment or Post-Increment

If the length of a bit-reversed buffer is $M = 2^N$ bytes, the last 'N' bits of the data buffer start address must be zeros.

XBREV<14:0> is the Bit-Reversed Addressing modifier, or 'pivot point', which is typically a constant. In the case of an FFT computation, its value is equal to half of the FFT data buffer size.

> **Note:** All bit-reversed EA calculations assume word-sized data (LSb of every EA is always clear). The XBREVx value is scaled accordingly to generate compatible (byte) addresses.

When enabled, Bit-Reversed Addressing is executed only for Register Indirect with Pre-Increment or Post-Increment Addressing and word-sized data writes. It does not function for any other addressing mode or for byte-sized data and normal addresses are generated instead. When Bit-Reversed Addressing is active, the W Address Pointer is always added to the address modifier (XBREVx) and the offset associated with the Register Indirect Addressing mode is ignored. In addition, as word-sized data is a requirement, the LSb of the EA is ignored (and always clear).

> **Note:** Modulo Addressing and Bit-Reversed Addressing can be enabled simultaneously using the same W register, but Bit-Reversed Addressing operation will always take precedence for data writes when enabled.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV<15>) bit, a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the Bit-Reversed Pointer.

FIGURE 8-2: **DMA CONTROLLER BLOCK DIAGRAM**

Note: CPU and DMA address buses are not shown for clarity.
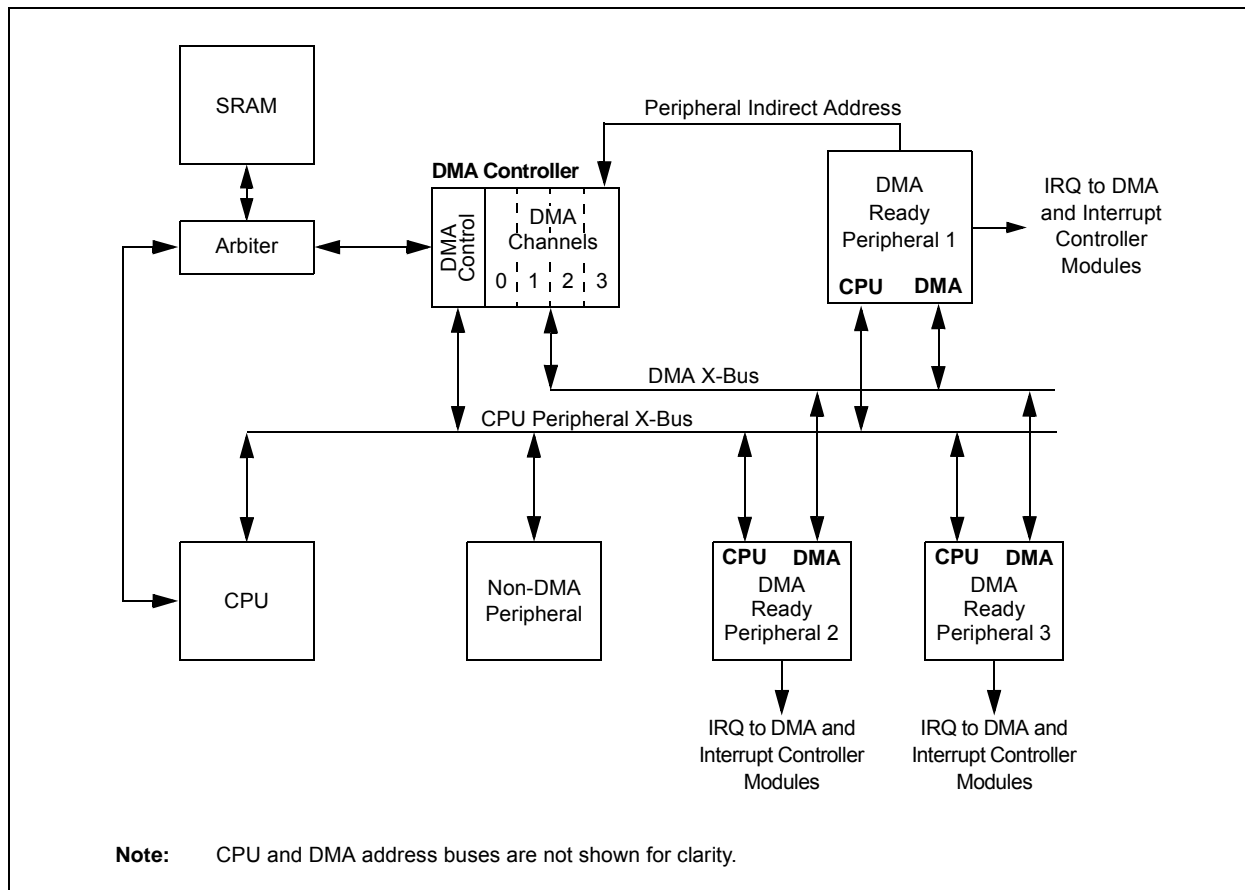
8.1 DMA Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page, which can be accessed using this link, contains the latest updates and additional information.

Note: In the event you are not able to access the product page using the link above, enter this URL in your browser: http://www.microchip.com/wwwproducts/ Devices.aspx?dDocName=en555464

8.1.1 KEY RESOURCES

- **Section 22. "Direct Memory Access (DMA)"** (DS70348) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

8.2 DMAC Registers

Each DMAC Channel x (where x = 0 through 3) contains the following registers:

- 16-Bit DMA Channel Control register (DMAxCON)
- 16-Bit DMA Channel IRQ Select register (DMAxREQ)
- 32-Bit DMA RAM Primary Start Address register (DMAxSTA)
- 32-Bit DMA RAM Secondary Start Address register (DMAxSTB)
- 16-Bit DMA Peripheral Address register (DMAxPAD)
- 14-Bit DMA Transfer Count register (DMAxCNT)

Additional status registers (DMAPWC, DMARQC, DMAPPS, DMALCA and DSADR) are common to all DMAC channels. These status registers provide information on write and request collisions, as well as on last address and channel access information.

The interrupt flags (DMAxIF) are located in an IFSx register in the interrupt controller. The corresponding interrupt enable control bits (DMAxIE) are located in an IECx register in the interrupt controller, and the corresponding interrupt priority control bits (DMAxIP) are located in an IPCx register in the interrupt controller.

REGISTER 9-2:   CLKDIV: CLOCK DIVISOR REGISTER (CONTINUED)

bit 4-0   **PLLPRE<4:0>:** PLL Phase Detector Input Divider Select bits (also denoted as 'N1', PLL prescaler)

11111 = Input divided by 33
•
•
•
00001 = Input divided by 3
00000 = Input divided by 2 (default)

Note 1:   The DOZE<2:0> bits can only be written to when the DOZEN bit is clear. If DOZEN = 1, any writes to DOZE<2:0> are ignored.

2:   This bit is cleared when the ROI bit is set and an interrupt occurs.

3:   The DOZEN bit cannot be set if DOZE<2:0> = 000. If DOZE<2:0> = 000, any attempt by user software to set the DOZEN bit is ignored.

10.2.1 SLEEP MODE

The following occurs in Sleep mode:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption is reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor does not operate, since the system clock source is disabled.
- The LPRC clock continues to run in Sleep mode if the WDT is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals can continue to operate. This includes items such as the Input Change Notification (ICN) on the I/O ports or peripherals that use an external clock input.
- Any peripheral that requires the system clock source for its operation is disabled.

The device wakes up from Sleep mode on any of these events:

- Any interrupt source that is individually enabled
- Any form of device Reset
- A WDT time-out

On wake-up from Sleep mode, the processor restarts with the same clock source that was active when Sleep mode was entered.

For optimal power savings, the internal regulator and the Flash regulator can be configured to go into Standby when Sleep mode is entered by clearing the VREGS (RCON<8>) and VREGSF (RCON<11>) bits (default configuration).

If the application requires a faster wake-up time, and can accept higher current requirements, the VREGS (RCON<8>) and VREGSF (RCON<11>) bits can be set to keep the internal regulator and the Flash regulator active during Sleep mode.

10.2.2 IDLE MODE

The following occurs in Idle mode:

- The CPU stops executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see **Section 10.4 "Peripheral Module Disable"**).
- If the WDT or FSCM is enabled, the LPRC also remains active.

The device wakes from Idle mode on any of these events:

- Any interrupt that is individually enabled
- Any device Reset
- A WDT time-out

On wake-up from Idle mode, the clock is reapplied to the CPU and instruction execution will begin (2-4 clock cycles later), starting with the instruction following the PWRSAV instruction or the first instruction in the Interrupt Service Routine (ISR).

All peripherals also have the option to discontinue operation when Idle mode is entered to allow for increased power savings. This option is selectable in the control register of each peripheral; for example, the TSIDL bit in the Timer1 Control register (T1CON<13>).

10.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS
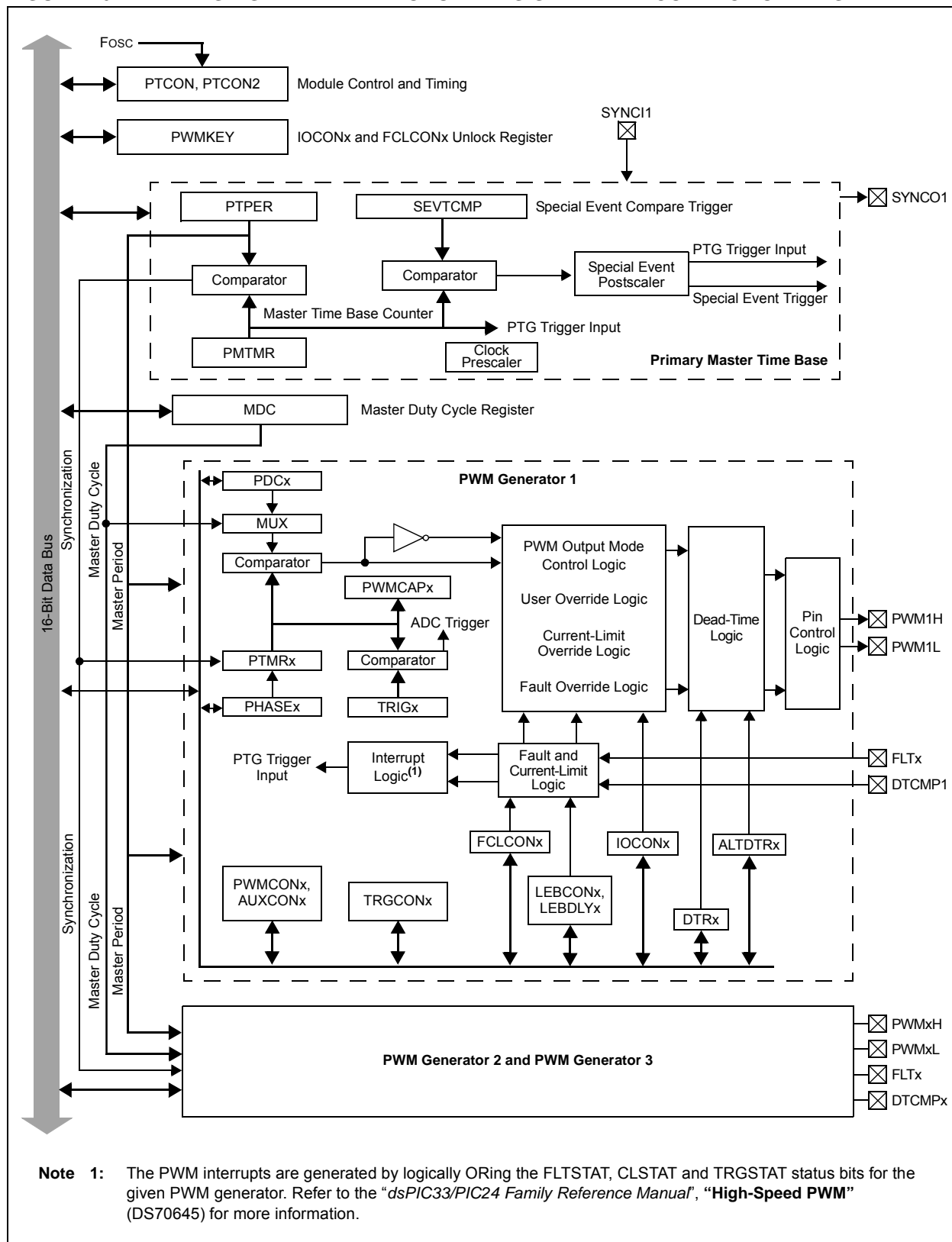
Any interrupt that coincides with the execution of a PWRSAV instruction is held off until entry into Sleep or Idle mode has completed. The device then wakes up from Sleep or Idle mode.

REGISTER 10-1:   PMD1: PERIPHERAL MODULE DISABLE CONTROL REGISTER 1 (CONTINUED)

bit 3   **SPI1MD:** SPI1 Module Disable bit
               1 = SPI1 module is disabled
               0 = SPI1 module is enabled

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

bit 1   **C1MD:** ECAN1 Module Disable bit[2]
               1 = ECAN1 module is disabled
               0 = ECAN1 module is enabled

bit 0   **AD1MD:** ADC1 Module Disable bit
               1 = ADC1 module is disabled
               0 = ADC1 module is enabled

Note 1:   This bit is available on dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X devices only.

2:   This bit is available on dsPIC33EPXXXGP50X and dsPIC33EPXXXMC50X devices only.

FIGURE 16-2: **HIGH-SPEED PWMx MODULE REGISTER INTERCONNECTION DIAGRAM**

Note 1: The PWM interrupts are generated by logically ORing the FLTSTAT, CLSTAT and TRGSTAT status bits for the given PWM generator. Refer to the "*dsPIC33/PIC24 Family Reference Manual*", **"High-Speed PWM"** (DS70645) for more information.

REGISTER 17-2:   QEI1IOC: QEI1 I/O CONTROL REGISTER (CONTINUED)

bit 2   **INDEX:** Status of INDXx Input Pin After Polarity Control

1 = Pin is at logic '1'
0 = Pin is at logic '0'

bit 1   **QEB:** Status of QEBx Input Pin After Polarity Control And SWPAB Pin Swapping

1 = Pin is at logic '1'
0 = Pin is at logic '0'

bit 0   **QEA:** Status of QEAx Input Pin After Polarity Control And SWPAB Pin Swapping

1 = Pin is at logic '1'
0 = Pin is at logic '0'

18.3   SPIx Control Registers

REGISTER 18-1:   SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	—	—	SPIBEC<2:0>		
bit 15							bit 8

R/W-0	R/C-0, HS	R/W-0	R/W-0	R/W-0	R/W-0	R-0, HS, HC	R-0, HS, HC
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

Legend:		C = Clearable bit	HS = Hardware Settable bit	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   **SPIEN:** SPIx Enable bit

1 = Enables the module and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables the module

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

bit 13   **SPISIDL:** SPIx Stop in Idle Mode bit

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

bit 12-11 **Unimplemented:** Read as '0'

bit 10-8   **SPIBEC<2:0>:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)

Master mode:
Number of SPIx transfers that are pending.

Slave mode:
Number of SPIx transfers that are unread.

bit 7   **SRMPT:** SPIx Shift Register (SPIxSR) Empty bit (valid in Enhanced Buffer mode)

1 = SPIx Shift register is empty and Ready-To-Send or receive the data
0 = SPIx Shift register is not empty

bit 6   **SPIROV:** SPIx Receive Overflow Flag bit

1 = A new byte/word is completely received and discarded; the user application has not read the previous
     data in the SPIxBUF register
0 = No overflow has occurred

bit 5   **SRXMPT:** SPIx Receive FIFO Empty bit (valid in Enhanced Buffer mode)

1 = RX FIFO is empty
0 = RX FIFO is not empty

bit 4-2   **SISEL<2:0>:** SPIx Buffer Interrupt Mode bits (valid in Enhanced Buffer mode)

111 = Interrupt when the SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when last bit is shifted into SPIxSR and as a result, the TX FIFO is empty
101 = Interrupt when the last bit is shifted out of SPIxSR and the transmit is complete
100 = Interrupt when one data is shifted into the SPIxSR and as a result, the TX FIFO has one open
      memory location
011 = Interrupt when the SPIx receive buffer is full (SPIRBF bit is set)
010 = Interrupt when the SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in the receive buffer (SRMPT bit is set)
000 = Interrupt when the last data in the receive buffer is read and as a result, the buffer is empty
      (SRXMPT bit is set)

REGISTER 20-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN**(1)**	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Legend:		
	HC = Hardware Clearable bit	C = 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 15,13 **UTXISEL<1:0>:** UARTx Transmission Interrupt Mode Selection bits

    11 = Reserved; do not use
    10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR) and as a result, the transmit buffer becomes empty
    01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed
    00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)

bit 14 **UTXINV:** UARTx Transmit Polarity Inversion bit

If IREN = 0:
1 = UxTX Idle state is '0'
0 = UxTX Idle state is '1'

If IREN = 1:
1 = IrDA encoded, UxTX Idle state is '1'
0 = IrDA encoded, UxTX Idle state is '0'

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

bit 11 **UTXBRK:** UARTx Transmit Break bit

    1 = Sends Sync Break on next transmission – Start bit, followed by twelve '0' bits, followed by Stop bit; cleared by hardware upon completion
    0 = Sync Break transmission is disabled or completed

bit 10 **UTXEN:** UARTx Transmit Enable bit**(1)**

    1 = Transmit is enabled, UxTX pin is controlled by UARTx
    0 = Transmit is disabled, any pending transmission is aborted and buffer is reset; UxTX pin is controlled by the PORT

bit 9 **UTXBF:** UARTx Transmit Buffer Full Status bit (read-only)

    1 = Transmit buffer is full
    0 = Transmit buffer is not full, at least one more character can be written

bit 8 **TRMT:** Transmit Shift Register Empty bit (read-only)

    1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)
    0 = Transmit Shift Register is not empty, a transmission is in progress or queued

bit 7-6 **URXISEL<1:0>:** UARTx Receive Interrupt Mode Selection bits

    11 = Interrupt is set on UxRSR transfer, making the receive buffer full (i.e., has 4 data characters)
    10 = Interrupt is set on UxRSR transfer, making the receive buffer 3/4 full (i.e., has 3 data characters)
    0x = Interrupt is set when any character is received and transferred from the UxRSR to the receive buffer; receive buffer has one or more characters

Note 1: Refer to the **"UART"** (DS70582) section in the *"dsPIC33/PIC24 Family Reference Manual"* for information on enabling the UARTx module for transmit operation.

REGISTER 23-6:   AD1CHS0: ADC1 INPUT CHANNEL 0 SELECT REGISTER (CONTINUED)

bit 4-0 **CH0SA<4:0>:** Channel 0 Positive Input Select for Sample MUXA bits[1]

11111 = Open; use this selection with CTMU capacitive and time measurement
11110 = Channel 0 positive input is connected to the CTMU temperature measurement diode (CTMU TEMP)
11101 = Reserved
11100 = Reserved
11011 = Reserved
11010 = Channel 0 positive input is the output of OA3/AN6[2,3]
11001 = Channel 0 positive input is the output of OA2/AN0[2]
11000 = Channel 0 positive input is the output of OA1/AN3[2]
10110 = Reserved
•
•
•
10000 = Reserved
01111 = Channel 0 positive input is AN15[1,3]
01110 = Channel 0 positive input is AN14[1,3]
01101 = Channel 0 positive input is AN13[1,3]
•
•
•
00010 = Channel 0 positive input is AN2[1,3]
00001 = Channel 0 positive input is AN1[1,3]
00000 = Channel 0 positive input is AN0[1,3]

Note 1: AN0 through AN7 are repurposed when comparator and op amp functionality is enabled. See Figure 23-1 to determine how enabling a particular op amp or comparator affects selection choices for Channels 1, 2 and 3.

2: The OAx input is used if the corresponding op amp is selected (OPMODE (CMxCON<10>) = 1); otherwise, the ANx input is used.

3: See the **"Pin Diagrams"** section for the available analog channels for each device.

26.0 PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

> **Note 1:** This data sheet summarizes the features of the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Programmable Cyclic Redundancy Check (CRC)"** (DS70346) of the "*dsPIC33/PIC24 Family Reference Manual*", which is available from the Microchip web site (www.microchip.com).
>
> **2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The programmable CRC generator offers the following features:

- User-programmable (up to 32nd order) polynomial CRC equation
- Interrupt output
- Data FIFO

The 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

A simplified block diagram of the CRC generator is shown in Figure 26-1. A simple version of the CRC shift engine is shown in Figure 26-2.
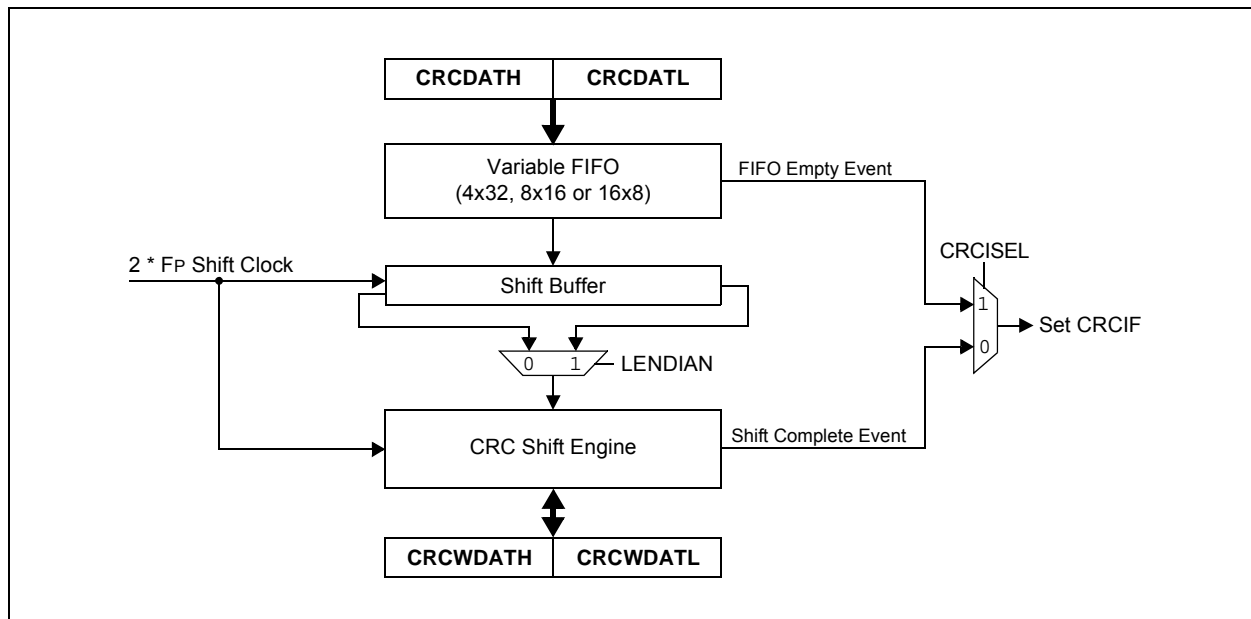
FIGURE 26-1: CRC BLOCK DIAGRAM

TABLE 28-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles(2)	Status Flags Affected
46	MOV	MOV	f,Wn	Move f to Wn	1	1	None
		MOV	f	Move f to f	1	1	None
		MOV	f,WREG	Move f to WREG	1	1	None
		MOV	#lit16,Wn	Move 16-bit literal to Wn	1	1	None
		MOV.b	#lit8,Wn	Move 8-bit literal to Wn	1	1	None
		MOV	Wn,f	Move Wn to f	1	1	None
		MOV	Wso,Wdo	Move Ws to Wd	1	1	None
		MOV	WREG,f	Move WREG to f	1	1	None
		MOV.D	Wns,Wd	Move Double from W(ns):W(ns + 1) to Wd	1	2	None
		MOV.D	Ws,Wnd	Move Double from Ws to W(nd + 1):W(nd)	1	2	None
47	MOVPAG	MOVPAG	#lit10,DSRPAG	Move 10-bit literal to DSRPAG	1	1	None
		MOVPAG	#lit9,DSWPAG	Move 9-bit literal to DSWPAG	1	1	None
		MOVPAG	#lit8,TBLPAG	Move 8-bit literal to TBLPAG	1	1	None
		MOVPAG	Ws, DSRPAG	Move Ws<9:0> to DSRPAG	1	1	None
		MOVPAG	Ws, DSWPAG	Move Ws<8:0> to DSWPAG	1	1	None
		MOVPAG	Ws, TBLPAG	Move Ws<7:0> to TBLPAG	1	1	None
48	MOVSAC	MOVSAC	Acc,Wx,Wxd,Wy,Wyd,AWB(1)	Prefetch and store accumulator	1	1	None
49	MPY	MPY	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd(1)	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd(1)	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
50	MPY.N	MPY.N	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd(1)	-(Multiply Wm by Wn) to Accumulator	1	1	None
51	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd,AWB(1)	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB

Note 1: These instructions are available in dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X devices only.
  2: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.

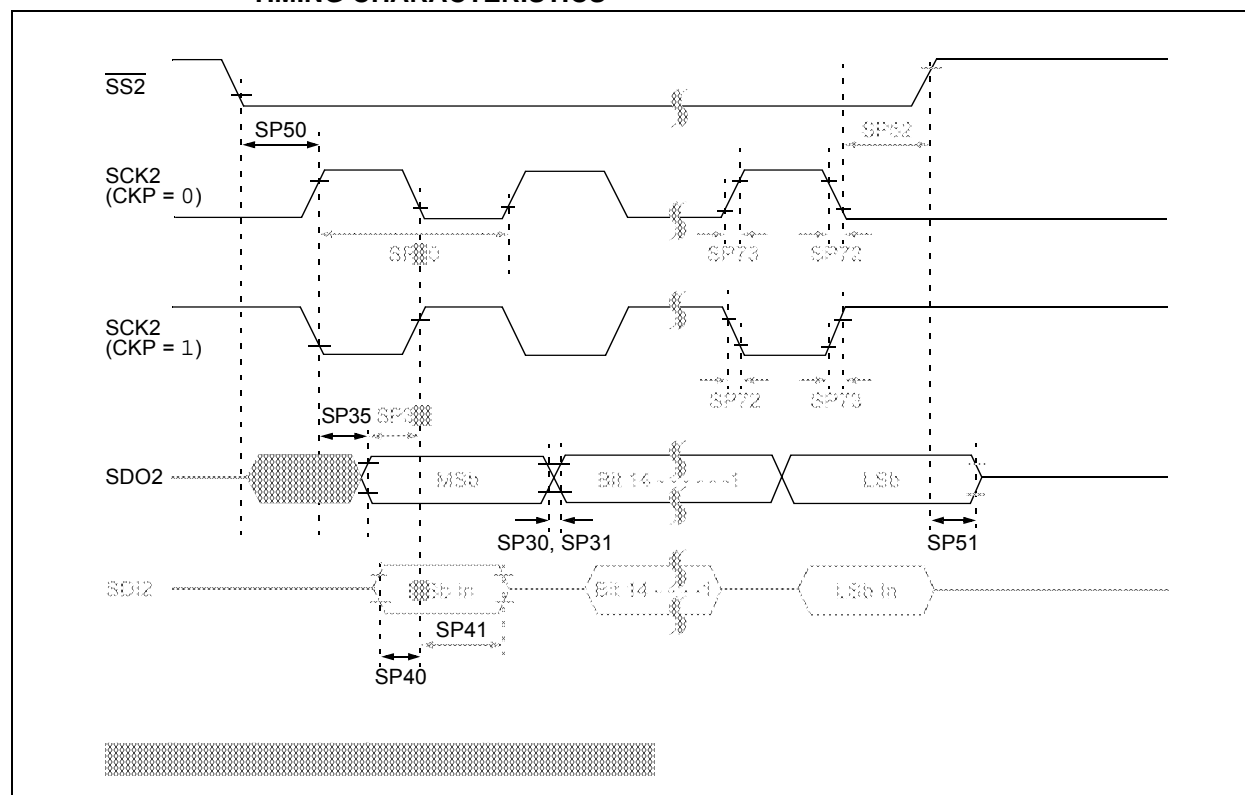**FIGURE 30-20:** **SPI2 SLAVE MODE (FULL-DUPLEX, CKE = 0, CKP = 1, SMP = 0) TIMING CHARACTERISTICS**

TABLE A-5:   MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description

Section 30.0 "Electrical Characteristics"	• Throughout: qualifies all footnotes relating to the operation of analog modules below V$_{DDMIN}$ (replaces "will have" with "may have")
	• Throughout: changes all references of SPI timing parameter symbol "TscP" to "FscP"
	• Table 30-1: changes V$_{DD}$ range to 3.0V to 3.6V
	• Table 30-4: removes Parameter DC12 (RAM Retention Voltage)
	• Table 30-7: updates Maximum values at 10 and 20 MIPS
	• Table 30-8: adds Maximum I$_{PD}$ values, and removes all ∆I$_{WDT}$ entries
	• Adds new Table 30-9 (Watchdog Timer Delta Current) with consolidated values removed from Table 30-8. All subsequent tables are renumbered accordingly.
	• Table 30-10: adds footnote for all parameters for 1:2 Doze ratio
	• Table 30-11:
	  - changes Minimum and Maximum values for D120 and D130
	  - adds Minimum and Maximum values for D131
	  - adds Minimum and Maximum values for D150 through D156, and removes Typical values
	• Table 30-12:
	  - reformats table for readability
	  - changes I$_{OL}$ conditions for DO10
	• Table 30-14: adds footnote to D135
	• Table 30-17: changes Minimum and Maximum values for OS30
	• Table 30-19:
	  - splits temperature range and adds new values for F20a
	  - reduces temperature range for F20b to extended temperatures only
	• Table 30-20:
	  - splits temperature range and adds new values for F21a
	  - reduces temperature range for F20b to extended temperatures only
	• Table 30-53:
	  - adds Maximum value to CM30
	  - adds footnote ("Parameter characterized...") to multiple parameters
	• Table 30-55: adds Minimum and Maximum values for all CTMUI specifications, and removes Typical values
	• Table 30-57: adds new footnote to AD09
	• Table 30-58:
	  - removes all specifications for accuracy with external voltage references
	  - removes Typical values for AD23a and AD24a
	  - replaces Minimum and Maximum values for AD21a, AD22a, AD23a and AD24a with new values, split by Industrial and Extended temperatures
	  - removes Maximum value of AD30
	  - removes Minimum values from AD31a and AD32a
	  - adds or changes Typical values for AD30, AD31a, AD32a and AD33a
	• Table 30-59:
	  - removes all specifications for accuracy with external voltage references
	  - removes Maximum value of AD30
	  - removes Typical values for AD23b and AD24b
	  - replaces Minimum and Maximum values for AD21b, AD22b, AD23b and AD24b with new values, split by Industrial and Extended temperatures
	  - removes Minimum and Maximum values from AD31b, AD32b, AD33b and AD34b
	  - adds or changes Typical values for AD30, AD31a, AD32a and AD33a
	• Table 30-61: Adds footnote to AD51

Section 32.0 "DC and AC Device Characteristics Graphs"	• Updates Figure 32-6 (Typical I$_{DD}$ @ 3.3V) with individual current vs. processor speed curves for the different program memory sizes
Section 33.0 "Packaging Information"	• Replaces drawing C04-149C (64-pin QFN, 7.15 x 7.15 exposed pad) with C04-154A (64-pin QFN, 5.4 x 5.4 exposed pad)