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	60 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	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 16
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 6x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24ep256mc202-e-so

TABLE 2: **dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X MOTOR CONTROL FAMILIES (CONTINUED)**

Device	Page Erase Size (Instructions)	Program Flash Memory (Kbytes)	RAM (Kbytes)	16-Bit/32-Bit Timers	Input Capture	Output Compare	Motor Control PWM[4] (Channels)	Quadrature Encoder Interface	UART	SPI[2]	ECAN™ Technology	External Interrupts[3]	I^2C™	CRC Generator	10-Bit/12-Bit ADC (Channels)	Op Amps/Comparators	CTMU	PTG	I/O Pins	Pins	Packages
								Remappable Peripherals													
dsPIC33EP32MC504	512	32	4																		
dsPIC33EP64MC504	1024	64	8																		VTLA[5], TQFP, QFN, UQFN
dsPIC33EP128MC504	1024	128	16	5	4	4	6	1	2	2	1	3	2	1	9	3/4	Yes	Yes	35	44/48	
dsPIC33EP256MC504	1024	256	32																		
dsPIC33EP512MC504	1024	512	48																		
dsPIC33EP64MC506	1024	64	8																		
dsPIC33EP128MC506	1024	128	16	5	4	4	6	1	2	2	1	3	2	1	16	3/4	Yes	Yes	53	64	TQFP, QFN
dsPIC33EP256MC506	1024	256	32																		
dsPIC33EP512MC506	1024	512	48																		

Note 1: On 28-pin devices, Comparator 4 does not have external connections. Refer to **Section 25.0 "Op Amp/Comparator Module"** for details.
2: Only SPI2 is remappable.
3: INT0 is not remappable.
4: Only the PWM Faults are remappable.
5: The SSOP and VTLA packages are not available for devices with 512 Kbytes of memory.

4.2.5   X AND Y DATA SPACES

The   dsPIC33EPXXXMC20X/50X   and dsPIC33EPXXXGP50X core has two Data Spaces, X and Y. These Data Spaces can be considered either separate (for some DSP instructions) or as one unified linear address range (for MCU instructions). The Data Spaces are accessed using two Address Generation Units (AGUs) and separate data paths. This feature allows certain instructions to concurrently fetch two words from RAM, thereby enabling efficient execution of DSP algorithms, such as Finite Impulse Response (FIR) filtering and Fast Fourier Transform (FFT).

The X Data Space is used by all instructions and supports all addressing modes. X Data Space has separate read and write data buses. The X read data bus is the read data path for all instructions that view Data Space as combined X and Y address space. It is also the X data prefetch path for the dual operand DSP instructions (MAC class).

The Y Data Space is used in concert with the X Data Space by the MAC class of instructions (CLR, ED, EDAC, MAC, MOVSAC, MPY, MPY.N and MSC) to provide two concurrent data read paths.

Both the X and Y Data Spaces support Modulo Addressing mode for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing mode is only supported for writes to X Data Space. Modulo Addressing and Bit-Reversed Addressing are not present in PIC24EPXXXGP/MC20X devices.

All data memory writes, including in DSP instructions, view Data Space as combined X and Y address space. The boundary between the X and Y Data Spaces is device-dependent and is not user-programmable.

4.3   Memory 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

4.3.1   KEY RESOURCES

- **"Program Memory"** (DS70613) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

TABLE 4-59:   PORTA REGISTER MAP FOR PIC24EPXXXGP/MC202 AND dsPIC33EPXXXGP/MC202/502 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
TRISA	0E00	—	—	—	—	—	—	—	—	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	001F
PORTA	0E02	—	—	—	—	—	—	—	—	—	—	—	RA4	RA3	RA2	RA1	RA0	0000
LATA	0E04	—	—	—	—	—	—	—	—	—	—	—	LATA4	LATA3	LATA2	LA1TA1	LA0TA0	0000
ODCA	0E06	—	—	—	—	—	—	—	—	—	—	—	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	0000
CNENA	0E08	—	—	—	—	—	—	—	—	—	—	—	CNIEA4	CNIEA3	CNIEA2	CNIEA1	CNIEA0	0000
CNPUA	0E0A	—	—	—	—	—	—	—	—	—	—	—	CNPUA4	CNPUA3	CNPUA2	CNPUA1	CNPUA0	0000
CNPDA	0E0C	—	—	—	—	—	—	—	—	—	—	—	CNPDA4	CNPDA3	CNPDA2	CNPDA1	CNPDA0	0000
ANSELA	0E0E	—	—	—	—	—	—	—	—	—	—	—	ANSA4	—	—	ANSA1	ANSA0	0013

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

TABLE 4-60:   PORTB REGISTER MAP FOR PIC24EPXXXGP/MC202 AND dsPIC33EPXXXGP/MC202/502 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
TRISB	0E10	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	0E12	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	0E14	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	0E16	ODCB15	ODCB14	ODCB13	ODCB12	ODCB11	ODCB10	ODCB9	ODCB8	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	0000
CNENB	0E18	CNIEB15	CNIEB14	CNIEB13	CNIEB12	CNIEB11	CNIEB10	CNIEB9	CNIEB8	CNIEB7	CNIEB6	CNIEB5	CNIEB4	CNIEB3	CNIEB2	CNIEB1	CNIEB0	0000
CNPUB	0E1A	CNPUB15	CNPUB14	CNPUB13	CNPUB12	CNPUB11	CNPUB10	CNPUB9	CNPUB8	CNPUB7	CNPUB6	CNPUB5	CNPUB4	CNPUB3	CNPUB2	CNPUB1	CNPUB0	0000
CNPDB	0E1C	CNPDB15	CNPDB14	CNPDB13	CNPDB12	CNPDB11	CNPDB10	CNPDB9	CNPDB8	CNPDB7	CNPDB6	CNPDB5	CNPDB4	CNPDB3	CNPDB2	CNPDB1	CNPDB0	0000
ANSELB	0E1E	—	—	—	—	—	—	—	ANSB8	—	—	—	—	ANSB3	ANSB2	ANSB1	ANSB0	010F

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

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.

4.8 Interfacing Program and Data Memory Spaces

The dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X architecture uses a 24-bit-wide Program Space (PS) and a 16-bit-wide Data Space (DS). The architecture is also a modified Harvard scheme, meaning that data can also be present in the Program Space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the architecture of the dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X devices provides two methods by which Program Space can be accessed during operation:
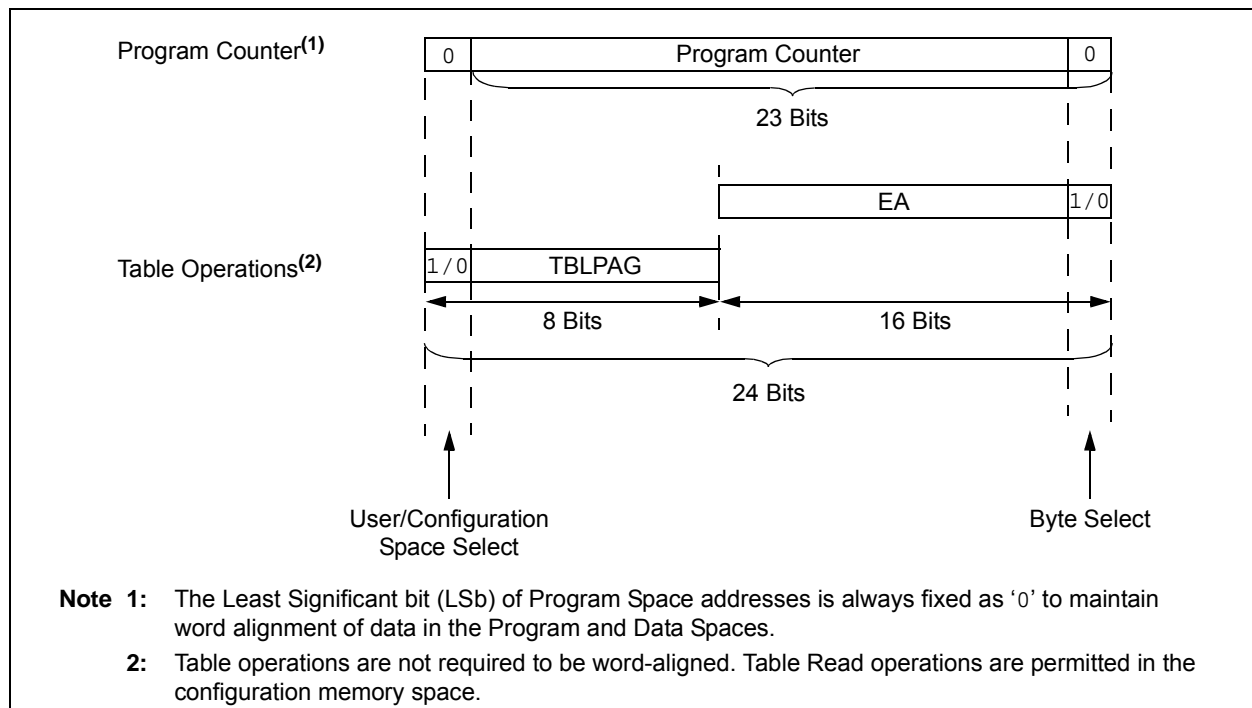
• Using table instructions to access individual bytes or words anywhere in the Program Space
• Remapping a portion of the Program Space into the Data Space (Program Space Visibility)

Table instructions allow an application to read or write to small areas of the program memory. This capability makes the method ideal for accessing data tables that need to be updated periodically. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data. The application can only access the least significant word of the program word.

TABLE 4-65: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		<23>	**<22:16>**	**<15>**	**<14:1>**	**<0>**
Instruction Access (Code Execution)	User	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
`TBLRD/TBLWT` (Byte/Word Read/Write)	User	TBLPAG<7:0>		Data EA<15:0>		
		0xxx xxxx xxxx xxxx xxxx xxxx				
	Configuration	TBLPAG<7:0>		Data EA<15:0>		
		1xxx xxxx xxxx xxxx xxxx xxxx				

FIGURE 4-22: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION

Note 1: The Least Significant bit (LSb) of Program Space addresses is always fixed as '0' to maintain word alignment of data in the Program and Data Spaces.

2: Table operations are not required to be word-aligned. Table Read operations are permitted in the configuration memory space.

In addition, DMA transfers can be triggered by timers as well as external interrupts. Each DMA channel is unidirectional. Two DMA channels must be allocated to read and write to a peripheral. If more than one channel receives a request to transfer data, a simple fixed priority scheme based on channel number, dictates which channel completes the transfer and which channel, or channels, are left pending. Each DMA channel moves a block of data, after which, it generates an interrupt to the CPU to indicate that the block is available for processing.

The DMA Controller provides these functional capabilities:

- Four DMA channels
- Register Indirect with Post-Increment Addressing mode
- Register Indirect without Post-Increment Addressing mode

- Peripheral Indirect Addressing mode (peripheral generates destination address)
- CPU interrupt after half or full block transfer complete
- Byte or word transfers
- Fixed priority channel arbitration
- Manual (software) or automatic (peripheral DMA requests) transfer initiation
- One-Shot or Auto-Repeat Block Transfer modes
- Ping-Pong mode (automatic switch between two SRAM start addresses after each block transfer is complete)
- DMA request for each channel can be selected from any supported interrupt source
- Debug support features

The peripherals that can utilize DMA are listed in Table 8-1.

TABLE 8-1:   DMA CHANNEL TO PERIPHERAL ASSOCIATIONS

Peripheral to DMA Association	DMAxREQ Register IRQSEL<7:0> Bits	DMAxPAD Register (Values to Read from Peripheral)	DMAxPAD Register (Values to Write to Peripheral)
INT0 – External Interrupt 0	00000000	—	—
IC1 – Input Capture 1	00000001	0x0144 (IC1BUF)	—
IC2 – Input Capture 2	00000101	0x014C (IC2BUF)	—
IC3 – Input Capture 3	00100101	0x0154 (IC3BUF)	—
IC4 – Input Capture 4	00100110	0x015C (IC4BUF)	—
OC1 – Output Compare 1	00000010	—	0x0906 (OC1R) 0x0904 (OC1RS)
OC2 – Output Compare 2	00000110	—	0x0910 (OC2R) 0x090E (OC2RS)
OC3 – Output Compare 3	00011001	—	0x091A (OC3R) 0x0918 (OC3RS)
OC4 – Output Compare 4	00011010	—	0x0924 (OC4R) 0x0922 (OC4RS)
TMR2 – Timer2	00000111	—	—
TMR3 – Timer3	00001000	—	—
TMR4 – Timer4	00011011	—	—
TMR5 – Timer5	00011100	—	—
SPI1 Transfer Done	00001010	0x0248 (SPI1BUF)	0x0248 (SPI1BUF)
SPI2 Transfer Done	00100001	0x0268 (SPI2BUF)	0x0268 (SPI2BUF)
UART1RX – UART1 Receiver	00001011	0x0226 (U1RXREG)	—
UART1TX – UART1 Transmitter	00001100	—	0x0224 (U1TXREG)
UART2RX – UART2 Receiver	00011110	0x0236 (U2RXREG)	—
UART2TX – UART2 Transmitter	00011111	—	0x0234 (U2TXREG)
ECAN1 – RX Data Ready	00100010	0x0440 (C1RXD)	—
ECAN1 – TX Data Request	01000110	—	0x0442 (C1TXD)
ADC1 – ADC1 Convert Done	00001101	0x0300 (ADC1BUF0)	—

FIGURE 8-2: **DMA CONTROLLER BLOCK DIAGRAM**
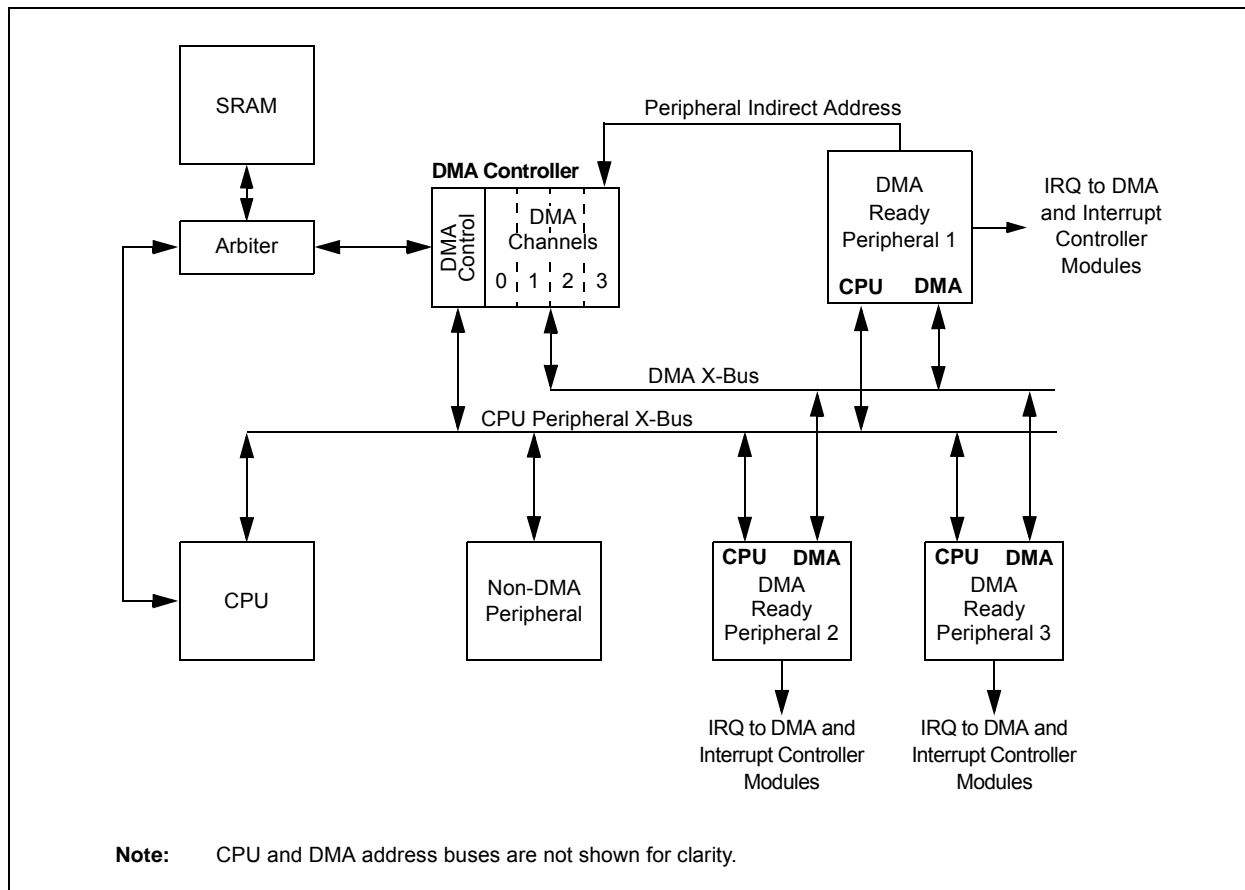
Note: CPU and DMA address buses are not shown for clarity.

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 8-14:   DMAPPS: DMA PING-PONG STATUS 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	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	PPST3	PPST2	PPST1	PPST0
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 15-4   **Unimplemented:** Read as '0'

bit 3   **PPST3:** DMA Channel 3 Ping-Pong Mode Status Flag bit
1 = DMASTB3 register is selected
0 = DMASTA3 register is selected

bit 2   **PPST2:** DMA Channel 2 Ping-Pong Mode Status Flag bit
1 = DMASTB2 register is selected
0 = DMASTA2 register is selected

bit 1   **PPST1:** DMA Channel 1 Ping-Pong Mode Status Flag bit
1 = DMASTB1 register is selected
0 = DMASTA1 register is selected

bit 0   **PPST0:** DMA Channel 0 Ping-Pong Mode Status Flag bit
1 = DMASTB0 register is selected
0 = DMASTA0 register is selected

11.4.4.2   Output Mapping

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Like the RPINRx registers, each register contains sets of 6-bit fields, with each set associated with one RPn pin (see Register 11-18 through Register 11-27). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 11-3 and Figure 11-3).

A null output is associated with the output register Reset value of '0'. This is done to ensure that remappable outputs remain disconnected from all output pins by default.

FIGURE 11-3:   MULTIPLEXING REMAPPABLE OUTPUT FOR RPn
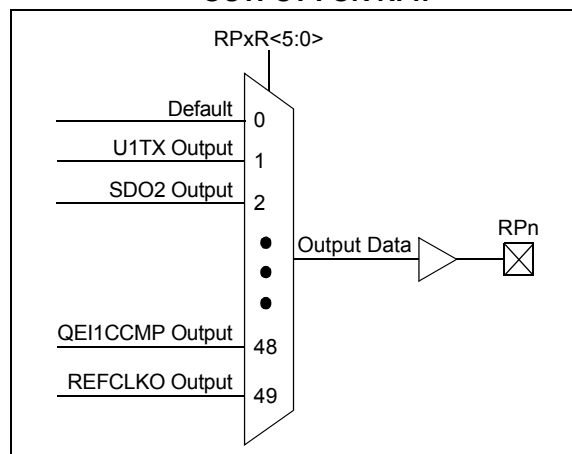
11.4.4.3   Mapping Limitations

The control schema of the peripheral select pins is not limited to a small range of fixed peripheral configurations. There are no mutual or hardware-enforced lockouts between any of the peripheral mapping SFRs. Literally any combination of peripheral mappings across any or all of the RPn pins is possible. This includes both many-to-one and one-to-many mappings of peripheral inputs and outputs to pins. While such mappings may be technically possible from a configuration point of view, they may not be supportable from an electrical point of view.

TABLE 11-3:   OUTPUT SELECTION FOR REMAPPABLE PINS (RPn)

Function	RPxR<5:0>	Output Name
Default PORT	000000	RPn tied to Default Pin
U1TX	000001	RPn tied to UART1 Transmit
U2TX	000011	RPn tied to UART2 Transmit
SDO2	001000	RPn tied to SPI2 Data Output
SCK2	001001	RPn tied to SPI2 Clock Output
$\overline{SS2}$	001010	RPn tied to SPI2 Slave Select
C1TX[2]	001110	RPn tied to CAN1 Transmit
OC1	010000	RPn tied to Output Compare 1 Output
OC2	010001	RPn tied to Output Compare 2 Output
OC3	010010	RPn tied to Output Compare 3 Output
OC4	010011	RPn tied to Output Compare 4 Output
C1OUT	011000	RPn tied to Comparator Output 1
C2OUT	011001	RPn tied to Comparator Output 2
C3OUT	011010	RPn tied to Comparator Output 3
SYNCO1[1]	101101	RPn tied to PWM Primary Time Base Sync Output
QEI1CCMP[1]	101111	RPn tied to QEI 1 Counter Comparator Output
REFCLKO	110001	RPn tied to Reference Clock Output
C4OUT	110010	RPn tied to Comparator Output 4

Note 1:   This function is available in dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X devices only.

   2:   This function is available in dsPIC33EPXXXGP/MC50X devices only.

14.2 Input Capture Registers

REGISTER 14-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/HC/HS-0	R/HC/HS-0	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0
bit 7							bit 0

Legend:		HC = Hardware 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	

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

bit 13 **ICSIDL:** Input Capture Stop in Idle Control bit

  1 = Input capture will Halt in CPU Idle mode
  0 = Input capture will continue to operate in CPU Idle mode

bit 12-10 **ICTSEL<2:0>:** Input Capture Timer Select bits

  111 = Peripheral clock (F$_P$) is the clock source of the ICx
  110 = Reserved
  101 = Reserved
  100 = T1CLK is the clock source of the ICx (only the synchronous clock is supported)
  011 = T5CLK is the clock source of the ICx
  010 = T4CLK is the clock source of the ICx
  001 = T2CLK is the clock source of the ICx
  000 = T3CLK is the clock source of the ICx

bit 9-7 **Unimplemented:** Read as '0'

bit 6-5 **ICI<1:0>:** Number of Captures per Interrupt Select bits (this field is not used if ICM<2:0> = 001 or 111)

  11 = Interrupt on every fourth capture event
  10 = Interrupt on every third capture event
  01 = Interrupt on every second capture event
  00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture Overflow Status Flag bit (read-only)

  1 = Input capture buffer overflow occurred
  0 = No input capture buffer overflow occurred

bit 3 **ICBNE:** Input Capture Buffer Not Empty Status bit (read-only)

  1 = Input capture buffer is not empty, at least one more capture value can be read
  0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits

  111 = Input capture functions as interrupt pin only in CPU Sleep and Idle modes (rising edge detect only, all other control bits are not applicable)
  110 = Unused (module is disabled)
  101 = Capture mode, every 16th rising edge (Prescaler Capture mode)
  100 = Capture mode, every 4th rising edge (Prescaler Capture mode)
  011 = Capture mode, every rising edge (Simple Capture mode)
  010 = Capture mode, every falling edge (Simple Capture mode)
  001 = Capture mode, every edge rising and falling (Edge Detect mode (ICI<1:0>) is not used in this mode)
  000 = Input capture module is turned off

REGISTER 16-13: IOCONx: PWMx I/O CONTROL REGISTER[(2)] (CONTINUED)

bit 1   **SWAP:** SWAP PWMxH and PWMxL Pins bit

1 = PWMxH output signal is connected to PWMxL pins; PWMxL output signal is connected to PWMxH pins

0 = PWMxH and PWMxL pins are mapped to their respective pins

bit 0   **OSYNC:** Output Override Synchronization bit

1 = Output overrides via the OVRDAT<1:0> bits are synchronized to the PWMx period boundary

0 = Output overrides via the OVDDAT<1:0> bits occur on the next CPU clock boundary

Note 1: These bits should not be changed after the PWMx module is enabled (PTEN = 1).

2: If the PWMLOCK Configuration bit (FOSCSEL<6>) is a '1', the IOCONx register can only be written after the unlock sequence has been executed.

17.0 QUADRATURE ENCODER INTERFACE (QEI) MODULE (dsPIC33EPXXXMC20X/50X and PIC24EPXXXMC20X DEVICES ONLY)

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 **"Quadrature Encoder Interface (QEI)"** (DS70601) in 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.

This chapter describes the Quadrature Encoder Interface (QEI) module and associated operational modes. The QEI module provides the interface to incremental encoders for obtaining mechanical position data.

The operational features of the QEI module include:

- 32-Bit Position Counter
- 32-Bit Index Pulse Counter
- 32-Bit Interval Timer
- 16-Bit Velocity Counter
- 32-Bit Position Initialization/Capture/Compare High register
- 32-Bit Position Compare Low register
- x4 Quadrature Count mode
- External Up/Down Count mode
- External Gated Count mode
- External Gated Timer mode
- Internal Timer mode

Figure 17-1 illustrates the QEI block diagram.

21.4 ECAN Control Registers

REGISTER 21-1: CxCTRL1: ECANx CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
—	—	CSIDL	ABAT	CANCKS	REQOP2	REQOP1	REQOP0
bit 15							bit 8

R-1	R-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
OPMODE2	OPMODE1	OPMODE0	—	CANCAP	—	—	WIN
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 **CSIDL:** ECANx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **ABAT:** Abort All Pending Transmissions bit
1 = Signals all transmit buffers to abort transmission
0 = Module will clear this bit when all transmissions are aborted

bit 11 **CANCKS:** ECANx Module Clock (F$_{CAN}$) Source Select bit
1 = F$_{CAN}$ is equal to 2 * F$_P$
0 = F$_{CAN}$ is equal to F$_P$

bit 10-8 **REQOP<2:0>:** Request Operation Mode bits
111 = Set Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Set Configuration mode
011 = Set Listen Only mode
010 = Set Loopback mode
001 = Set Disable mode
000 = Set Normal Operation mode

bit 7-5 **OPMODE<2:0>:** Operation Mode bits
111 = Module is in Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Module is in Configuration mode
011 = Module is in Listen Only mode
010 = Module is in Loopback mode
001 = Module is in Disable mode
000 = Module is in Normal Operation mode

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

bit 3 **CANCAP:** CAN Message Receive Timer Capture Event Enable bit
1 = Enables input capture based on CAN message receive
0 = Disables CAN capture

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **WIN:** SFR Map Window Select bit
1 = Uses filter window
0 = Uses buffer window

REGISTER 21-6: CxINTF: ECANx INTERRUPT FLAG REGISTER

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	TXBO	TXBP	RXBP	TXWAR	RXWAR	EWARN
bit 15							bit 8

R/C-0	R/C-0	R/C-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0
IVRIF	WAKIF	ERRIF	—	FIFOIF	RBOVIF	RBIF	TBIF
bit 7							bit 0

Legend:		C = Writable bit, but only '0' can be written to clear the 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   **TXBO:** Transmitter in Error State Bus Off bit

1 = Transmitter is in Bus Off state
0 = Transmitter is not in Bus Off state

bit 12   **TXBP:** Transmitter in Error State Bus Passive bit

1 = Transmitter is in Bus Passive state
0 = Transmitter is not in Bus Passive state

bit 11   **RXBP:** Receiver in Error State Bus Passive bit

1 = Receiver is in Bus Passive state
0 = Receiver is not in Bus Passive state

bit 10   **TXWAR:** Transmitter in Error State Warning bit

1 = Transmitter is in Error Warning state
0 = Transmitter is not in Error Warning state

bit 9   **RXWAR:** Receiver in Error State Warning bit

1 = Receiver is in Error Warning state
0 = Receiver is not in Error Warning state

bit 8   **EWARN:** Transmitter or Receiver in Error State Warning bit

1 = Transmitter or receiver is in Error Warning state
0 = Transmitter or receiver is not in Error Warning state

bit 7   **IVRIF:** Invalid Message Interrupt Flag bit

1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6   **WAKIF:** Bus Wake-up Activity Interrupt Flag bit

1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5   **ERRIF:** Error Interrupt Flag bit (multiple sources in CxINTF<13:8>)

1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

bit 3   **FIFOIF:** FIFO Almost Full Interrupt Flag bit

1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2   **RBOVIF:** RX Buffer Overflow Interrupt Flag bit

1 = Interrupt request has occurred
0 = Interrupt request has not occurred

BUFFER 21-3: **ECAN™ MESSAGE BUFFER WORD 2**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID5	EID4	EID3	EID2	EID1	EID0	RTR	RB1

bit 15   bit 8

U-x	U-x	U-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	RB0	DLC3	DLC2	DLC1	DLC0

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   **EID<5:0>:** Extended Identifier bits

bit 9   **RTR:** Remote Transmission Request bit

                    When IDE = 1:
                    1 = Message will request remote transmission
                    0 = Normal message

                    When IDE = 0:
                    The RTR bit is ignored.

bit 8   **RB1:** Reserved Bit 1

                    User must set this bit to '0' per CAN protocol.

bit 7-5   **Unimplemented:** Read as '0'

bit 4   **RB0:** Reserved Bit 0

                    User must set this bit to '0' per CAN protocol.

bit 3-0   **DLC<3:0>:** Data Length Code bits

BUFFER 21-4: **ECAN™ MESSAGE BUFFER WORD 3**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			Byte 1				

bit 15   bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			Byte 0				

bit 7   bit 0
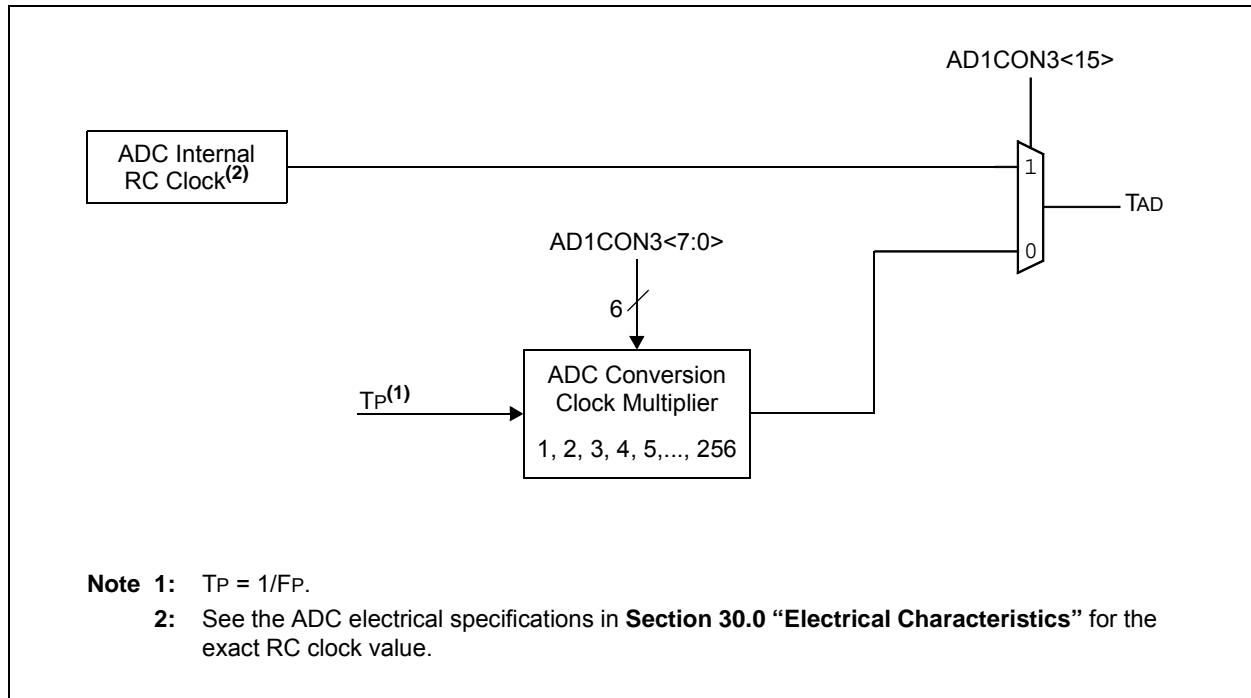
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8   **Byte 1<15:8>:** ECAN Message Byte 1 bits

bit 7-0   **Byte 0<7:0>:** ECAN Message Byte 0 bits

FIGURE 23-2: **ADC CONVERSION CLOCK PERIOD BLOCK DIAGRAM**

AD1CON3<15>

ADC Internal
RC Clock[2]

T_{AD}

AD1CON3<7:0>

6

ADC Conversion
Clock Multiplier

1, 2, 3, 4, 5,..., 256

T_P[1]

Note 1: T_P = 1/F_P.

2: See the ADC electrical specifications in **Section 30.0 "Electrical Characteristics"** for the exact RC clock value.

Most instructions are a single word. Certain double-word instructions are designed to provide all the required information 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 executes as a NOP.

The double-word instructions execute in two instruction cycles.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true, or the Program Counter is changed as a result of the instruction, or a PSV or Table Read is performed, or an SFR register is read. In these cases, the execution takes multiple instruction cycles with the additional instruction cycle(s) executed as a NOP. 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.

> **Note:** For more details on the instruction set, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).
>
> For more information on instructions that take more than one instruction cycle to execute, refer to **"CPU"** (DS70359) in the *"dsPIC33/PIC24 Family Reference Manual"*, particularly the **"Instruction Flow Types"** section.

TABLE 28-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
a ∈ {b, c, d}	a is selected from the set of values b, c, d
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write back destination address register ∈ {W13, [W13]+ = 2}
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x0000...0x1FFF}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSb must be '0'
None	Field does not require an entry, can be blank
OA, OB, SA, SB	DSP Status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0...W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }

29.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

29.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

29.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

29.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

44-Terminal Very Thin Leadless Array Package (TL) – 6x6x0.9 mm Body With Exposed Pad [VTLA]

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

(DATUM A OR B)

e/2

L

44X b

⊕ | 0.10 (M) | C | A | B |
| 0.05 (M) | C |
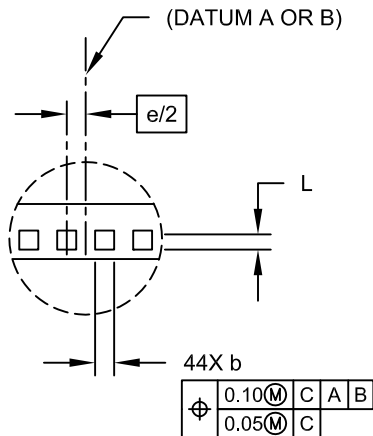
DETAIL A
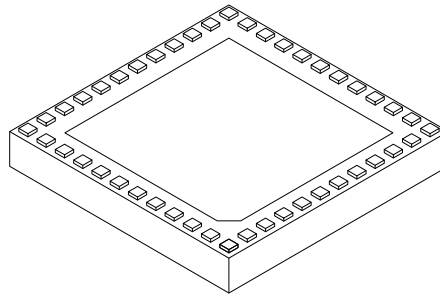
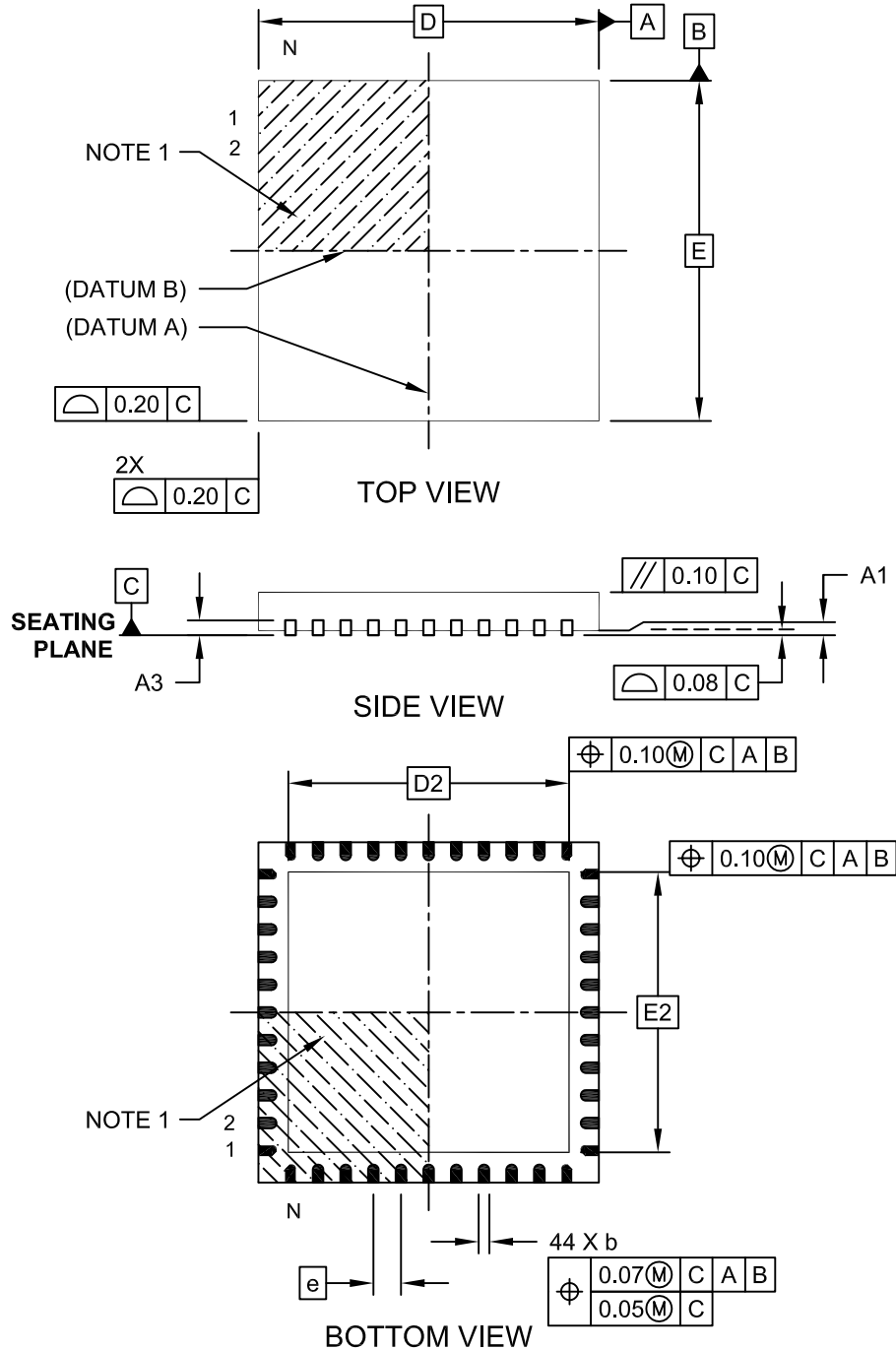|		Units	MILLIMETERS		
Dimension	Limits		MIN	NOM	MAX
Number of Pins	N			44	
Number of Pins per Side	ND			12	
Number of Pins per Side	NE			10	
Pitch	e			0.50 BSC	
Overall Height	A		0.80	0.90	1.00
Standoff	A1		0.025	-	0.075
Overall Width	E			6.00 BSC	
Exposed Pad Width	E2		4.40	4.55	4.70
Overall Length	D			6.00 BSC	
Exposed Pad Length	D2		4.40	4.55	4.70
Contact Width	b		0.20	0.25	0.30
Contact Length	L		0.20	0.25	0.30
Contact-to-Exposed Pad	K		0.20	-	-

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.
    BSC: Basic Dimension. Theoretically exact value shown without tolerances.
    REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-157C Sheet 2 of 2

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

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

D | A | B
N

NOTE 1

1
2

(DATUM B)
(DATUM A)

E

⌀ 0.20 C

2X
⌀ 0.20 C

TOP VIEW

C
SEATING PLANE
A3

// 0.10 C

A1

⌀ 0.08 C

SIDE VIEW

⊕ 0.10Ⓜ C A B

D2

⊕ 0.10Ⓜ C A B

E2

NOTE 1

2
1

N

e

44 X b

⊕ | 0.07Ⓜ C A B
| 0.05Ⓜ C

BOTTOM VIEW

Microchip Technology Drawing C04-103C Sheet 1 of 2