

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.

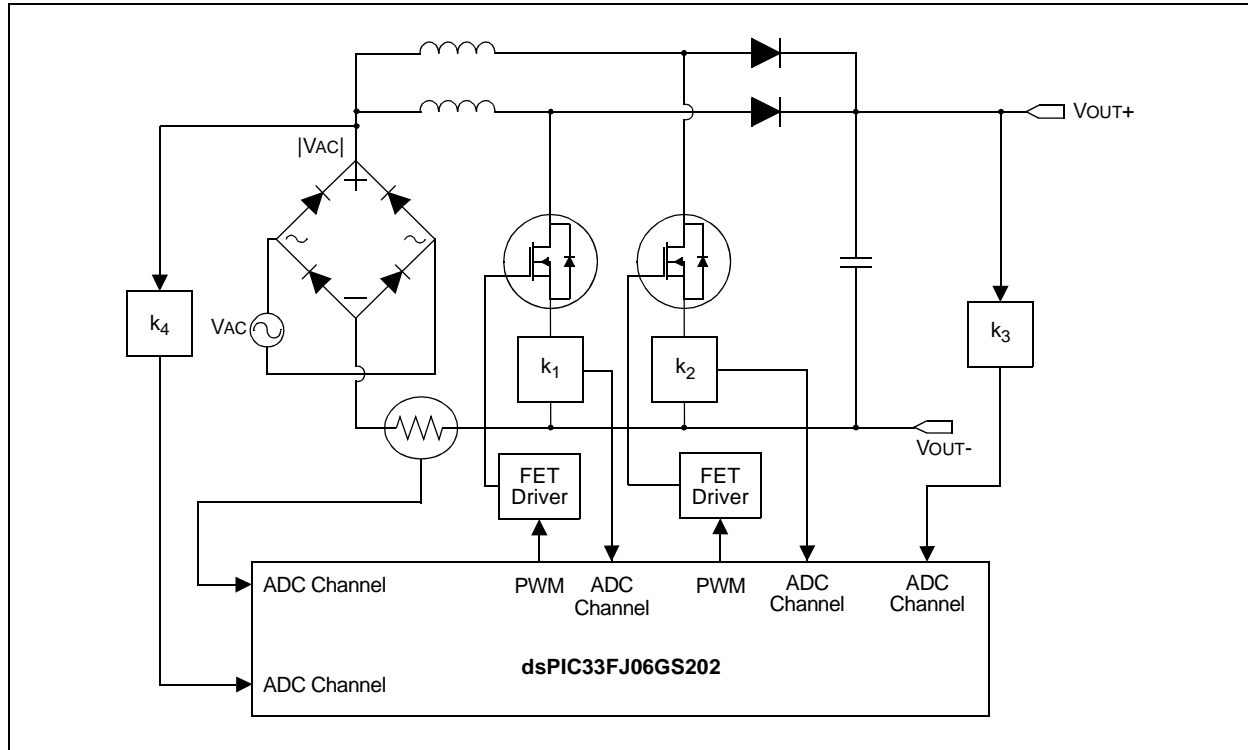
Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Last Time Buy
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPS
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	6KB (6K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj06gs101-i-so

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

FIGURE 2-9: INTERLEAVED PFC



dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

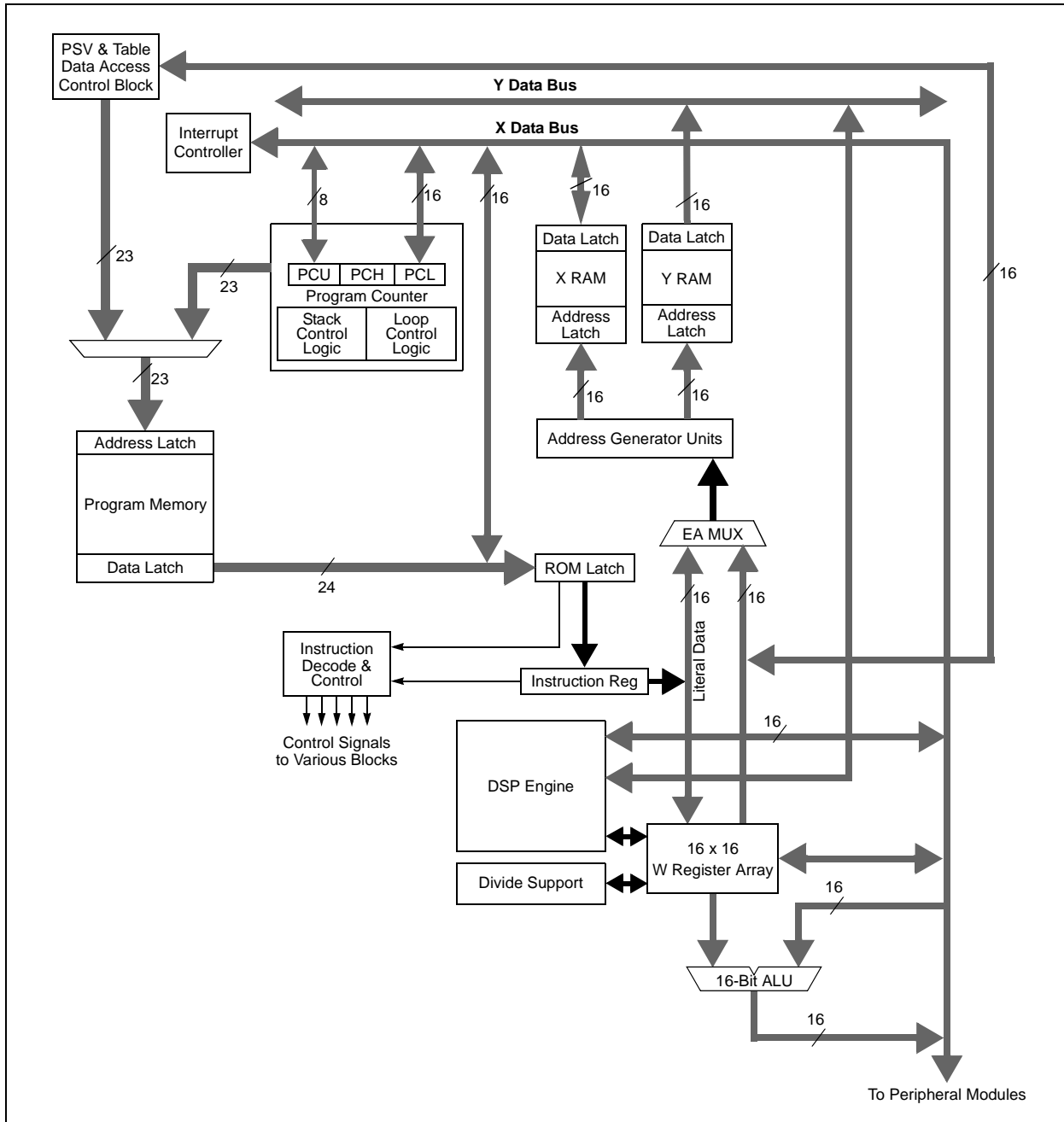
3.3 Special MCU Features

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices feature a 17-bit by 17-bit single-cycle multiplier that is shared by both the MCU ALU and DSP engine. The multiplier can perform signed, unsigned and mixed sign multiplication. Using a 17-bit by 17-bit multiplier for 16-bit by 16-bit multiplication not only allows you to perform mixed sign multiplication, it also achieves accurate results for special operations, such as $(-1.0) \times (1.0)$.

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices support 16/16 and 32/16 divide operations, both fractional and integer. All divide instructions are iterative operations. They must be executed within a REPEAT loop, resulting in a total execution time of 19 instruction cycles. The divide operation can be interrupted during any of those 19 cycles without loss of data.

A 40-bit barrel shifter is used to perform up to a 16-bit left or right shift in a single cycle. The barrel shifter can be used by both MCU and DSP instructions.

FIGURE 3-1: dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 CPU CORE BLOCK DIAGRAM



4.2 Data Address Space

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 CPU has a separate, 16-bit-wide data memory space. The data space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory maps is shown in Figure 4-3.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This arrangement gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when $EA_{<15>} = 0$) is used for implemented memory addresses, while the upper half ($EA_{<15>} = 1$) is reserved for the Program Space Visibility area (see **Section 4.6.3 “Reading Data from Program Memory Using Program Space Visibility”**).

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices implement up to 2 Kbytes of data memory. Should an EA point to a location outside of this area, an all zero word or byte will be returned.

4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC[®] MCU devices and improve data space memory usage efficiency, the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] that results in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word that contains the byte, using the LSB of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel byte-wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register that matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap is generated. If the error occurred on a read, the instruction underway is completed. If the error occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the Least Significant Byte. The Most Significant Byte is not modified.

A sign-extend (SE) instruction is provided to allow user applications to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, user applications can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

4.2.3 SFR SPACE

The first 2 Kbytes of the Near Data Space, from 0x0000 to 0x07FF, is primarily occupied by Special Function Registers (SFRs). These are used by the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control, and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'.

Note: The actual set of peripheral features and interrupts varies by the device. Refer to the corresponding device tables and pinout diagrams for device-specific information.

4.2.4 NEAR DATA SPACE

The 8-Kbyte area, between 0x0000 and 0x1FFF, is referred to as the Near Data Space. Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field or by using Indirect Addressing mode using a Working register as an Address Pointer.

TABLE 4-11: TIMER REGISTER MAP FOR dsPIC33FJ06GS101 AND dsPIC33FJ06GSX02

File Name	SFR 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
TMR1	0100	Timer1 Register																0000
PR1	0102	Period Register 1																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
PR2	010C	Period Register 2																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

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

TABLE 4-12: TIMER REGISTER MAP FOR dsPIC33FJ16GSX02 AND dsPIC33FJ16GSX04

File Name	SFR 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
TMR1	0100	Timer1 Register																0000
PR1	0102	Period Register 1																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																xxxx
TMR3	010A	Timer3 Register																0000
PR2	010C	Period Register 2																FFFF
PR3	010E	Period Register 3																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

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

TABLE 4-13: INPUT CAPTURE REGISTER MAP FOR dsPIC33FJ06GS202

File Name	SFR 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
IC1BUF	0140	Input Capture 1 Register																xxxx
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000

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

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

4.4.3 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

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.

4.5 Bit-Reversed Addressing

Bit-Reversed Addressing mode is intended to simplify data re-ordering 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.

4.5.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

Bit-Reversed Addressing mode is enabled in any of these situations:

- BWM bits (W register selection) in the MODCON register are any value other than 15 (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.

XB<14:0> is the Bit-Reversed Address 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 XB 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 will 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 (XB), 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 should not be enabled together. If an application attempts to do so, Bit-Reversed Addressing will assume priority when active for the X WAGU and X WAGU; Modulo Addressing will be disabled. However, Modulo Addressing will continue to function in the X RAGU.

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.

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 7-17: IEC6: INTERRUPT ENABLE CONTROL REGISTER 6

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
ADCP1IE	ADCP0IE	—	—	—	—	AC4IE	AC3IE
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
AC2IE	—	—	—	—	—	PWM4IE	PWM3IE
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 **ADCP1IE:** ADC Pair 1 Conversion Done Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 14 **ADCP0IE:** ADC Pair 0 Conversion Done Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 13-10 **Unimplemented:** Read as '0'
- bit 9 **AC4IE:** Analog Comparator 4 Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 8 **AC3IE:** Analog Comparator 3 Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 7 **AC2IE:** Analog Comparator 2 Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 6-2 **Unimplemented:** Read as '0'
- bit 1 **PWM4IE:** PWM4 Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled
- bit 0 **PWM3IE:** PWM3 Interrupt Enable bit
 1 = Interrupt request is enabled
 0 = Interrupt request is not enabled

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 7-22: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADIP2	ADIP1	ADIP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

Legend:

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

- bit 15-7 **Unimplemented:** Read as '0'
- bit 6-4 **ADIP<2:0>:** ADC1 Conversion Complete Interrupt Priority bits
 - 111 = Interrupt is Priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is Priority 1
 - 000 = Interrupt source is disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
 - 111 = Interrupt is Priority 7 (highest priority interrupt)
 -
 -
 -
 - 001 = Interrupt is Priority 1
 - 000 = Interrupt source is disabled

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 7-26: IPC14: INTERRUPT PRIORITY CONTROL REGISTER 14

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PSEMIP2	PSEMIP1	PSEMIP0	—	—	—	—
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 15-7 **Unimplemented:** Read as '0'
 bit 6-4 **PSEMIP<2:0>:** PWM Special Event Match Interrupt Priority bits
 111 = Interrupt is Priority 7 (highest priority interrupt)
 •
 •
 •
 001 = Interrupt is Priority 1
 000 = Interrupt source is disabled
 bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-27: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1EIP2	U1EIP1	U1EIP0	—	—	—	—
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 15-7 **Unimplemented:** Read as '0'
 bit 6-4 **U1EIP<2:0>:** UART1 Error Interrupt Priority bits
 111 = Interrupt is Priority 7 (highest priority interrupt)
 •
 •
 •
 001 = Interrupt is Priority 1
 000 = Interrupt source is disabled
 bit 3-0 **Unimplemented:** Read as '0'

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 9-5: PMD6: PERIPHERAL MODULE DISABLE CONTROL REGISTER 6

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PWM4MD	PWM3MD	PWM2MD	PWM1MD
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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 15-12 **Unimplemented:** Read as '0'
- bit 11 **PWM4MD:** PWM Generator 4 Module Disable bit
 1 = PWM Generator 4 module is disabled
 0 = PWM Generator 4 module is enabled
- bit 10 **PWM3MD:** PWM Generator 3 Module Disable bit
 1 = PWM Generator 3 module is disabled
 0 = PWM Generator 3 module is enabled
- bit 9 **PWM2MD:** PWM Generator 2 Module Disable bit
 1 = PWM Generator 2 module is disabled
 0 = PWM Generator 2 module is enabled
- bit 8 **PWM1MD:** PWM Generator 1 Module Disable bit
 1 = PWM Generator 1 module is disabled
 0 = PWM Generator 1 module is enabled
- bit 7-0 **Unimplemented:** Read as '0'

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 10-3: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
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 13-8 **T3CKR<5:0>:** Assign Timer3 External Clock (T3CK) to the Corresponding RPn Pin bits

111111 = Input tied to Vss
 100011 = Input tied to RP35
 100010 = Input tied to RP34
 100001 = Input tied to RP33
 100000 = Input tied to RP32

-
-
-

00000 = Input tied to RP0

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

bit 5-0 **T2CKR<5:0>:** Assign Timer2 External Clock (T2CK) to the Corresponding RPn Pin bits

111111 = Input tied to Vss
 100011 = Input tied to RP35
 100010 = Input tied to RP34
 100001 = Input tied to RP33
 100000 = Input tied to RP32

-
-
-

00000 = Input tied to RP0

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

13.0 INPUT CAPTURE

Note 1: This data sheet summarizes the features of the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to “**Input Capture**” (DS70198) in the “*dsPIC33F/PIC24H Family Reference Manual*”, which is available on 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 input capture module is useful in applications requiring frequency (period) and pulse measurement. dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices support up to two input capture channels.

The input capture module captures the 16-bit value of the selected Time Base register when an event occurs at the ICx pin. The events that cause a capture event are listed below in three categories:

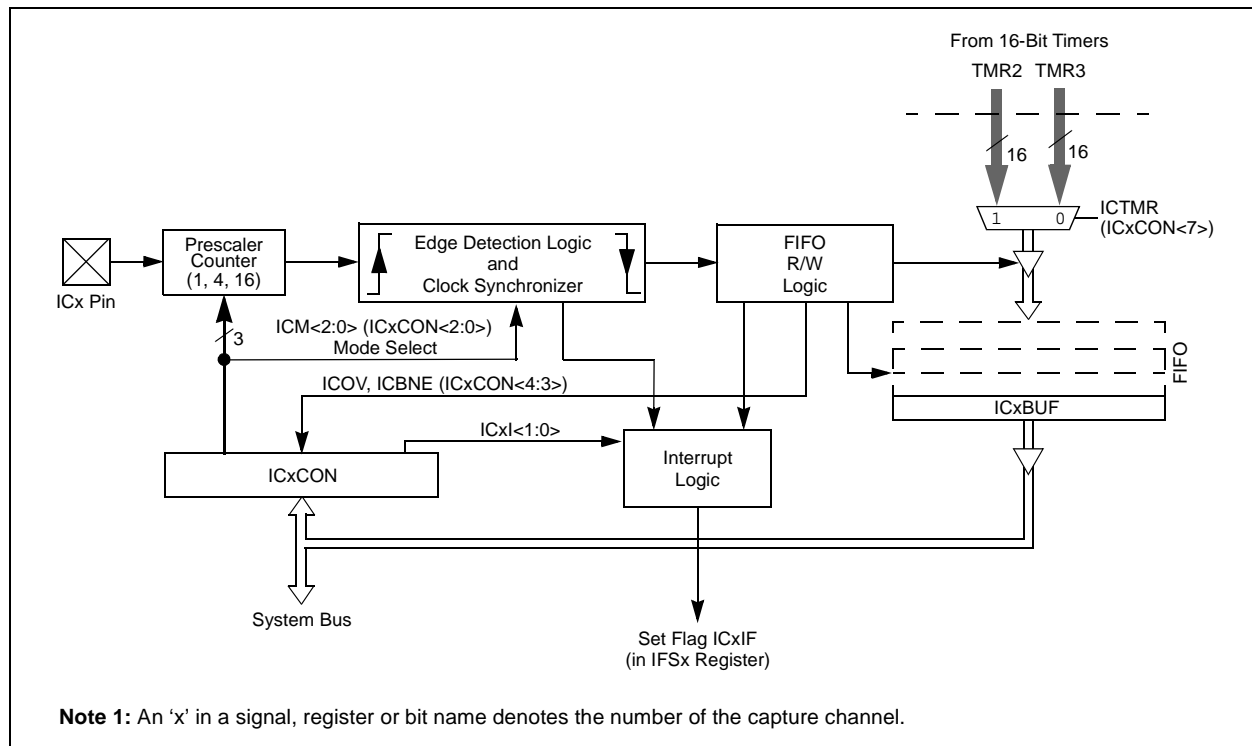
- Simple Capture Event modes:
 - Capture timer value on every falling edge of input at ICx pin
 - Capture timer value on every rising edge of input at ICx pin
- Capture timer value on every edge (rising and falling)
- Prescaler Capture Event modes:
 - Capture timer value on every 4th rising edge of input at ICx pin
 - Capture timer value on every 16th rising edge of input at ICx pin

Each input capture channel can select one of the two 16-bit timers (Timer2 or Timer3) for the time base. The selected timer can use either an internal or external clock.

Other operational features include:

- Device wake-up from capture pin during CPU Sleep and Idle modes
- Interrupt on input capture event
- 4-word FIFO buffer for capture values
 - Interrupt optionally generated after 1, 2, 3 or 4 buffer locations are filled
- Use of input capture to provide additional sources of external interrupts

FIGURE 13-1: INPUT CAPTURE x BLOCK DIAGRAM



dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 15-19: PWMCAPx: PRIMARY PWMx TIME BASE CAPTURE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWMCAP<15:8> ^(1,2)							
bit 15							
bit 8							

R-0	R-0	R-0	R-0	R-0	U-0	U-0	U-0
PWMCAP<7:3> ^(1,2)					—	—	—
bit 7							
bit 0							

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-3 **PWMCAP<15:3>**: Captured PWM Time Base Value bits^(1,2)

The value in this register represents the captured PWM time base value when a leading edge is detected on the current-limit input.

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

Note 1: The capture feature is only available on the primary output (PWMxH).

2: This feature is active only after LEB processing on the current-limit input signal is complete.

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

16.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note 1: This data sheet summarizes the features of the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to “**Serial Peripheral Interface (SPI)**” (DS70206) in the “*dsPIC33F/PIC24H Family Reference Manual*”, which is available on 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 Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices can be serial EEPROMs, shift registers, display drivers, Analog-to-Digital Converters and so on. The SPI module is compatible with SPI and SIOP from Motorola®.

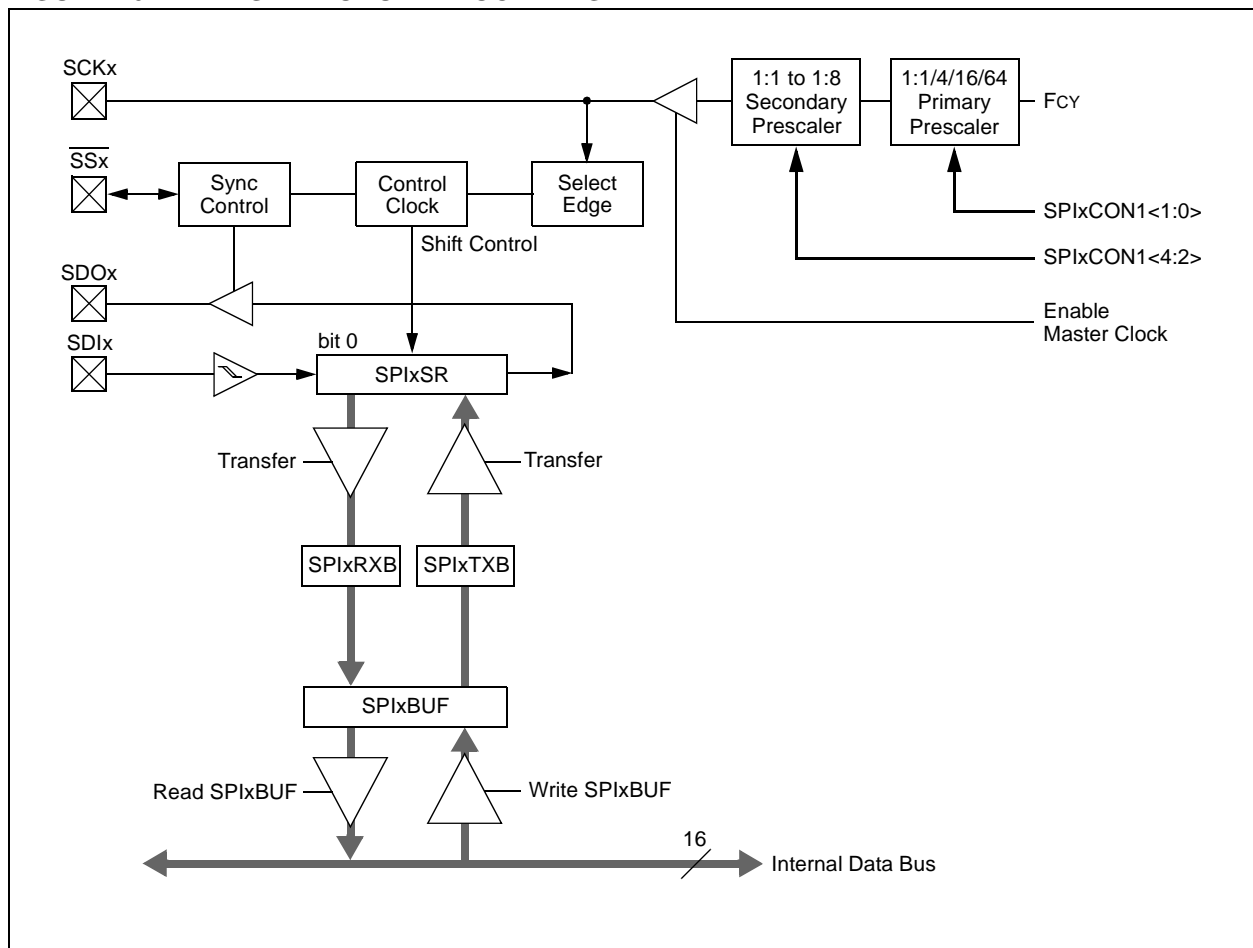
The SPI module consists of a 16-bit shift register, SPIxSR (where x = 1), used for shifting data in and out, and a buffer register, SPIxBUF. A control register, SPIxCON, configures the module. Additionally, a status register, SPIxSTAT, indicates status conditions.

The serial interface consists of the following four pins:

- SDIx (Serial Data Input)
- SDOx (Serial Data Output)
- SCKx (Shift Clock Input Or Output)
- SSx (Active-Low Slave Select).

In Master mode operation, SCK is a clock output; in Slave mode, it is a clock input.

FIGURE 16-1: SPIx MODULE BLOCK DIAGRAM



17.2 I²C Registers

I2CxCON and I2CxSTAT are control and status registers. The I2CxCON register is readable and writable. The lower six bits of I2CxSTAT are read-only. The remaining bits of the I2CxSTAT are read/write:

- I2CxRSR is the shift register used for shifting data internal to the module and the user application has no access to it
- I2CxRCV is the receive buffer and the register to which data bytes are written, or from which data bytes are read
- I2CxTRN is the transmit register to which bytes are written during a transmit operation
- The I2CxADD register holds the slave address
- A status bit, ADD10, indicates 10-Bit Addressing mode
- The I2CxBRG acts as the Baud Rate Generator (BRG) reload value

In receive operations, I2CxRSR and I2CxRCV together form a double-buffered receiver. When I2CxRSR receives a complete byte, it is transferred to I2CxRCV, and an interrupt pulse is generated.

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 17-2: I2CxSTAT: I2Cx STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HSC	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15						bit 8	

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF
bit 7						bit 0	

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

- bit 15 **ACKSTAT:** Acknowledge Status bit
(when operating as I²C™ master, applicable to master transmit operation)
1 = NACK received from slave
0 = ACK received from slave
Hardware is set or clear at end of slave Acknowledge.
- bit 14 **TRSTAT:** Transmit Status bit (when operating as I²C master, applicable to master transmit operation)
1 = Master transmit is in progress (8 bits + ACK)
0 = Master transmit is not in progress
Hardware is set at beginning of master transmission. Hardware is clear at end of slave Acknowledge.
- bit 13-11 **Unimplemented:** Read as '0'
- bit 10 **BCL:** Master Bus Collision Detect bit
1 = A bus collision has been detected during a master operation
0 = No collision
Hardware is set at detection of bus collision.
- bit 9 **GCSTAT:** General Call Status bit
1 = General call address was received
0 = General call address was not received
Hardware is set when address matches general call address. Hardware is clear at Stop detection.
- bit 8 **ADD10:** 10-Bit Address Status bit
1 = 10-bit address was matched
0 = 10-bit address was not matched
Hardware is set at match of 2nd byte of matched 10-bit address. Hardware is clear at Stop detection.
- bit 7 **IWCOL:** I2Cx Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I²C module is busy
0 = No collision
Hardware is set at occurrence of write to I2CxTRN while busy (cleared by software).
- bit 6 **I2COV:** I2Cx Receive Overflow Flag bit
1 = A byte was received while the I2CxRCV register is still holding the previous byte
0 = No overflow
Hardware is set at attempt to transfer I2CxRSR to I2CxRCV (cleared by software).
- bit 5 **D_A:** Data/Address bit (when operating as I²C slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was a device address
Hardware is clear at device address match. Hardware is set by reception of slave byte.

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 17-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

- bit 4 **P:** Stop bit
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last
Hardware is set or clear when Start, Repeated Start or Stop is detected.
- bit 3 **S:** Start bit
1 = Indicates that a Start (or Repeated Start) bit has been detected last
0 = Start bit was not detected last
Hardware is set or clear when Start, Repeated Start or Stop is detected.
- bit 2 **R_W:** Read/Write Information bit (when operating as I²C slave)
1 = Read – indicates data transfer is output from slave
0 = Write – indicates data transfer is input to slave
Hardware is set or clear after reception of an I²C device address byte.
- bit 1 **RBF:** Receive Buffer Full Status bit
1 = Receive is complete, I2CxRCV is full
0 = Receive is not complete, I2CxRCV is empty
Hardware is set when I2CxRCV is written with a received byte. Hardware is clear when software reads I2CxRCV.
- bit 0 **TBF:** Transmit Buffer Full Status bit
1 = Transmit is in progress, I2CxTRN is full
0 = Transmit is complete, I2CxTRN is empty
Hardware is set when software writes to I2CxTRN. Hardware is clear at completion of data transmission.

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

NOTES:

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

REGISTER 19-7: ADCPC2: ANALOG-TO-DIGITAL CONVERT PAIR CONTROL REGISTER 2⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRQEN5	PEND5	SWTRG5	TRGSRC54	TRGSRC53	TRGSRC52	TRGSRC51	TRGSRC50
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRQEN4	PEND4	SWTRG4	TRGSRC44	TRGSRC43	TRGSRC42	TRGSRC41	TRGSRC40
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 **IRQEN5:** Interrupt Request Enable 5 bit
 1 = Enables IRQ generation when requested conversion of Channels AN11 and AN10 is completed
 0 = IRQ is not generated
- bit 14 **PEND5:** Pending Conversion Status 5 bit
 1 = Conversion of Channels AN11 and AN10 is pending; set when selected trigger is asserted
 0 = Conversion is complete
- bit 13 **SWTRG5:** Software Trigger 5 bit
 1 = Starts conversion of AN11 and AN10 (if selected by the TRGSRCx bits)⁽²⁾
 This bit is automatically cleared by hardware when the PEND5 bit is set.
 0 = Conversion has not started

- Note 1:** This register is only implemented in the dsPIC33FJ16GS504 devices.
- 2:** The trigger source must be set as a global software trigger prior to setting this bit to '1'. If other conversions are in progress, then the conversion will be performed when the conversion resources are available.

23.2 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

23.3 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

23.4 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

23.5 MPLAB Assembler, Linker and Librarian for Various Device Families

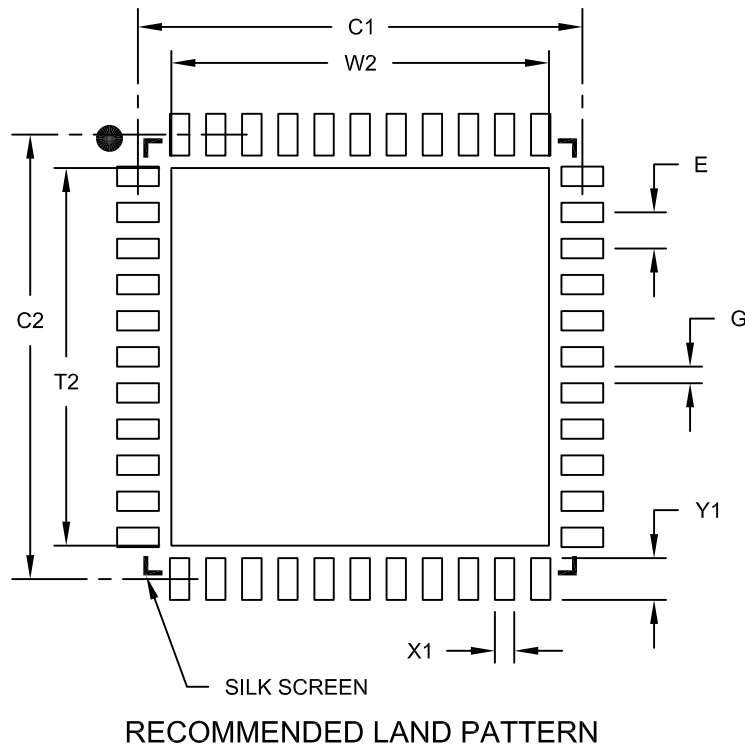
MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object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

dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04

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>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			6.60
Optional Center Pad Length	T2			6.60
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.85
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2103B