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.

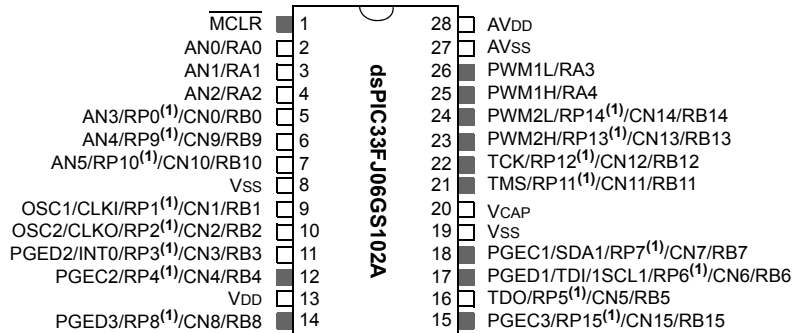**Applications of "Embedded - Microcontrollers"**

Details

Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	6KB (2K x 24)
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 ~ 125°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/dspic33fj06gs001t-e-so

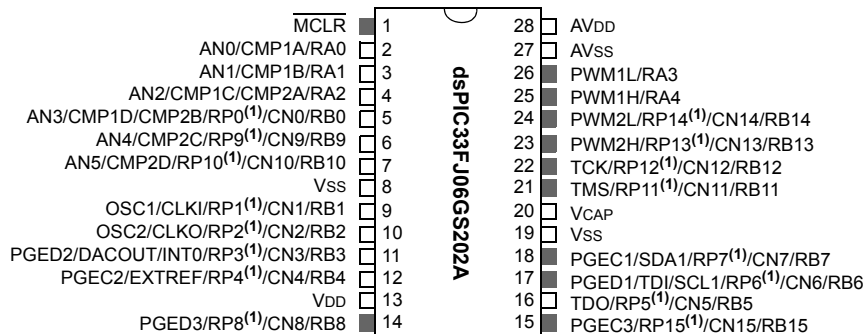# dsPIC33FJ06GS001/101A/102A/202A and dsPIC33FJ09GS302

Pin Diagrams (Continued)
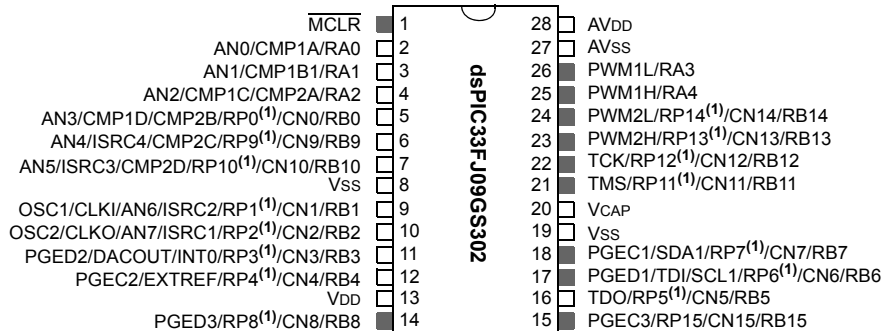
28-Pin SOIC, SPDIP, SSOP

■ = Pins are up to 5V tolerant

dsPIC33FJ06GS102A

Left Pin	#	#	Right Pin
$\overline{MCLR}$	1	28	AV_{DD}
AN0/RA0	2	27	AV_{SS}
AN1/RA1	3	26	PWM1L/RA3
AN2/RA2	4	25	PWM1H/RA4
AN3/RP0[1]/CN0/RB0	5	24	PWM2L/RP14[1]/CN14/RB14
AN4/RP9[1]/CN9/RB9	6	23	PWM2H/RP13[1]/CN13/RB13
AN5/RP10[1]/CN10/RB10	7	22	TCK/RP12[1]/CN12/RB12
V_{SS}	8	21	TMS/RP11[1]/CN11/RB11
OSC1/CLKI/RP1[1]/CN1/RB1	9	20	V_{CAP}
OSC2/CLKO/RP2[1]/CN2/RB2	10	19	V_{SS}
PGED2/INT0/RP3[1]/CN3/RB3	11	18	PGEC1/SDA1/RP7[1]/CN7/RB7
PGEC2/RP4[1]/CN4/RB4	12	17	PGED1/TDI/1SCL1/RP6[1]/CN6/RB6
V_{DD}	13	16	TDO/RP5[1]/CN5/RB5
PGED3/RP8[1]/CN8/RB8	14	15	PGEC3/RP15[1]/CN15/RB15

28-Pin SPDIP, SOIC, SSOP

■ = Pins are up to 5V tolerant

dsPIC33FJ06GS202A

Left Pin	#	#	Right Pin
$\overline{MCLR}$	1	28	AV_{DD}
AN0/CMP1A/RA0	2	27	AV_{SS}
AN1/CMP1B/RA1	3	26	PWM1L/RA3
AN2/CMP1C/CMP2A/RA2	4	25	PWM1H/RA4
AN3/CMP1D/CMP2B/RP0[1]/CN0/RB0	5	24	PWM2L/RP14[1]/CN14/RB14
AN4/CMP2C/RP9[1]/CN9/RB9	6	23	PWM2H/RP13[1]/CN13/RB13
AN5/CMP2D/RP10[1]/CN10/RB10	7	22	TCK/RP12[1]/CN12/RB12
V_{SS}	8	21	TMS/RP11[1]/CN11/RB11
OSC1/CLKI/RP1[1]/CN1/RB1	9	20	V_{CAP}
OSC2/CLKO/RP2[1]/CN2/RB2	10	19	V_{SS}
PGED2/DACOUT/INT0/RP3[1]/CN3/RB3	11	18	PGEC1/SDA1/RP7[1]/CN7/RB7
PGEC2/EXTREF/RP4[1]/CN4/RB4	12	17	PGED1/TDI/SCL1/RP6[1]/CN6/RB6
V_{DD}	13	16	TDO/RP5[1]/CN5/RB5
PGED3/RP8[1]/CN8/RB8	14	15	PGEC3/RP15[1]/CN15/RB15

28-Pin SPDIP, SOIC, SSOP

■ = Pins are up to 5V tolerant

dsPIC33FJ09GS302

Left Pin	#	#	Right Pin
$\overline{MCLR}$	1	28	AV_{DD}
AN0/CMP1A/RA0	2	27	AV_{SS}
AN1/CMP1B1/RA1	3	26	PWM1L/RA3
AN2/CMP1C/CMP2A/RA2	4	25	PWM1H/RA4
AN3/CMP1D/CMP2B/RP0[1]/CN0/RB0	5	24	PWM2L/RP14[1]/CN14/RB14
AN4/ISRC4/CMP2C/RP9[1]/CN9/RB9	6	23	PWM2H/RP13[1]/CN13/RB13
AN5/ISRC3/CMP2D/RP10[1]/CN10/RB10	7	22	TCK/RP12[1]/CN12/RB12
V_{SS}	8	21	TMS/RP11[1]/CN11/RB11
OSC1/CLKI/AN6/ISRC2/RP1[1]/CN1/RB1	9	20	V_{CAP}
OSC2/CLKO/AN7/ISRC1/RP2[1]/CN2/RB2	10	19	V_{SS}
PGED2/DACOUT/INT0/RP3[1]/CN3/RB3	11	18	PGEC1/SDA1/RP7[1]/CN7/RB7
PGEC2/EXTREF/RP4[1]/CN4/RB4	12	17	PGED1/TDI/SCL1/RP6[1]/CN6/RB6
V_{DD}	13	16	TDO/RP5[1]/CN5/RB5
PGED3/RP8[1]/CN8/RB8	14	15	PGEC3/RP15/CN15/RB15

Note 1: The RPn pins can be used by any remappable peripheral. See **Table 1** for the list of available peripherals.

2.5 ICSP™ Pins

The PGECx and PGEDx pins are used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ohms, not to exceed 100 Ohms.

Pull-up resistors, series diodes, and capacitors on the PGECx and PGEDx pins, are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and Input Voltage High (V_{IH}) and Input Voltage Low (V_{IL}) pin requirements.

Ensure that the "Communication Channel Select" (i.e., PGECx/PGEDx pins), programmed into the device matches the physical connections for the ICSP to MPLAB® ICD 3 or MPLAB REAL ICE™.

For more information on MPLAB ICD 3 and REAL ICE connection requirements, refer to the following documents that are available on the Microchip web site (**www.microchip.com):**

- *"Using MPLAB® ICD 3"* (poster) (DS51765)
- *"Multi-Tool Design Advisory"* (DS51764)
- *"MPLAB® REAL ICE™ In-Circuit Emulator User's Guide"* (DS51616)
- *"Using MPLAB® REAL ICE™"* (poster) (DS51749)

2.6 External Oscillator Pins

Many DSCs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 8.0 "Oscillator Configuration"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is shown in Figure 2-3.

FIGURE 2-3: **SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**
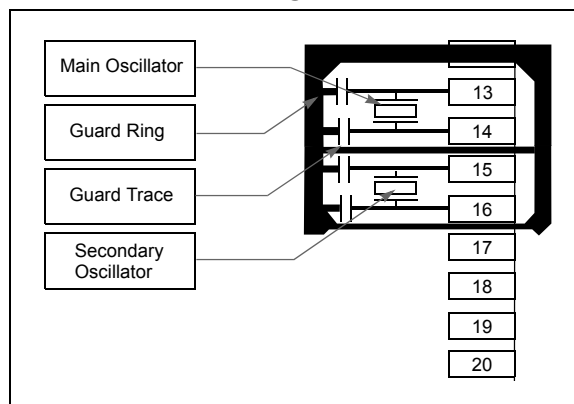
NOTES:

3.0 CPU

Note 1: This data sheet summarizes the features of the dsPIC33FJ06GS001/101A/102A/202A and dsPIC33FJ09GS302 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 2. "CPU"** (DS70204) in the *"dsPIC33F/PIC24H 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 CPU module has a 16-bit (data) modified Harvard architecture with an enhanced instruction set, including significant support for DSP. The CPU has a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M x 24 bits of user program memory space. The actual amount of program memory implemented varies from device to device. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the DO and REPEAT instructions, both of which are interruptible at any point.

The dsPIC33FJ06GS001/101A/102A/202A and dsPIC33FJ09GS302 devices have sixteen, 16-bit working registers in the programmer's model. Each of the working registers can serve as a Data, Address or Address Offset register. The sixteenth working register (W15) operates as a software Stack Pointer (SP) for interrupts and calls.

There are two classes of instruction: MCU and DSP. These two instruction classes are seamlessly integrated into a single CPU. The instruction set includes many addressing modes and is designed for optimum C compiler efficiency. For most instructions, the devices are capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing A + B = C operations to be executed in a single cycle.

A block diagram of the CPU is shown in Figure 3-1, and the programmer's model is shown in Figure 3-2.

3.1 Data Addressing Overview

The data space can be addressed as 32K words or 64 Kbytes and is split into two blocks, referred to as X and Y data memory. Each memory block has its own independent Address Generation Unit (AGU). The MCU class of instructions operates solely through the X memory AGU, which accesses the entire memory map as one linear data space. Certain DSP instructions operate through the X and Y AGUs to support dual operand reads, which splits the data address space into two parts. The X and Y data space boundary is device-specific.

Overhead-free circular buffers (Modulo Addressing mode) are supported in both X and Y address spaces. The Modulo Addressing removes the software boundary checking overhead for DSP algorithms. Furthermore, the X AGU Circular Addressing can be used with any of the MCU class of instructions. The X AGU also supports Bit-Reversed Addressing to greatly simplify input or output data reordering for radix-2 FFT algorithms.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K program word boundary defined by the 8-bit Program Space Visibility Page (PSVPAG) register. The program-to-data space mapping feature lets any instruction access program space as if it were data space.

3.2 DSP Engine Overview

The DSP engine features a high-speed, 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bidirectional barrel shifter. The barrel shifter is capable of shifting a 40-bit value up to 16 bits, right or left, in a single cycle. The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance. The MAC instruction and other associated instructions can concurrently fetch two data operands from memory while multiplying two W registers and accumulating and optionally saturating the result in the same cycle. This instruction functionality requires that the RAM data space be split for these instructions and linear for all others. Data space partitioning is achieved in a transparent and flexible manner through dedicating certain working registers to each address space.

TABLE 4-16: I2C1 REGISTER MAP

SFR 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
I2C1RCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	Baud Rate Generator Register									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	—	—	—	—	—	—	Address Register										0000
I2C1MSK	020C	—	—	—	—	—	—	AMSK<9:0>										0000

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

TABLE 4-17: UART1 REGISTER MAP FOR dsPIC33FJ06GS101A, dsPIC33FJ06GS102A, dsPIC33FJ06GS202A AND dsPIC33FJ09GS302

SFR 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
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	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	—	—	—	—	—	—	—	UART Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART Receive Register									0000
U1BRG	0228	Baud Rate Generator Prescaler																0000

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

TABLE 4-18: SPI1 REGISTER MAP FOR dsPIC33FJ06GS101A, dsPIC33FJ06GS102A, dsPIC33FJ06GS202A AND dsPIC33FJ09GS302

SFR 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
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>		0000
SPI1CON2	0244	FRMEN	SPIFSD	FRMPOL	—	—	—	—	—	—	—	—	—	—	—	FRMDLY	—	0000
SPI1BUF	0248	SPI1 Transmit and Receive Buffer Register																0000

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

TABLE 4-32: SYSTEM CONTROL REGISTER MAP

SFR 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
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	xxxx[1]
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	IOLOCK	LOCK	—	CF	—	—	OSWEN	0300[2]
CLKDIV	0744	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>		—	PLLPRE<4:0>					3040
PLLFBD	0746	—	—	—	—	—	—	PLLDIV<8:0>										0030
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000
LFSR	074C	—	LFSR<14:0>															0000
REFOCON	074E	ROON	—	ROSSLP	ROSEL	RODIV<3:0>				—	—	—	—	—	—	—	—	0000
ACLKCON	0750	ENAPLL	APLLCK	SELACLK	—	—	APSTSCLR<2:0>			ASRCSEL	FRCSEL	—	—	—	—	—	—	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The RCON register Reset values are dependent on the type of Reset.
2: The OSCCON register Reset values are dependent on the FOSCx Configuration bits and on type of Reset.

TABLE 4-33: NVM 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
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP<3:0>				0000[1]
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY<7:0>								0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Reset value shown is for POR only. Value on other Reset states is dependent on the state of memory write or erase operations at the time of Reset.

4.7.2 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the lower word of any address within the program space without going through data space. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to data space addresses. Program memory can thus be regarded as two 16-bit wide word address spaces, residing side by side, each with the same address range. TBLRDL and TBLWTL access the space that contains the least significant data word. TBLRDH and TBLWTH access the space that contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

- TBLRDL (Table Read Low):
  - In Word mode, this instruction maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>)

- In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when byte select is '1'; the lower byte is selected when it is '0'.

- TBLRDH (Table Read High):
  - In Word mode, this instruction maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the 'phantom byte', will always be '0'.
  - In Byte mode, this instruction maps the upper or lower byte of the program word to D<7:0> of the data address, in the TBLRDL instruction. The data is always '0' when the upper 'phantom' byte is selected (Byte Select = 1).

Similarly, two table instructions, TBLWTH and TBLWTL, are used to write individual bytes or words to a program space address. The details of their operation are explained in **Section 5.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Page register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = 0, the table page is located in the user memory space. When TBLPAG<7> = 1, the page is located in configuration space.

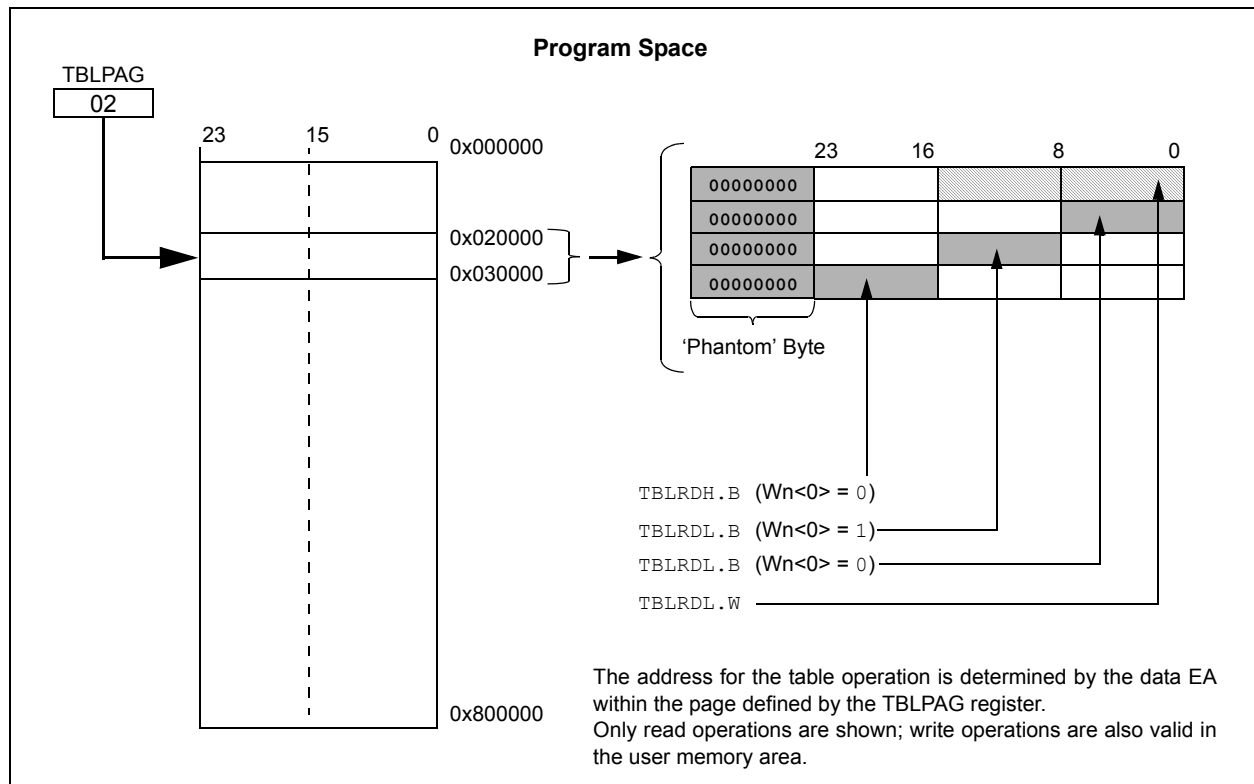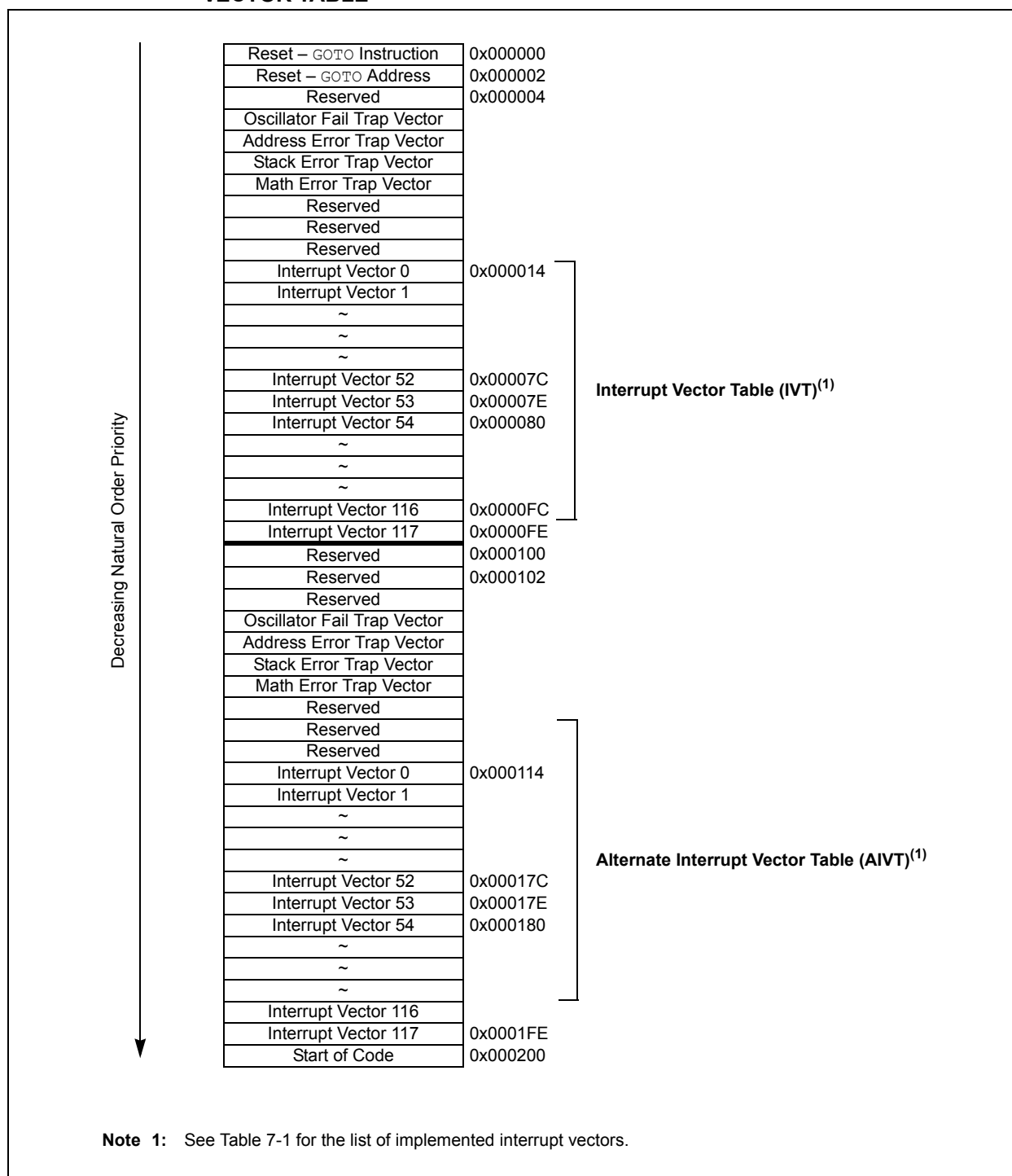**FIGURE 4-9: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS**

The address for the table operation is determined by the data EA within the page defined by the TBLPAG register.
Only read operations are shown; write operations are also valid in the user memory area.

FIGURE 7-1: **dsPIC33FJ06GS001/101A/102A/202A and dsPIC33FJ09GS302 INTERRUPT VECTOR TABLE**

Reset – `GOTO` Instruction	0x000000
Reset – `GOTO` Address	0x000002
Reserved	0x000004
Oscillator Fail Trap Vector	
Address Error Trap Vector	
Stack Error Trap Vector	
Math Error Trap Vector	
Reserved	
Reserved	
Reserved	
Interrupt Vector 0	0x000014
Interrupt Vector 1	
~	
~	
~	
Interrupt Vector 52	0x00007C
Interrupt Vector 53	0x00007E
Interrupt Vector 54	0x000080
~	
~	
~	
Interrupt Vector 116	0x0000FC
Interrupt Vector 117	0x0000FE
Reserved	0x000100
Reserved	0x000102
Reserved	
Oscillator Fail Trap Vector	
Address Error Trap Vector	
Stack Error Trap Vector	
Math Error Trap Vector	
Reserved	
Reserved	
Reserved	
Interrupt Vector 0	0x000114
Interrupt Vector 1	
~	
~	
~	
Interrupt Vector 52	0x00017C
Interrupt Vector 53	0x00017E
Interrupt Vector 54	0x000180
~	
~	
~	
Interrupt Vector 116	
Interrupt Vector 117	0x0001FE
Start of Code	0x000200

Decreasing Natural Order Priority

Interrupt Vector Table (IVT)[1]

Alternate Interrupt Vector Table (AIVT)[1]

Note 1: See Table 7-1 for the list of implemented interrupt vectors.

REGISTER 7-11:   IFS7: INTERRUPT FLAG STATUS REGISTER 7

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	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	—	ADCP6IF	—	—	ADCP3IF[1]	ADCP2IF[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 15-5   **Unimplemented:** Read as '0'

bit 4   **ADCP6IF:** ADC Pair 6 Conversion Done Interrupt Flag Status bit
　　　1 = Interrupt request has occurred
　　　0 = Interrupt request has not occurred

bit 3-2   **Unimplemented:** Read as '0'

bit 1   **ADCP3IF:** ADC Pair 3 Conversion Done Interrupt Flag Status bit[1]
　　　1 = Interrupt request has occurred
　　　0 = Interrupt request has not occurred

bit 0   **ADCP2IF:** ADC Pair 2 Conversion Done Interrupt Flag Status bit[2]
　　　1 = Interrupt request has occurred
　　　0 = Interrupt request has not occurred

Note 1: This bit is not implemented in dsPIC33FJ06GS102A/202A devices.
　　2: This bit is not implemented in dsPIC33FJ06GS001/101A devices.

10.2 Open-Drain Configuration

In addition to the PORT, LAT and TRIS registers for data control, some digital only port pins can also be individually configured for either digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than V_{DD} (for example, 5V), on any desired 5V tolerant pins, by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum V_{IH} specification.

Refer to the **"Pin Diagrams"** section for the available pins and their functionality.

10.3 Configuring Analog Port Pins

The ADPCFG and TRIS registers control the operation of the Analog-to-Digital port pins. The port pins that are to function as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

The ADPCFG register has a default value of 0x0000; therefore, all pins that share ANx functions are analog (not digital) by default.

When the PORT register is read, all pins configured as analog input channels will read as cleared (a low level).

Pins configured as digital inputs will not convert an analog input. Analog levels on any pin, defined as a digital input (including the ANx pins), can cause the input buffer to consume current that exceeds the device specifications.

10.4 I/O Port Write/Read Timing

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a `NOP`. An example is shown in Example 10-1.

10.5 Input Change Notification

The Input Change Notification function of the I/O ports allows devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature can detect input Change-of-States (COS), even in Sleep mode when the clocks are disabled. Depending on the device pin count, up to 16 external signals (CNx pin) can be selected (enabled) for generating an interrupt request on a Change-of-State.

Four control registers are associated with the CN module. The CNEN1 register contains the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pin.

Each CN pin also has a weak pull-up connected to it. The pull-ups act as a current source connected to the pin and eliminate the need for external resistors when the push button or keypad devices are connected. The pull-ups are enabled separately, using the CNPU1 register, which contains the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-ups for the corresponding pins.

Note:	Pull-ups on change notification pins should always be disabled when the port pin is configured as a digital output.

EXAMPLE 10-1: PORT WRITE/READ EXAMPLE

```
MOV   0xFF00, W0        ; Configure PORTB<15:8> as inputs
MOV   W0, TRISBB        ; and PORTB<7:0> as outputs
NOP                     ; Delay 1 cycle
BTSS  PORTB, #13        ; Next Instruction
```

REGISTER 15-13: TRGCONx: PWMx TRIGGER CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
	TRGDIV	<3:0>		—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTM[1]	—			TRGSTRT	<5:0>		
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 **TRGDIV<3:0>:** Trigger # Output Divider bits

1111 = Trigger output for every 16th trigger event
1110 = Trigger output for every 15th trigger event
1101 = Trigger output for every 14th trigger event
1100 = Trigger output for every 13th trigger event
1011 = Trigger output for every 12th trigger event
1010 = Trigger output for every 11th trigger event
1001 = Trigger output for every 10th trigger event
1000 = Trigger output for every 9th trigger event
0111 = Trigger output for every 8th trigger event
0110 = Trigger output for every 7th trigger event
0101 = Trigger output for every 6th trigger event
0100 = Trigger output for every 5th trigger event
0011 = Trigger output for every 4th trigger event
0010 = Trigger output for every 3rd trigger event
0001 = Trigger output for every 2nd trigger event
0000 = Trigger output for every trigger event

bit 11-8 **Unimplemented:** Read as '0'

bit 7 **DTM:** Dual Trigger Mode bit[1]

1 = Secondary trigger event is combined with the primary trigger event to create the PWM trigger.
0 = Secondary trigger event is not combined with the primary trigger event to create the PWM trigger; two separate PWM triggers are generated

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

bit 5-0 **TRGSTRT<5:0>:** Trigger Postscaler Start Enable Select bits

111111 = Wait 63 PWM cycles before generating the first trigger event after the module is enabled
•
•
•
000010 = Wait 2 PWM cycles before generating the first trigger event after the module is enabled
000001 = Wait 1 PWM cycle before generating the first trigger event after the module is enabled
000000 = Wait 0 PWM cycle before generating the first trigger event after the module is enabled

Note 1: The secondary generator cannot generate PWM trigger interrupts.

16.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note 1: This data sheet summarizes the features of the dsPIC33FJ06GS001/101A/102A/202A and dsPIC33FJ09GS302 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 18. "Serial Peripheral Interface (SPI)"** (DS70206) of the "*dsPIC33F/PIC24H 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 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, etc. The SPI module is compatible with Motorola® SPI and SIOP.

Each SPI module consists of a 16-bit shift register, SPIxSR (where x = 1 or 2), 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 4 pins:

- SDIx (serial data input)
- SDOx (serial data output)
- SCKx (shift clock input or output)
- $\overline{SSx}$ (active-low slave select)

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

FIGURE 16-1:   SPI MODULE BLOCK DIAGRAM
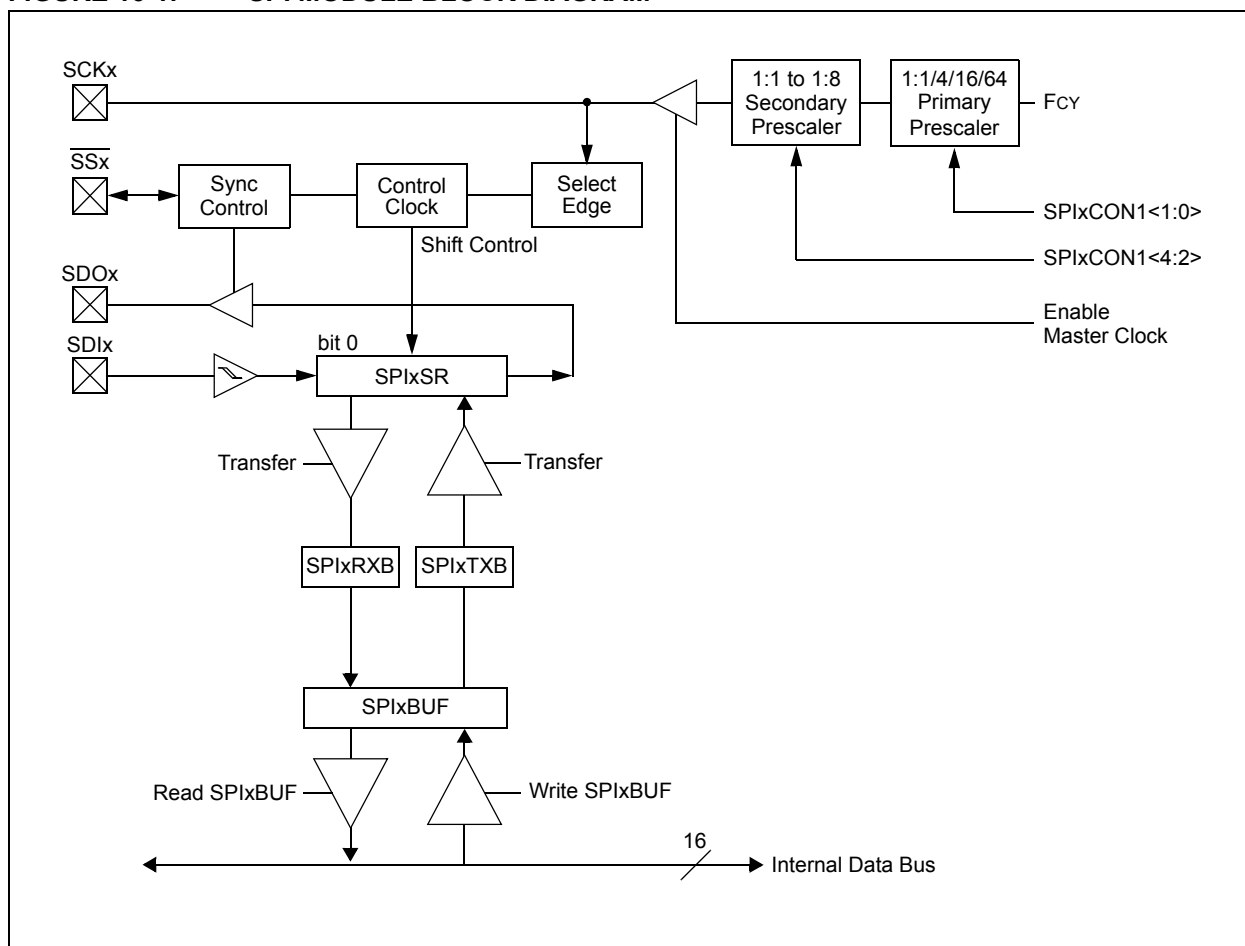
18.1   UART Helpful Tips

1. In multinode, direct-connect UART networks, UART receive inputs react to the complementary logic level defined by the URXINV bit (U1MODE<4>), which defines the Idle state, the default of which is logic high, (i.e., URXINV = 0). Because remote devices do not initialize at the same time, it is likely that one of the devices, because the RX line is floating, will trigger a Start bit detection and will cause the first byte received after the device has been initialized to be invalid. To avoid this situation, the user should use a pull-up or pull-down resistor on the RX pin, depending on the value of the URXINV bit.

    a) If URXINV = 0, use a pull-up resistor on the RX pin.

    b) If URXINV = 1, use a pull-down resistor on the RX pin.

2. The first character received on a wake-up from Sleep mode, caused by activity on the UxRX pin of the UART module, will be invalid. In Sleep mode, peripheral clocks are disabled. By the time the oscillator system has restarted and stabilized from Sleep mode, the baud rate bit sampling clock, relative to the incoming UxRX bit timing, is no longer synchronized. This results in the first character being invalid; this is to be expected.

18.2   UART Resources

Many useful resources related to UART are provided on the Microchip web site (www.microchip.com).

18.2.1   KEY RESOURCES

*   **Section 17. "UART"** (DS70188) in the *"dsPIC33F/PIC24H Family Reference Manual"*
*   Code Samples
*   Application Notes
*   Software Libraries
*   Webinars
*   All related *"dsPIC33F/PIC24H Family Reference Manual"* Sections
*   Development Tools

19.4   ADC Control Registers

The ADC module uses the following control and status registers:

- ADCON: ADC Control Register
- ADSTAT: ADC Status Register
- ADBASE: ADC Base Register(1)
- ADPCFG: ADC Port Configuration Register
- ADCPC0: ADC Convert Pair Control Register 0
- ADCPC1: ADC Convert Pair Control Register 1
- ADCPC3: ADC Convert Pair Control Register 3(1)

The ADCON register controls the operation of the ADC module. The ADSTAT register displays the status of the conversion processes. The ADPCFG register configures the port pins as analog inputs or as digital I/Os. The ADCPCx registers control the triggering of the ADC conversions. See Register 19-1 through Register 19-7 for detailed bit configurations.

Note:	A unique feature of the ADC module is its ability to sample inputs in an asynchronous manner. Individual Sample-and-Hold circuits can be triggered independently of each other.

REGISTER 19-1:   ADCON: ADC CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0
ADON	—	ADSIDL	SLOWCLK(1)	—	GSWTRG	—	FORM(1)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-1	R/W-1
EIE(1)	ORDER(1)	SEQSAMP(1)	ASYNCSAMP(1)	—	ADCS<2:0>(1)		
bit 7							bit 0

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

bit 15   **ADON:** ADC Operating Mode bit
1 = ADC module is operating
0 = ADC module is off

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

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

bit 12   **SLOWCLK:** Enable Slow Clock Divider bit(1)
1 = ADC is clocked by the auxiliary PLL (ACLK)
0 = ADC is clocked by the primary PLL (F$_{VCO}$)

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

bit 10   **GSWTRG:** Global Software Trigger bit
When this bit is set by the user, it will trigger conversions if selected by the TRGSRC<4:0> bits in the ADCPCx registers. This bit must be cleared by the user prior to initiating another global trigger (i.e., this bit is not auto-clearing).

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

Note 1:   This control bit can only be changed while the ADC is disabled (ADON = 0).

REGISTER 19-5:   ADCPC0: ADC CONVERT PAIR CONTROL REGISTER 0 (CONTINUED)

bit 7   **IRQEN0:** Interrupt Request Enable 0 bit

1 = Enables IRQ generation when requested conversion of channels AN1 and AN0 is completed
0 = IRQ is not generated

bit 6   **PEND0:** Pending Conversion Status 0 bit

1 = Conversion of channels AN1 and AN0 is pending; set when selected trigger is asserted
0 = Conversion is complete

bit 5   **SWTRG0:** Software Trigger 0 bit
1 = Starts conversion of AN1 and AN0 (if selected by the TRGSRCx bits)[1]
This bit is automatically cleared by hardware when the PEND0 bit is set.
0 = Conversion has not started

bit 4-0   **TRGSRC0<4:0>:** Trigger 0 Source Selection bits
Selects trigger source for conversion of analog channels AN1 and AN0.
11111 = Timer2 period match
•
•
•
11011 = Reserved
11010 = PWM Generator 4 current-limit ADC trigger
11001 = Reserved
11000 = PWM Generator 2 current-limit ADC trigger
10111 = PWM Generator 1 current-limit ADC trigger
10110 = Reserved
•
•
•
10010 = Reserved
10001 = PWM Generator 4 secondary trigger is selected
10000 = Reserved
01111 = PWM Generator 2 secondary trigger is selected
01110 = PWM Generator 1 secondary trigger is selected
01101 = Reserved
01100 = Timer1 period match
•
•
•
01000 = Reserved
00111 = PWM Generator 4 primary trigger is selected
00110 = Reserved
00101 = PWM Generator 2 primary trigger is selected
00100 = PWM Generator 1 primary trigger is selected
00011 = PWM Special Event Trigger is selected
00010 = Global software trigger is selected
00001 = Individual software trigger is selected
00000 = No conversion is enabled

Note 1:   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 conversion will be performed when the conversion resources are available.

TABLE 22-3: dsPIC33F CONFIGURATION BITS DESCRIPTION

Bit Field	Description
GCP	General Segment Code-Protect bit `1` = User program memory is not code-protected `0` = Code protection is enabled for the entire program memory space
GWRP	General Segment Write-Protect bit `1` = User program memory is not write-protected `0` = User program memory is write-protected
IESO	Two-Speed Oscillator Start-up Enable bit `1` = Start up device with FRC, then automatically switch to the user-selected oscillator source when ready `0` = Start up device with user-selected oscillator source
FNOSC<2:0>	Oscillator Selection bits `111` = Fast RC Oscillator with divide-by-N (FRCDIVN) `110` = Reserved; do not use `101` = Low-Power RC Oscillator (LPRC) `100` = Reserved; do not use `011` = Primary Oscillator with PLL module (MS + PLL, EC + PLL) `010` = Primary Oscillator (MS, HS, EC) `001` = Fast RC Oscillator with divide-by-N with PLL module (FRCDIVN + PLL) `000` = Fast RC Oscillator (FRC)
FCKSM<1:0>	Clock Switching Mode bits `1x` = Clock switching is disabled, Fail-Safe Clock Monitor is disabled `01` = Clock switching is enabled, Fail-Safe Clock Monitor is disabled `00` = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
IOL1WAY	Peripheral Pin Select Configuration bit `1` = Allow only one reconfiguration `0` = Allow multiple reconfigurations
OSCIOFNC	OSC2 Pin Function bit (except in MS and HS modes) `1` = OSC2 is the clock output `0` = OSC2 is the general purpose digital I/O pin
POSCMD<1:0>	Primary Oscillator Mode Select bits `11` = Primary Oscillator is disabled `10` = HS Crystal Oscillator mode (10 MHz-32 MHz) `01` = MS Crystal Oscillator mode (3 MHz-10 MHz) `00` = EC (External Clock) mode (DC-32 MHz)
FWDTEN	Watchdog Timer Enable bit `1` = Watchdog Timer is always enabled (LPRC oscillator cannot be disabled; clearing the SWDTEN bit in the RCON register will have no effect) `0` = Watchdog Timer is enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register)
WDTPRE	Watchdog Timer Prescaler bit `1` = 1:128 `0` = 1:32
WDTPOST<3:0>	Watchdog Timer Postscaler bits `1111` = 1:32,768 `1110` = 1:16,384 • • • `0001` = 1:2 `0000` = 1:1

24.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C® for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

24.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

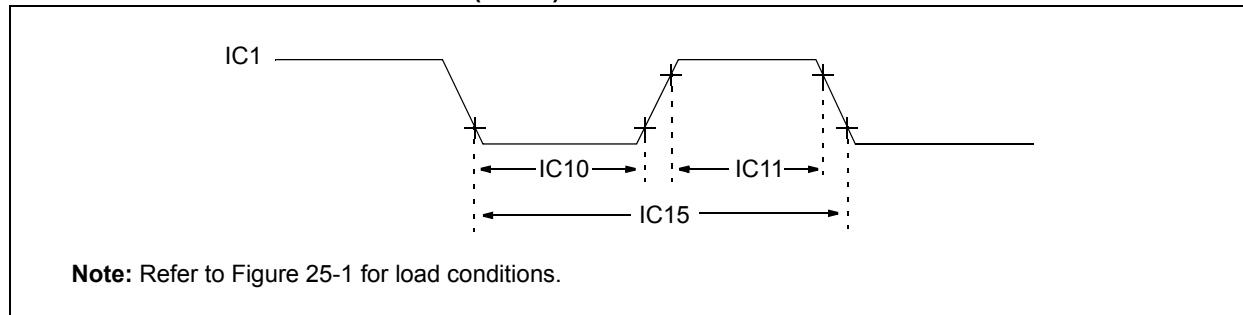**FIGURE 25-6:** INPUT CAPTURE (CAP1) TIMING CHARACTERISTICS

Note: Refer to Figure 25-1 for load conditions.

TABLE 25-25: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param.	Symbol	Characteristic[1]		Min.	Max.	Units	Conditions
IC10	TccL	IC1 Input Low Time	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
IC11	TccH	IC1 Input High Time	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
IC15	TccP	IC1 Input Period		(TCY + 40)/N	—	ns	N = prescale value (1, 4, 16)

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

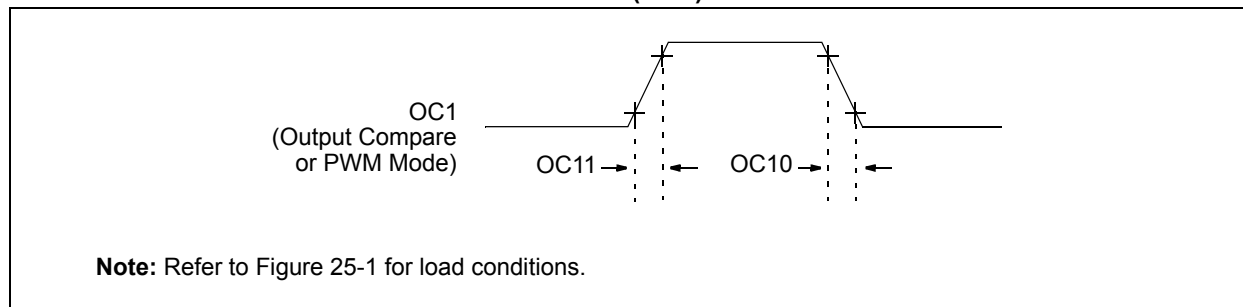**FIGURE 25-7:** OUTPUT COMPARE MODULE (OC1) TIMING CHARACTERISTICS

Note: Refer to Figure 25-1 for load conditions.

TABLE 25-26: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

| AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param.	Symbol	Characteristic[1]	Min.	Typ.	Max.	Units	Conditions
OC10	TccF	OC1 Output Fall Time	—	—	—	ns	See Parameter DO32
OC11	TccR	OC1 Output Rise Time	—	—	—	ns	See Parameter DO31

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

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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

RECOMMENDED LAND PATTERN
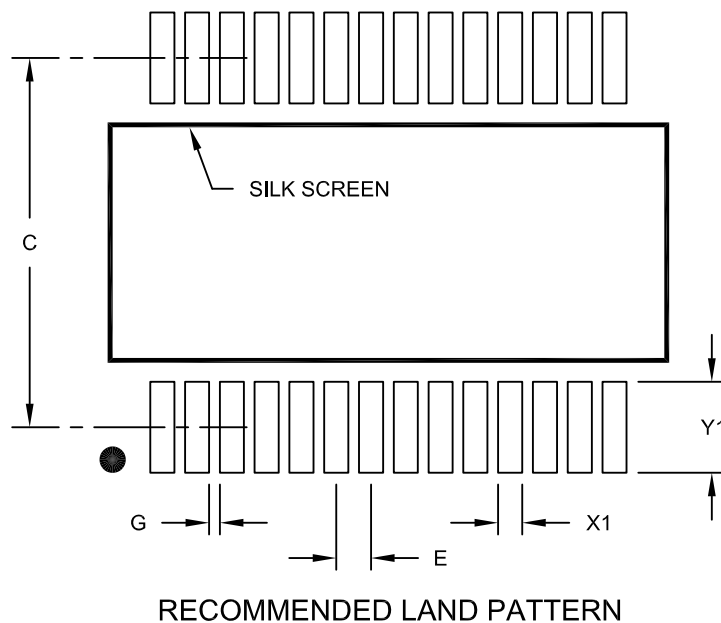
	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

TOP VIEW

SIDE VIEW

VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2