Microchip Technology - DSPIC33FJ06GS202A-E/SS Datasheet

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	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	21
Program Memory Size	6KB (2K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 6x10b; D/A 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj06gs202a-e-ss

3.3 Special MCU Features

A 17-bit by 17-bit single-cycle multiplier 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) x (-1.0).

The 16/16 and 32/16 divide operations are supported, 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: CPU CORE BLOCK DIAGRAM

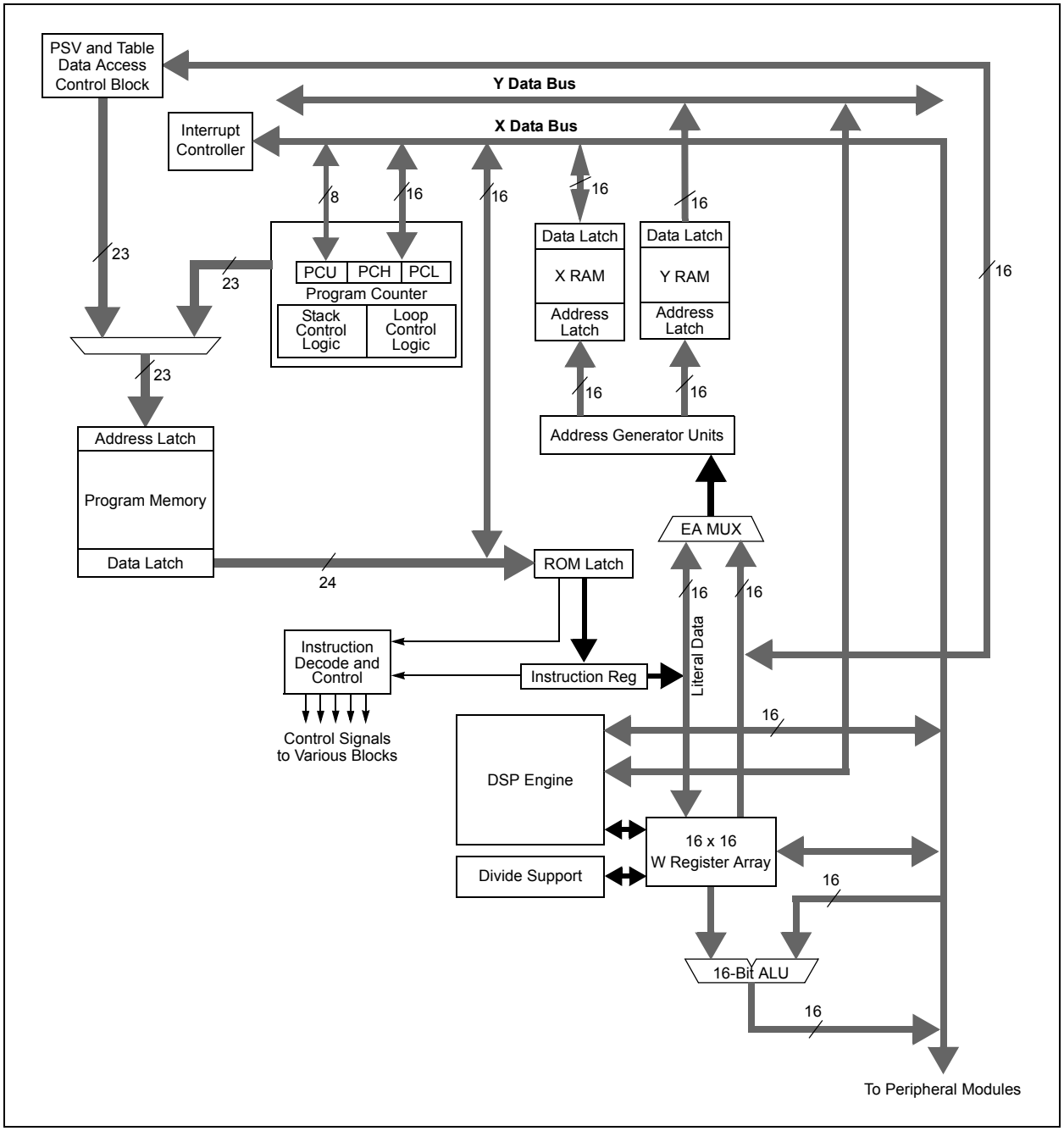


TABLE 4-7: INTERRUPT CONTROLLER REGISTER MAP FOR dsPIC33FJ06GS202A DEVICES ONLY

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
INTCON1	0080	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	—	T2IF	—	—	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	AC1IF	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	JTAGIF	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	AC2IF	—	—	—	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	ADCP6IF	—	—	—	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	—	T2IE	—	—	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	AC1IE	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	JTAGIE	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	AC2IE	—	—	—	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	ADCP6IE	—	—	—	ADCP2IE	0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			4444
IPC1	00A6	—	T2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	—	—	—	4440
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	AC1IP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC20	00CC	—	—	—	—	—	—	—	—	—	—	—	—	—	JTAGIP<2:0>			0004
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC25	00D6	—	AC2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IP<2:0>			0004
IPC29	00DE	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP6IP<2:0>			0004
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							0000

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

TABLE 4-15: HIGH-SPEED PWM GENERATOR 4 REGISTER MAP FOR dsPIC33FJ06GS001, dsPIC33FJ06GS101A AND dsPIC33FJ09GS302

File Name	Addr Offset	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
PWMCON4	0480	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
IOCON4	0482	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
FCLCON4	0484	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
PDC4	0486	PDC4<15:0>																0000
PHASE4	0488	PHASE4<15:0>																0000
DTR4	048A	—	—	DTR4<13:0>														0000
ALTDTR4	048C	—	—	ALTDTR4<13:0>														0000
SDC4	048E	SDC4<15:0>																0000
SPHASE4	0490	SPHASE4<15:0>																0000
TRIG4	0492	TRGCMP<15:3>													—	—	—	0000
TRGCON4	0494	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
STRIG4	0496	STRGCMP<15:3>													—	—	—	0000
PWMCAP4	0498	PWMCAP4<15:3>													—	—	—	0000
LEBCON4	049A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<6:0>							—	—	—	0000
AUXCON4	049E	HRPDIS	HRDDIS	—	—	—	—	—	—	—	—	CHOPSEL<3:0>				CHOPHEN	CHOPLEN	0000

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

REGISTER 7-9: IFS5: INTERRUPT FLAG STATUS REGISTER 5

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PWM2IF(1)	PWM1IF	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	JTAGIF
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 **PWM2IF:** PWM2 Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **PWM1IF:** PWM1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-1 **Unimplemented:** Read as '0'

bit 0 **JTAGIF:** JTAG Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 7-35: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	ILR<3:0>			
bit 15							bit 8

U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM<6:0>						
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 11-8 **ILR<3:0>:** New CPU Interrupt Priority Level bits
1111 = CPU Interrupt Priority Level is 15
•
•
•
0001 = CPU Interrupt Priority Level is 1
0000 = CPU Interrupt Priority Level is 0

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

bit 6-0 **VECNUM<6:0>:** Vector Number of Pending Interrupt bits
0111111 = Interrupt vector pending is Number 135
•
•
•
0000001 = Interrupt vector pending is Number 9
0000000 = Interrupt vector pending is Number 8

8.4 Oscillator Control Registers

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER(1,3)

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC<2:0>			—	NOSC<2:0>(2)		
bit 15							bit 8

R/W-0	R/W-0	R-0	U-0	R/C-0	U-0	U-0	R/W-0
CLKLOCK	IOLOCK	LOCK	—	CF	—	—	OSWEN
bit 7							bit 0

Legend:	y = Value set from Configuration bits on POR		
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 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits (read-only)
111 = Fast RC Oscillator (FRC) with divide-by-n
110 = Fast RC Oscillator (FRC) with divide-by-16
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL
000 = Fast RC Oscillator (FRC)

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

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(2)
111 = Fast RC Oscillator (FRC) with divide-by-n
110 = Fast RC Oscillator (FRC) with divide-by-16
101 = Low-Power RC Oscillator (LPRC)
100 = Reserved
011 = Primary Oscillator (XT, HS, EC) with PLL
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator (FRC) with PLL
000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Lock Enable bit
If clock switching is enabled and FSCM is disabled, FCKSM<1:0> (FOSC<7:6>) bits = 0b01):
1 = Clock switching is disabled, system clock source is locked
0 = Clock switching is enabled, system clock source can be modified by clock switching

bit 6 **IOLOCK:** Peripheral Pin Select Lock bit
1 = Peripherial Pin Select is locked, write to Peripheral Pin Select registers is not allowed
0 = Peripherial Pin Select is not locked, write to Peripheral Pin Select registers is allowed

bit 5 **LOCK:** PLL Lock Status bit (read-only)
1 = Indicates that PLL is in lock or PLL start-up timer is satisfied
0 = Indicates that PLL is out of lock, start-up timer is in progress or PLL is disabled

Note 1: Writes to this register require an unlock sequence. Refer to **Section 42. "Oscillator (Part IV)"** (DS70307) in the *"dsPIC33F/PIC24H Family Reference Manual"* for details.

2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.

3: This register is reset only on a Power-on Reset (POR).

8.6 Fail-Safe Clock Monitor (FSCM)

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate, even in the event of an oscillator failure. The FSCM function is enabled by programming. If the FSCM function is enabled, the LPRC internal oscillator runs at all times (except during Sleep mode) and is not subject to control by the Watchdog Timer.

In the event of an oscillator failure, the FSCM generates a clock failure trap event and switches the system clock over to the FRC oscillator. Then, the application program can either attempt to restart the oscillator or execute a controlled shutdown. The trap can be treated as a warm Reset by simply loading the Reset address into the oscillator fail trap vector.

If the PLL multiplier is used to scale the system clock, the internal FRC is also multiplied by the same factor on clock failure. Essentially, the device switches to FRC with PLL on a clock failure.

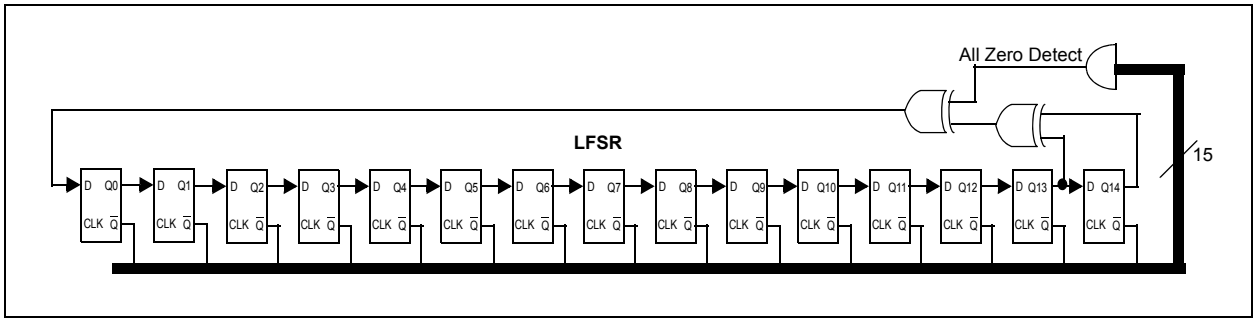
8.7 Pseudo-Random Generator

The pseudo-random generator is implemented with a 15-bit Linear Feedback Shift Register (LFSR), which is a shift register with a few exclusive OR gates. The shift register is clocked by the PWM clock and is a read-only register. The purpose of this feature is to provide the ability to randomly change the period or the active portion of the PWM.

A firmware routine can be used to read “n” random bits from the LFSR register and combine them, by either summing or performing another logical operation with the PWM period of the Duty Cycle registers. The result will be a PWM signal whose nominal period (or duty cycle) is the desired one, but whose effective value changes randomly. This capability will help in reducing the EMI/EMC emissions by spreading the power over a wider frequency range.

Figure 8-3 provides a block diagram of the LFSR.

FIGURE 8-3: LFSR BLOCK DIAGRAM



REGISTER 10-3: RPINR2: PERIPHERAL PIN SELECT INPUT REGISTER 2

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T1CKR<5:0>					
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **T1CKR<5:0>:** Assign Timer1 External Clock (T1CK) 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 7-0 **Unimplemented:** Read as '0'

REGISTER 10-8: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R<5:0>(1)					
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
—	—	SDI1R<5:0>(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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **SCK1R<5:0>:** Assign SPI1 Clock Input (SCK1) to the Corresponding RPn Pin bits(1)

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 **SDI1R<5:0>:** Assign SPI1 Data Input (SDI1) to the Corresponding RPn Pin bits(1)

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

Note 1: These bits are not implemented in the dsPIC33FJ06GS001 device.

REGISTER 10-13: RPINR32: PERIPHERAL PIN SELECT INPUT REGISTER 32

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT7R<5:0>					
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
—	—	FLT6R<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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **FLT7R<5:0>:** Assign PWM Fault Input 7 (FLT7) 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 **FLT6R<5:0>:** Assign PWM Fault Input 6 (FLT6) 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

REGISTER 10-20: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R<5:0>(1)					
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP8R<5:0>(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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP9R<5:0>:** Peripheral Output Function is Assigned to RP9 Output Pin bits(1)
(see Table 10-2 for peripheral function numbers)

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

bit 5-0 **RP8R<5:0>:** Peripheral Output Function is Assigned to RP8 Output Pin bits(1)
(see Table 10-2 for peripheral function numbers)

Note 1: These bits are not implemented in dsPIC33FJ06GS001/101A devices.

REGISTER 10-21: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R<5:0>(1)					
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP10R<5:0>(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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP11R<5:0>:** Peripheral Output Function is Assigned to RP11 Output Pin bits(1)
(see Table 10-2 for peripheral function numbers)

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

bit 5-0 **RP10R<5:0>:** Peripheral Output Function is Assigned to RP10 Output Pin bits(1)
(see Table 10-2 for peripheral function numbers)

Note 1: These bits are not implemented in dsPIC33FJ06GS001/101A devices.

14.0 OUTPUT COMPARE

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 13. "Output Compare"** (DS70209) 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 output compare module can select either Timer1 or Timer2 for its time base. The module compares the value of the timer with the value of one or two Compare registers, depending on the operating mode selected. The state of the output pin changes when the timer value matches the Output Compare register value. The output compare module generates either a single output pulse, or a sequence of output pulses, by changing the state of the output pin on the compare match events. The output compare module can also generate interrupts on compare match events.

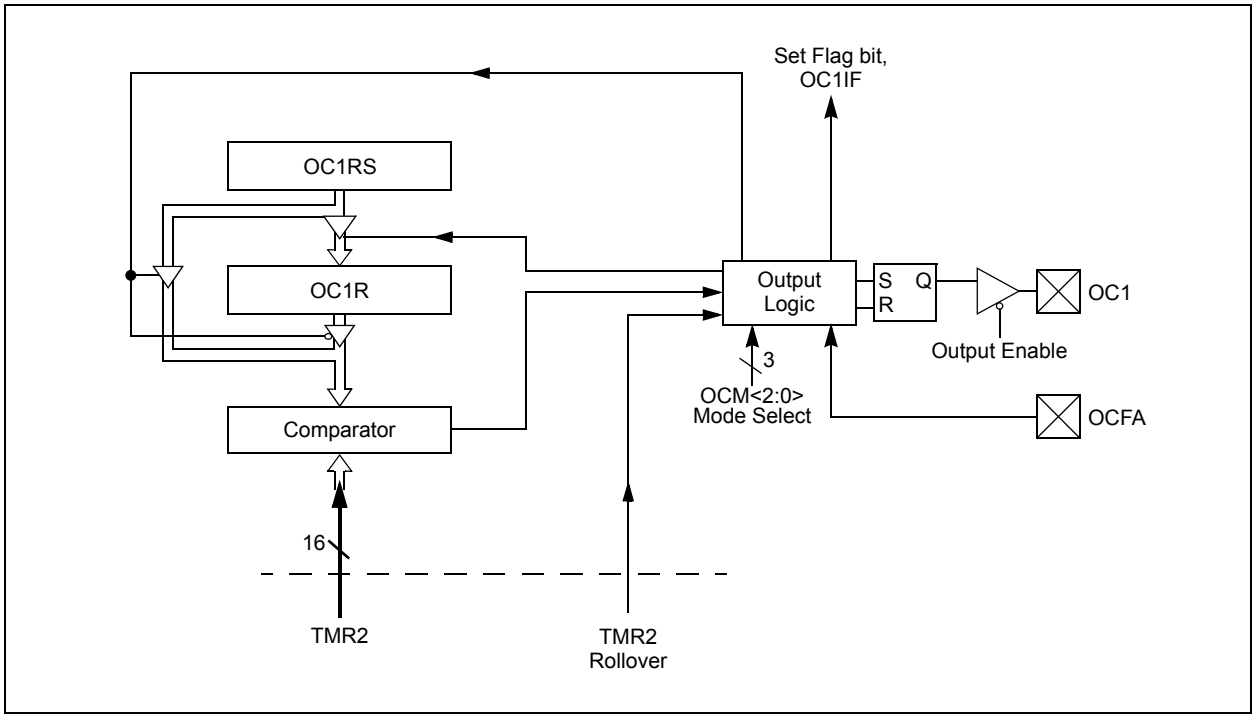
The output compare module has multiple operating modes:

- Active-Low One-Shot mode
- Active-High One-Shot mode
- Toggle mode
- Delayed One-Shot mode
- Continuous Pulse mode
- PWM mode without Fault Protection
- PWM mode with Fault Protection

Note: The output compare module is not implemented in the dsPIC33FJ06GS001 device.

If a Fault condition is detected on the OCFA pin, the output pin(s) of the output compare module are placed in tri-state. The user may elect to use a pull-down or pull-up resistor on the PWM pin to provide for a desired state if a Fault condition occurs.

FIGURE 14-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM



15.2 Feature Description

The PWM module is designed for applications that require:

- High-resolution at high PWM frequencies
- The ability to drive Standard, Edge-Aligned, Center-Aligned Complementary mode, and Push-Pull mode outputs
- The ability to create multiphase PWM outputs

For Center-Aligned mode, the duty cycle, period, phase and dead-time resolutions will be 8.32 ns.

Two common, medium power converter topologies are push-pull and half-bridge. These designs require the PWM output signal to be switched between alternate pins, as provided by the Push-Pull PWM mode.

A phase-shifted PWM describes the situation where each PWM generator provides outputs, but the phase relationship between the generator outputs is specifiable and changeable.

A multiphase PWM is often used to improve DC/DC Converter load transient response, and reduce the size of output filter capacitors and inductors. Multiple DC/DC Converters are often operated in parallel, but phase-shifted in time. A single PWM output, operating at 250 kHz, has a period of 4 μs, but an array of four PWM channels staggered by 1 μs each, yields an effective switching frequency of 1 MHz. Multiphase PWM applications typically use a fixed-phase relationship.

A variable phase PWM is useful in Zero Voltage Transition (ZVT) power converters. Here, the PWM duty cycle is always 50%, and the power flow is controlled by varying the relative phase-shift between the two PWM generators.

REGISTER 15-15: FCLCONx: PWMx FAULT CURRENT-LIMIT CONTROL REGISTER (CONTINUED)

bit 7-3 **FLTSRC<4:0>:** Fault Control Signal Source Select for PWM Generator # bits(2,3)
11111 = Reserved
•
•
•
01000 = Reserved

00111 = Fault 8
00110 = Fault 7
00101 = Fault 6
00100 = Fault 5
00011 = Fault 4
00010 = Fault 3
00001 = Fault 2
00000 = Fault 1

bit 2 **FLTPOL:** Fault Polarity for PWMx Generator # bit(1)
1 = The selected Fault source is active-low
0 = The selected Fault source is active-high

bit 1-0 **FLTMOD<1:0>:** Fault Mode for PWMx Generator # bits
11 = Fault input is disabled
10 = Reserved
01 = The selected Fault source forces PWMxH, PWMxL pins to FLTDAT values (cycle)
00 = The selected Fault source forces PWMxH, PWMxL pins to FLTDAT values (latched condition)

Note 1: These bits should be changed only when PTEN = 0. Changing the clock selection during operation will yield unpredictable results.

2: When Independent Fault mode is enabled (IFLTMOD<1:0> = 1), and Fault 1 is used for Current-Limit mode (CLSRC<4:0> = b0000), the Fault Control Source Select bits (FLTSRC<4:0>) should be set to an unused Fault source to prevent Fault 1 from disabling both the PWMxL and PWMxH outputs.

3: When Independent Fault mode is enabled (IFLTMOD<1:0> = 1) and Fault 1 is used for Fault mode (FLTSRC<4:0> = b0000), the Current-Limit Control Source Select bits (CLSRC<4:0>) should be set to an unused current-limit source to prevent the current-limit source from disabling both the PWMxH and PWMxL outputs.

REGISTER 17-2: I2C1STAT: I2C1 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:	HS = Hardware Settable bit'	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	C = Clearable bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ACKSTAT:** Acknowledge Status bit
(when operating as I^2C master, applicable to master transmit operation)
1 = NACK is received from slave
0 = ACK is received from slave
Hardware is set or clear at end of slave Acknowledge.

bit 14 **TRSTAT:** Transmit Status bit (when operating as I^2C 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:** Write Collision Detect bit
1 = An attempt to write to the I2C1TRN register failed because the I^2C module is busy
0 = No collision
Hardware is set at occurrence of write to I2C1TRN while busy (cleared by software).

bit 6 **I2COV:** Receive Overflow Flag bit
1 = A byte was received while the I2C1RCV register is still holding the previous byte
0 = No overflow
Hardware is set at attempt to transfer I2C1RSR to I2C1RCV (cleared by software).

bit 5 **D_A:** Data/Address bit (when operating as I^2C slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was the device address
Hardware is clear at device address match. Hardware is set by reception of slave byte.

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.

REGISTER 22-1: CONSTANT CURRENT SOURCE CALIBRATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

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

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	CCSCAL<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 23-6 **Unimplemented:** Read as ‘0’

bit 5-0 **CCSCAL<5:0>:** Constant Current Source Calibration bits

The value of these bits must be copied into the ISRCCAL<5:0> bits (ISRCCON<5:0>). Refer to the Current Source Control register (Register 21-1) in **Section 21.0 “Constant Current Source”**.

22.4 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

22.4.1 PRESCALER/POSTSCALER

The nominal WDT clock source from LPRC is 32 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the WDTPRE Configuration bit (FWDT<4>). With a 32 kHz input, the prescaler yields a nominal WDT time-out period (TWDT) of 1 ms in 5-bit mode or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPOST<3:0> Configuration bits (FWDT<3:0>), which allow the selection of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler, time-out periods ranging from 1 ms to 131 seconds can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSC<2:0> bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a PWRSAV instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a CLRWDT instruction during normal execution

Note: The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

22.4.2 SLEEP AND IDLE MODES

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP bit (RCON<3>) or IDLE bit (RCON<2>) will need to be cleared in software after the device wakes up.

22.4.3 ENABLING WDT

The WDT is enabled or disabled by the FWDTEN Configuration bit in the FWDT Configuration register (FWDT<7>). When the FWDTEN Configuration bit is set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN Configuration bit has been programmed to '0'. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user application to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

The WDT flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

FIGURE 22-2: WDT BLOCK DIAGRAM

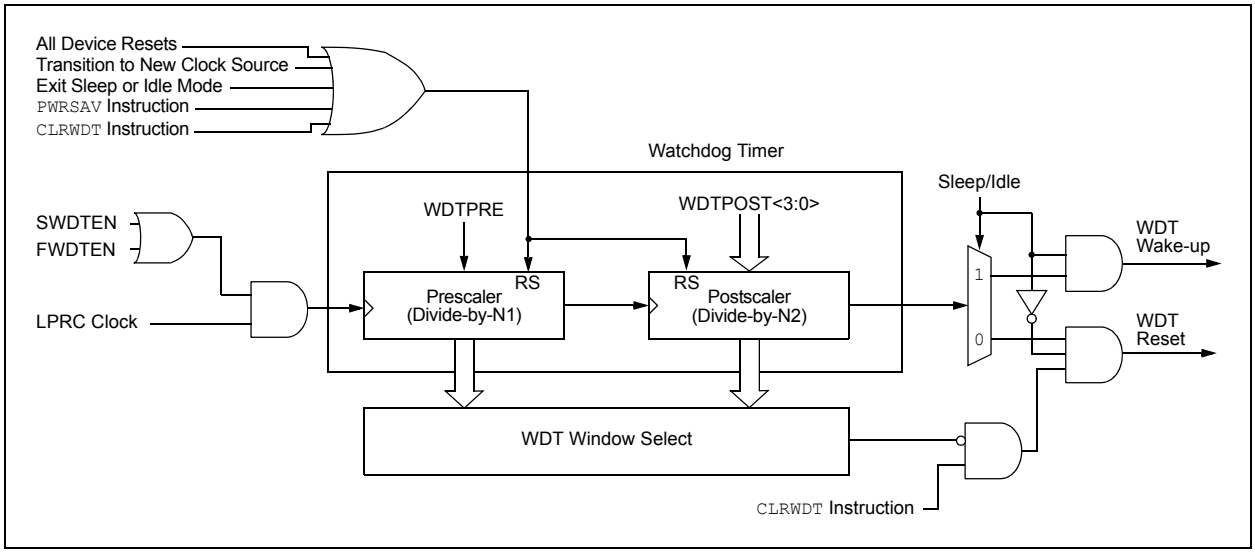


TABLE 23-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
10	BTSC	BTSC f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
12	BTST	BTST f,#bit4	Bit Test f	1	1	Z
		BTST.C Ws,#bit4	Bit Test Ws to C	1	1	C
		BTST.Z Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
		BTST.Z Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
13	BTSTS	BTSTS f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL lit23	Call Subroutine	2	2	None
		CALL Wn	Call Indirect Subroutine	1	2	None
15	CLR	CLR f	f = 0x0000	1	1	None
		CLR WREG	WREG = 0x0000	1	1	None
		CLR Ws	Ws = 0x0000	1	1	None
		CLR Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT	Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM f	$f = \overline{f}$	1	1	N,Z
		COM f,WREG	$WREG = \overline{f}$	1	1	N,Z
		COM Ws,Wd	$Wd = \overline{Ws}$	1	1	N,Z
18	CP	CP f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP Wb,#lit5	Compare Wb with lit5	1	1	C,DC,N,OV,Z
		CP Wb,Ws	Compare Wb with Ws (Wb – Ws)	1	1	C,DC,N,OV,Z
19	CP0	CP0 f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0 Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
20	CPB	CPB f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,Ws	Compare Wb with Ws, with Borrow ($Wb – Ws – \overline{C}$)	1	1	C,DC,N,OV,Z
21	CPSEQ	CPSEQ Wb, Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
22	CPSGT	CPSGT Wb, Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
23	CPSLT	CPSLT Wb, Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
24	CPSNE	CPSNE Wb, Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
25	DAW	DAW Wn	Wn = Decimal Adjust Wn	1	1	C
26	DEC	DEC f	f = f – 1	1	1	C,DC,N,OV,Z
		DEC f,WREG	WREG = f – 1	1	1	C,DC,N,OV,Z
		DEC Ws,Wd	Wd = Ws – 1	1	1	C,DC,N,OV,Z
27	DEC2	DEC2 f	f = f – 2	1	1	C,DC,N,OV,Z
		DEC2 f,WREG	WREG = f – 2	1	1	C,DC,N,OV,Z
		DEC2 Ws,Wd	Wd = Ws – 2	1	1	C,DC,N,OV,Z
28	DISI	DISI #lit14	Disable Interrupts for k Instruction Cycles	1	1	None

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

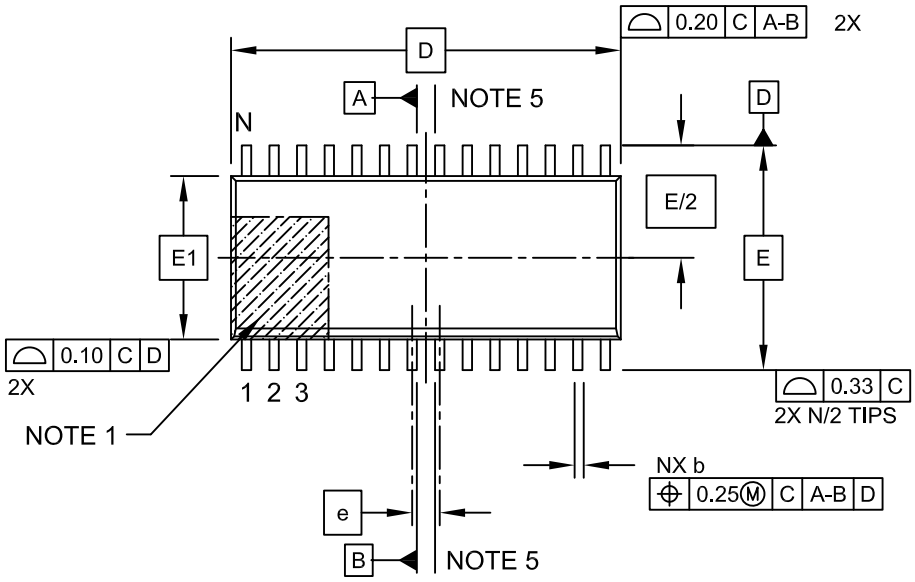
The MPLAB IDE allows you to:

- 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

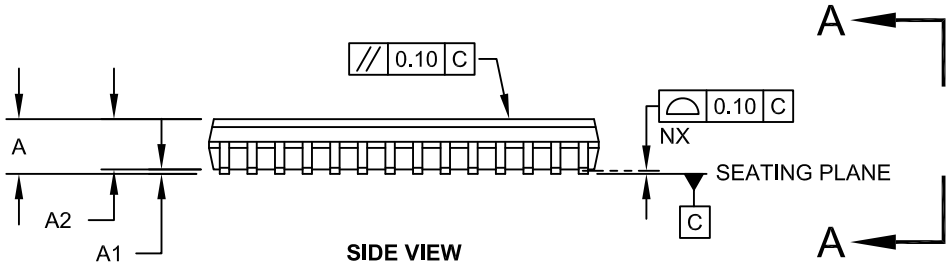
MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

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

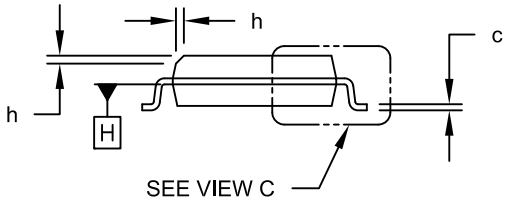
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2