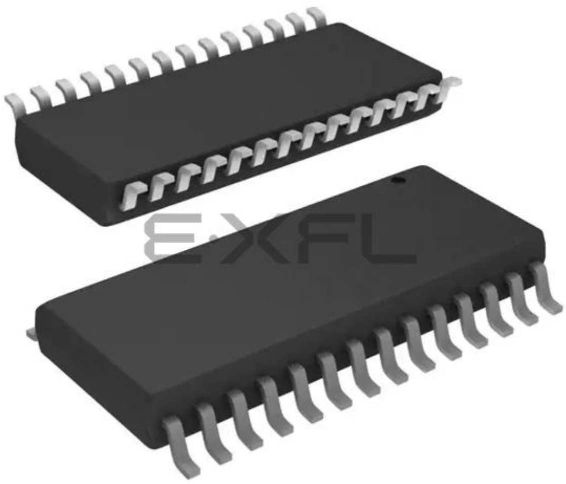


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	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	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	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj06gs102a-i-so

REGISTER 3-2: CORCON: CORE CONTROL REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT(1)	DL<2:0>		
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3(2)	PSV	RND	IF
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **US:** DSP Multiply Unsigned/Signed Control bit
1 = DSP engine multiplies are unsigned
0 = DSP engine multiplies are signed

bit 11 **EDT:** Early DO Loop Termination Control bit(1)
1 = Terminate executing DO loop at end of current loop iteration
0 = No effect

bit 10-8 **DL<2:0>:** DO Loop Nesting Level Status bits
111 = 7 DO loops active
•
•
•
001 = 1 DO loop active
000 = 0 DO loops active

bit 7 **SATA:** ACCA Saturation Enable bit
1 = Accumulator A saturation is enabled
0 = Accumulator A saturation is disabled

bit 6 **SATB:** ACCB Saturation Enable bit
1 = Accumulator B saturation is enabled
0 = Accumulator B saturation is disabled

bit 5 **SATDW:** Data Space Write from DSP Engine Saturation Enable bit
1 = Data space write saturation is enabled
0 = Data space write saturation is disabled

bit 4 **ACCSAT:** Accumulator Saturation Mode Select bit
1 = 9.31 saturation (super saturation)
0 = 1.31 saturation (normal saturation)

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit 3(2)
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

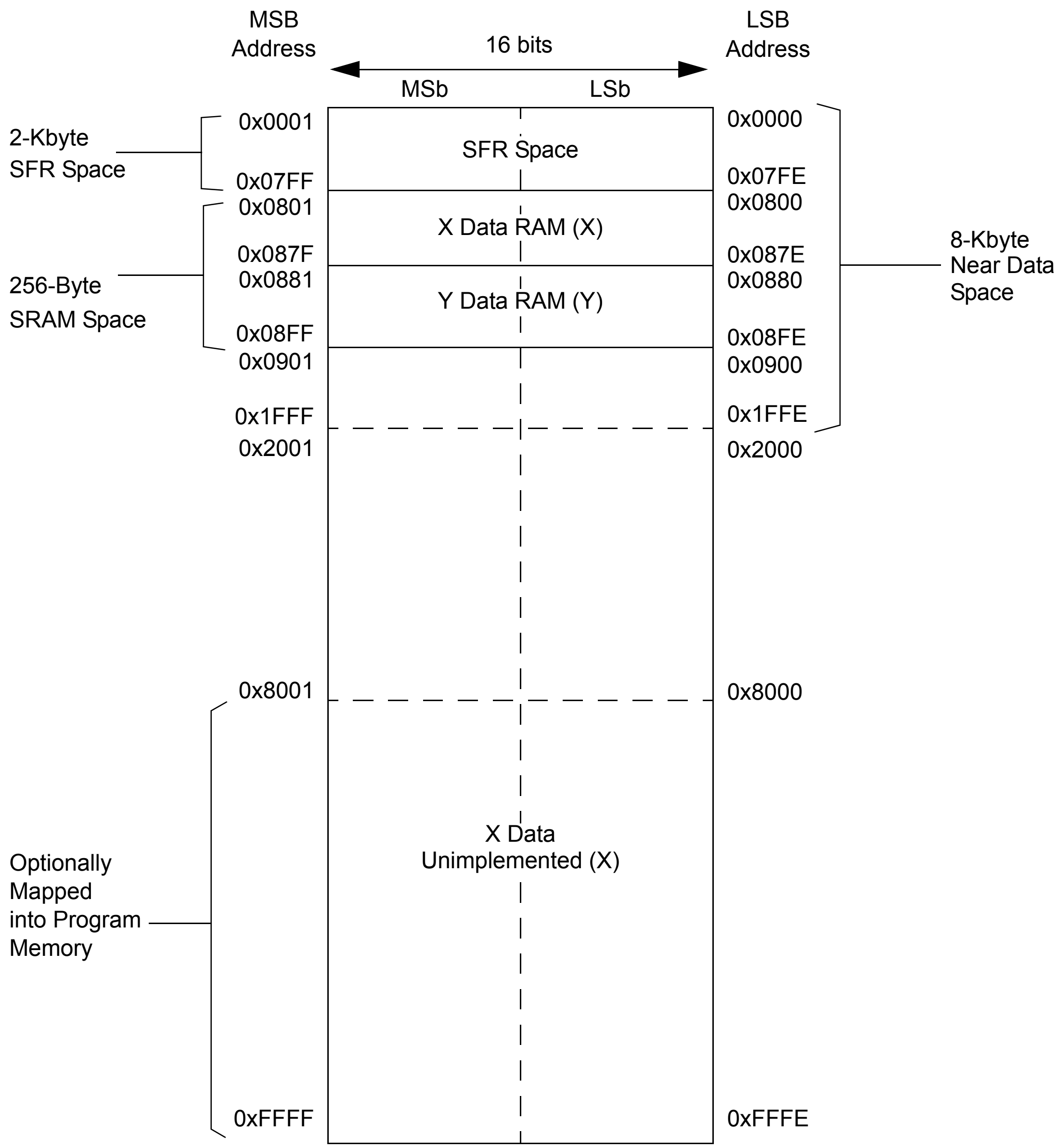
bit 2 **PSV:** Program Space Visibility in Data Space Enable bit
1 = Program space is visible in data space
0 = Program space is not visible in data space

bit 1 **RND:** Rounding Mode Select bit
1 = Biased (conventional) rounding is enabled
0 = Unbiased (convergent) rounding is enabled

bit 0 **IF:** Integer or Fractional Multiplier Mode Select bit
1 = Integer mode enabled for DSP multiply ops
0 = Fractional mode enabled for DSP multiply ops

Note 1: This bit will always read as ‘0’.
2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

FIGURE 4-3: DATA MEMORY MAP FOR dsPIC33FJ06GS001/101A/102A DEVICES WITH 256 BYTES OF RAM



4.2.5 X AND Y DATA SPACES

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

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

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

Both the X and Y data spaces support Modulo Addressing mode for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing mode is only supported for writes to X data space.

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

All Effective Addresses are 16 bits wide and point to bytes within the data space. Therefore, the data space address range is 64 Kbytes, or 32K words, though the implemented memory locations vary by device.

TABLE 4-19: CONSTANT CURRENT SOURCE REGISTER MAP

File Name	ADR	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
ISRCCON	0500	ISRCEN	—	—	—	—	OUTSEL<2:0>			—	—	ISRCCAL<5:0>						0000

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

TABLE 4-20: HIGH-SPEED 10-BIT ADC REGISTER MAP FOR dsPIC33FJ06GS001 AND dsPIC33FJ06GS101A

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
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS<2:0>			0003
ADPCFG	0302	—	—	—	—	—	—	—	—	PCFG7	PCFG6	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0000
ADSTAT	0306	—	—	—	—	—	—	—	—	—	P6RDY	—	—	P3RDY	—	P1RDY	P0RDY	0000
ADBASE	0308	ADBASE<15:1>															—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC3<4:0>					—	—	—	—	—	—	—	—	0000
ADCPC3	0310	—	—	—	—	—	—	—	—	IRQEN6	PEND6	SWTRG6	TRGSRC6<4:0>					0000
ADCBUF0	0320	ADC Data Buffer 0																xxxx
ADCBUF1	0322	ADC Data Buffer 1																xxxx
ADCBUF2	0324	ADC Data Buffer 2																xxxx
ADCBUF3	0326	ADC Data Buffer 3																xxxx
ADCBUF6	032C	ADC Data Buffer 6																xxxx
ADCBUF7	032E	ADC Data Buffer 7																xxxx
ADCBUF12	0338	ADC Data Buffer 12																xxxx
ADCBUF13	033A	ADC Data Buffer 13																xxxx

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

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
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 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Uses alternate vector table
0 = Uses standard (default) vector table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-3 **Unimplemented:** Read as ‘0’

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	ADIF	U1TXIF(1)	U1RXIF(1)	SPI1IF(1)	SPI1EIF(1)	—
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF	—	—	—	T1IF	OC1IF(1)	IC1IF(2)	INT0IF
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 **ADIF:** ADC Group Conversion Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **U1TXIF:** UART1 Transmitter Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **U1RXIF:** UART1 Receiver Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPI1IF:** SPI1 Event Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **SPI1EIF:** SPI1 Error Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

bit 7 **T2IF:** Timer2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **T1IF:** Timer1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit(2)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Note 1: This bit is not implemented in the dsPIC33FJ06GS001 device.
2: This bit is not implemented in dsPIC33FJ06GS001/101A/102A devices.

REGISTER 7-24: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

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

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP<2:0>		
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 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 7-25: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

U-0	U-1	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
—	INT2IP<2:0>			—	—	—	—
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 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **INT2IP<2:0>:** External Interrupt 2 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’

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

Complete the following steps to configure an interrupt source at initialization:

1. Set the NSTDIS bit (INTCON1<15>) if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources can be programmed to the same non-zero value.

Note:	At a device Reset, the IPCx registers are initialized such that all user interrupt sources are assigned to Priority Level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

7.4.2 INTERRUPT SERVICE ROUTINE

The method used to declare an ISR and initialize the IVT with the correct vector address depends on the programming language (C or assembler) and the language development toolsuite used to develop the application.

In general, the user application must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles; otherwise, the program will re-enter the ISR immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and the old CPU priority level.

7.4.3 TRAP SERVICE ROUTINE

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

7.4.4 INTERRUPT DISABLE

The following steps outline the procedure to disable all user interrupts:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to Priority Level 7 by inclusive ORing the value, 0xE0 with SRL.

To enable user interrupts, the `POP` instruction can be used to restore the previous SR value.

Note:	Only user interrupts with a priority level of 7 or lower can be disabled. Trap sources (Level 8-Level 15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of Priority Levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

8.5 Clock Switching Operation

Applications are free to switch among any of the four clock sources (primary, LP, FRC and LPRC) under software control at any time. To limit the possible side effects of this flexibility, devices have a safeguard lock built into the switch process.

Note:	Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMD<1:0> Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch among the different primary submodes without reprogramming the device.

8.5.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the FOSC Configuration register must be programmed to '0'. (Refer to **Section 22.1 "Configuration Bits"** for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and Fail-Safe Clock Monitor function are disabled. This is the default setting.

The NOSC<2:0> control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSC<2:0> bits (OSCCON<14:12>) reflect the clock source selected by the FNOSC Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled. It is held at '0' at all times.

8.5.2 OSCILLATOR SWITCHING SEQUENCE

To perform a clock switch, the following basic sequence is required:

1. If desired, read the COSC<2:0> bits to determine the current oscillator source.
2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
3. Write the appropriate value to the NOSC<2:0> control bits for the new oscillator source.
4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
5. Set the OSWEN bit (OSCCON<0>) to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically as follows:

1. The clock switching hardware compares the COSC<2:0> status bits with the new value of the NOSC<2:0> control bits. If they are the same, the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and the CF (OSCCON<3>) status bits are cleared.
3. The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware waits until the Oscillator Start-up Timer (OST) expires. If the new source is using the PLL, the hardware waits until a PLL lock is detected (LOCK = 1).
4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSC<2:0> bit values are transferred to the COSC<2:0> status bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM is enabled).

Note 1:	The processor continues to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.
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:	Refer to **Section 42. "Oscillator (Part IV)"** (DS70307) in the *"dsPIC33F/PIC24H Family Reference Manual"* for details.

REGISTER 10-10: RPINR29: PERIPHERAL PIN SELECT INPUT REGISTER 29

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT1R<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 **FLT1R<5:0>:** Assign PWM Fault Input 1 (FLT1) 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 15-10: SPHASEx: PWMx SECONDARY PHASE SHIFT REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPHASEx<15:8>(1,2)							
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
SPHASEx<7:0>(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 15-0 **SPHASEx<15:0>:** Secondary Phase Offset for PWMxL Output Pin bits(1,2)
(used in Independent PWM mode only)

Note 1: If the ITB (PWMCONx<9>) bit = 0, the following applies based on the mode of operation:
- Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01 or 10), SPHASEx<15:0> = Not used.
- True Independent Output mode PMOD<1:0> (IOCONx<11:10>) = 11), PHASEx<15:0> = Phase shift value for PWMxL only.

2: If the ITB (PWMCONx<9>) bit = 1, the following applies based on the mode of operation:
- Complementary, Redundant and Push-Pull Output mode (PMOD<1:0> (IOCONx<11:10>) = 00, 01 or 10), SPHASEx<15:0> = Not used.
- True Independent Output mode PMOD<1:0> (IOCONx<11:10>) = 11), PHASEx<15:0> = Independent time base period value for PWMxL only.

REGISTER 15-18: LEBCONx: PWMx LEADING-EDGE BLANKING CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<6:5>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
LEB<4:0>					—	—	—
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 **PHR:** PWMxH Rising Edge Trigger Enable bit
1 = Rising edge of PWMxH will trigger LEB counter
0 = LEB ignores rising edge of PWMxH

bit 14 **PHF:** PWMxH Falling Edge Trigger Enable bit
1 = Falling edge of PWMxH will trigger LEB counter
0 = LEB ignores falling edge of PWMxH

bit 13 **PLR:** PWMxL Rising Edge Trigger Enable bit
1 = Rising edge of PWMxL will trigger LEB counter
0 = LEB ignores rising edge of PWMxL

bit 12 **PLF:** PWMxL Falling Edge Trigger Enable bit
1 = Falling edge of PWMxL will trigger LEB counter
0 = LEB ignores falling edge of PWMxL

bit 11 **FLTLEBEN:** Fault Input LEB Enable bit
1 = Leading-edge blanking is applied to selected Fault input
0 = Leading-edge blanking is not applied to selected Fault input

bit 10 **CLLEBEN:** Current-Limit LEB Enable bit
1 = Leading-edge blanking is applied to selected current-limit input
0 = Leading-edge blanking is not applied to selected current-limit input

bit 9-3 **LEB<6:0>:** Leading-Edge Blanking for Current-Limit and Fault Inputs bits
The value is 8.32 nsec increments.

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

17.0 INTER-INTEGRATED CIRCUIT™ (I^2C™)

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 19. "Inter-Integrated Circuit (I^2C™)"** (DS70195) 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 Inter-Integrated Circuit™ (I^2C™) module provides complete hardware support for both Slave and Multi-Master modes of the I^2C serial communication standard with a 16-bit interface.

The I^2C module has a 2-pin interface:

- The SCL1 pin is the clock
- The SDA1 pin is data

The I^2C module offers the following key features:

- I^2C interface supporting both Master and Slave modes of operation
- I^2C Slave mode supports 7-bit and 10-bit addressing
- I^2C Master mode supports 7-bit and 10-bit addressing
- I^2C port allows bidirectional transfers between master and slaves
- Serial clock synchronization for I^2C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control)
- I^2C supports multi-master operation, detects bus collision and arbitrates accordingly

17.1 Operating Modes

The hardware fully implements all the master and slave functions of the I^2C Standard and Fast mode specifications, as well as 7-bit and 10-bit addressing.

The I^2C module can operate either as a slave or a master on an I^2C bus.

The following types of I^2C operation are supported:

- I^2C slave operation with 7-bit addressing
- I^2C slave operation with 10-bit addressing
- I^2C master operation with 7-bit or 10-bit addressing

For details about the communication sequence in each of these modes, please see the Microchip web site (www.microchip.com) for the latest *"dsPIC33F/PIC24H Family Reference Manual"* sections.

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.

21.3 Current Source Control Register

REGISTER 21-1: ISRCCON: CONSTANT CURRENT SOURCE CONTROL REGISTER(1)

R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ISRCEN	—	—	—	—	OUTSEL<2:0>		
bit 15							bit 8

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	ISRCCAL<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 **ISRCEN:** Current Source Enable bit
1 = Current source is enabled
0 = Current source is disabled

bit 14-11 **Unimplemented:** Read as ‘0’

bit 10-8 **OUTSEL<2:0>:** Output Current Select bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = Select input pin, ISRC4 (AN4)
011 = Select input pin, ISRC3 (AN5)
010 = Select input pin, ISRC2 (AN6)
001 = Select input pin, ISRC1 (AN7)
000 = No output is selected

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **ISRCCAL<5:0>:** Current Source Calibration bits
The calibration value must be copied from Flash address, 0x800840, into these bits. Refer to the Constant Current Source Calibration Register (Register 22-1) in **Section 22.0 “Special Features”** for more information.

Note 1: This register is available in the dsPIC33FJ09GS302 device only.

TABLE 23-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

Field	Description
Wm*Wm	Multiplicand and Multiplier Working register pair for Square instructions ∈ {W4 * W4,W5 * W5,W6 * W6,W7 * W7}
Wm*Wn	Multiplicand and Multiplier Working register pair for DSP instructions ∈ {W4 * W5,W4 * W6,W4 * W7,W5 * W6,W5 * W7,W6 * W7}
Wn	One of 16 Working registers ∈ {W0..W15}
Wnd	One of 16 Destination Working registers ∈ {W0...W15}
Wns	One of 16 Source Working registers ∈ {W0...W15}
WREG	W0 (Working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }
Wx	X Data Space Prefetch Address register for DSP instructions ∈ {[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] - = 6, [W8] - = 4, [W8] - = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] - = 6, [W9] - = 4, [W9] - = 2, [W9 + W12], none}
Wxd	X Data Space Prefetch Destination register for DSP instructions ∈ {W4...W7}
Wy	Y Data Space Prefetch Address register for DSP instructions ∈ {[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] - = 6, [W11] - = 4, [W11] - = 2, [W11 + W12], none}
Wyd	Y Data Space Prefetch Destination register for DSP instructions ∈ {W4...W7}

24.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

24.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

24.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

NOTES:

Revision C (August 2012)

This revision includes minor typographical updates and content corrections. Major changes include new figures in **Section 26.0 "DC and AC Device Characteristics Graphs"**, updated values in Table 25-39 in **Section 25.0 "Electrical Characteristics"** and updated package drawings in **Section 27.0 "Packaging Information"**.

NOTES: