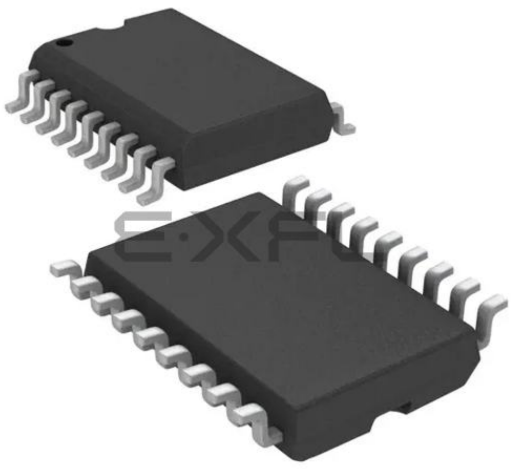
E·XFL

Microchip Technology - DSPIC33FJ12GP201-I/SO 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	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	12KB (12K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 6x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj12gp201-i-so

Email: info@E-XFL.COM Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

3.0 CPU

Note 1:	This data sheet summarizes the features of the dsPIC33FJ12GP201/202 family 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) of the *"dsPIC33F/PIC24H Family Reference Manual"*, which is available from the Microchip website (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 dsPIC33FJ12GP201/202 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 by 24 bits of user program memory space. The actual amount of program memory implemented varies by 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 dsPIC33FJ12GP201/202 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 16th working register (W15) operates as a software Stack Pointer (SP) for interrupts and calls.

The dsPIC33FJ12GP201/202 instruction set has two classes of instructions: 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, dsPIC33FJ12GP201/202 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. The programmer's model for the dsPIC33FJ12GP201/202 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.

3.3 Special MCU Features

The dsPIC33FJ12GP201/202 features a 17-bit by 17-bit single-cycle multiplier that is shared by both the MCU ALU and DSP engine. The multiplier can perform signed, unsigned and mixed-sign multiplication. Using a 17-bit by 17-bit multiplier for 16-bit by 16-bit multiplication not only allows you to perform mixed-sign multiplication, it also achieves accurate results for special operations, such as (-1.0) x (-1.0).

TABLE 4-11: PERIPHERAL PIN SELECT INPUT 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
RPINR0	0680	—	—	—	INT1R<4:0>					—	—	—	—	—	—	—	—	1F00
RPINR1	0682	—	—	—	—	—	—	—	—	—	—	—	INT2R<4:0>					001F
RPINR3	0686	—	—	—	T3CKR<4:0>					—	—	—	T2CKR<4:0>					1F1F
RPINR7	068E	—	—	—	IC2R<4:0>					—	—	—	IC1R<4:0>					1F1F
RPINR10	0694	—	—	—	IC8R<4:0>					—	—	—	IC7R<4:0>					1F1F
RPINR11	0696	—	—	—	—	—	—	—	—	—	—	—	OCFAR<4:0>					001F
RPINR18	06A4	—	—	—	U1CTSR<4:0>					—	—	—	U1RXR<4:0>					1F1F
RPINR20	06A8	—	—	—	SCK1R<4:0>					—	—	—	SDI1R<4:0>					1F1F
RPINR21	06AA	—	—	—	—	—	—	—	—	—	—	—	SS1R<4:0>					001F

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

TABLE 4-12: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP FOR dsPIC33FJ12GP202

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
RPOR0	06C0	—	—	—	RP1R<4:0>					—	—	—	RP0R<4:0>					0000
RPOR1	06C2	—	—	—	RP3R<4:0>					—	—	—	RP2R<4:0>					0000
RPOR2	06C4	—	—	—	RP5R<4:0>					—	—	—	RP4R<4:0>					0000
RPOR3	06C6	—	—	—	RP7R<4:0>					—	—	—	RP6R<4:0>					0000
RPOR4	06C8	—	—	—	RP9R<4:0>					—	—	—	RP8R<4:0>					0000
RPOR5	06CA	—	—	—	RP11R<4:0>					—	—	—	RP10R<4:0>					0000
RPOR6	06CC	—	—	—	RP13R<4:0>					—	—	—	RP12R<4:0>					0000
RPOR7	06CE	—	—	—	RP15R<4:0>					—	—	—	RP14R<4:0>					0000

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

TABLE 4-13: PERIPHERAL PIN SELECT OUTPUT REGISTER MAP FOR dsPIC33FJ12GP201

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
RPOR0	06C0	—	—	—	RP1R<4:0>					—	—	—	RP0R<4:0>					0000
RPOR2	06C4	—	—	—	—	—	—	—	—	—	—	—	RP4R<4:0>					0000
RPOR3	06C6	—	—	—	RP7R<4:0>					—	—	—	—	—	—	—	—	0000
RPOR4	06C8	—	—	—	RP9R<4:0>					—	—	—	RP8R<4:0>					0000
RPOR7	06CE	—	—	—	RP15R<4:0>					—	—	—	RP14R<4:0>					0000

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

TABLE 4-20: 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.

TABLE 4-21: PMD 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
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	—	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	AD1MD	0000
PMD2	0772	IC8MD	IC7MD	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000

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

© 2007-2011 Microchip Technology Inc.

4.6.2 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>).

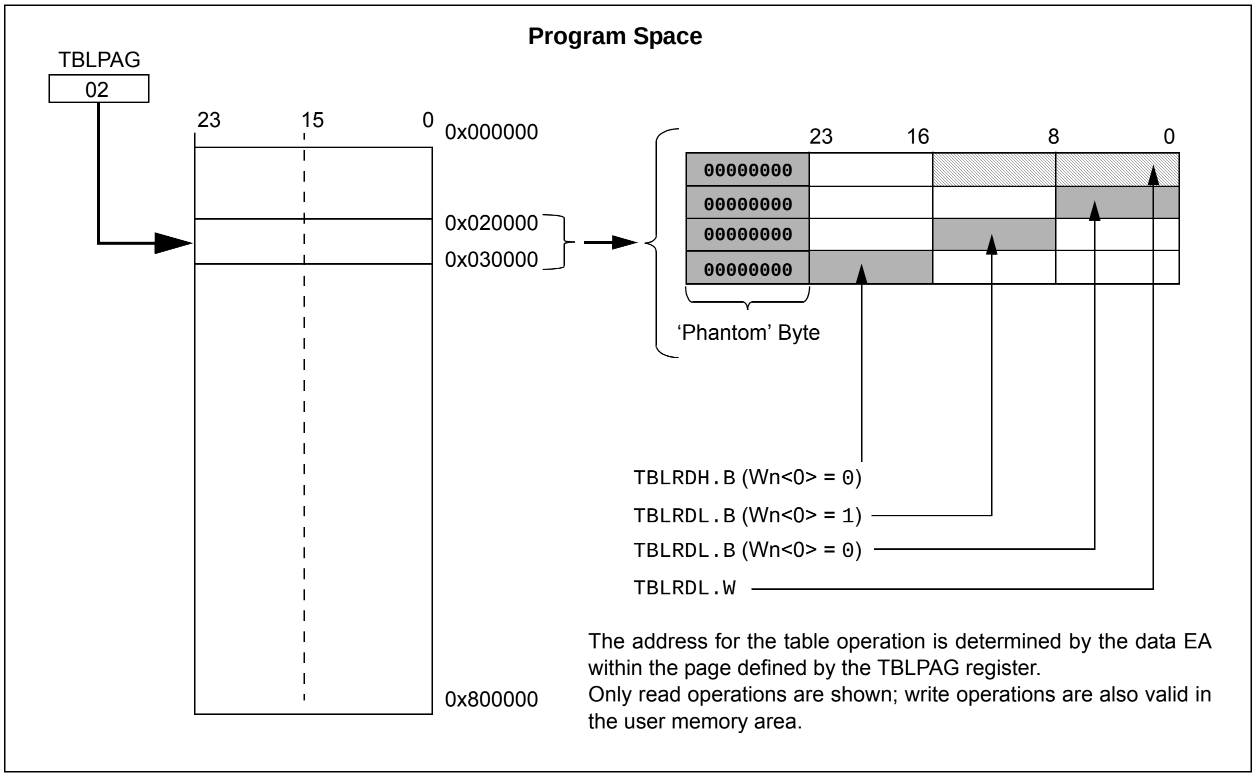
  In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when Byte Select is ‘`1`’; the lower byte is selected when it is ‘`0`’.
- `TBLRDH` (Table Read High): In Word mode, this instruction maps the entire upper word 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, as in the `TBLRDL` instruction. Note that the data will always be ‘`0`’ when the upper ‘phantom’ byte is selected (Byte Select = `1`).

In a similar fashion, 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 4-8: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS


Program Space
TBLPAG
02
23
15
0
0x000000
0x020000
0x030000
0x800000
23
16
8
0
00000000
00000000
00000000
00000000
‘Phantom’ Byte
TBLRDH.B (Wn<0> = 0)
TBLRDL.B (Wn<0> = 1)
TBLRDL.B (Wn<0> = 0)
TBLRDL.W
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.

5.4.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

Programmers can program one row of program Flash memory at a time. To do this, it is necessary to erase the 8-row erase page that contains the desired row. The general process is:

1. Read eight rows of program memory (512 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase the block (see Example 5-1):
   a) Set the NVMOP bits (NVMCON<3:0>) to '`0010`' to configure for block erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the page to be erased into the TBLPAG and W registers.
   c) Write 0x55 to NVMKEY.
   d) Write 0xAA to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 64 instructions from data RAM into the program memory buffers (see Example 5-2).
5. Write the program block to Flash memory:
   a) Set the NVMOP bits to '`0001`' to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 0x55 to NVMKEY.
   c) Write 0xAA to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.
6. Repeat steps 4 and 5, using the next available 64 instructions from the block in data RAM by incrementing the value in TBLPAG, until all 512 instructions are written back to Flash memory.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user application must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be `NOP`s, as shown in Example 5-3.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY PAGE

```
; Set up NVMCON for block erase operation
       MOV     #0x4042, W0                 ;
       MOV     W0, NVMCON                  ; Initialize NVMCON
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR), W0     ;
       MOV     W0, TBLPAG                  ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR), W0   ; Initialize in-page EA[15:0] pointer
       TBLWTL  W0, [W0]                    ; Set base address of erase block
       DISI    #5                          ; Block all interrupts with priority <7
                                           ; for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                  ; Write the 55 key
       MOV     #0xAA, W1                   ;
       MOV     W1, NVMKEY                  ; Write the AA key
       BSET    NVMCON, #WR                 ; Start the erase sequence
       NOP                                 ; Insert two NOPs after the erase
       NOP                                 ; command is asserted
```

TABLE 7-1: INTERRUPT VECTORS (CONTINUED)

Vector Number	Interrupt Request (IRQ) Number	IVT Address	AIVT Address	Interrupt Source
54	46	0x000070	0x000170	Reserved
55	47	0x000072	0x000172	Reserved
56	48	0x000074	0x000174	Reserved
57	49	0x000076	0x000176	Reserved
58	50	0x000078	0x000178	Reserved
59	51	0x00007A	0x00017A	Reserved
60	52	0x00007C	0x00017C	Reserved
61	53	0x00007E	0x00017E	Reserved
62	54	0x000080	0x000180	Reserved
63	55	0x000082	0x000182	Reserved
64	56	0x000084	0x000184	Reserved
65	57	0x000086	0x000186	Reserved
66	58	0x000088	0x000188	Reserved
67	59	0x00008A	0x00018A	Reserved
68	60	0x00008C	0x00018C	Reserved
69	61	0x00008E	0x00018E	Reserved
70	62	0x000090	0x000190	Reserved
71	63	0x000092	0x000192	Reserved
72	64	0x000094	0x000194	Reserved
73	65	0x000096	0x000196	U1E – UART1 Error
74	66	0x000098	0x000198	Reserved
75	67	0x00009A	0x00019A	Reserved
76	68	0x00009C	0x00019C	Reserved
77	69	0x00009E	0x00019E	Reserved
78	70	0x0000A0	0x0001A0	Reserved
79	71	0x0000A2	0x0001A2	Reserved
80-125	72-117	0x0000A4- 0x0000FE	0x0001A4- 0x0001FE	Reserved

TABLE 7-2: TRAP VECTORS

Vector Number	IVT Address	AIVT Address	Trap Source
0	0x000004	0x000104	Reserved
1	0x000006	0x000106	Oscillator Failure
2	0x000008	0x000108	Address Error
3	0x00000A	0x00010A	Stack Error
4	0x00000C	0x00010C	Math Error
5	0x00000E	0x00010E	Reserved
6	0x000010	0x000110	Reserved
7	0x000012	0x000112	Reserved

© 2007-2011 Microchip Technology Inc.

8.0 OSCILLATOR CONFIGURATION

Note 1: This data sheet summarizes the features of the dsPIC33FJ12GP201/202 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 7. "Oscillator"** (DS70186) of the *"dsPIC33F/PIC24H Family Reference Manual"*, which is available from the Microchip website (www.microchip.com).

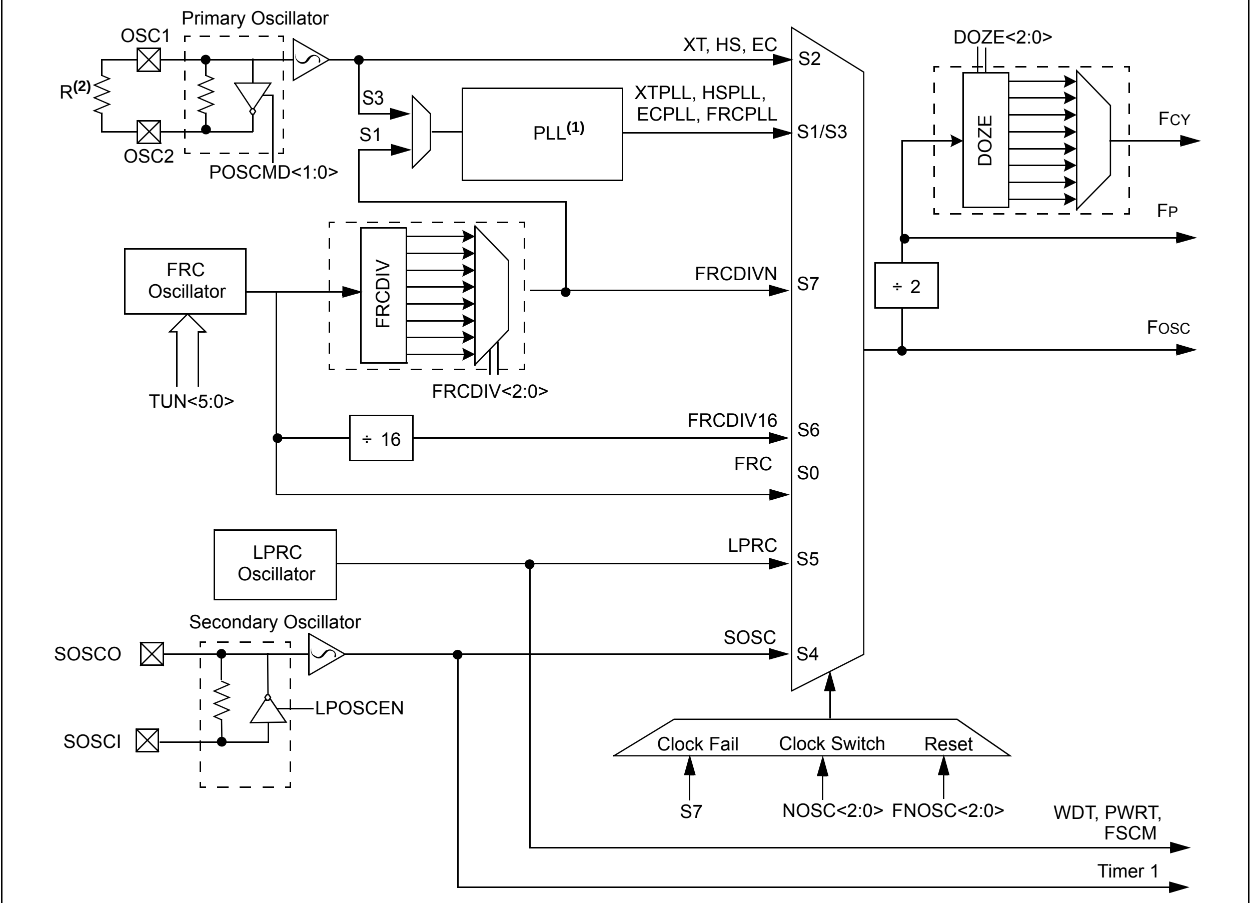
2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33FJ12GP201/202 oscillator system provides:

- External and internal oscillator options as clock sources
- An on-chip PLL to scale the internal operating frequency to the required system clock frequency
- An internal FRC oscillator that can also be used with the PLL, thereby allowing full-speed operation without any external clock generation hardware
- Clock switching between various clock sources
- Programmable clock postscaler for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and takes fail-safe measures
- An Oscillator Control register (OSCCON)
- Nonvolatile Configuration bits for main oscillator selection.

A simplified diagram of the oscillator system is shown in Figure 8-1.

FIGURE 8-1: dsPIC33FJ12GP201/202 OSCILLATOR SYSTEM DIAGRAM



Note 1: See Figure 8-2 for PLL details.

2: If the Oscillator is used with XT or HS modes, an external parallel resistor with the value of 1 MW must be connected.

3: The term FP refers to the clock source for all peripherals, while FCY refers to the clock source for the CPU. Throughout this document, FCY and FP are used interchangeably, except in the case of Doze mode. FP and FCY will be different when Doze mode is used with a Doze ratio of 1:2 or lower.

REGISTER 8-3: PLLFBD: PLL FEEDBACK DIVISOR REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0(1)
—	—	—	—	—	—	—	PLLDIV<8>
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
PLLDIV<7:0>							
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 15-9	**Unimplemented:** Read as ‘0’
bit 8-0	**PLLDIV<8:0>:** PLL Feedback Divisor bits (also denoted as ‘M’, PLL multiplier)

111111111 = 513
•
•
•
000110000 = 50 (default)
•
•
•
000000010 = 4
000000001 = 3
000000000 = 2

Note 1: This register is reset only on a Power-on Reset (POR).

© 2007-2011 Microchip Technology Inc.

10.1.1 OPEN-DRAIN CONFIGURATION

In addition to the PORT, LAT, and TRIS registers for data control, some 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} (e.g., 5V) on any 5V-tolerant pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum V_{IH} specification.

See **"Pin Diagrams"** for the available pins and their functionality.

10.2 Configuring Analog Port Pins

The AD1PCFG and TRIS registers control the operation of the Analog-to-Digital (A/D) port pins. The port pins that are desired 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 AD1PCFGL 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 that is defined as a digital input (including the ANx pins) can cause the input buffer to consume current that exceeds the device specifications.

10.2.1 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.

10.3 Input Change Notification

The input change notification function of the I/O ports allows the dsPIC33FJ12GP201/202 devices to generate interrupt requests to the processor in response to a change-of-state on selected input pins. This feature can detect input change-of-states even in Sleep mode, when the clocks are disabled. Depending on the device pin count, up to 21 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 and CNEN2 registers contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

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 push-button or keypad devices are connected. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain 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
```

© 2007-2011 Microchip Technology Inc.

REGISTER 10-1: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT1R<4:0>				
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 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **INT1R<4:0>:** Assign External Interrupt 1 (INTR1) to the corresponding RPn pin bits

11111 = Input tied to Vss
01111 = Input tied to RP15
•
•
•
00001 = Input tied to RP1
00000 = Input tied to RP0

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

© 2007-2011 Microchip Technology Inc.

REGISTER 10-11: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTERS 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP3R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP2R<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 15-13	**Unimplemented:** Read as ‘0’
bit 12-8	**RP3R<4:0>:** Peripheral Output Function is Assigned to RP3 Output Pin bits (see Table 10-2 for peripheral function numbers)
bit 7-5	**Unimplemented:** Read as ‘0’
bit 4-0	**RP2R<4:0>:** Peripheral Output Function is Assigned to RP2 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 10-12: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTERS 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP4R<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 15-13	**Unimplemented:** Read as ‘0’
bit 12-8	**RP5R<4:0>:** Peripheral Output Function is Assigned to RP5 Output Pin bits (see Table 10-2 for peripheral function numbers)
bit 7-5	**Unimplemented:** Read as ‘0’
bit 4-0	**RP4R<4:0>:** Peripheral Output Function is Assigned to RP4 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

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

© 2007-2011 Microchip Technology Inc.

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

bit 5	**ADDEN:** Address Character Detect bit (bit 8 of received data = `1`) `1` = Address Detect mode enabled. If 9-bit mode is not selected, this does not take effect `0` = Address Detect mode disabled
bit 4	**RIDLE:** Receiver Idle bit (read-only) `1` = Receiver is Idle `0` = Receiver is active
bit 3	**PERR:** Parity Error Status bit (read-only) `1` = Parity error has been detected for the current character (character at the top of the receive FIFO) `0` = Parity error has not been detected
bit 2	**FERR:** Framing Error Status bit (read-only) `1` = Framing error has been detected for the current character (character at the top of the receive FIFO) `0` = Framing error has not been detected
bit 1	**OERR:** Receive Buffer Overrun Error Status bit (read-only/clear-only) `1` = Receive buffer has overflowed `0` = Receive buffer has not overflowed. Clearing a previously set OERR bit (`1` $\rightarrow$ `0` transition) will reset the receiver buffer and the UxRSR to the empty state
bit 0	**URXDA:** Receive Buffer Data Available bit (read-only) `1` = Receive buffer has data, at least one more character can be read `0` = Receive buffer is empty

Note 1: Refer to **Section 17. "UART"** (DS70188) in the *"dsPIC33F/PIC24H Family Reference Manual"* for information on enabling the UART module for transmit operation.

TABLE 20-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
1	ADD	`ADD Acc`	Add Accumulators	1	1	OA,OB,SA,SB
		`ADD f`	f = f + WREG	1	1	C,DC,N,OV,Z
		`ADD f,WREG`	WREG = f + WREG	1	1	C,DC,N,OV,Z
		`ADD #lit10,Wn`	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		`ADD Wb,Ws,Wd`	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		`ADD Wb,#lit5,Wd`	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		`ADD Wso,#Slit4,Acc`	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	`ADDC f`	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		`ADDC f,WREG`	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		`ADDC #lit10,Wn`	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		`ADDC Wb,Ws,Wd`	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		`ADDC Wb,#lit5,Wd`	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	`AND f`	f = f .AND. WREG	1	1	N,Z
		`AND f,WREG`	WREG = f .AND. WREG	1	1	N,Z
		`AND #lit10,Wn`	Wd = lit10 .AND. Wd	1	1	N,Z
		`AND Wb,Ws,Wd`	Wd = Wb .AND. Ws	1	1	N,Z
		`AND Wb,#lit5,Wd`	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	`ASR f`	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		`ASR f,WREG`	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		`ASR Ws,Wd`	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		`ASR Wb,Wns,Wnd`	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		`ASR Wb,#lit5,Wnd`	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	`BCLR f,#bit4`	Bit Clear f	1	1	None
		`BCLR Ws,#bit4`	Bit Clear Ws	1	1	None
6	BRA	`BRA C,Expr`	Branch if Carry	1	1 (2)	None
		`BRA GE,Expr`	Branch if greater than or equal	1	1 (2)	None
		`BRA GEU,Expr`	Branch if unsigned greater than or equal	1	1 (2)	None
		`BRA GT,Expr`	Branch if greater than	1	1 (2)	None
		`BRA GTU,Expr`	Branch if unsigned greater than	1	1 (2)	None
		`BRA LE,Expr`	Branch if less than or equal	1	1 (2)	None
		`BRA LEU,Expr`	Branch if unsigned less than or equal	1	1 (2)	None
		`BRA LT,Expr`	Branch if less than	1	1 (2)	None
		`BRA LTU,Expr`	Branch if unsigned less than	1	1 (2)	None
		`BRA N,Expr`	Branch if Negative	1	1 (2)	None
		`BRA NC,Expr`	Branch if Not Carry	1	1 (2)	None
		`BRA NN,Expr`	Branch if Not Negative	1	1 (2)	None
		`BRA NOV,Expr`	Branch if Not Overflow	1	1 (2)	None
		`BRA NZ,Expr`	Branch if Not Zero	1	1 (2)	None
		`BRA OA,Expr`	Branch if Accumulator A overflow	1	1 (2)	None
		`BRA OB,Expr`	Branch if Accumulator B overflow	1	1 (2)	None
		`BRA OV,Expr`	Branch if Overflow	1	1 (2)	None
		`BRA SA,Expr`	Branch if Accumulator A saturated	1	1 (2)	None
		`BRA SB,Expr`	Branch if Accumulator B saturated	1	1 (2)	None
		`BRA Expr`	Branch Unconditionally	1	2	None
		`BRA Z,Expr`	Branch if Zero	1	1 (2)	None
		`BRA Wn`	Computed Branch	1	2	None
7	BSET	`BSET f,#bit4`	Bit Set f	1	1	None
		`BSET Ws,#bit4`	Bit Set Ws	1	1	None
8	BSW	`BSW.C Ws,Wb`	Write C bit to Ws<Wb>	1	1	None
		`BSW.Z Ws,Wb`	Write Z bit to Ws<Wb>	1	1	None
9	BTG	`BTG f,#bit4`	Bit Toggle f	1	1	None
		`BTG Ws,#bit4`	Bit Toggle Ws	1	1	None

© 2007-2011 Microchip Technology Inc.

21.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

21.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

21.4 MPASM Assembler

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

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

The MPASM Assembler features include:

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

21.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

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

The object linker/library features include:

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

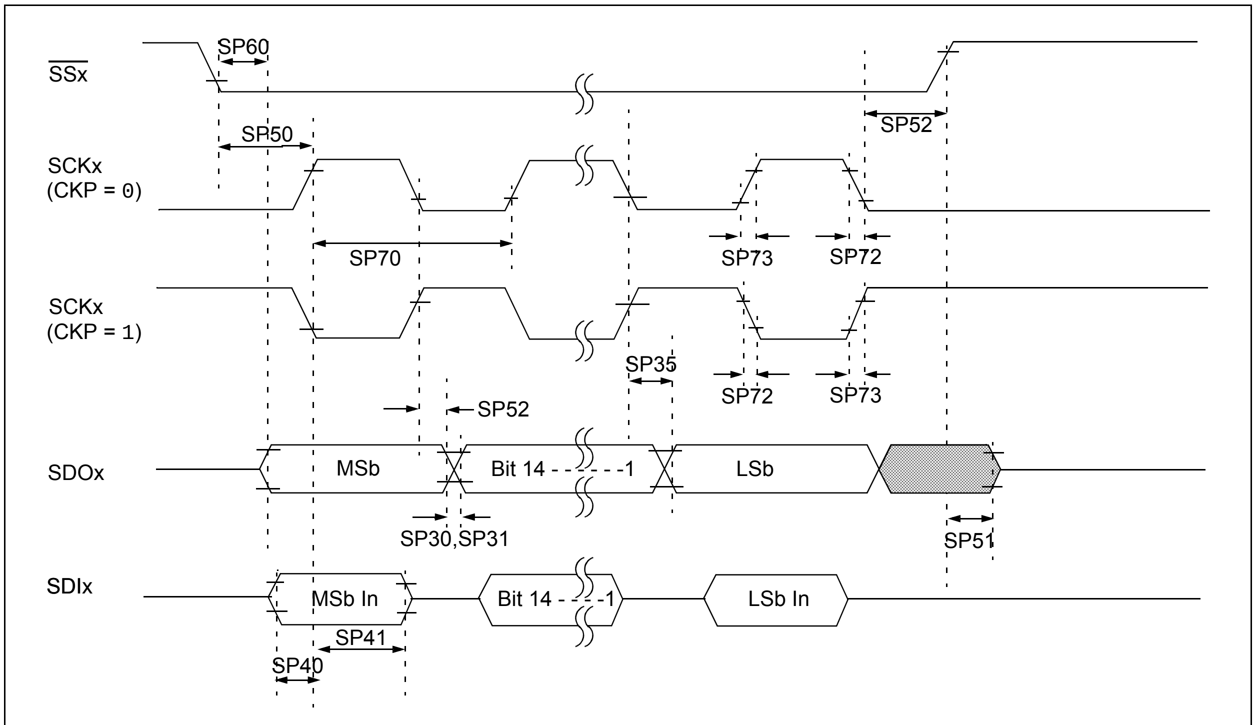
21.6 MPLAB Assembler, Linker and Librarian for Various Device Families

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

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

© 2007-2011 Microchip Technology Inc.

FIGURE 22-14: SPIx SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = 1, SMP = 0) TIMING CHARACTERISTICS

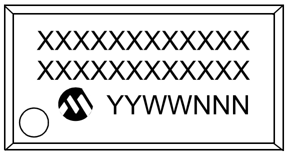


Note: Refer to Figure 22-1 for load conditions.

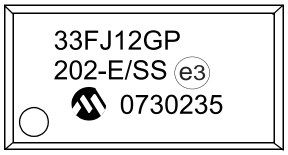
© 2007-2011 Microchip Technology Inc.

23.1 Package Marking Information (Continued)

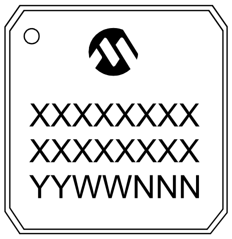
28-Lead SSOP



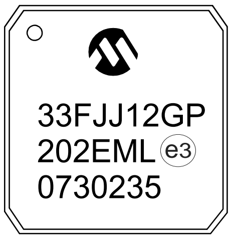
Example



28-Lead QFN



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: If the full Microchip part number cannot be marked on one line, it is carried over to the next line, thus limiting the number of available characters for customer-specific information.

© 2007-2011 Microchip Technology Inc.

TABLE 23-1: MAJOR SECTION UPDATES

Section Name	Update Description
Section 10.0 “I/O Ports”	Added paragraph and Table 10-1 to **Section 10.1.1 “Open-Drain Configuration”**, which provides details on I/O pins and their functionality. Removed the following sections, which are now available in the related section of the *“dsPIC33F Family Reference Manual”*: • 9.4.2 “Available Peripherals” • 9.4.3.3 “Mapping” • 9.4.5 “Considerations for Peripheral Pin Selection”
Section 14.0 “Output Compare”	Replaced sections 13.1, 13.2, and 13.3 and related figures and tables with entirely new content.
Section 15.0 “Serial Peripheral Interface (SPI)”	Removed the following sections, which are now available in the related section of the *“dsPIC33F Family Reference Manual”*: • 14.1 “Interrupts” • 14.2 “Receive Operations” • 14.3 “Transmit Operations” • 14.4 “SPI Setup” (retained Figure 15-1: SPI Module Block Diagram)
Section 16.0 “Inter-Integrated Circuit™ (I^2C™)”	Removed the following sections, which are now available in the related section of the *“dsPIC33F Family Reference Manual”*: • 15.3 “I^2C Interrupts” • 15.4 “Baud Rate Generator” (retained Figure 16-1: I^2C Block Diagram) • 15.5 “I^2C Module Addresses • 15.6 “Slave Address Masking” • 15.7 “IPMI Support” • 15.8 “General Call Address Support” • 15.9 “Automatic Clock Stretch” • 15.10 “Software Controlled Clock Stretching (STREN = 1)” • 15.11 “Slope Control” • 15.12 “Clock Arbitration” • 15.13 “Multi-Master Communication, Bus Collision, and Bus Arbitration • 15.14 “Peripheral Pin Select Limitations
Section 17.0 “Universal Asynchronous Receiver Transmitter (UART)”	Removed the following sections, which are now available in the related section of the *“dsPIC33F Family Reference Manual”*: • 16.1 “UART Baud Rate Generator” • 16.2 “Transmitting in 8-bit Data Mode • 16.3 “Transmitting in 9-bit Data Mode • 16.4 “Break and Sync Transmit Sequence” • 16.5 “Receiving in 8-bit or 9-bit Data Mode” • 16.6 “Flow Control Using $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Pins” • 16.7 “Infrared Support” Removed IrDA references and Note 1, and updated the bit and bit value descriptions for UTXINV (UxSTA<14>) in the UARTx Status and Control Register (see Register 17-2).

TABLE 23-1: MAJOR SECTION UPDATES

Section Name	Update Description
Section 22.0 "Electrical Characteristics"	Updated Max MIPS value for -40ºC to +125ºC temperature range in Operating MIPS vs. Voltage (see Table 22-1). Added 28-pin SSOP package information to Thermal Packaging Characteristics and updated Typical values for all devices (see Table 22-3). Removed Typ value for parameter DC12 (see Table 22-4). Updated Note 2 in Table 22-7: DC Characteristics: Power-Down Current (IPD). Updated MIPS conditions for parameters DC24c, DC44c, DC72a, DC72f, and DC72g (see Table 22-5, Table 22-6, and Table 22-8). Added Note 4 (reference to new table containing digital-only and analog pin information to I/O Pin Input Specifications (see Table 22-9). Updated Program Memory parameters (D136a, D136b, D137a, D137b, D138a, and D138b) and added Note 2 (see Table 22-12). Updated Max value for Internal RC Accuracy parameter F21 for -40°C ≤TA ≤ +125°C condition and added Note 2 (see Table 22-19). Removed all values for Reset, Watchdog Timer, Oscillator Start-up Timer, and Power-up Timer parameter SY20 and updated conditions, which now refers to **Section 19.4 "Watchdog Timer (WDT)"** and LPRC parameter F21 (see Table 22-21). The following changes were made to the ADC Module Specifications (Table 22-34): • Updated Min value for ADC Module Specification parameter AD07 • Updated Typ value for parameter AD08 • Removed parameter AD10 • Added references to Note 1 for parameters AD12 and AD13 • Removed Note 2. The following changes were made to the ADC Module Specifications (12-bit Mode) (Table 22-35): • Updated Min and Max values for both AD21a parameters (measurements with *internal* and *external* VREF+/VREF-). • Updated Min, Typ, and Max values for parameter AD24a. • Updated Max value for parameter AD32a. • Removed Note 1. • Removed VREFL from Conditions for parameters AD21a, AD22a, AD23a, and AD24a (measurements with *internal* VREF+/VREF-). The following changes were made to the ADC Module Specifications (10-bit Mode) (Table 22-36): • Updated Min and Max values for parameter AD21b (measurements with *external* VREF+/VREF-). • Removed ± symbol from Min, Typ, and Max values for parameters AD23b and AD24b (measurements with *internal* VREF+/VREF-). • Updated Typ and Max values for parameter AD32b. • Removed Note 1. • Removed VREFL from Conditions for parameters AD21a, AD22a, AD23a, and AD24a (measurements with *internal* VREF+/VREF-). Updated Min and Typ values for parameters AD60, AD61, AD62, and AD63 and removed Note 3 (see Table 22-37 and Table 22-38).

INDEX