



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	16 MIPs
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj16gp101-i-p

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong







FIGURE 1-1: dsPIC33FJ16(GP/MC)101/102 AND dsPIC33FJ32(GP/MC)101/102/104 BLOCK DIAGRAM



2.7 Oscillator Value Conditions on Device Start-up

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to 4 MHz < FIN < 8 MHz (for MSPLL mode) or 3 MHz < FIN < 8 MHz (for ECPLL mode) to comply with device PLL start-up conditions. HSPLL mode is not supported. This means that if the external oscillator frequency is outside this range, the application must start-up in the FRC mode first. The fixed PLL settings of 4x after a POR with an oscillator frequency outside this range will violate the device operating speed.

Once the device powers up, the application firmware can enable the PLL and then perform a clock switch to the Oscillator + PLL clock source. Note that clock switching must be enabled in the device Configuration Word.

2.8 Configuration of Analog and Digital Pins During ICSP Operations

If MPLAB ICD 3 or MPLAB REAL ICE in-circuit emulator is selected as a debugger, it automatically initializes all of the Analog-to-Digital input pins (ANx) as "digital" pins, by setting all bits in the AD1PCFGL register.

The bits in the register that correspond to the Analog-to-Digital pins that are initialized by MPLAB ICD 3 or MPLAB REAL ICE in-circuit emulator, must not be cleared by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain Analog-to-Digital pins as analog input pins during the debug session, the user application must clear the corresponding bits in the AD1PCFGL register during initialization of the ADC module.

When MPLAB ICD 3 or MPLAB REAL ICE in-circuit emulator is used as a programmer, the user application firmware must correctly configure the AD1PCFGL register. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all Analog-to-Digital pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.9 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic-low state.

Alternately, connect a 1k to 10k resistor between Vss and unused pins.

3.3 Special MCU Features

The dsPIC33FJ16(GP/MC)101/102 and dsPIC33FJ32(GP/MC)101/102/104 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).

The dsPIC33FJ16(GP/MC)101/102 and dsPIC33FJ32(GP/MC)101/102/104 supports 16/16 and 32/16 divide operations, both fractional and integer. All divide instructions are iterative operations. They must be executed within a REPEAT loop, resulting in a total execution time of 19 instruction cycles. The divide operation can be interrupted during any of those 19 cycles without loss of data.

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

FIGURE 3-1: dsPIC33FJ16(GP/MC)101/102 AND dsPIC33FJ32(GP/MC)101/102/104 CPU CORE BLOCK DIAGRAM



NOTES:

FIGURE 4-2: PROGRAM MEMORY MAP FOR dsPIC33FJ32(GP/MC)101/102/104 DEVICES



4.4.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the Effective Address (EA) calculation associated with any W register. Address boundaries check for addresses equal to:

- The upper boundary addresses for incrementing buffers
- The lower boundary addresses for decrementing buffers

It is important to realize that the address boundaries check for addresses less than or greater than the upper (for incrementing buffers) and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes can, therefore, jump beyond boundaries and still be adjusted correctly.

Note: The modulo corrected Effective Address is written back to the register only when Pre-Modify or Post-Modify Addressing mode is used to compute the Effective Address. When an address offset (such as [W7 + W2]) is used, Modulo Addressing correction is performed, but the contents of the register remain unchanged.

4.5 Bit-Reversed Addressing

Bit-Reversed Addressing mode is intended to simplify data reordering for radix-2 FFT algorithms. It is supported by the X AGU for data writes only.

The modifier, which can be a constant value or register contents, is regarded as having its bit order reversed. The address source and destination are kept in normal order. Thus, the only operand requiring reversal is the modifier.

4.5.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

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

- BWM<3:0> bits (W register selection) in the MODCON register are any value other than '15' (the stack cannot be accessed using Bit-Reversed Addressing)
- The BREN bit is set in the XBREV register
- The addressing mode used is Register Indirect with Pre-Increment or Post-Increment

If the length of a bit-reversed buffer is $M = 2^N$ bytes, the last 'N' bits of the data buffer start address must be zeros.

XB<14:0> is the bit-reversed address modifier, or 'pivot point,' which is typically a constant. In the case of an FFT computation, its value is equal to half of the FFT data buffer size.

Note: All bit-reversed EA calculations assume word-sized data (LSb of every EA is always clear). The XB<14:0> value is scaled accordingly to generate compatible (byte) addresses.

When enabled, Bit-Reversed Addressing is executed only for Register Indirect with Pre-Increment or Post-Increment Addressing and word-sized data writes. It will not function for any other addressing mode or for byte-sized data and normal addresses are generated instead. When Bit-Reversed Addressing is active, the W Address Pointer is always added to the address modifier (XB) and the offset associated with the Register Indirect Addressing mode is ignored. In addition, as word-sized data is a requirement, the LSb of the EA is ignored (and always clear).

Note: Modulo Addressing and Bit-Reversed Addressing should not be enabled together. If an application attempts to do so, Bit-Reversed Addressing will assume priority when active. For the X WAGU and Y AGU, Modulo Addressing will be disabled. However, Modulo Addressing will continue to function in the X RAGU.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV<15>) bit, a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the Bit-Reversed Pointer.

REGISTER 6-1: RCON: RESET CONTROL REGISTER⁽¹⁾ (CONTINUED)

- bit 2 IDLE: Wake-up from Idle Flag bit
 - 1 = Device has been in Idle mode
 - 0 = Device has not been in Idle mode
- bit 1 BOR: Brown-out Reset Flag bit
 - 1 = A Brown-out Reset has occurred
 - 0 = A Brown-out Reset has not occurred
- bit 0 **POR:** Power-on Reset Flag bit
 - 1 = A Power-on Reset has occurred
 - 0 = A Power-on Reset has not occurred
- **Note 1:** All of the Reset status bits can be set or cleared in software. Setting one of these bits in software does not cause a device Reset.
 - 2: If the FWDTEN Configuration bit is set to '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	_	_	_		_		_
bit 15	·						bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2 ⁽¹⁾	T5IP1 ⁽¹⁾	T5IP0 ⁽¹⁾
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimple	mented bit, read	l as '0'	
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cleared		x = Bit is unkr	nown
bit 15-7	Unimplemen	ted: Read as '	0'				
bit 6-4	INT2IP<2:0>:	External Inter	rupt 2 Priority	bits			
	111 = Interru	pt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	001 = Interru	pt is Priority 1					
	000 = Interru	pt source is dis	abled				
bit 3	Unimplemen	ted: Read as '	0'				
bit 2-0	T5IP<2:0>: ⊤	imer5 Interrupt	Priority bits ⁽¹⁾				
	111 = Interru	pt is Priority 7 (highest priorit	y interrupt)			
	•						
	•						
	001 = Interru	pt is Priority 1					
	000 = Interru	pt source is dis	abled				

REGISTER 7-22: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

Note 1: These bits are available in dsPIC33FJ32(GP/MC)10X devices only.

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—
bit 7					•		bit 0
Legend:							

REGISTER 7-23: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

1.	~~	~ .	~ ~	۱.
Le	эg	e	nc	1.

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	IC3IP<2:0>: External Interrupt 3 Priority bits
	111 = Interrupt is Priority 7 (highest priority interrupt)
	•
	•
	•
	001 = Interrupt is Priority 1
	000 = Interrupt source is disabled
bit 3-0	Unimplemented: Read as '0'

REGISTER 7-24: IPC14: INTERRUPT PRIORITY CONTROL REGISTER 14

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PWM1IP2 ⁽¹⁾	PWM1IP1 ⁽¹⁾	PWM1IP0 ⁽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 15-7	Unimplemented: Read as '0'
bit 6-4	PWM1IP<2:0>: PWM1 Interrupt Priority bits ⁽¹⁾
	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'
bit 3-0	Unimplemented: Read as '0'

Note 1: These bits are available in dsPIC(16/32)MC10X devices only.





EXAMPLE 15-1: ASSEMBLY CODE FOR WRITE-PROTECTED REGISTER UNLOCK AND FAULT CLEARING SEQUENCE

; FLTA1 pin must be pulled high externally in order to clear and disable the Fault ; Writing to P1FLTBCON register requires unlock sequence						
<pre>mov #0xabcd,w10 mov #0x4321,w11 mov #0x0000,w0 mov w10, PWM1KEY mov w11, PWM1KEY mov w0,P1FLTACON</pre>	<pre>; Load first unlock key to w10 register ; Load second unlock key to w11 register ; Load desired value of P1FLTACON register in w0 ; Write first unlock key to PWM1KEY register ; Write second unlock key to PWM1KEY register ; Write desired value to P1FLTACON register</pre>					
; FLTB1 pin must be pu ; Writing to P1FLTBCON	ulled high externally in order to clear and disable the Fault N register requires unlock sequence					
mov #0xabcd,w10	; Load first unlock key to w10 register					
mov #0x4321,w11	; Load second unlock key to w11 register					
mov #0x0000,w0	; Load desired value of P1FLTBCON register in w0					
mov w10, PWM1KEY	; Write first unlock key to PWM1KEY register					
mov w11, PWM1KEY	; Write second unlock key to PWM1KEY register					
mov w0, P1FLTBCON	; Write desired value to P1FLTBCON register					
; Enable all PWMs usir	g PWM1CON1 register					
; Writing to PWM1CON1	register requires unlock sequence					
mov #0xabcd,w10	; Load first unlock key to w10 register					
mov #0x4321,w11	; Load second unlock key to w11 register					
mov #0x0077,w0	; Load desired value of PWM1CON1 register in w0					
mov w10, PWM1KEY	; Write first unlock key to PWM1KEY register					
mov w11, PWM1KEY	; Write second unlock key to PWM1KEY register					
mov w0,PWM1CON1	; Write desired value to PWM1CON1 register					

EXAMPLE 15-2: C CODE FOR WRITE-PROTECTED REGISTER UNLOCK AND FAULT CLEARING SEQUENCE

// FLTAl pin must be pulled high externally in order to clear and disable the Fault // Writing to PIFLTACON register requires unlock sequence // Use builtin function to write 0x0000 to PIFLTACON register __builtin_write_PWMSFR(&PIFLTACON, 0x0000, &PWM1KEY); // FLTBl pin must be pulled high externally in order to clear and disable the Fault // Writing to PIFLTBCON register requires unlock sequence // Use builtin function to write 0x0000 to PIFLTBCON register __builtin_write_PWMSFR(&PIFLTBCON, 0x0000, &PWM1KEY); // Enable all PWMs using PWM1CON1 register // Writing to PWM1CON1 register requires unlock sequence // Use builtin function to write 0x0077 to PWM1CON1 register __builtin_write_PWMSFR(&PWM1CON1, 0x0077, &PWM1KEY);

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, read	as '0'	
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
bit 15	FRMEN: Frar	med SPIx Supp	ort bit				
	1 = Framed S	SPIx support is	enabled (<mark>SS</mark> x	pin is used as	Frame Sync pu	Ilse input/outpu	t)
	0 = Framed S	SPIx support is	disabled				
bit 14	SPIFSD: Fran	me Sync Pulse	Direction Cor	ntrol bit			
	1 = Frame Sy	nc pulse input	(slave)				
hit 12		ama Svna Bula	o Dolority bit				
DIL 13	1 - Erame Sv	une oyne ruis vne oulse is act					
	0 = Frame Sy	nc pulse is act	ive-low				
bit 12-2	Unimplemen	ted: Read as '	0'				
bit 1	FRMDLY: Fra	ame Sync Pulse	e Edge Select	bit			
	1 = Frame Sy	nc pulse coinci	ides with first	bit clock			
	0 = Frame Sy	nc pulse prece	des first bit cl	ock			
bit 0	Unimplemen	ted: This bit m	ust not be set	to '1' by the us	ser application		

REGISTER 16-3: SPIxCON2: SPIx CONTROL REGISTER 2

17.0 INTER-INTEGRATED CIRCUIT™ (I²C™)

- Note 1: This data sheet summarizes the features of the dsPIC33FJ16(GP/MC)101/102 and dsPIC33FJ32(GP/MC)101/102/104 family devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to "Inter-Integrated CircuitTM (I²CTM)" (DS70195) in the "dsPIC33/ PIC24 Family Reference Manual", which is available from the Microchip web site (www.microchip.com).
 - 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 CircuitTM (I^2C^{TM}) 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 I²C module has a 2-pin interface:

- The SCLx pin is clock
- The SDAx pin is data

The I²C module offers the following key features:

- I²C interface supporting both Master and Slave modes of operation
- I²C Slave mode supports 7-bit and 10-bit addresses
- I²C Master mode supports 7-bit and 10-bit addresses
- I²C port allows bidirectional transfers between master and slaves
- Serial clock synchronization for I²C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control)
- I²C 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 I²C module can operate either as a slave or a master on an I²C bus.

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

- I²C slave operation with 7-Bit Addressing
- I²C slave operation with 10-Bit Addressing
- I²C master operation with 7-Bit or 10-Bit Addressing

For details about the communication sequence in each of these modes, refer to the Microchip web site (www.microchip.com) for the latest *"dsPIC33/PIC24 Family Reference Manual"* sections.

17.2 I²C Registers

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

- I2CxRSR is the shift register used for shifting data
- I2CxRCV is the receive buffer and the register to which data bytes are written or from which data bytes are read
- I2CxTRN is the transmit register to which bytes are written during a transmit operation
- · I2CxADD register holds the slave address
- ADD10 status bit indicates 10-Bit Addressing mode
- I2CxBRG acts as the Baud Rate Generator (BRG) reload value

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

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0	R-1	
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN ⁽¹⁾	UTXBF	TRMT	
bit 15							bit 8	
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0	
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	
bit 7							bit 0	
Legend:		C = Clearable bit		HC = Hardware Clearable bit				
R = Readable bit		W = Writable bit		U = Unimplemented bit, read as '0'				
-n = Value at POR		'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown		
bit 15,13 UTXISEL<1:0>: UARTx Transmission Interrupt Mode Selection bits 11 = Reserved; do not use								
								10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR) and as transmit buffer becomes empty
01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operation are completed								

REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least
one character open in the transmit buffer)

bit 14	UTXINV: UARTx Transmit Polarity Inversion bit				
	If IREN = 0:				
	1 = UxTX Idle state is '0'				
	0 = UxTX Idle state is '1'				
	<u>If IREN = 1:</u>				
	1 = IrDA encoded, UxTX Idle state is '1'				
	0 = IrDA encoded, UxTX Idle state is '0'				
bit 12	Unimplemented: Read as '0'				
bit 11	UTXBRK: UARTx Transmit Break bit				
	1 = Sends Sync Break on next transmission – Start bit, followed by twelve '0' bits, followed by Stop bit; cleared by hardware upon completion				
	0 = Sync Break transmission is disabled or completed				
bit 10	UTXEN: UARTx Transmit Enable bit ⁽¹⁾				
	1 = Transmit is enabled, UxTX pin is controlled by UARTx				
	0 = Transmit is disabled, any pending transmission is aborted and the buffer is reset; UxTX pin is controlled by port				
bit 9	UTXBF: UARTx Transmit Buffer Full Status bit (read-only)				
	1 = Transmit buffer is full				
	0 = Transmit buffer is not full, at least one more character can be written				
bit 8	TRMT: Transmit Shift Register Empty bit (read-only)				
	1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)				
bit 7 6	UDVISEL 41.0 + LIADTy Dessive Interrupt Mode Selection hits				
DIT 7-6	URAISEL<1:0>: UARTX Receive interrupt wode Selection bits				
	11 = Interrupt is set on UXRSR transfer, making the receive buffer full (i.e., has 4 data characters)				
	10 = Interrupt is set on OXRSR transfer, making the received and transferred from the LIXRSR to the received				
	buffer; receive buffer has one or more characters				

Note 1: Refer to "**UART**" (DS70188) in the "*dsPIC33/PIC24 Family Reference Manual*" for information on enabling the UART module for transmit operation.

23.5 In-Circuit Serial Programming[™] (ICSP[™])

Devices can be serially programmed while in the end application circuit. This is done with two lines for clock and data and three other lines for power, ground and the programming sequence. Serial programming allows customers to manufacture boards with unprogrammed devices and then program the Digital Signal Controller just before shipping the product. Serial programming also allows the most recent firmware or a custom firmware to be programmed. Refer to the *"dsPIC33F Flash Programming Specification for Devices with Volatile Configuration Bits"* (DS70659) for details about In-Circuit Serial Programming (ICSP).

Any of the three pairs of programming clock/data pins can be used:

- PGEC1 and PGED1
- PGEC2 and PGED2
- PGEC3 and PGED3

23.6 In-Circuit Debugger

When MPLAB[®] ICD 3 is selected as a debugger, the incircuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx (Emulation/Debug Clock) and PGEDx (Emulation/Debug Data) pin functions.

Any of the three pairs of debugging clock/data pins can be used:

- PGEC1 and PGED1
- PGEC2 and PGED2
- PGEC3 and PGED3

To use the in-circuit debugger function of the device, the design must implement ICSP connections to \overline{MCLR} , VDD, Vss and the PGECx/PGEDx pin pair. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins.

24.0 INSTRUCTION SET SUMMARY

Note: This data sheet summarizes the features of the dsPIC33FJ16(GP/ MC)101/102 and dsPIC33FJ32(GP/ MC)101/102/104 devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the latest family reference sections of the "dsPIC33/PIC24 Family Reference Manual", which are available from the Microchip web site (www.microchip.com).

The dsPIC33F instruction set is identical to that of the dsPIC30F.

Most instructions are a single program memory word (24 bits). Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word, divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into five basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- DSP operations
- Control operations

Table 24-1 shows the general symbols used in describing the instructions.

The dsPIC33F instruction set summary in Table 24-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register 'Wb' without any address modifier
- The second source operand, which is typically a register 'Ws' with or without an address modifier
- The destination of the result, which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value 'f'
- The destination, which could be either the file register 'f' or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/ shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register 'Wb')

The literal instructions that involve data movement can use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register 'Wb' without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register 'Wd' with or without an address modifier

The MAC class of DSP instructions can use some of the following operands:

- The accumulator (A or B) to be used (required operand)
- The W registers to be used as the two operands
- · The X and Y address space prefetch operations
- The X and Y address space prefetch destinations
- The accumulator write-back destination

The other DSP instructions do not involve any multiplication and can include:

- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift specified by a W register 'Wn' or a literal value

The control instructions can use some of the following operands:

- A program memory address
- The mode of the Table Read and Table Write instructions





44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

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



	MILLIMETERS			
Dimension	Dimension Limits		NOM	MAX
Number of Leads	Ν	44		
Lead Pitch	е	0.80 BSC		
Overall Height	А	-	-	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	—	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	φ	0°	3.5°	7°
Overall Width	Е	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	с	0.09	-	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom		11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Chamfers at corners are optional; size may vary.

3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.

- 4. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B