

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

E·XFI

Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DCI, DMA, I ² S, POR, PWM, WDT
Number of I/O	21
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 10x10b/12b
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/dspic33fj32gp202-i-so

Email: info@E-XFL.COM

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

Pin Name	Pin Type	Buffer Type	PPS	Description			
VCAP	Р	—	No	CPU logic filter capacitor connection.			
Vss	Р	—	No	Ground reference for logic and I/O pins.			
VREF+	I	Analog	No	nalog voltage reference (high) input.			
VREF-	I	Analog	No	Analog voltage reference (low) input.			
Avdd	Р	Р	No	Positive supply for analog modules. This pin must be connected at all times.			
MCLR	I/P	ST	No	Master Clear (Reset) input. This pin is an active-low Reset to the device.			
Avss	Р	Р	No	Ground reference for analog modules.			
Vdd	Р	_	No	Positive supply for peripheral logic and I/O pins.			
Legend: $CMOS = CMOS$ compatible input or output: Analog = Analog input: P = Power							

TABLE 1-1: PINOUT I/O DESCRIPTIONS (CONTINUED)

Legend: CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; PPS = Peripheral Pin Select Analog = Analog input; P = Power O = Output; I = Input

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 \leq 8 MHz for start-up with PLL enabled to comply with device PLL start-up conditions. This means that if the external oscillator frequency is outside this range, the application must start-up in FRC mode first. The default PLL settings 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 initialize the PLL SFRs, CLKDIV and PLLDBF to a suitable value, 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 2, MPLAB ICD 3, or MPLAB REAL ICE in-circuit emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as "digital" pins, by setting all bits in the AD1PCFGL register.

The bits in the registers that correspond to the A/D 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 A/D 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 A/D 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.

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

4.6.3 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 also check for addresses less than or greater than these addresses. 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 Address correction is performed but the contents of the register remain unchanged.

4.7 Bit-Reversed Addressing

Bit-Reversed Addressing mode is intended to simplify data re-ordering 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.

4.7.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

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

- BWM 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 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 X WAGU 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 bit (XBREV<15>), 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.

5.5.1 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 bit (NVMCON<6>) and the WREN bit (NVMCON<14>).
 - 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.
- 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 NOPS, as shown in Example 5-3.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY PAGE

	; Set up NVMCO	N for block erase operation		
	MOV	#0x4042, W0	;	
	MOV	W0, NVMCON	;	Initialize NVMCON
	; Init pointer	to row to be ERASED		
	MOV	<pre>#tblpage(PROG_ADDR), W0</pre>	;	
	MOV	W0, TBLPAG	;	Initialize PM Page Boundary SFR
	MOV	<pre>#tbloffset(PROG_ADDR), W0</pre>	;	Initialize in-page EA[15:0] pointer
	TBLWTL	WO, [WO]	;	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
1				

REGISTER 7-8: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0 (CONTINUED)

bit 1	IC1IE: Input Capture Channel 1 Interrupt Enable bit
-------	---

- 1 = Interrupt request enabled
- 0 = Interrupt request not enabled
- INTOIE: External Interrupt 0 Enable bit
- 1 = Interrupt request enabled

bit 0

0 = Interrupt request not enabled

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			
—		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	Unimplemen	ted: Read as '	כי							
bit 6-4	INT2IP<2:0>:	: External Interr	upt 2 Priority	bits						
	111 = Interru	pt is priority 7 (I	highest priorit	y interrupt)						
	•									
	•									
	•									
		pt is priority 1 pt source is dis	abled							
bit 3-0	Unimplemen	ted: Read as '	ר <u>י</u> ייייייייייייייייייייייייייייייייייי							
511 0 0	- mpionion		<i>.</i>							

REGISTER 7-17: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

7.5 Interrupt Setup Procedures

7.5.1 INITIALIZATION

To configure an interrupt source at initialization:

- 1. Set the NSTDIS bit (INTCON1<15>) if nested interrupts are not desired.
- 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.

7.5.2 INTERRUPT SERVICE ROUTINE

The method used to declare an Interrupt Service Routine (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 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 old CPU priority level.

7.5.3 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.

7.5.4 INTERRUPT DISABLE

All user interrupts can be disabled using this procedure:

- 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 OEh 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.

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

bit 3	CF: Clock Fail Detect bit (read/clear by application) 1 = FSCM has detected clock failure 0 = FSCM has not detected clock failure
bit 2	Unimplemented: Read as '0'
bit 1	LPOSCEN: Secondary (LP) Oscillator Enable bit
	1 = Enable secondary oscillator0 = Disable secondary oscillator
bit 0	OSWEN: Oscillator Switch Enable bit
	1 = Request oscillator switch to selection specified by NOSC<2:0> bits

- 0 = Oscillator switch is complete
- Note 1: Writes to this register require an unlock sequence. Refer to Section 7. "Oscillator" (DS70186) in the "dsPIC33F/PIC24H Family Reference Manual" for details.
 - 2: Direct clock switches between any primary oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.
 - **3:** This register is reset only on a Power-on Reset (POR).

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
	_	_	_	—	_	PLLDIV<8>
						bit 8
R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
		PLLD	IV<7:0>			
						bit 0
R = Readable bit W = Writable			U = Unimple	mented bit, rea	d as '0'	
POR	'1' = Bit is set		'0' = Bit is cleared		x = Bit is unknown	
Unimplemer	ted: Read as ')'				
PLLDIV<8:0	>: PLL Feedbac	k Divisor bits	(also denoted	as 'M', PLL mu	ıltiplier)	
111111111	= 513					
•						
•						
•						
000110000	= 50 (default)					
•						
•						
•						
00000010	= 4					
000000000	= 3 = 2					
	U-0 	U-0 U-0 R/W-0 R/W-1 bit W = Writable POR '1' = Bit is set Unimplemented: Read as '0 PLLDIV<8:0>: PLL Feedbac 11111111 = 513	U-0 U-0 U-0 - - - R/W-0 R/W-1 R/W-1 PLLD PLLD bit W = Writable bit POR '1' = Bit is set Unimplemented: Read as '0' PLLDIV<8:0>: PLL Feedback Divisor bits 11111111 = 513 . . 000110000 = 50 (default) . <tr< td=""><td>U-0 U-0 U-0 U-0 - - - - - R/W-0 R/W-1 R/W-1 R/W-0 PLLDIV<7:0> bit W = Writable bit U = Unimplemented: POR '1' = Bit is set '0' = Bit is cle Unimplemented: Read as '0' PLLDIV<8:0>: PLL Feedback Divisor bits (also denoted 111111111 = 513 . . .</td><td>U-0 U-0 U-0 U-0 U-0 - - - - - - R/W-0 R/W-1 R/W-1 R/W-0 R/W-0 PLLDIV<7:0> PLLDIV<7:0></td><td>U-0 U-0 U-0 U-0 U-0 U-0 - <</td></tr<>	U-0 U-0 U-0 U-0 - - - - - R/W-0 R/W-1 R/W-1 R/W-0 PLLDIV<7:0> bit W = Writable bit U = Unimplemented: POR '1' = Bit is set '0' = Bit is cle Unimplemented: Read as '0' PLLDIV<8:0>: PLL Feedback Divisor bits (also denoted 111111111 = 513 . . .	U-0 U-0 U-0 U-0 U-0 - - - - - - R/W-0 R/W-1 R/W-1 R/W-0 R/W-0 PLLDIV<7:0> PLLDIV<7:0>	U-0 U-0 U-0 U-0 U-0 U-0 - <

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

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

13.0 INPUT CAPTURE

- **Note 1:** This data sheet summarizes the features of the dsPIC33FJ32GP202/204 and dsPIC33FJ16GP304 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 12. "Input Capture" (DS70198) "dsPIC33F/PIC24H Family of the Reference Manual", which is available from the Microchip web site (www.microchip.com).
 - 2: Some registers and associated bits described in this section may not be available on all devices. Refer to Section 4.0 "Memory Organization" in this data sheet for device-specific register and bit information.

The input capture module is useful in applications requiring frequency (period) and pulse measurement. The dsPIC33FJ32GP202/204 and dsPIC33FJ16GP304 devices support up to eight input capture channels.

The input capture module captures the 16-bit value of the selected Time Base register when an event occurs at the ICx pin. The events that cause a capture event are listed below in three categories:

- Simple Capture Event modes:
 - Capture timer value on every falling edge of input at ICx pin
 - Capture timer value on every rising edge of input at ICx pin
- Capture timer value on every edge (rising and falling).
- Prescaler Capture Event modes:
 - Capture timer value on every 4th rising edge of input at ICx pin
 - Capture timer value on every 16th rising edge of input at ICx pin

Each input capture channel can select one of two 16-bit timers (Timer2 or Timer3) for the time base. The selected timer can use either an internal or external clock.

Other operational features include:

- Device wake-up from capture pin during CPU Sleep and Idle modes
- Interrupt on input capture event
- · 4-word FIFO buffer for capture values
 - Interrupt optionally generated after 1, 2, 3 or 4 buffer locations are filled
- Use of input capture to provide additional sources of external interrupts



NOTES:

14.1 Output Compare Modes

Configure the Output Compare modes by setting the appropriate Output Compare Mode bits (OCM<2:0>) in the Output Compare Control register (OCxCON<2:0>). Table 14-1 lists the different bit settings for the Output Compare modes. Figure 14-2 illustrates the output compare operation for various modes. The user

TABLE 14-1: OUTPUT COMPARE MODES

application must disable the associated timer when writing to the output compare control registers to avoid malfunctions.

Note: See Section 13. "Output Compare" (DS70209) in the "dsPIC33F/PIC24H Family Reference Manual" for OCxR and OCxRS register restrictions.

OCM<2:0>	Mode	OCx Pin Initial State	OCx Interrupt Generation
000	Module Disabled	Controlled by GPIO register	—
001	Active-Low One-Shot	0	OCx Rising edge
010	Active-High One-Shot	1	OCx Falling edge
011	Toggle Mode	Current output is maintained	OCx Rising and Falling edge
100	Delayed One-Shot	0	OCx Falling edge
101	Continuous Pulse mode	0	OCx Falling edge
110	PWM mode without fault protection	0, if OCxR is zero 1, if OCxR is non-zero	No interrupt
111	PWM mode with fault protection	0, if OCxR is zero 1, if OCxR is non-zero	OCFA Falling edge for OC1 to OC4

FIGURE 14-2: OUTPUT COMPARE 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	Evor	Branch Linconditionally	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
ľ	2001	BSET	Ws.#bit4	Bit Set Ws	1	1	None
8	BSW	BSW C	Ws.Wb	Write C bit to Ws <wb></wb>	1	1	None
Ĭ	2011	BSW 7	Ws.Wb	Write Z bit to Ws <wb></wb>	1	1	None
9	BTG	BTG	f.#bit4	Bit Togale f	1	1	None
Ĵ	210	BTC	Ws #hit4	Bit Toggle Ws	1	1	None
L		510			· ·	ı .	None

IADL	L 20-2.	INSTR					
Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
48	MPY	MPY Wm*Wn,A	cc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,A	cc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
49	MPY.N	MPY.N Wm*Wn,Ad	cc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
50	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd , AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
51	MUL	MUL.SS	Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
		MUL.SU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.US	Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.UU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
		MUL.SU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.UU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL	f	W3:W2 = f * WREG	1	1	None
52	NEG	NEG	Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG	f	$f = \overline{f} + 1$	1	1	C,DC,N,OV,Z
		NEG	f,WREG	WREG = \overline{f} + 1	1	1	C,DC,N,OV,Z
		NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C,DC,N,OV,Z
53	NOP	NOP		No Operation	1	1	None
		NOPR		No Operation	1	1	None
54	POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
54		POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
		POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd + 1)	1	2	None
		POP.S		Pop Shadow Registers	1	1	All
55	PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
		PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
		PUSH.D	Wns	Push W(ns):W(ns + 1) to Top-of-Stack (TOS)	1	2	None
		PUSH.S		Push Shadow Registers	1	1	None
56	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO,Sleep
57	RCALL	RCALL	Expr	Relative Call	1	2	None
		RCALL	Wn	Computed Call	1	2	None
58	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
		REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
59	RESET	RESET		Software device Reset	1	1	None
60	RETFIE	RETFIE	#3 1 1 0 m	Return from Interrupt	1	3 (2)	None
62	RETLW	RETLW	#11110,Wn	Return from Subrouting	1	3 (Z)	None
63	RETURN	RETURN	£	f = Potato Loft through Carry f	1	3 (2)	
03	RLC	RLC	f NDEC	WPEG = Potete Loft through Carry f	1	1	
		RLC	Ne Wd	Wd = Rotate Left through Carry Ws	1	1	C N Z
64	RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N 7
		RLNC	- f,WREG	WREG = Rotate Left (No Carry) f	1	1	N.Z
		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N.Z
65	RRC	RRC	f	f = Rotate Right through Carry f	1	1	C,N,Z
		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C,N,Z
		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C,N,Z

TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)

TABLE 22-17:PLL CLOCK TIMING SPECIFICATIONS (VDD = 3.0V TO 3.6V)

AC CHARACTERISTICS			Standard C Operating f	Operating temperatu	y Conditic ure -40° -40°	O NS: 3.0V C ≤TA ≤+8 C ≤TA ≤+1	to 3.6V 5°C for I 25°C for	(unless otherwise stated) ndustrial Extended
Param No.	Symbol	Characteris	Min	Тур ⁽¹⁾	Мах	Units	Conditions	
OS50	Fplli	PLL Voltage Controlled Oscillator (VCO) Input Frequency Range ⁽²⁾		0.8	_	8	MHz	ECPLL, XTPLL modes
OS51	Fsys	On-Chip VCO System Frequency ⁽³⁾		100	—	200	MHz	_
OS52	TLOCK	PLL Start-up Time (Lock Time) ⁽³⁾		0.9	1.5	3.1	ms	—
OS53	DCLK	CLKO Stability (Jitter) ⁽³⁾		-3	0.5	3	%	Measured over 100 ms period

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

2: These parameters are characterized by similarity, but are tested in manufacturing at 7.7 MHz input only.

3: These parameters are characterized by similarity, but are not tested in manufacturing. This specification is based on clock cycle by clock cycle measurements. To calculate the effective jitter for individual time bases or communication clocks use this formula:

$$Peripheral Clock Jitter = \frac{DCLK}{\sqrt{\frac{FOSC}{Peripheral Bit Rate Clock}}}$$

For example: Fosc = 32 MHz, DCLK = 3%, SPI bit rate clock, (i.e., SCK) is 2 MHz.

$$SPI SCK Jitter = \left[\frac{D_{CLK}}{\sqrt{\left(\frac{32 MHz}{2 MHz}\right)}}\right] = \left[\frac{3\%}{\sqrt{16}}\right] = \left[\frac{3\%}{4}\right] = 0.75\%$$

TABLE 22-18: AC CHARACTERISTICS: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤TA ≤+85°C for industrial -40°C ≤TA ≤+125°C for Extended							
Param No.	Characteristic	Min	Тур	Мах	Units	Conditions			
	Internal FRC Accuracy @ FRC Frequency = 7.37 MHz ⁽¹⁾								
F20a	FRC	-2	—	+2	%	-40°C ≤TA ≤+85°C	VDD = 3.0-3.6V		
F20b	FRC	-5	_	+5	%	-40°C ≤TA ≤+125°C	VDD = 3.0-3.6V		

Note 1: Frequency calibrated at 25°C and 3.3V. TUN bits can be used to compensate for temperature drift.

TABLE 22-19: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤TA ≤+85°C for Industrial -40°C ≤TA ≤+125°C for Extended						
Param No.	Characteristic	Min	Тур	Мах	Units	S Conditions		
	LPRC @ 32.768 kHz ^(1,2)							
F21a	LPRC	-15	±6	+15	%	-40°C ≤TA ≤+85°C	VDD = 3.0-3.6V	
F21b	LPRC	-40	_	+40	%	-40°C ≤TA ≤+125°C	VDD = 3.0-3.6V	

Note 1: Change of LPRC frequency as VDD changes.

2: LPRC impacts the Watchdog Timer Time-out Period (TWDT1). See Section 19.4 "Watchdog Timer (WDT)" for more information.

TABLE 22-34:SPIX SLAVE MODE (FULL-DUPLEX, CKE = 0, CKP = 1, SMP = 0) TIMING
REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V(unless otherwise stated)Operating temperature $-40^{\circ}C \leq TA \leq +85^{\circ}C$ for Industrial $-40^{\circ}C \leq TA \leq +125^{\circ}C$ for Extended					
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур ⁽²⁾	Max	Units	Conditions	
SP70	TscP	Maximum SCK Input Frequency			15	MHz	See Note 3	
SP72	TscF	SCKx Input Fall Time	—			ns	See parameter DO32 and Note 4	
SP73	TscR	SCKx Input Rise Time	—			ns	See parameter DO31 and Note 4	
SP30	TdoF	SDOx Data Output Fall Time	—			ns	See parameter DO32 and Note 4	
SP31	TdoR	SDOx Data Output Rise Time	—			ns	See parameter DO31 and Note 4	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	6	20	ns	—	
SP36	TdoV2scH, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	_		ns	_	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	30			ns	—	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	30			ns	—	
SP50	TssL2scH, TssL2scL	$\overline{\text{SSx}} \downarrow \text{to SCKx} \uparrow \text{ or SCKx Input}$	120	—	—	ns	—	
SP51	TssH2doZ	SSx	10	_	50	ns	_	
SP52	TscH2ssH TscL2ssH	SSx after SCKx Edge	1.5 TCY + 40	_		ns	See Note 4	

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

2: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

3: The minimum clock period for SCKx is 66.7 ns. Therefore, the SCK clock generated by the Master must not violate this specification.

4: Assumes 50 pF load on all SPIx pins.

25.0 PACKAGING INFORMATION

25.1 Package Marking Information



dsPIC33FJ32GP 202-E/SP@3 0730235	\bigcirc

28-Lead SOIC



Example



28-Lead SSOP



Example



Legend	: XXX Y YY WW NNN @3	Customer-specific information Year code (last digit of calendar year) Year code (last 2 digits of calendar year) Week code (week of January 1 is week '01') Alphanumeric traceability code 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 N line, thus I	licrochip part number cannot be marked on one line, it is carried over to the next imiting the number of available characters for customer-specific information.

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

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



Microchip Technology Drawing C04-052C Sheet 1 of 2

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

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



RECOMMENDED LAND PATTERN

		-		
	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	С		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2052A

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

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



	Units	MILLIMETERS				
	Dimension Limits	MIN	NOM	MAX		
Number of Leads	N	44				
Lead Pitch	e	0.80 BSC				
Overall Height	A		_	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	E	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	C.	11°	12°	13°		
Mold Draft Angle Bottom	β	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