



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	I <sup>2</sup> C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, QEI, POR, PWM, WDT
Number of I/O	85
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj32gs610-e-pt">https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj32gs610-e-pt</a>

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com). We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

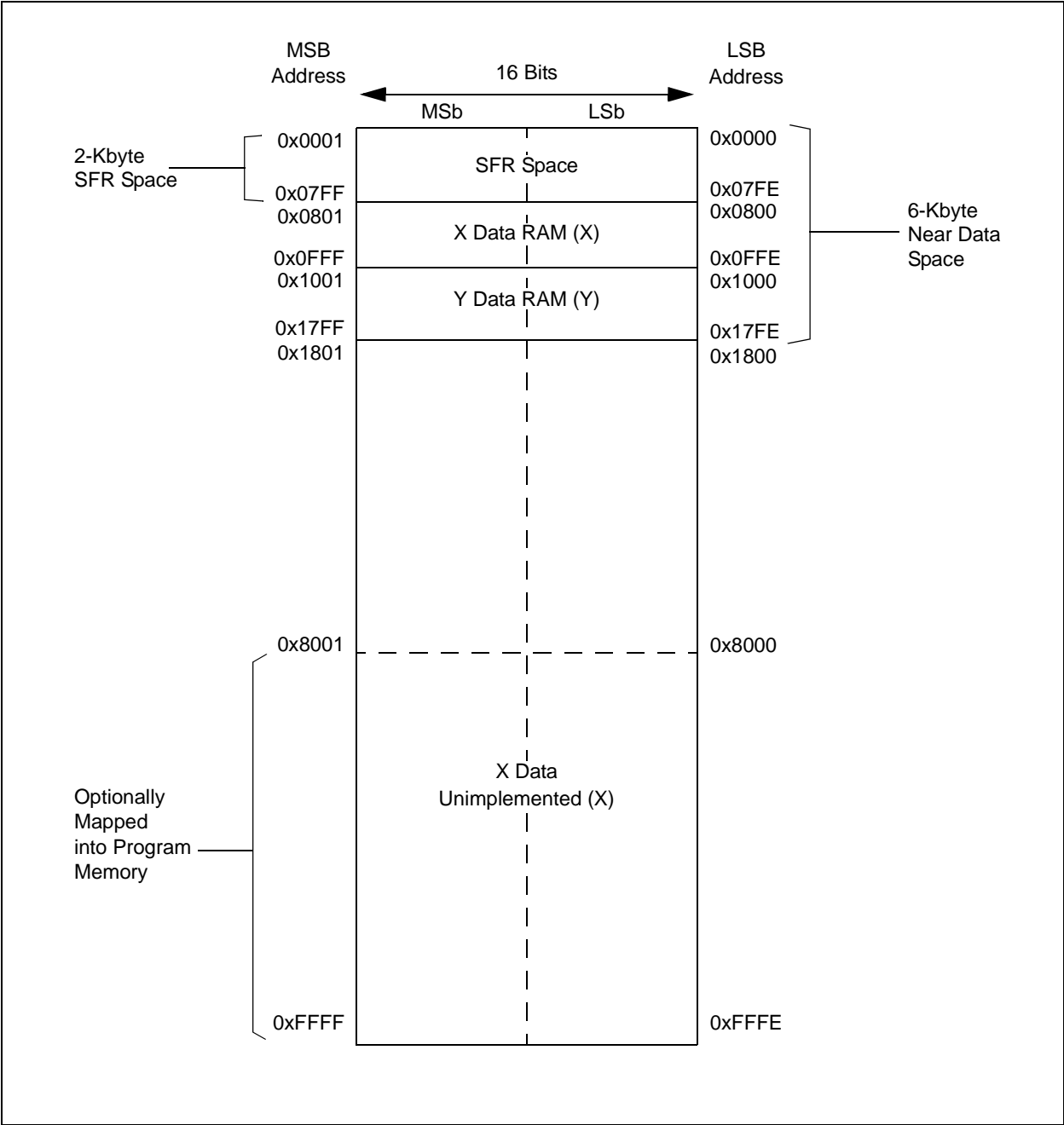
TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN23	I	Analog	Analog input channels.
CLKI CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
OSC1 OSC2	I I/O	ST/CMOS —	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.
SOSCI SOSCO	I O	ST/CMOS —	32.768 kHz low-power oscillator crystal input; CMOS otherwise. 32.768 kHz low-power oscillator crystal output.
CN0-CN23	I	ST	Change Notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
C1RX C1TX	I O	ST —	ECAN1 bus receive pin. ECAN1 bus transmit pin.
IC1-IC4	I	ST	Capture Inputs 1 through 4.
INDX1, INDX2, AINDX1 QEA1, QEA2, AQEA1 QEB1, QEB2, AQEB1 UPDN1	I I I O	ST ST ST CMOS	Quadrature Encoder Index Pulse input. Quadrature Encoder Phase A input in QE1 mode. Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QE1 mode. Auxiliary Timer External Clock/Gate input in Timer mode. Position Up/Down Counter Direction State.
OCFA OC1-OC4	I O	ST —	Compare Fault A input. Compare Outputs 1 through 4.
INT0 INT1 INT2 INT3 INT4	I I I I I	ST ST ST ST ST	External Interrupt 0. External Interrupt 1. External Interrupt 2. External Interrupt 3. External Interrupt 4.
RA0-RA15	I/O	ST	PORTA is a bidirectional I/O port.
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC0-RC15	I/O	ST	PORTC is a bidirectional I/O port.
RD0-RD15	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE9	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF13	I/O	ST	PORTF is a bidirectional I/O port.
RG0-RG15	I/O	ST	PORTG is a bidirectional I/O port.
T1CK T2CK T3CK T4CK T5CK	I I I I I	ST ST ST ST ST	Timer1 external clock input. Timer2 external clock input. Timer3 external clock input. Timer4 external clock input. Timer5 external clock input.

**Legend:** CMOS = CMOS compatible input or output      Analog = Analog input      I = Input  
ST = Schmitt Trigger input with CMOS levels      P = Power      O = Output  
TTL = Transistor-Transistor Logic

NOTES:

FIGURE 4-3: DATA MEMORY MAP FOR DEVICES WITH 4-KBYTE RAM



#### 4.6.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word page of the program space. This option provides transparent access to stored constant data from the data space without the need to use special instructions (such as TBLRDL/H).

Program space access through the data space occurs if the Most Significant bit of the data space EA is '1' and program space visibility is enabled by setting the PSV bit in the Core Control register (CORCON<2>). The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page register (PSVPAG). This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with the 15 bits of the EA functioning as the lower bits. By incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads to this area add a cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address 8000h and higher maps directly into a corresponding program memory address (see Figure 4-11), only the lower 16 bits of the

24-bit program word are used to contain the data. The upper 8 bits of any program space location used as data should be programmed with '1111 1111' or '0000 0000' to force a NOP. This prevents possible issues should the area of code ever be accidentally executed.

**Note:** PSV access is temporarily disabled during Table Reads/Writes.

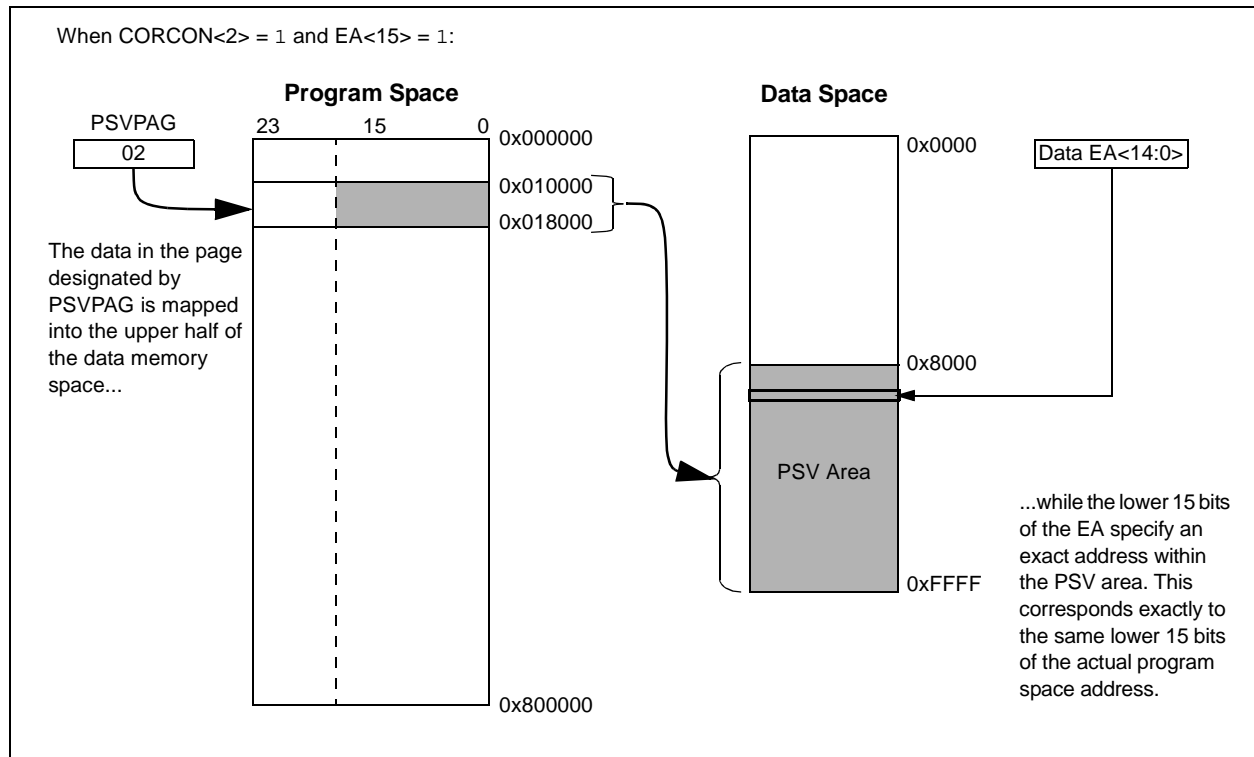
For operations that use PSV and are executed outside a REPEAT loop, the MOV and MOV.D instructions require one instruction cycle in addition to the specified execution time. All other instructions require two instruction cycles in addition to the specified execution time.

For operations that use PSV and are executed inside a REPEAT loop, these instances require two instruction cycles in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction using PSV to access data, to execute in a single cycle.

**FIGURE 4-11: PROGRAM SPACE VISIBILITY OPERATION**



#### 5.4.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

One row of program Flash memory can be programmed at a time. To achieve 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 NVMOPx 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 NVMOPx 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 NOPs, 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
```

FIGURE 6-2: SYSTEM RESET TIMING

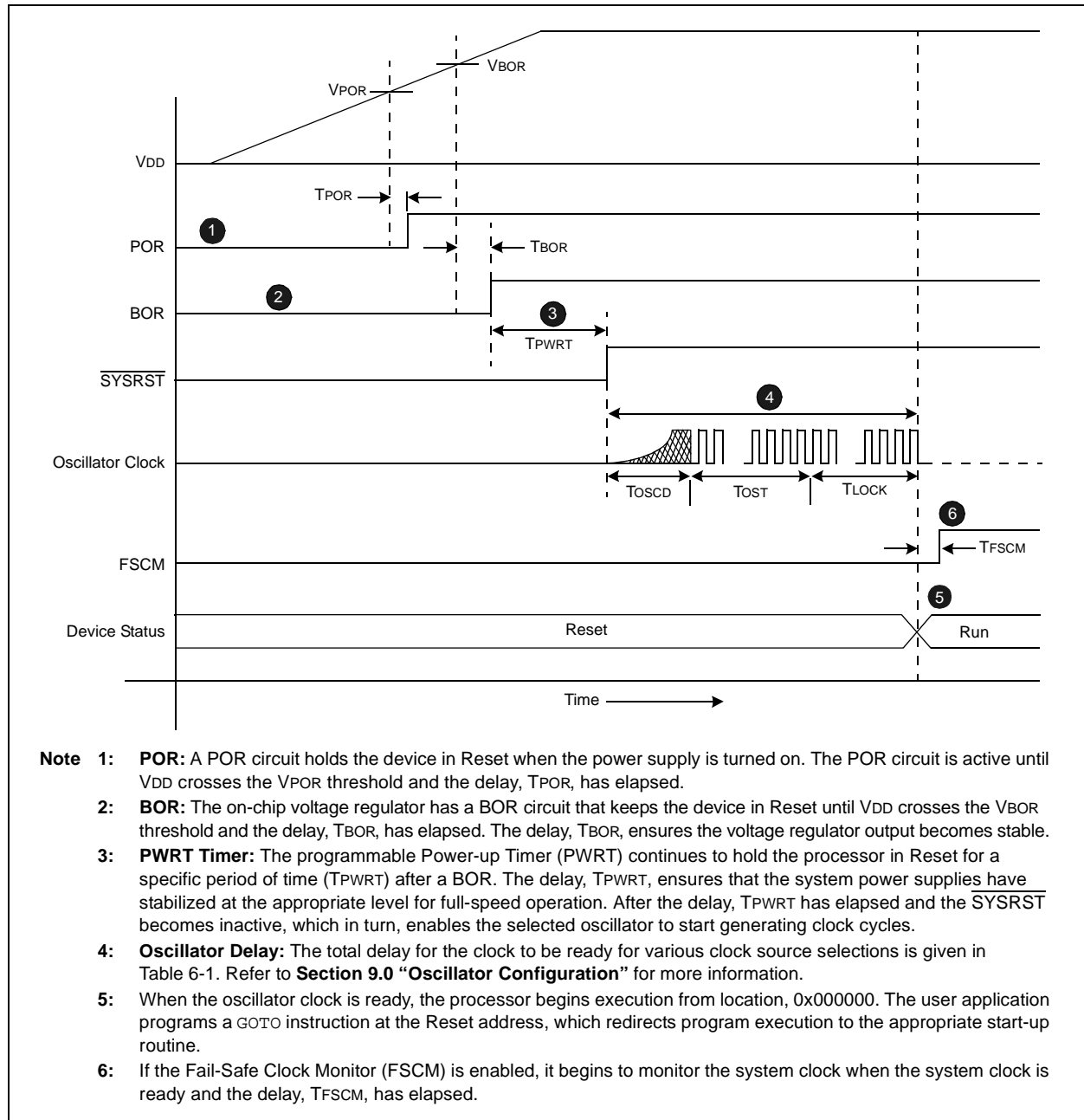


TABLE 6-2: OSCILLATOR DELAY

Symbol	Parameter	Value
$V_{POR}$	POR Threshold	1.8V nominal
$T_{POR}$	POR Extension Time	30 $\mu$ s maximum
$V_{BOR}$	BOR Threshold	2.5V nominal
$T_{BOR}$	BOR Extension Time	100 $\mu$ s maximum
$T_{PWRT}$	Programmable Power-up Time Delay	0-128 ms nominal
$T_{FSCM}$	Fail-Safe Clock Monitor Delay	900 $\mu$ s maximum

**Note:** When the device exits the Reset condition (begins normal operation), the device operating parameters (voltage, frequency, temperature, etc.) must be within their operating ranges; otherwise, the device may not function correctly. The user application must ensure that the delay between the time power is first applied, and the time **SYSRST** becomes inactive, is long enough to get all operating parameters within specification.



# dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610

## REGISTER 7-1: SR: CPU STATUS REGISTER<sup>(1)</sup>

R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R-0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8

R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2)</sup>	IPL1 <sup>(2)</sup>	IPL0 <sup>(2)</sup>	RA	N	OV	Z	C
bit 7							bit 0

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

bit 7-5 **IPL<2:0>**: CPU Interrupt Priority Level Status bits<sup>(2,3)</sup>

111 = CPU Interrupt Priority Level is 7 (15), user interrupts are disabled  
110 = CPU Interrupt Priority Level is 6 (14)  
101 = CPU Interrupt Priority Level is 5 (13)  
100 = CPU Interrupt Priority Level is 4 (12)  
011 = CPU Interrupt Priority Level is 3 (11)  
010 = CPU Interrupt Priority Level is 2 (10)  
001 = CPU Interrupt Priority Level is 1 (9)  
000 = CPU Interrupt Priority Level is 0 (8)

- Note 1:** For complete register details, see Register 3-1.
- 2:** The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.
- 3:** The IPL<2:0> Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

## REGISTER 7-2: CORCON: CORE CONTROL REGISTER<sup>(1)</sup>

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL2	DL1	DL0
bit 15							bit 8

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

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

bit 3 **IPL3**: CPU Interrupt Priority Level Status bit<sup>(2)</sup>

1 = CPU Interrupt Priority Level is greater than 7  
0 = CPU Interrupt Priority Level is 7 or less

- Note 1:** For complete register details, see Register 3-2.
- 2:** The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

## REGISTER 7-11: IFS6: INTERRUPT FLAG STATUS REGISTER 6

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
ADCP1IF	ADCP0IF	—	—	—	—	AC4IF	AC3IF
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AC2IF	PWM9IF	PWM8IF	PWM7IF	PWM6IF	PWM5IF	PWM4IF	PWM3IF
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                      **ADCP1IF:** ADC Pair 1 Conversion Done Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 14                      **ADCP0IF:** ADC Pair 0 Conversion Done Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 13-10                      **Unimplemented:** Read as '0'

bit 9                      **AC4IF:** Analog Comparator 4 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 8                      **AC3IF:** Analog Comparator 3 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 7                      **AC2IF:** Analog Comparator 2 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 6                      **PWM9IF:** PWM9 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 5                      **PWM8IF:** PWM8 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 4                      **PWM7IF:** PWM7 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 3                      **PWM6IF:** PWM6 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 2                      **PWM5IF:** PWM5 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 1                      **PWM4IF:** PWM4 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

bit 0                      **PWM3IF:** PWM3 Interrupt Flag Status bit

1 = Interrupt request has occurred  
 0 = Interrupt request has not occurred

## dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610

**REGISTER 8-2: DMAxREQ: DMA CHANNEL x IRQ SELECT REGISTER**

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

U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	IRQSEL6 <sup>(2)</sup>	IRQSEL5 <sup>(2)</sup>	IRQSEL4 <sup>(2)</sup>	IRQSEL3 <sup>(2)</sup>	IRQSEL2 <sup>(2)</sup>	IRQSEL1 <sup>(2)</sup>	IRQSEL0 <sup>(2)</sup>
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 **FORCE:** Force DMA Transfer bit<sup>(1)</sup>

1 = Forces a single DMA transfer (Manual mode)

0 = Automatic DMA transfer initiation by DMA request

bit 14-7 **Unimplemented:** Read as '0'

bit 6-0 **IRQSEL<6:0>:** DMA Peripheral IRQ Number Select bits<sup>(2)</sup>

0000000-1111111 = DMAIRQ0-DMAIRQ127 are selected to be Channel DMAREQ

**Note 1:** The FORCE bit cannot be cleared by the user. The FORCE bit is cleared by hardware when the forced DMA transfer is complete.

**2:** See Table 8-1 for a complete listing of IRQ numbers for all interrupt sources.

**REGISTER 8-3: DMAxSTA: DMA CHANNEL x RAM START ADDRESS OFFSET REGISTER A**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STA<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STA<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 15-0 **STA<15:0>:** Primary DMA RAM Start Address bits (source or destination)

### 9.1.3 PLL CONFIGURATION

The primary oscillator and internal FRC oscillator can optionally use an on-chip PLL to obtain higher speeds of operation. The PLL provides significant flexibility in selecting the device operating speed. A block diagram of the PLL is shown in Figure 9-2.

The output of the primary oscillator or FRC, denoted as 'FIN', is divided down by a prescale factor (N1) of 2, 3, ... or 33 before being provided to the PLL's Voltage Controlled Oscillator (VCO). The input to the VCO must be selected in the range of 0.8 MHz to 8 MHz. The prescale factor 'N1' is selected using the PLLPRE<4:0> bits (CLKDIV<4:0>).

The PLL Feedback Divisor, selected using the PLLDIV<8:0> bits (PLLFBF<8:0>), provides a factor, 'M', by which the input to the VCO is multiplied. This factor must be selected such that the resulting VCO output frequency is in the range of 100 MHz to 200 MHz.

The VCO output is further divided by a postscale factor, 'N2'. This factor is selected using the PLLPOST<1:0> bits (CLKDIV<7:6>). 'N2' can be either 2, 4 or 8, and must be selected such that the PLL output frequency (Fosc) is in the range of 12.5 MHz to 100 MHz, which generates device operating speeds of 6.25-50 MIPS.

For a primary oscillator or FRC oscillator, output 'FIN', the PLL output 'Fosc' is given by Equation 9-2.

#### EQUATION 9-2: Fosc CALCULATION

$$F_{OSC} = F_{IN} * \left( \frac{M}{N1 * N2} \right)$$

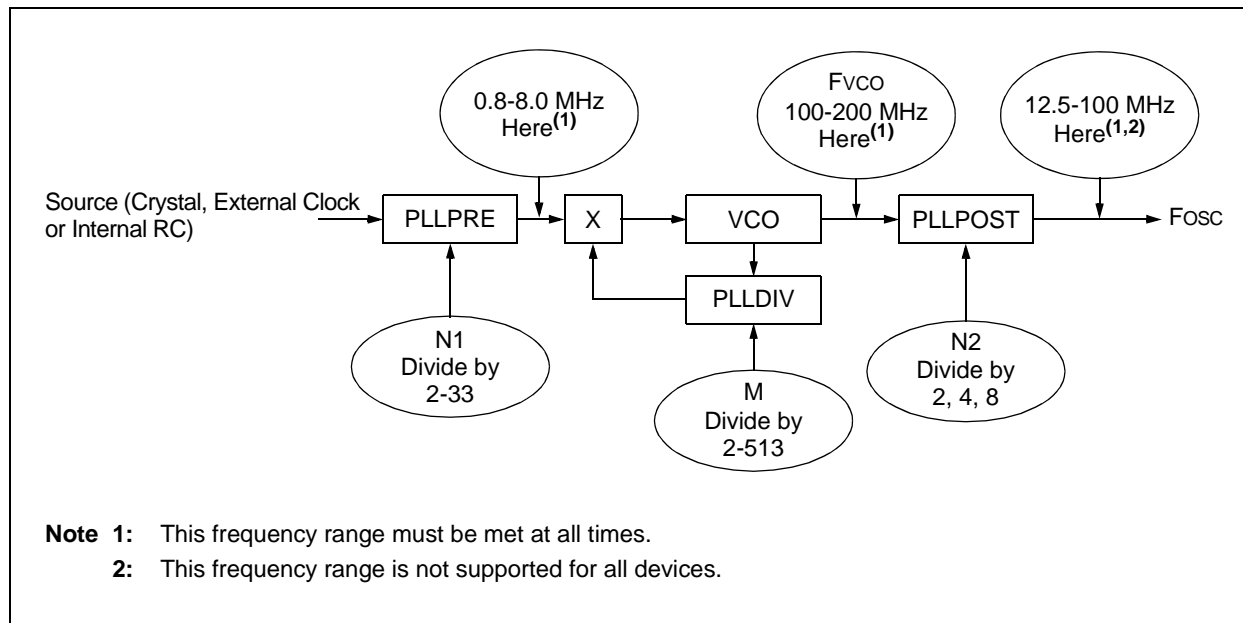
For example, suppose a 10 MHz crystal is being used with the selected oscillator mode of XT with PLL (see Equation 9-3).

- If PLLPRE<4:0> = 0000, then N1 = 2. This yields a VCO input of 10/2 = 5 MHz, which is within the acceptable range of 0.8-8 MHz.
- If PLLDIV<8:0> = 0x26, then M = 40. This yields a VCO output of 5 x 40 = 200 MHz, which is within the 100-200 MHz ranged needed.
- If PLLPOST<1:0> = 00, then N2 = 2. This provides a Fosc of 200/2 = 100 MHz. The resultant device operating speed is 100/2 = 40 MIPS.

#### EQUATION 9-3: XT WITH PLL MODE EXAMPLE

$$F_{CY} = \frac{F_{OSC}}{2} = \frac{1}{2} \left( \frac{10000000 * 40}{2 * 2} \right) = 50 \text{ MIPS}$$

FIGURE 9-2: PLL BLOCK DIAGRAM

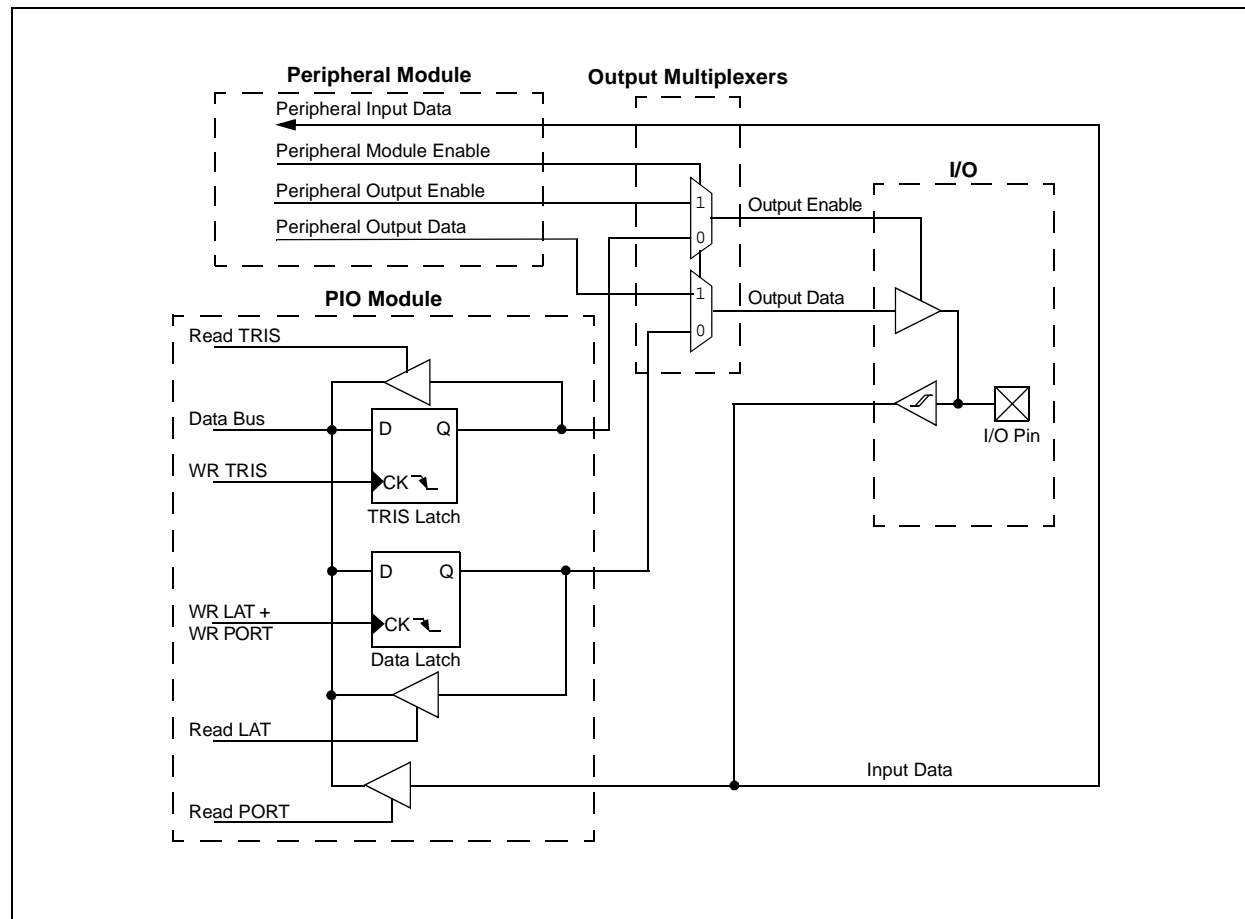


## REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER<sup>(1)</sup> (CONTINUED)

bit 3	<b>CF:</b> Clock Fail Detect bit (read/clear by application) 1 = FSCM has detected clock failure 0 = FSCM has not detected clock failure
bit 2-1	<b>Unimplemented:</b> Read as '0'
bit 0	<b>OSWEN:</b> Oscillator Switch Enable bit 1 = Requests oscillator switch to the selection specified by the NOSC<2:0> bits 0 = Oscillator switch is complete

- Note 1:** Writes to this register require an unlock sequence. Refer to “**Oscillator (Part IV)**” (DS70307) in the “*dsPIC33/PIC24 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.

FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



## dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610

### REGISTER 16-16: DTRx: PWM DEAD-TIME x REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTRx<13:8>					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTRx<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 15-14      **Unimplemented:** Read as '0'  
bit 13-0      **DTRx<13:0>:** Unsigned 14-Bit Value for PWMx Dead-Time Unit bits

### REGISTER 16-17: ALTDTRx: PWM ALTERNATE DEAD-TIME x REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ALTDTRx<13:8>					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALTDTRx<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 15-14      **Unimplemented:** Read as '0'  
bit 13-0      **ALTDTRx<13:0>:** Unsigned 14-Bit Value for PWMx Dead-Time Unit bits

NOTES:



**REGISTER 21-4: CxFCTRL: ECANx FIFO CONTROL REGISTER**

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
DMABS2	DMABS1	DMABS0	—	—	—	—	—
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
—	—	—	FSA4 <sup>(1)</sup>	FSA3 <sup>(1)</sup>	FSA2 <sup>(1)</sup>	FSA1 <sup>(1)</sup>	FSA0 <sup>(1)</sup>
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 **DMABS<2:0>**: DMA Buffer Size bits

111 = Reserved

110 = 32 buffers in DMA RAM

101 = 24 buffers in DMA RAM

100 = 16 buffers in DMA RAM

011 = 12 buffers in DMA RAM

010 = 8 buffers in DMA RAM

001 = 6 buffers in DMA RAM

000 = 4 buffers in DMA RAM

bit 12-5 **Unimplemented**: Read as '0'

bit 4-0 **FSA<4:0>**: FIFO Area Starts with Buffer bits<sup>(1)</sup>

11111 = Reads Buffer RB31

11110 = Reads Buffer RB30

•

•

•

00001 = TX/RX Buffer TRB1

00000 = TX/RX Buffer TRB0

**Note 1:** FSA<4:0> bits are used to specify the start of the FIFO within the buffer area.

**REGISTER 22-4: ADPCFG: ADC PORT CONFIGURATION REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG<15:8> <sup>(1)</sup>							
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG<7:0> <sup>(1)</sup>							
bit 7				bit 0			

**Legend:**

R = Readable bit      W = Writable bit      U = Unimplemented bit, read as '0'  
 -n = Value at POR      '1' = Bit is set      '0' = Bit is cleared      x = Bit is unknown

bit 15-0      **PCFG<15:0>:** ADC Port Configuration Control bits<sup>(1)</sup>

1 = Port pin in Digital mode, port read input is enabled; Analog-to-Digital input multiplexer is connected to AVss

0 = Port pin in Analog mode, port read input is disabled; Analog-to-Digital samples the pin voltage

**Note 1:** Not all PCFGx bits are available on all devices. See Figure 22-1, Figure 22-2, Figure 22-3 and Figure 22-4 for the available analog inputs (PCFGx = ANx, where x = 0-15).

**REGISTER 22-5: ADPCFG2: ADC PORT CONFIGURATION REGISTER 2**

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG<23:16> <sup>(1)</sup>							
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 15-8      **Unimplemented:** Read as '0'

bit 7-0      **PCFG<23:16>:** ADC Port Configuration Control bits<sup>(1)</sup>

1 = Port pin in Digital mode, port read input is enabled; Analog-to-Digital input multiplexer is connected to AVss

0 = Port pin in Analog mode, port read input is disabled; Analog-to-Digital samples the pin voltage

**Note 1:** Not all PCFGx bits are available on all devices. See Figure 22-1, Figure 22-2, Figure 22-3 and Figure 22-4 for the available analog inputs (PCFGx = ANx, where x can be 0 through 15).

## 25.0 INSTRUCTION SET SUMMARY

**Note:** This data sheet summarizes the features of the dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610 devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the “dsPIC33/PIC24 Family Reference Manual”. Please see the Microchip web site ([www.microchip.com](http://www.microchip.com)) for the latest “dsPIC33F/PIC24H Family Reference Manual” sections. The information in this data sheet supersedes the information in the FRM.

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 25-1 shows the general symbols used in describing the instructions.

The dsPIC33F instruction set summary in Table 25-2 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

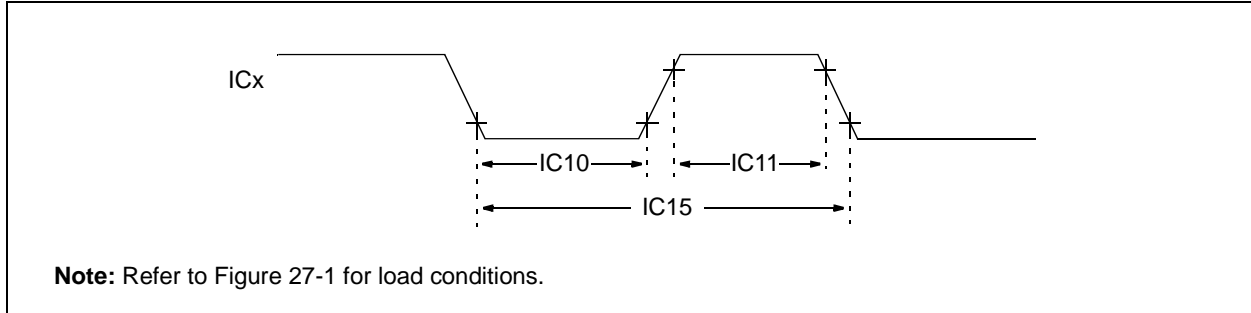
The other DSP instructions do not involve any multiplication and can include:

- 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

**FIGURE 27-6: INPUT CAPTURE x (ICx) TIMING CHARACTERISTICS**

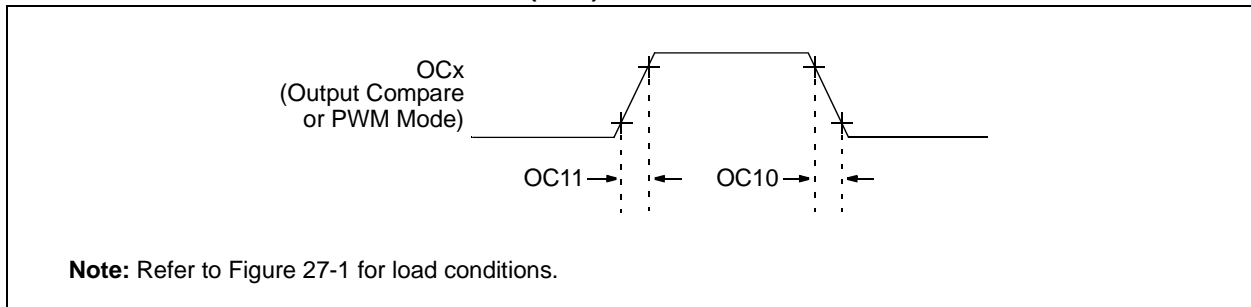


**TABLE 27-26: INPUT CAPTURE x TIMING REQUIREMENTS**

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.	Symbol	Characteristic <sup>(1)</sup>		Min	Max	Units	Conditions
IC10	TccL	ICx Input Low Time	No Prescaler	0.5 Tcy + 20	—	ns	
			With Prescaler	10	—	ns	
IC11	TccH	ICx Input High Time	No Prescaler	0.5 Tcy + 20	—	ns	
			With Prescaler	10	—	ns	
IC15	TccP	ICx Input Period		(Tcy + 40)/N	—	ns	N = Prescale value (1, 4, 16)

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

**FIGURE 27-7: OUTPUT COMPARE x (OCx) MODULE TIMING CHARACTERISTICS**



**TABLE 27-27: OUTPUT COMPARE x MODULE TIMING REQUIREMENTS**

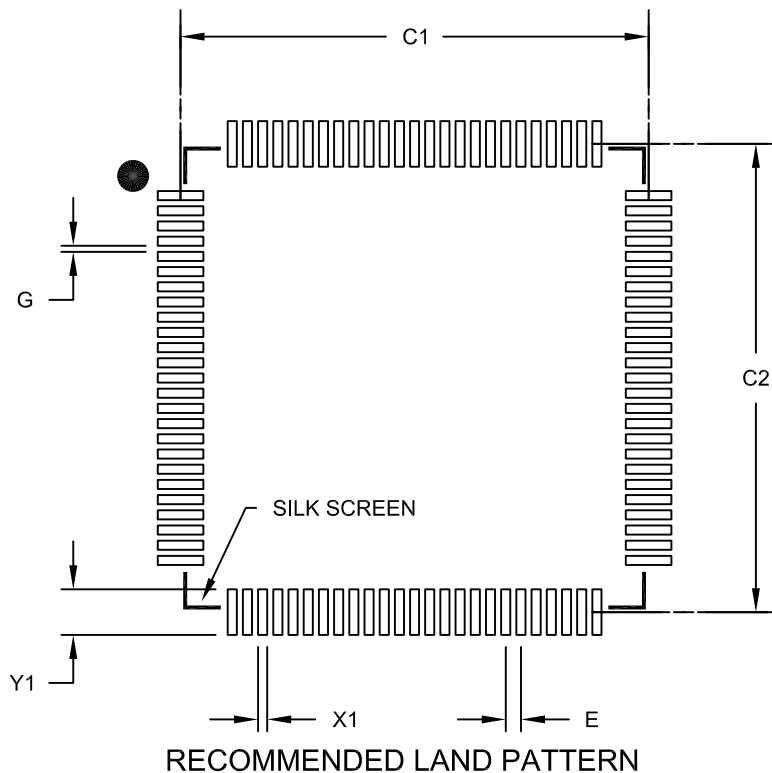
AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended				
Param No.	Symbol	Characteristic <sup>(1)</sup>	Min	Typ	Max	Units	Conditions
OC10	TccF	OCx Output Fall Time	—	—	—	ns	See Parameter DO32
OC11	TccR	OCx Output Rise Time	—	—	—	ns	See Parameter DO31

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

# dsPIC33FJ32GS406/606/608/610 and dsPIC33FJ64GS406/606/608/610

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 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>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (Y100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2110B