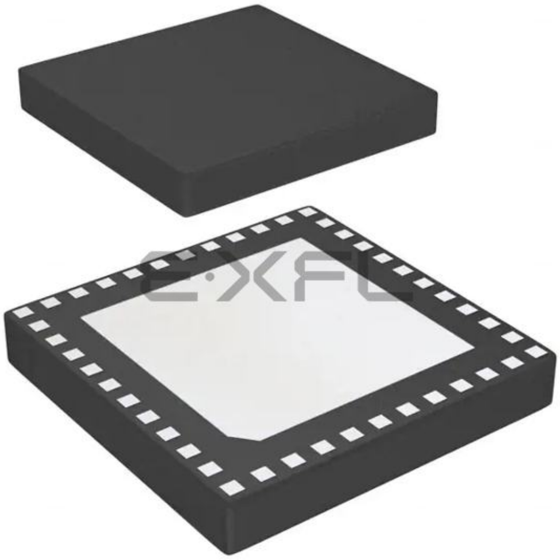
Microchip Technology - DSPIC33FJ16GS504-I/TL Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

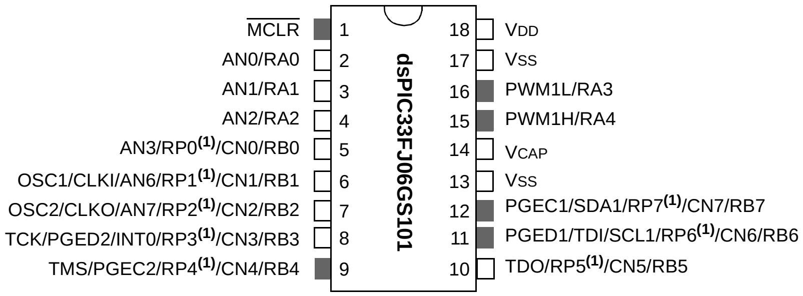
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	35
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10b; D/A 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VFTLA Exposed Pad
Supplier Device Package	44-VTLA (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj16gs504-i-tl

Pin Diagrams

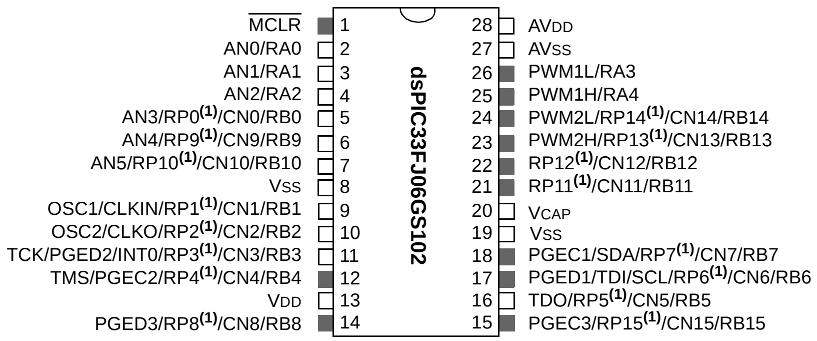
18-Pin SOIC

■ = Pins are up to 5V tolerant


dsPIC33FJ06GS101
MCLR 1
AN0/RA0 2
AN1/RA1 3
AN2/RA2 4
AN3/RP0(1)/CN0/RB0 5
OSC1/CLKI/AN6/RP1(1)/CN1/RB1 6
OSC2/CLKO/AN7/RP2(1)/CN2/RB2 7
TCK/PGED2/INT0/RP3(1)/CN3/RB3 8
TMS/PGEC2/RP4(1)/CN4/RB4 9
18 VDD
17 VSS
16 PWM1L/RA3
15 PWM1H/RA4
14 VCAP
13 VSS
12 PGEC1/SDA1/RP7(1)/CN7/RB7
11 PGED1/TDI/SCL1/RP6(1)/CN6/RB6
10 TDO/RP5(1)/CN5/RB5

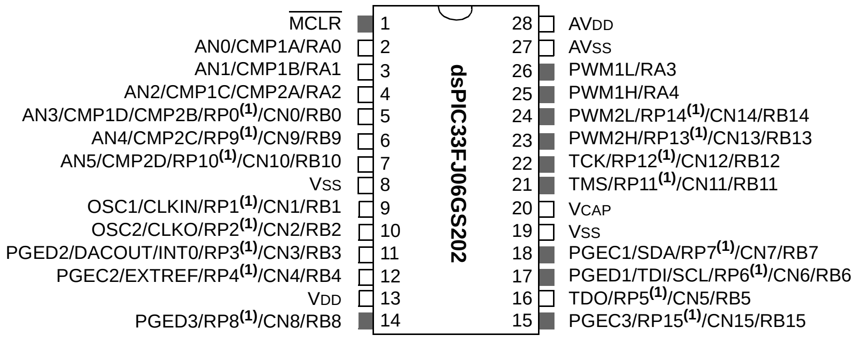
28-Pin SOIC, SPDIP

■ = Pins are up to 5V tolerant



28-Pin SPDIP, SOIC

■ = Pins are up to 5V tolerant


dsPIC33FJ06GS202
MCLR 1
AN0/CMP1A/RA0 2
AN1/CMP1B/RA1 3
AN2/CMP1C/CMP2A/RA2 4
AN3/CMP1D/CMP2B/RP0(1)/CN0/RB0 5
AN4/CMP2C/RP9(1)/CN9/RB9 6
AN5/CMP2D/RP10(1)/CN10/RB10 7
VSS 8
OSC1/CLKIN/RP1(1)/CN1/RB1 9
OSC2/CLKO/RP2(1)/CN2/RB2 10
PGED2/DACOUT/INT0/RP3(1)/CN3/RB3 11
PGEC2/EXTREF/RP4(1)/CN4/RB4 12
VDD 13
PGED3/RP8(1)/CN8/RB8 14
28 AVDD
27 AVSS
26 PWM1L/RA3
25 PWM1H/RA4
24 PWM2L/RP14(1)/CN14/RB14
23 PWM2H/RP13(1)/CN13/RB13
22 TCK/RP12(1)/CN12/RB12
21 TMS/RP11(1)/CN11/RB11
20 VCAP
19 VSS
18 PGEC1/SDA/RP7(1)/CN7/RB7
17 PGED1/TDI/SCL/RP6(1)/CN6/RB6
16 TDO/RP5(1)/CN5/RB5
15 PGEC3/RP15(1)/CN15/RB15

Note 1: The RPn pins can be used by any remappable peripheral. See **Table 1** for the list of available peripherals.

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

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 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.

4.5.1 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 (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.

4.6 Interfacing Program and Data Memory Spaces

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 architecture uses a 24-bit-wide program space and a 16-bit-wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (Program Space Visibility)

Table instructions allow an application to read or write to small areas of the program memory. This capability makes the method ideal for accessing data tables that need to be updated periodically. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look ups from a large table of static data. The application can only access the least significant word of the program word.

4.6.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Page register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the Most Significant bit of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 8-bit Program Space Visibility register (PSVPAG) is used to define a 16K word page in the program space. When the Most Significant bit of the EA is '1', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike table operations, this limits remapping operations strictly to the user memory area.

Table 4-50 and Figure 4-9 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, and D<15:0> refers to a data space word.

TABLE 4-50: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access (Code Execution)	User	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT (Byte/Word Read/Write)	User	TBLPAG<7:0>		Data EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	Configuration	TBLPAG<7:0>		Data EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
Program Space Visibility (Block Remap/Read)	User	0	PSVPAG<7:0>		Data EA<14:0>[(1)]	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.

5.2 RTSP Operation

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user application to erase a page of memory, which consists of eight rows (512 instructions) at a time, and to program one row or one word at a time. Table 24-12 shows typical erase and programming times. The 8-row erase pages and single row write rows are edge-aligned from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

The program memory implements holding buffers that can contain 64 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the buffers sequentially. The instruction words loaded must always be from a group of 64 boundary.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register. A total of 64 `TBLWTL` and `TBLWTH` instructions are required to load the instructions.

All of the Table Write operations are single-word writes (two instruction cycles) because only the buffers are written. A programming cycle is required for programming each row.

5.3 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. The processor stalls (waits) until the programming operation is finished.

The programming time depends on the FRC accuracy (see Table 24-20) and the value of the FRC Oscillator Tuning register (see Register 8-4). Use the following formula to calculate the minimum and maximum values for the Row Write Time, Page Erase Time, and Word Write Cycle Time parameters (see Table 24-12).

EQUATION 5-1: PROGRAMMING TIME

$$\frac{T}{7.37\ MHz \times (FRC\ Accuracy)\% \times (FRC\ Tuning)\%}$$

For example, if the device is operating at +125°C, the FRC accuracy will be ±5%. If the TUN<5:0> bits (see Register 8-4) are set to `'b111111`, the minimum row write time is equal to Equation 5-2.

EQUATION 5-2: MINIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 + 0.05) \times (1 - 0.00375)} = 1.435ms$$

The maximum row write time is equal to Equation 5-3.

EQUATION 5-3: MAXIMUM ROW WRITE TIME

$$T_{RW} = \frac{11064\ Cycles}{7.37\ MHz \times (1 - 0.05) \times (1 - 0.00375)} = 1.586ms$$

Setting the WR bit (NVMCON<15>) starts the operation, and the WR bit is automatically cleared when the operation is finished.

5.4 Control Registers

Two SFRs are used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and the start of the programming cycle.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user application must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 5.3 "Programming Operations"** for further details.

EXAMPLE 5-2: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV     #0x4001, W0               ;
       MOV     W0, NVMCON                ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0               ;
       MOV     W0, TBLPAG                ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0               ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2           ;
       MOV     #HIGH_BYTE_0, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2           ;
       MOV     #HIGH_BYTE_1, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2           ;
       MOV     #HIGH_BYTE_2, W3          ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV     #LOW_WORD_31, W2          ;
       MOV     #HIGH_BYTE_31, W3         ;
       TBLWTL  W2, [W0]                  ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                ; Write PM high byte into program latch
```

EXAMPLE 5-3: INITIATING A PROGRAMMING SEQUENCE

```
       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
       NOP                               ; erase command is asserted
```

REGISTER 7-10: 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	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
AC2IF	—	—	—	—	—	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 6-2 **Unimplemented:** Read as ‘0’

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

REGISTER 7-12: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
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 **ADIE:** ADC1 Conversion Complete Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **U1TXIE:** UART1 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **U1RXIE:** UART1 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **SPI1IE:** SPI1 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **SPI1EIE:** SPI1 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **T3IE:** Timer3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **T2IE:** Timer2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **OC2IE:** Output Compare Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **IC2IE:** Input Capture Channel 2 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2 **OC1IE:** Output Compare Channel 1 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-23: IPC4: INTERRUPT PRIORITY CONTROL REGISTER 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	AC1IP2	AC1IP1	AC1IP0
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
—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
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 **Unimplemented:** Read as ‘0’

bit 14-12 **CNIP<2:0>:** Change Notification Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **AC1IP<2:0>:** Analog Comparator 1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **MI2C1IP<2:0>:** I2C1 Master Events Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SI2C1IP<2:0>:** I2C1 Slave Events Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

9.0 POWER-SAVING FEATURES

> **Note 1:** This data sheet summarizes the features of the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Watchdog Timer and Power-Saving Modes"** (DS70196) in the *"dsPIC33F/PIC24H Family 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 dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices provide the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices can manage power consumption in four different ways:

- Clock Frequency
- Instruction-Based Sleep and Idle modes
- Software-Controlled Doze mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption while still maintaining critical application features, such as timing-sensitive communications.

9.1 Clock Frequency and Clock Switching

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices allow a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC<2:0> bits (OSCCON<10:8>). The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 8.0 "Oscillator Configuration"**.

9.2 Instruction-Based Power-Saving Modes

The dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 devices have two special power-saving modes that are entered through the execution of a special `PWRSAV` instruction. Sleep mode stops clock operation and halts all code execution. Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembler syntax of the `PWRSAV` instruction is shown in Example 9-1.

> **Note:** `SLEEP_MODE` and `IDLE_MODE` are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to wake-up.

9.2.1 SLEEP MODE

The following occur in Sleep mode:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off
- The device current consumption is reduced to a minimum, provided that no I/O pin is sourcing current
- The Fail-Safe Clock Monitor does not operate, since the system clock source is disabled
- The LPRC clock continues to run in Sleep mode if the WDT is enabled
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode
- Some device features or peripherals may continue to operate. This includes the items such as the Input Change Notification on the I/O ports or peripherals that use an external clock input.
- Any peripheral that requires the system clock source for its operation is disabled

The device will wake-up from Sleep mode on any of these events:

- Any interrupt source that is individually enabled
- Any form of device Reset
- A WDT time-out

On wake-up from Sleep mode, the processor restarts with the same clock source that was active when Sleep mode was entered.

EXAMPLE 9-1: `PWRSAV` INSTRUCTION SYNTAX

```
PWRSAV  #SLEEP_MODE    ; Put the device into SLEEP mode
PWRSAV  #IDLE_MODE     ; Put the device into IDLE mode
```

10.0 I/O PORTS

Note 1: This data sheet summarizes the features of the dsPIC33FJ06GS101/X02 and dsPIC33FJ16GSX02/X04 families of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"I/O Ports"** (DS70193) in the *"dsPIC33F/PIC24H Family Reference Manual"*, which is available on 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.

All of the device pins (except VDD, VSS, $\overline{\text{MCLR}}$ and OSC1/CLKI) are shared among the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

10.1 Parallel I/O (PIO) Ports

Generally a parallel I/O port that shares a pin with a peripheral is subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 10-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin can be read, but the output driver for the parallel port bit is disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin can be driven by a port.

All port pins have three registers directly associated with their operation as digital I/O. The data direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the latch (LATx) read the latch. Writes to the latch write the latch. Reads from the port (PORTx) read the port pins, while writes to the port pins write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs.

FIGURE 10-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE

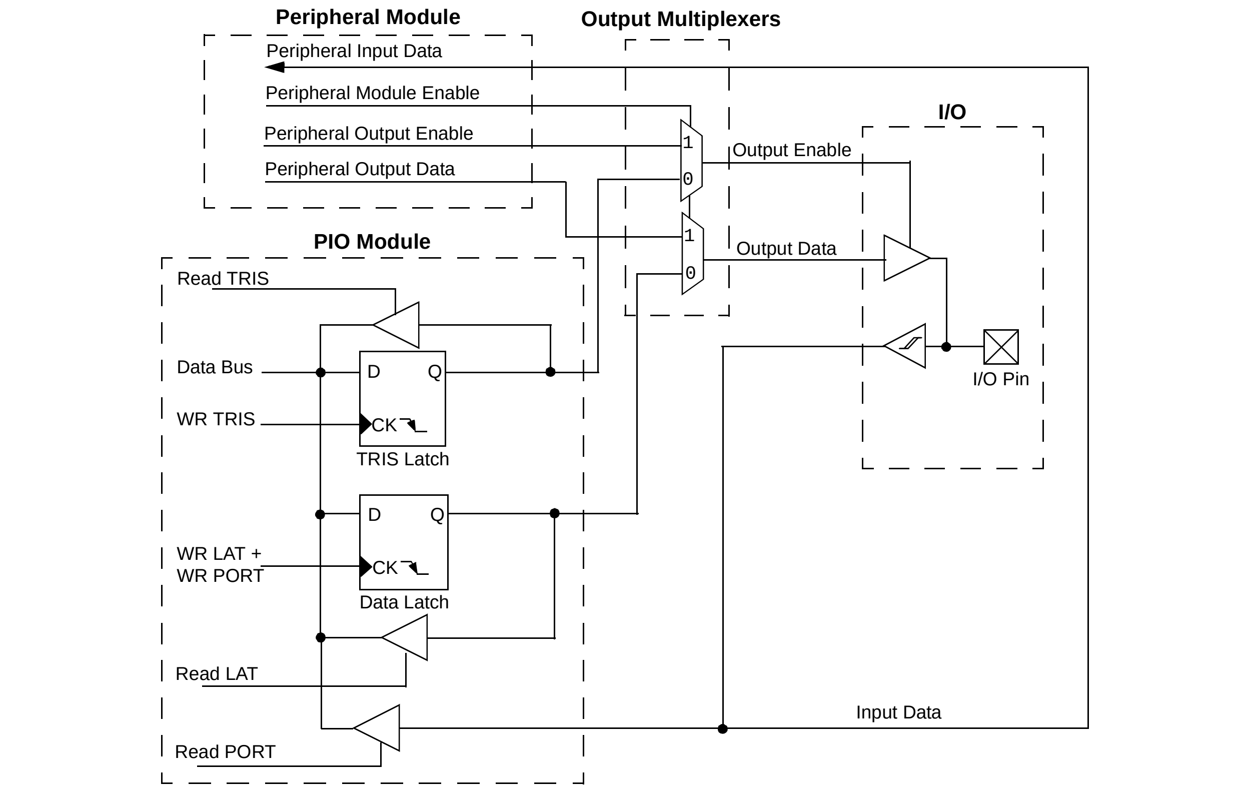
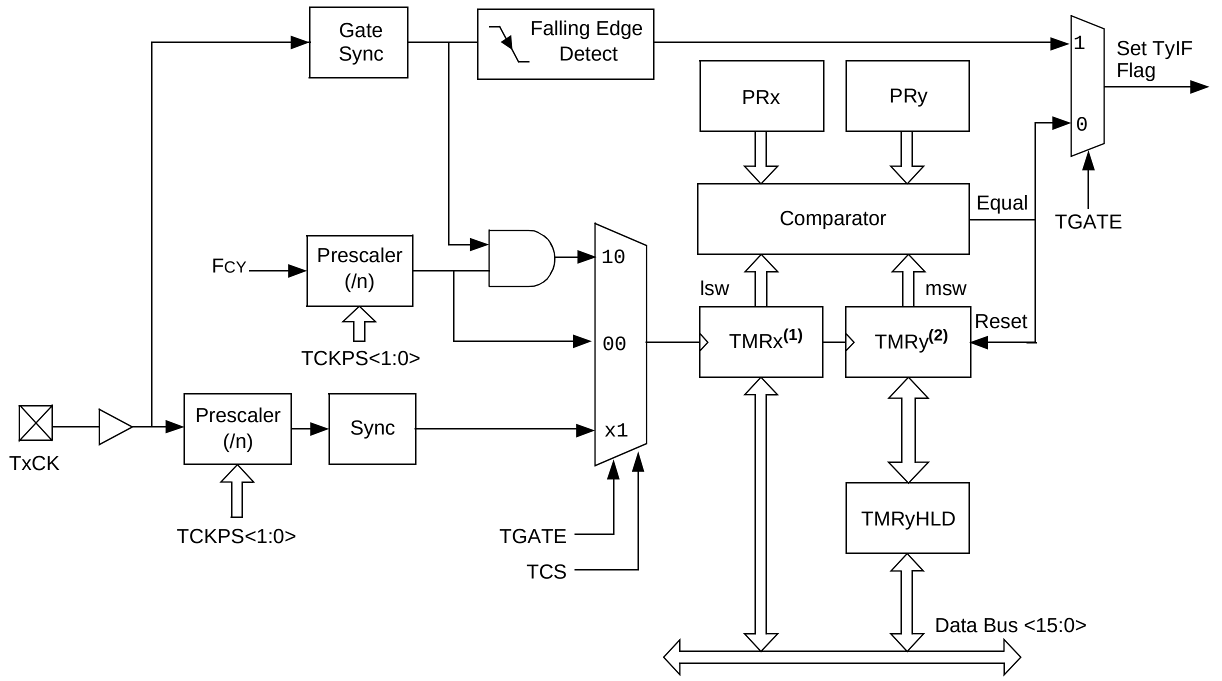


FIGURE 12-3: 32-BIT TIMER BLOCK DIAGRAM



Note 1: Timerx is a Type B Timer (x = 2).
2: Timery is a Type C Timer (y = 3).

REGISTER 15-4: SEVTCMP: PWM SPECIAL EVENT COMPARE 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
SEVTCMP <15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
SEVTCMP <7:3>					—	—	—
bit 7							bit 0

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

bit 15-3 **SEVTCMP<12:0>**: Special Event Compare Count Value bits

bit 2-0 **Unimplemented:** Read as ‘0’

REGISTER 15-5: MDC: PWM MASTER DUTY CYCLE REGISTER(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<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
MDC<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 **MDC<15:0>:** Master PWM Duty Cycle Value bits

Note 1: The smallest pulse width that can be generated on the PWM output corresponds to a value of 0x0009, while the maximum pulse width generated corresponds to a value of Period – 0x0008.

2: As the duty cycle gets closer to 0% or 100% of the PWM period (0 ns-40 ns, depending on the mode of operation), the PWM duty cycle resolution will degrade from 1 LSB to 3 LSBs.

REGISTER 16-3: SPIxCON2: SPIx CONTROL REGISTER 2

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 = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **FRMEN**: Framed SPIx Support bit
1 = Framed SPIx support is enabled ($\overline{\text{SSx}}$ pin used as frame sync pulse input/output)
0 = Framed SPIx support is disabled

bit 14 **SPIFSD**: SPIx Frame Sync Pulse Direction Control bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

bit 13 **FRMPOL**: Frame Sync Pulse Polarity bit
1 = Frame sync pulse is active-high
0 = Frame sync pulse is active-low

bit 12-2 **Unimplemented:** Read as ‘0’

bit 1 **FRMDLY**: Frame Sync Pulse Edge Select bit
1 = Frame sync pulse coincides with first bit clock
0 = Frame sync pulse precedes first bit clock

bit 0 **Unimplemented:** This bit must not be set to ‘1’ by the user application

REGISTER 19-7: ADCPC2: ANALOG-TO-DIGITAL CONVERT PAIR CONTROL REGISTER 2[(1)]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRQEN5	PEND5	SWTRG5	TRGSRC54	TRGSRC53	TRGSRC52	TRGSRC51	TRGSRC50
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
IRQEN4	PEND4	SWTRG4	TRGSRC44	TRGSRC43	TRGSRC42	TRGSRC41	TRGSRC40
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 **IRQEN5:** Interrupt Request Enable 5 bit

1 = Enables IRQ generation when requested conversion of Channels AN11 and AN10 is completed
0 = IRQ is not generated

bit 14 **PEND5:** Pending Conversion Status 5 bit

1 = Conversion of Channels AN11 and AN10 is pending; set when selected trigger is asserted
0 = Conversion is complete

bit 13 **SWTRG5:** Software Trigger 5 bit

1 = Starts conversion of AN11 and AN10 (if selected by the TRGSRCx bits)[(2)]
This bit is automatically cleared by hardware when the PEND5 bit is set.
0 = Conversion has not started

Note 1: This register is only implemented in the dsPIC33FJ16GS504 devices.

2: The trigger source must be set as a global software trigger prior to setting this bit to ‘1’. If other conversions are in progress, then the conversion will be performed when the conversion resources are available.

TABLE 22-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
10	BTSC	BTSC f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
12	BTST	BTST f,#bit4	Bit Test f	1	1	Z
		BTST.C Ws,#bit4	Bit Test Ws to C	1	1	C
		BTST.Z Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
		BTST.Z Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
13	BTSTS	BTSTS f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL lit23	Call Subroutine	2	2	None
		CALL Wn	Call Indirect Subroutine	1	2	None
15	CLR	CLR f	f = 0x0000	1	1	None
		CLR WREG	WREG = 0x0000	1	1	None
		CLR Ws	Ws = 0x0000	1	1	None
		CLR Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT	Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM f	f = $\overline{f}$	1	1	N,Z
		COM f,WREG	WREG = $\overline{f}$	1	1	N,Z
		COM Ws,Wd	Wd = $\overline{Ws}$	1	1	N,Z
18	CP	CP f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP Wb,#lit5	Compare Wb with lit5	1	1	C,DC,N,OV,Z
		CP Wb,Ws	Compare Wb with Ws (Wb – Ws)	1	1	C,DC,N,OV,Z
19	CP0	CP0 f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0 Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
20	CPB	CPB f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,Ws	Compare Wb with Ws, with Borrow (Wb – Ws – $\overline{C}$)	1	1	C,DC,N,OV,Z
21	CPSEQ	CPSEQ Wb, Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
22	CPSGT	CPSGT Wb, Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
23	CPSLT	CPSLT Wb, Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
24	CPSNE	CPSNE Wb, Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
25	DAW	DAW Wn	Wn = Decimal Adjust Wn	1	1	C
26	DEC	DEC f	f = f – 1	1	1	C,DC,N,OV,Z
		DEC f,WREG	WREG = f – 1	1	1	C,DC,N,OV,Z
		DEC Ws,Wd	Wd = Ws – 1	1	1	C,DC,N,OV,Z
27	DEC2	DEC2 f	f = f – 2	1	1	C,DC,N,OV,Z
		DEC2 f,WREG	WREG = f – 2	1	1	C,DC,N,OV,Z
		DEC2 Ws,Wd	Wd = Ws – 2	1	1	C,DC,N,OV,Z
28	DISI	DISI #lit14	Disable Interrupts for k Instruction Cycles	1	1	None

TABLE 25-6: DC CHARACTERISTICS: PROGRAM MEMORY

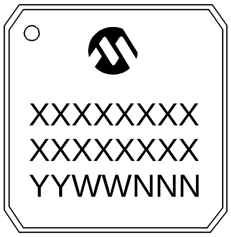
DC CHARACTERISTICS			**Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +150°C for High Temperature				
Param No.	**Symbol**	**Characteristic[1]**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
		Program Flash Memory					
HD130	EP	Cell Endurance	10,000	—	—	E/W	-40°C to +150°C[2]
HD134	TRETD	Characteristic Retention	20	—	—	Year	1000 E/W cycles or less and no other specifications are violated

Note 1: These parameters are assured by design, but are not characterized or tested in manufacturing.

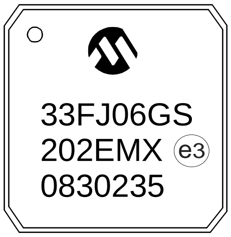
2: Programming of the Flash memory is not allowed above +125°C.

28.1 Package Marking Information (Continued)

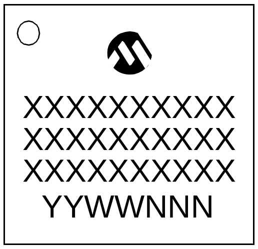
28-Lead UQFN



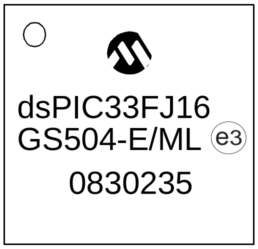
Example



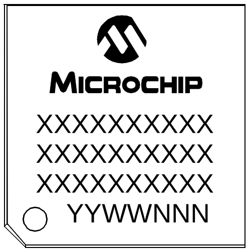
44-Lead QFN



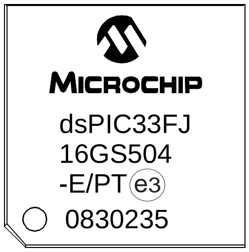
Example



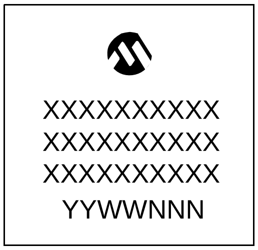
44-Lead TQFP



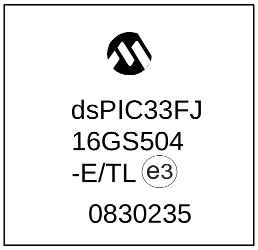
Example



44-Lead VTLA (TLA)

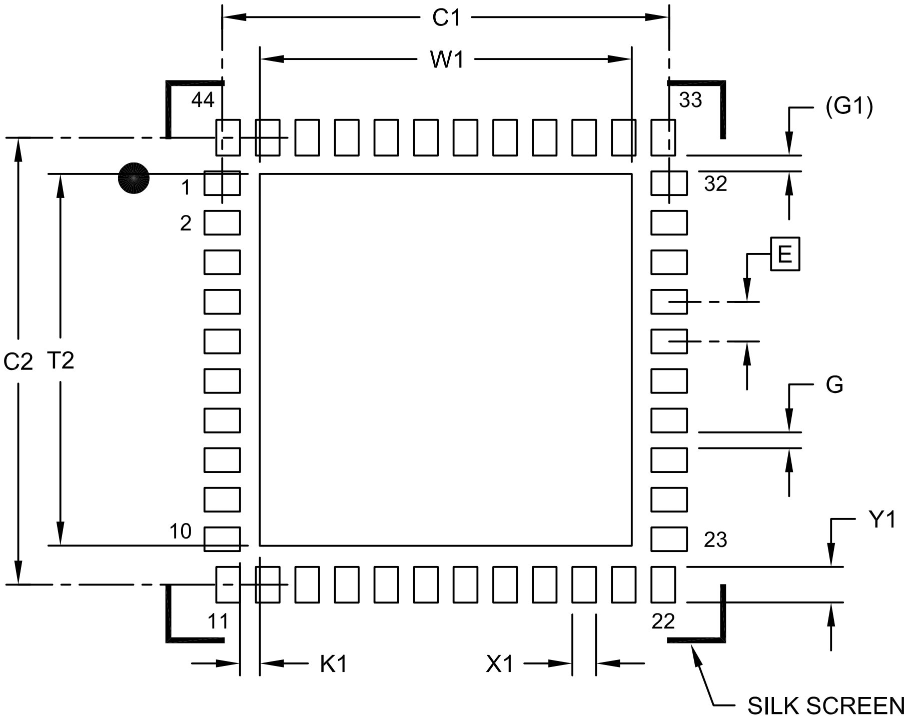


Example



44-Terminal Very Thin Leadless Array Package (TL) – 6x6x0.9 mm Body With Exposed Pad [VTLA]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Terminal Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			4.70
Optional Center Pad Length	T2			4.70
Terminal Pad Spacing	C1		5.65	
Terminal Pad Spacing	C2		5.65	
Terminal Pad Width (X44)	X1			0.30
Terminal Pad Length (X44)	Y1			0.45
Distance Between Pads	(G1)		0.20 REF.	
Distance Between Pads	G	0.20		
Distance Between Pads	K1	0.267		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2157A

TABLE A-4: MAJOR SECTION UPDATES (CONTINUED)

Section Name	Update Description
Section 24.0 “Electrical Characteristics”	Updated the Absolute Maximum Ratings. Updated the Operating MIPS vs. Voltage (see Table 24-1). Updated Parameter DC10 and Note 4, and removed Parameter DC18 from the DC Temperature and Voltage Specifications (see Table 24-4). Updated Note 2 in the IDD Operating Current specification (see Table 24-5). Updated all Typical values and Note 2 in the IIDLE Operating Current specification (see Table 24-6). Updated Typical values for Parameters DC60d, DC60a, DC60b, and DC60c, and Note 2 in the IPD Operating Current specification (see Table 24-7). Added all Typical values and Note 2 in the IDOZE Operating Current specification (see Table 24-8). Updated Parameters DI19 and DI50, added Parameters DI128, DI129, DI60a, DI60b, and DI60c, and removed Parameter DI57 in the I/O Pin Input Specifications (see Table 24-9). Revised all I/O Pinout Output Specifications (see Table 24-10). Added Notes 2 and 3 to the BOR Electrical Characteristics (see Table 24-11). Added Note 1 to Internal Voltage Regulator Specifications (see Table 24-13). Updated the External Clock Timing diagram (see Figure 24-2). Added Note 2 to the PLL Clock Timing Specifications (see Table 24-17). Removed Note 2 from the Internal FRC Accuracy (see Table 24-19). Updated Parameters DO31 and DO32 in the I/O Timing Requirements (see Table 24-21). Updated the External Clock Timing Requirements for Timer1, Timer2, and Timer3 (see Table 24-23, Table 24-24, and Table 24-25, respectively). Updated Parameters OC15 and OC20 in the Simple OC/PWM Mode Timing Requirements (see Table 24-28). Revised all SPIx Module Timing Characteristics diagrams and all Timing Requirements (see Figure 24-11 through Figure 24-18 and Table 24-30 through Table 24-37, respectively). Added Note 2 to the 10-bit High-Speed ADC Module Specifications (see Table 24-40). Added Note 2 to the 10-bit High-Speed ADC Module Timing Requirements (see Table 24-41). Added Note 2 to the Comparator Module Specifications (see Table 24-42). Added Parameter DA08 and Note 2 in the DAC Module Specifications (see Table 24-43). Updated Parameter DA16 and Note 2 in the DAC Output Buffer DC Specifications (see Table 24-44).