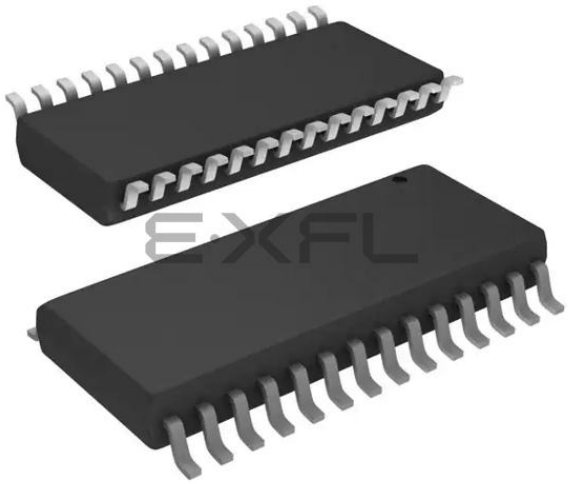
Microchip Technology - DSPIC33EP16GS202T-I/SO Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	70 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	21
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 12x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
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/dspic33ep16gs202t-i-so

3.8 Arithmetic Logic Unit (ALU)

The dsPIC33EPXXGS202 family ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. Depending on the operation, the ALU can affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

Refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157) for information on the SR bits affected by each instruction.

The core CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.8.1 MULTIPLIER

Using the high-speed 17-bit x 17-bit multiplier, the ALU supports unsigned, signed, or mixed-sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit signed x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.8.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 32-bit signed/16-bit signed divide
- 32-bit unsigned/16-bit unsigned divide
- 16-bit signed/16-bit signed divide
- 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn) and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.9 DSP Engine

The DSP engine consists of a high-speed 17-bit x 17-bit multiplier, a 40-bit barrel shifter and a 40-bit adder/subtracter (with two target accumulators, round and saturation logic).

The DSP engine can also perform inherent accumulator-to-accumulator operations that require no additional data. These instructions are ADD, SUB and NEG.

The DSP engine has options selected through bits in the CPU Core Control register (CORCON), as listed below:

- Fractional or Integer DSP Multiply (IF)
- Signed, unsigned or mixed-sign DSP multiply (USx)
- Conventional or Convergent Rounding (RND)
- Automatic Saturation On/Off for ACCA (SATA)
- Automatic Saturation On/Off for ACCB (SATB)
- Automatic Saturation On/Off for Writes to Data Memory (SATDW)
- Accumulator Saturation mode Selection (ACCSAT)

TABLE 3-2: DSP INSTRUCTIONS SUMMARY

Instruction	Algebraic Operation	ACC Write Back
CLR	$A = 0$	Yes
ED	$A = (x - y)^2$	No
EDAC	$A = A + (x - y)^2$	No
MAC	$A = A + (x \cdot y)$	Yes
MAC	$A = A + x^2$	No
MOVSAC	No change in A	Yes
MPY	$A = x \cdot y$	No
MPY	$A = x^2$	No
MPY.N	$A = -x \cdot y$	No
MSC	$A = A - x \cdot y$	Yes

FIGURE 4-9: BIT-REVERSED ADDRESSING EXAMPLE

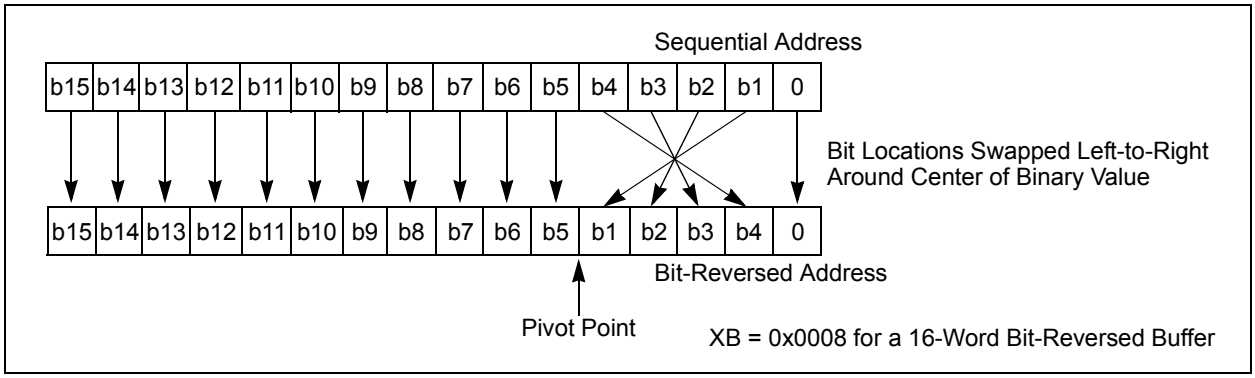


TABLE 4-26: BIT-REVERSED ADDRESSING SEQUENCE (16-ENTRY)

Normal Address					Bit-Reversed Address				
A3	**A2**	**A1**	**A0**	**Decimal**	**A3**	**A2**	**A1**	**A0**	**Decimal**
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	8
0	0	1	0	2	0	1	0	0	4
0	0	1	1	3	1	1	0	0	12
0	1	0	0	4	0	0	1	0	2
0	1	0	1	5	1	0	1	0	10
0	1	1	0	6	0	1	1	0	6
0	1	1	1	7	1	1	1	0	14
1	0	0	0	8	0	0	0	1	1
1	0	0	1	9	1	0	0	1	9
1	0	1	0	10	0	1	0	1	5
1	0	1	1	11	1	1	0	1	13
1	1	0	0	12	0	0	1	1	3
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	0	1	1	1	7
1	1	1	1	15	1	1	1	1	15

REGISTER 10-18: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
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 13-8 **RP37R<5:0>:** Peripheral Output Function is Assigned to RP37 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP36R<5:0>:** Peripheral Output Function is Assigned to RP36 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 10-19: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP38R5	RP38R4	RP38R3	RP38R2	RP38R1	RP38R0
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 13-8 **RP39R<5:0>:** Peripheral Output Function is Assigned to RP39 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP38R<5:0>:** Peripheral Output Function is Assigned to RP38 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 10-22: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
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 13-8 **RP45R<5:0>:** Peripheral Output Function is Assigned to RP45 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP44R<5:0>:** Peripheral Output Function is Assigned to RP44 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 10-23: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
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 13-8 **RP47R<5:0>:** Peripheral Output Function is Assigned to RP47 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP46R<5:0>:** Peripheral Output Function is Assigned to RP46 Output Pin bits (see Table 10-2 for peripheral function numbers)

REGISTER 12-2: T3CON: TIMER3 CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(2)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1)	—
bit 7							bit 0

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

bit 15 **TON:** Timer3 On bit(1)
1 = Starts 16-bit Timer3
0 = Stops 16-bit Timer3

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

bit 13 **TSIDL:** Timer3 Stop in Idle Mode bit(2)
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer3 Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation is enabled
0 = Gated time accumulation is disabled

bit 5-4 **TCKPS<1:0>:** Timer3 Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timer3 Clock Source Select bit(1)
1 = External clock is from pin, T3CK (on the rising edge)
0 = Peripheral Clock (FP)

bit 0 **Unimplemented:** Read as '0'

Note 1: When 32-bit operation is enabled (T2CON<3> = 1), these bits have no effect on Timer3 operation; all timer functions are set through T2CON.

2: When 32-bit timer operation is enabled (T32 = 1) in the Timer2 Control register (T2CON<3>), the TSIDL bit must be cleared to operate the 32-bit timer in Idle mode.

REGISTER 15-26: AUXCONx: PWMx AUXILIARY CONTROL REGISTER

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
HRPDIS	HRDDIS	—	—	BLANKSEL3	BLANKSEL2	BLANKSEL1	BLANKSEL0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHOPSEL3	CHOPSEL2	CHOPSEL1	CHOPSEL0	CHOPHEN	CHOPLEN
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 **HRPDIS:** High-Resolution PWMx Period Disable bit
1 = High-resolution PWMx period is disabled to reduce power consumption
0 = High-resolution PWMx period is enabled

bit 14 **HRDDIS:** High-Resolution PWMx Duty Cycle Disable bit
1 = High-resolution PWMx duty cycle is disabled to reduce power consumption
0 = High-resolution PWMx duty cycle is enabled

bit 13-12 **Unimplemented:** Read as '0'

bit 11-8 **BLANKSEL<3:0>:** PWMx State Blank Source Select bits
The selected state blank signal will block the current-limit and/or Fault input signals (if enabled via the BCH and BCL bits in the LEBCONx register).
1001 = Reserved
1000 = Reserved
0111 = Reserved
0110 = Reserved
0101 = Reserved
0100 = Reserved
0011 = PWM3H is selected as the state blank source
0010 = PWM2H is selected as the state blank source
0001 = PWM1H is selected as the state blank source
0000 = No state blanking

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHOPSEL<3:0>:** PWMx Chop Clock Source Select bits
The selected signal will enable and disable (chop) the selected PWMx outputs.
1001 = Reserved
1000 = Reserved
0111 = Reserved
0110 = Reserved
0101 = Reserved
0100 = Reserved
0011 = PWM3H is selected as the chop clock source
0010 = PWM2H is selected as the chop clock source
0001 = PWM1H is selected as the chop clock source
0000 = Chop clock generator is selected as the chop clock source

bit 1 **CHOPHEN:** PWMxH Output Chopping Enable bit
1 = PWMxH chopping function is enabled
0 = PWMxH chopping function is disabled

bit 0 **CHOPLEN:** PWMxL Output Chopping Enable bit
1 = PWMxL chopping function is enabled
0 = PWMxL chopping function is disabled

16.1 SPI Helpful Tips

1. In Frame mode, if there is a possibility that the master may not be initialized before the slave:
   a) If FRMPOL (SPI1CON2<13>) = 1, use a pull-down resistor on $\overline{SS1}$.
   b) If FRMPOL = 0, use a pull-up resistor on $\overline{SS1}$.

Note:	This ensures that the first frame transmission after initialization is not shifted or corrupted.

2. In Non-Framed 3-Wire mode (i.e., not using $\overline{SS1}$ from a master):
   a) If CKP (SPI1CON1<6>) = 1, always place a pull-up resistor on $\overline{SS1}$.
   b) If CKP = 0, always place a pull-down resistor on $\overline{SS1}$.

Note:	This will ensure that during power-up and initialization, the master/slave will not lose synchronization due to an errant SCK1 transition that would cause the slave to accumulate data shift errors for both transmit and receive, appearing as corrupted data.

3. FRMEN (SPI1CON2<15>) = 1 and SSEN (SPI1CON1<7>) = 1 are exclusive and invalid. In Frame mode, SCK1 is continuous and the frame sync pulse is active on the $\overline{SS1}$ pin, which indicates the start of a data frame.

Note:	Not all third-party devices support Frame mode timing. Refer to the SPI1 specifications in **Section 25.0 "Electrical Characteristics"** for details.

4. In Master mode only, set the SMP bit (SPI1CON1<9>) to a '1' for the fastest SPI1 data rate possible. The SMP bit can only be set at the same time or after the MSTEN bit (SPI1CON1<5>) is set.

To avoid invalid slave read data to the master, the user's master software must ensure enough time for slave software to fill its write buffer before the user application initiates a master write/read cycle. It is always advisable to preload the SPI1BUF Transmit register in advance of the next master transaction cycle. SPI1BUF is transferred to the SPI1 Shift register and is empty once the data transmission begins.

16.2 SPI Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page contains the latest updates and additional information.

16.2.1 KEY RESOURCES

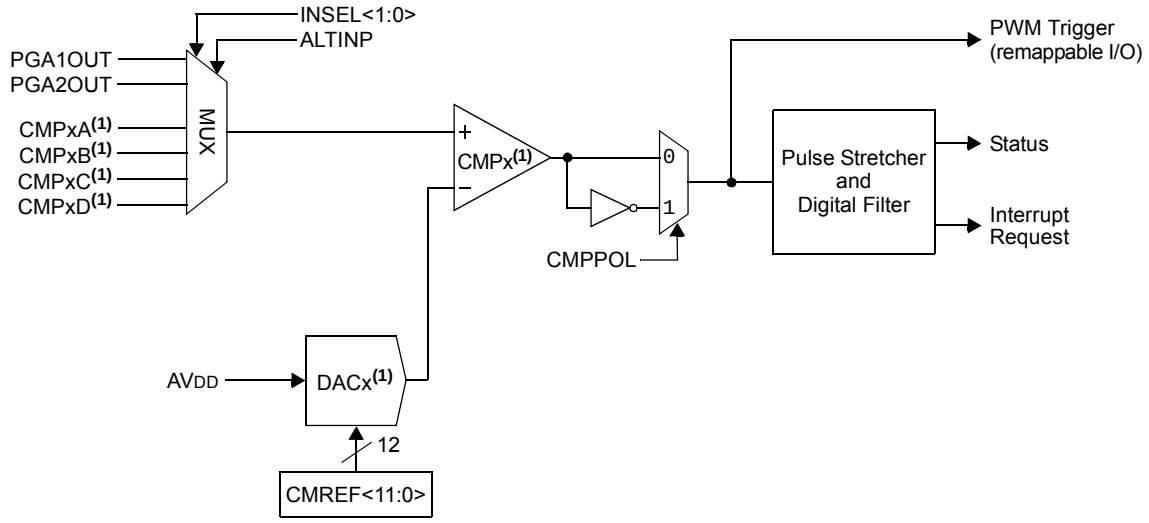
- **"Serial Peripheral Interface (SPI)"** (DS70005185) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

20.2 Module Description

Figure 20-1 shows a functional block diagram of one analog comparator from the high-speed analog comparator module. The analog comparator provides high-speed operation with a typical delay of 15 ns. The negative input of the comparator is always connected to the DACx circuit. The positive input of the comparator is connected to an analog multiplexer that selects the desired source pin.

The analog comparator input pins are typically shared with pins used by the Analog-to-Digital Converter (ADC) module. Both the comparator and the ADC can use the same pins at the same time. This capability enables a user to measure an input voltage with the ADC and detect voltage transients with the comparator.

FIGURE 20-1: HIGH-SPEED ANALOG COMPARATOR x MODULE BLOCK DIAGRAM



Note 1: x = 1-2

REGISTER 20-1: CMPxCON: COMPARATOR x CONTROL REGISTER (x = 1,2) (CONTINUED)

bit 2 **ALTINP:** Alternate Input Select bit
1 = INSEL<1:0> bits select alternate inputs
0 = INSEL<1:0> bits select comparator inputs

bit 1 **CMPPOL:** Comparator Output Polarity Control bit
1 = Output is inverted
0 = Output is non-inverted

bit 0 **Unimplemented:** Read as '0'

REGISTER 20-2: CMPxDAC: COMPARATOR DACx CONTROL REGISTER (x = 1,2)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CMREF<11:8>			
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
CMREF<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 15-12 **Unimplemented:** Read as '0'

bit 11-0 **CMREF<11:0>:** Comparator Reference Voltage Select bits
111111111111 = (CMREF<11:0> * (AVDD)/4096)
•
•
•
000000000000 = 0.0 volts

TABLE 22-2: CONFIGURATION BITS DESCRIPTION (CONTINUED)

Bit Field	Description
FNOSC<2:0>	Oscillator Selection bits 111 = Fast RC Oscillator with Divide-by-N (FRCDIVN) 110 = Fast RC Oscillator with Divide-by-16 101 = Low-Power RC Oscillator (LPRC) 100 = Reserved; do not use 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL) 010 = Primary Oscillator (XT, HS, EC) 001 = Fast RC Oscillator with Divide-by-N with PLL module (FRCPLL) 000 = Fast RC Oscillator (FRC)
FCKSM<1:0>	Clock Switching Mode bits 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
IOL1WAY	Peripheral Pin Select Configuration bit 1 = Allows only one reconfiguration 0 = Allows multiple reconfigurations
OSCIOFNC	OSC2 Pin Function bit (except in XT and HS modes) 1 = OSC2 is the clock output 0 = OSC2 is a general purpose digital I/O pin
POSCMD<1:0>	Primary Oscillator Mode Select bits 11 = Primary Oscillator is disabled 10 = HS Crystal Oscillator mode 01 = XT Crystal Oscillator mode 00 = EC (External Clock) mode
WDTEN<1:0>	Watchdog Timer Enable bits 11 = Watchdog Timer is always enabled (LPRC oscillator cannot be disabled; clearing the SWDTEN bit in the RCON register will have no effect) 10 = Watchdog Timer is enabled/disabled by user software (LPRC can be disabled by clearing the SWDTEN bit in the RCON register) 01 = Watchdog Timer is enabled only while device is active and is disabled while in Sleep mode; software control is disabled in this mode 00 = Watchdog Timer and the SWDTEN bit are disabled
WINDIS	Watchdog Timer Window Enable bit 1 = Watchdog Timer is in Non-Window mode 0 = Watchdog Timer is in Window mode
PLLKEN	PLL Lock Enable bit 1 = PLL lock is enabled 0 = PLL lock is disabled
WDTPRE	Watchdog Timer Prescaler bit 1 = 1:128 0 = 1:32
WDTPOST<3:0>	Watchdog Timer Postscaler bits 1111 = 1:32,768 1110 = 1:16,384 • • • 0001 = 1:2 0000 = 1:1

Note 1: The Boot Segment must be present to use the Alternate Interrupt Vector Table.

NOTES:

Most instructions are a single word. Certain double-word instructions are designed to provide all the required information in these 48 bits. In the second word, the 8 MSbs are '0's. If this second word is executed as an instruction (by itself), it executes as a NOP.

The double-word instructions execute in two instruction cycles.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter is changed as a result of the instruction, or a PSV or Table Read is performed. In these cases, the execution takes multiple instruction cycles, with the additional instruction cycle(s) executed as a NOP. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles.

Note: For more details on the instruction set, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).

TABLE 23-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
a ∈ {b, c, d}	a is selected from the set of values b, c, d
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write-back destination address register ∈ {W13, [W13]+ = 2}
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x0000...0x1FFF}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSb must be '0'
None	Field does not require an entry, can be blank
OA, OB, SA, SB	DSP Status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0...W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor Working register pair (Direct Addressing)

TABLE 23-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
47	MAC	MAC Wm*Wn,Acc,Wx,Wxd,Wy,Wyd,AWB	Multiply and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
		MAC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
48	MOV	MOV f,Wn	Move f to Wn	1	1	None
		MOV f	Move f to f	1	1	None
		MOV f,WREG	Move f to WREG	1	1	None
		MOV #lit16,Wn	Move 16-bit literal to Wn	1	1	None
		MOV.b #lit8,Wn	Move 8-bit literal to Wn	1	1	None
		MOV Wn,f	Move Wn to f	1	1	None
		MOV Wso,Wdo	Move Ws to Wd	1	1	None
		MOV WREG,f	Move WREG to f	1	1	None
		MOV.D Wns,Wd	Move Double from W(ns):W(ns + 1) to Wd	1	2	None
		MOV.D Ws,Wnd	Move Double from Ws to W(nd + 1):W(nd)	1	2	None
49	MOVPAG	MOVPAG #lit10,DSRPAG	Move 10-bit literal to DSRPAG	1	1	None
		MOVPAG #lit8,TBLPAG	Move 8-bit literal to TBLPAG	1	1	None
		MOVPAGW Ws, DSRPAG	Move Ws<9:0> to DSRPAG	1	1	None
		MOVPAGW Ws, TBLPAG	Move Ws<7:0> to TBLPAG	1	1	None
50	MOVSAC	MOVSAC Acc,Wx,Wxd,Wy,Wyd,AWB	Prefetch and store accumulator	1	1	None
51	MPY	MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
52	MPY.N	MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
53	MSC	MSC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd,AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
54	MUL	MUL.SS Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
		MUL.SS Wb,Ws,Acc	Accumulator = signed(Wb) * signed(Ws)	1	1	None
		MUL.SU Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.SU Wb,Ws,Acc	Accumulator = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Acc	Accumulator = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.US Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.US Wb,Ws,Acc	Accumulator = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.UU Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
		MUL.UU Wb,#lit5,Acc	Accumulator = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL.UU Wb,Ws,Acc	Accumulator = unsigned(Wb) * unsigned(Ws)	1	1	None
		MULW.SS Wb,Ws,Wnd	Wnd = signed(Wb) * signed(Ws)	1	1	None
		MULW.SU Wb,Ws,Wnd	Wnd = signed(Wb) * unsigned(Ws)	1	1	None
		MULW.US Wb,Ws,Wnd	Wnd = unsigned(Wb) * signed(Ws)	1	1	None
		MULW.UU Wb,Ws,Wnd	Wnd = unsigned(Wb) * unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.SU Wb,#lit5,Wnd	Wnd = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	Wnd = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL f	W3:W2 = f * WREG	1	1	None

Note: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.

24.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

24.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

24.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

24.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

24.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

24.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

24.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

TABLE 25-20: INTERNAL FRC 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.**	**Typ.**	**Max.**	**Units**	**Conditions**	
Internal FRC Accuracy @ FRC Frequency = 7.37 MHz(1,2)							
F20a	FRC	-2	0.5	+2	%	-40°C ≤ TA ≤ -10°C	VDD = 3.0-3.6V
		-0.9	0.5	+0.9	%	-10°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
F20b	FRC	-2	1	+2	%	+85°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

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

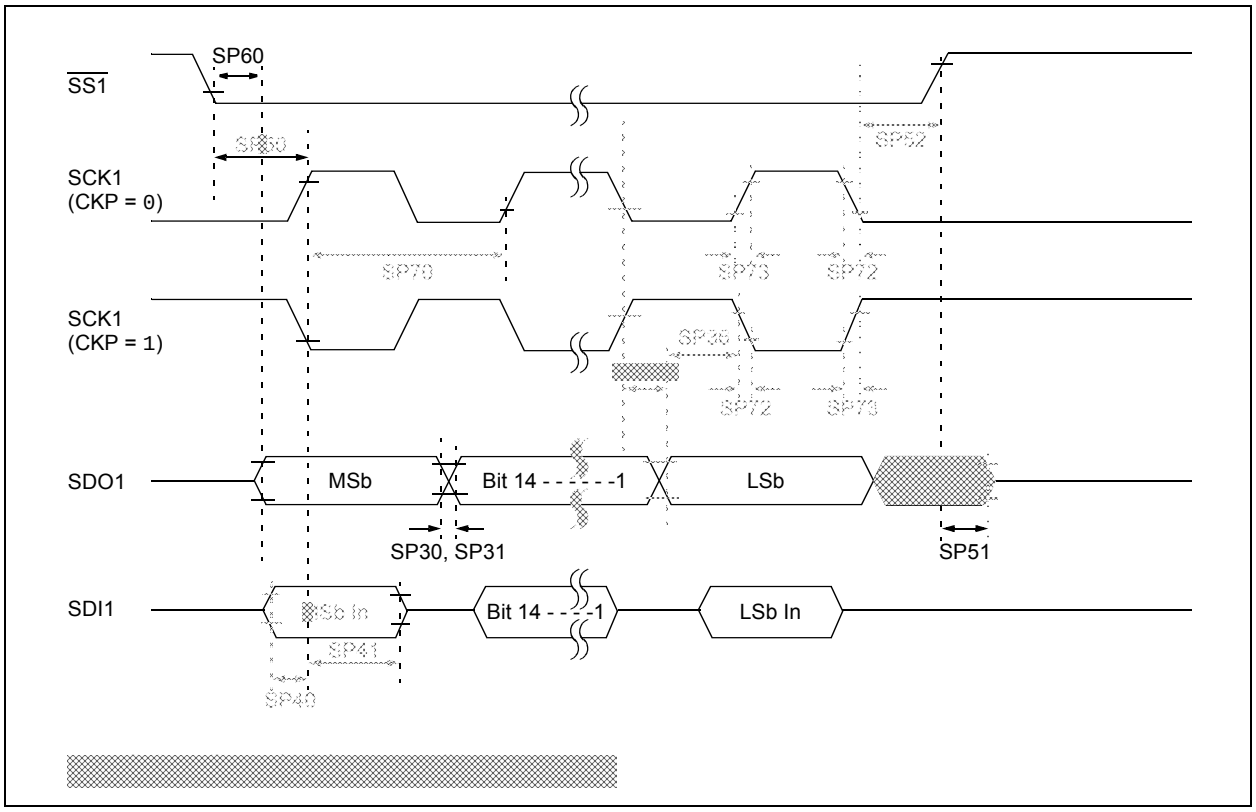
2: Over the lifetime of the 28-Lead 4x4 UQFN package device, the internal FRC accuracy could vary between ±4%.

TABLE 25-21: INTERNAL LPRC 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.**	**Typ.**	**Max.**	**Units**	**Conditions**	
LPRC @ 32.768 kHz(1)							
F21a	LPRC	-30	—	+30	%	-40°C ≤ TA ≤ -10°C	VDD = 3.0-3.6V
		-20	—	+20	%	-10°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
F21b	LPRC	-30	—	+30	%	+85°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

Note 1: This is the change of the LPRC frequency as VDD changes.

FIGURE 25-16: SPI1 SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = 1, SMP = 0) TIMING CHARACTERISTICS



26.0 DC AND AC DEVICE CHARACTERISTICS GRAPHS

Note: The graphs provided following this note are a statistical summary based on a limited number of samples and are provided for design guidance purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

FIGURE 26-1: VOH – 4x DRIVER PINS

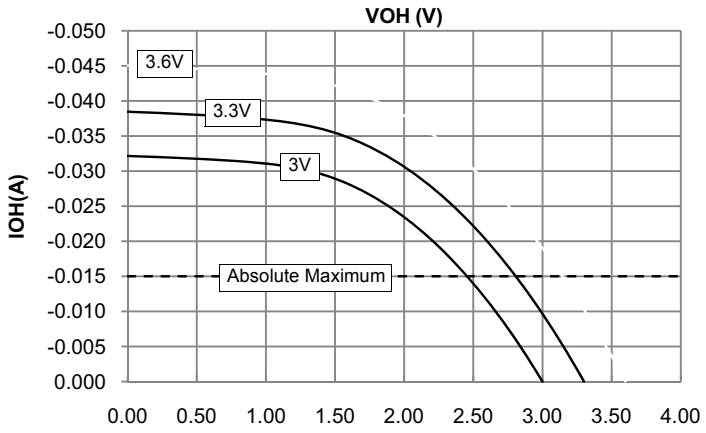


FIGURE 26-2: VOH – 8x DRIVER PINS

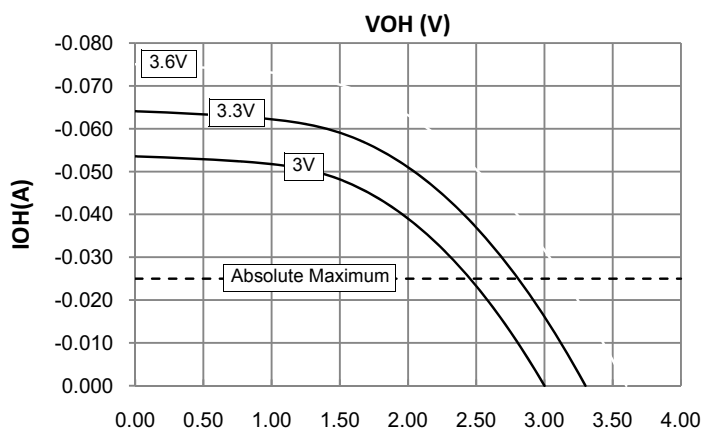


FIGURE 26-3: VOL – 4x DRIVER PINS

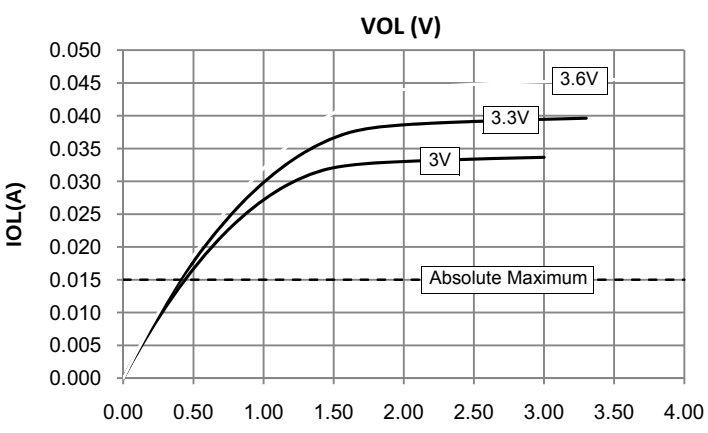
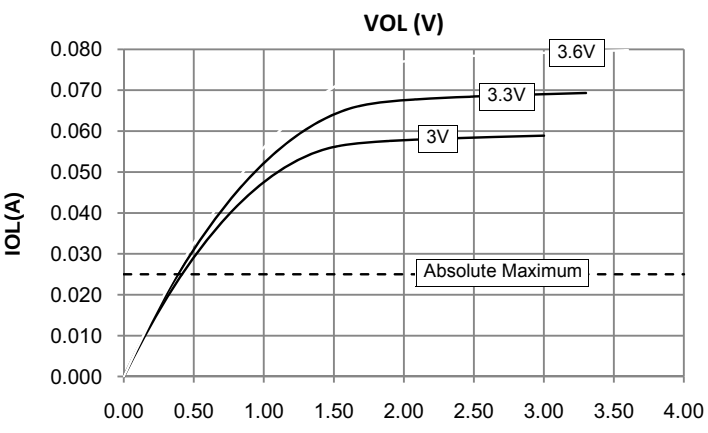
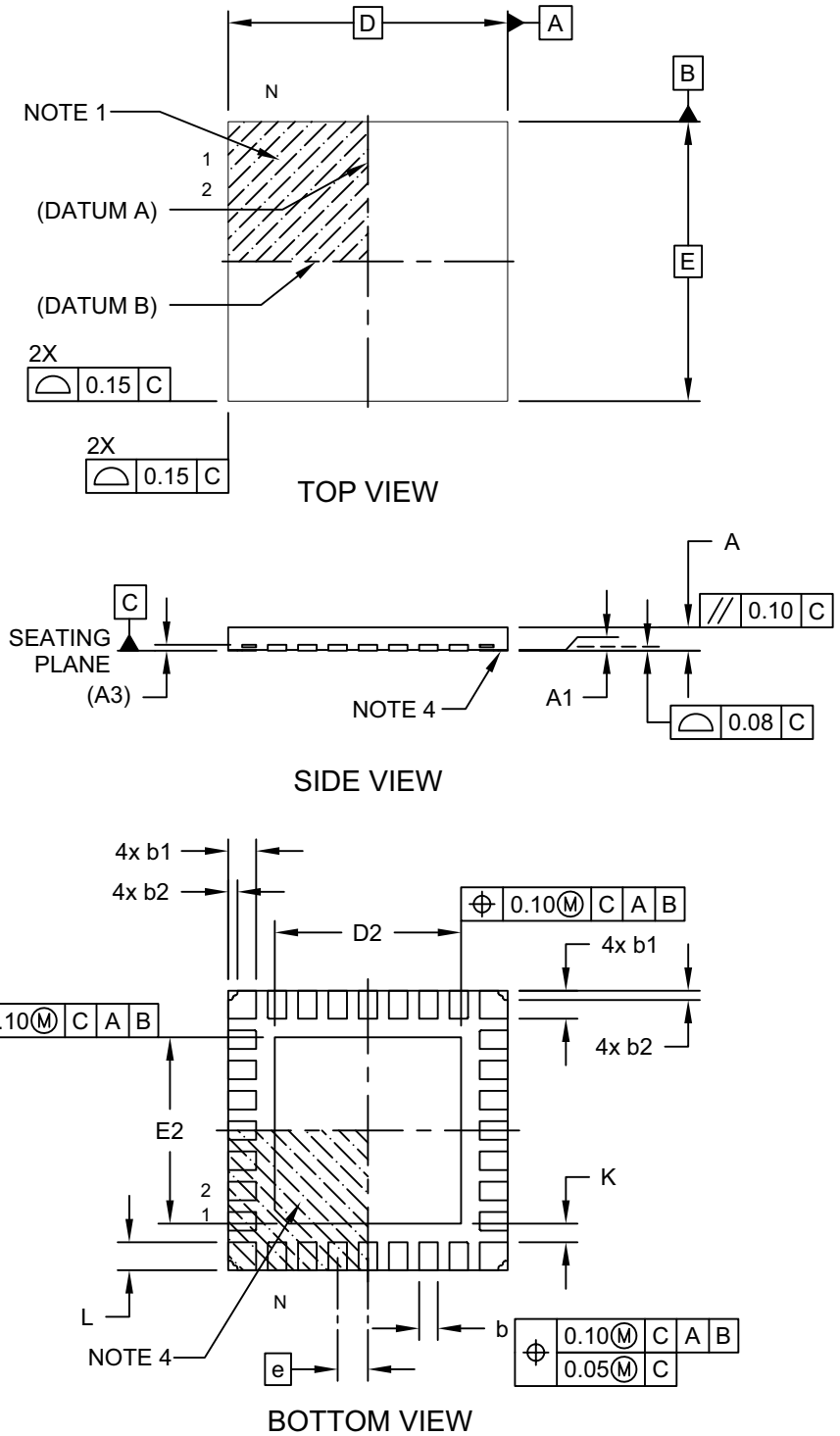


FIGURE 26-4: VOL – 8x DRIVER PINS



28-Lead Plastic Quad Flat, No Lead Package (MX) - 6x6x0.5mm Body [UQFN]
Ultra-Thin with 0.40 x 0.60 mm Terminal Width/Length and Corner Anchors

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



Microchip Technology Drawing C04-0209 Rev C Sheet 1 of 2

Registers
ACLKCON (Auxiliary Clock Divisor Control) 96
ADCAL0L (ADC Calibration 0 Low) 222
ADCAL1H (ADC Calibration 1 High) 223
ADCMPxCON (ADC Digital Comparator x Control) 224
ADCMPxENL (ADC Digital Comparator x Channel Enable Low) 225
ADCON1H (ADC Control 1 High) 203
ADCON1L (ADC Control 1 Low) 202
ADCON2H (ADC Control 2 High) 205
ADCON2L (ADC Control 2 Low) 204
ADCON3H (ADC Control 3 High) 207
ADCON3L (ADC Control 3 Low) 206
ADCON4H (ADC Control 4 High) 209
ADCON4L (ADC Control 4 Low) 208
ADCON5H (ADC Control 5 High) 211
ADCON5L (ADC Control 5 Low) 210
ADCORExH (Dedicated ADC Core x Control High) 213
ADCORExL (Dedicated ADC Core x Control Low) 212
ADEIEL (ADC Early Interrupt Enable Low) 215
ADEISTATL (ADC Early Interrupt Status Low) 215
ADFL0CON (ADC Digital Filter 0 Control) 226
ADIEL (ADC Interrupt Enable Low) 217
ADLVLTRGL (ADC Level-Sensitive Trigger Control Low) 214
ADMOD0H (ADC Input Mode Control 0 High) 216
ADMOD0L (ADC Input Mode Control 0 Low) 216
ADSTATL (ADC Data Ready Status Low) 217
ADTRIGxH (ADC Channel Trigger x Selection High) 220
ADTRIGxL (ADC Channel Trigger x Selection Low) 218
ALTDTRx (PWMx Alternate Dead-Time) 167
AUXCONx (PWMx Auxiliary Control) 175
CHOP (PWM Chop Clock Generator) 160
CLKDIV (Clock Divisor) 93
CMPxCON (Comparator x Control) 233
CMPxDAC (Comparator DACx Control) 234
CORCON (Core Control) 24, 80
CTXTSTAT (CPU W Register Context Status) 25
DEVID (Device ID) 245
DEVREV (Device Revision) 245
DTRx (PWMx Dead-Time) 167
FCLCONx (PWMx Fault Current-Limit Control) 171
I2C1CONH (I2C1 Control High) 189
I2C1CONL (I2C1 Control Low) 187
I2C1MSK (I2C1 Slave Mode Address Mask) 192
I2C1STAT (I2C1 Status) 190
IC1CON1 (Input Capture Control 1) 142
IC1CON2 (Input Capture Control 2) 143
INTCON1 (Interrupt Control 1) 81
INTCON2 (Interrupt Control 2) 83
INTCON3 (Interrupt Control 3) 84
INTCON4 (Interrupt Control 4) 84
INTTREG (Interrupt Control and Status) 85
IOCONx (PWMx I/O Control) 169
LEBCONx (PWMx Leading-Edge Blanking Control) 173
LEBDLYx (PWMx Leading-Edge Blanking Delay) 174
LFSR (Linear Feedback Shift) 97
MDC (PWM Master Duty Cycle) 161
NVMADR (Nonvolatile Memory Lower Address) 65
NVMADRU (Nonvolatile Memory Upper Address) 66
NVMCON (Nonvolatile Memory (NVM) Control) 64
NVMKEY (Nonvolatile Memory Key) 66
NVMSRCADRH (NVM Source Data Address High) 67
NVMSRCADRL (NVM Source Data Address Low) 67
OC1CON1 (Output Compare Control 1) 146
OC1CON2 (Output Compare Control 2) 148
OSCCON (Oscillator Control) 91
OSCTUN (FRC Oscillator Tuning) 95
PDCx (PWMx Generator Duty Cycle) 164
PGAxCAL (PGAx Calibration) 238
PGAxCON (PGAx Control) 237
PHASEx (PWMx Primary Phase-Shift) 165
PLLFBD (PLL Feedback Divisor) 94
PMD1 (Peripheral Module Disable Control 1) 102
PMD2 (Peripheral Module Disable Control 2) 103
PMD3 (Peripheral Module Disable Control 3) 103
PMD6 (Peripheral Module Disable Control 6) 104
PMD7 (Peripheral Module Disable Control 7) 105
PMD8 (Peripheral Module Disable Control 8) 105
PTCON (PWM Time Base Control) 155
PTCON2 (PWM Clock Divider Select 2) 156
PTPER (PWM Primary Master Time Base Period) 157
PWMCAPx (PWMx Primary Time Base Capture) 176
PWMCONx (PWMx Control) 162
PWMKEY (PWM Protection Lock/Unlock Key) 161
RCON (Reset Control) 71
RPINR0 (Peripheral Pin Select Input 0) 115
RPINR1 (Peripheral Pin Select Input 1) 115
RPINR11 (Peripheral Pin Select Input 11) 118
RPINR12 (Peripheral Pin Select Input 12) 119
RPINR13 (Peripheral Pin Select Input 13) 120
RPINR18 (Peripheral Pin Select Input 18) 121
RPINR2 (Peripheral Pin Select Input 2) 116
RPINR20 (Peripheral Pin Select Input 20) 122
RPINR21 (Peripheral Pin Select Input 21) 123
RPINR3 (Peripheral Pin Select Input 3) 117
RPINR37 (Peripheral Pin Select Input 37) 123
RPINR38 (Peripheral Pin Select Input 38) 124
RPINR42 (Peripheral Pin Select Input 42) 125
RPINR43 (Peripheral Pin Select Input 43) 126
RPINR7 (Peripheral Pin Select Input 7) 118
RPOR0 (Peripheral Pin Select Output 0) 127
RPOR1 (Peripheral Pin Select Output 1) 127
RPOR10 (Peripheral Pin Select Output 10) 132
RPOR2 (Peripheral Pin Select Output 2) 128
RPOR3 (Peripheral Pin Select Output 3) 128
RPOR4 (Peripheral Pin Select Output 4) 129
RPOR5 (Peripheral Pin Select Output 5) 129
RPOR6 (Peripheral Pin Select Output 6) 130
RPOR7 (Peripheral Pin Select Output 7) 130
RPOR8 (Peripheral Pin Select Output 8) 131
RPOR9 (Peripheral Pin Select Output 9) 131
SDCx (PWMx Secondary Duty Cycle) 164
SEVTCMP (PWM Special Event Compare) 157
SPHASEx (PWMx Secondary Phase-Shift) 166
SPI1CON1 (SPI1 Control 1) 181
SPI1CON2 (SPI1 Control 2) 183
SPI1STAT (SPI1 Status and Control) 179