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 | 60 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 19x12b; D/A 1x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-TQFP Exposed Pad |
| Supplier Device Package | 48-TQFP-EP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic33ep16gs505-e-pt |

## 3.6 CPU 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.

### 3.6.1 KEY RESOURCES

- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All related *“dsPIC33/PIC24 Family Reference Manual”* Sections
- Development Tools

## 4.5 Special Function Register Maps

### TABLE 4-2: CPU CORE REGISTER MAP

| SFR Name | Addr. | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W0 | 0000 | W0 (WREG) | | | | | | | | | | | | | | | | xxxx |
| W1 | 0002 | W1 | | | | | | | | | | | | | | | | xxxx |
| W2 | 0004 | W2 | | | | | | | | | | | | | | | | xxxx |
| W3 | 0006 | W3 | | | | | | | | | | | | | | | | xxxx |
| W4 | 0008 | W4 | | | | | | | | | | | | | | | | xxxx |
| W5 | 000A | W5 | | | | | | | | | | | | | | | | xxxx |
| W6 | 000C | W6 | | | | | | | | | | | | | | | | xxxx |
| W7 | 000E | W7 | | | | | | | | | | | | | | | | xxxx |
| W8 | 0010 | W8 | | | | | | | | | | | | | | | | xxxx |
| W9 | 0012 | W9 | | | | | | | | | | | | | | | | xxxx |
| W10 | 0014 | W10 | | | | | | | | | | | | | | | | xxxx |
| W11 | 0016 | W11 | | | | | | | | | | | | | | | | xxxx |
| W12 | 0018 | W12 | | | | | | | | | | | | | | | | xxxx |
| W13 | 001A | W13 | | | | | | | | | | | | | | | | xxxx |
| W14 | 001C | W14 | | | | | | | | | | | | | | | | xxxx |
| W15 | 001E | W15 | | | | | | | | | | | | | | | | xxxx |
| SPLIM | 0020 | SPLIM | | | | | | | | | | | | | | | | 0000 |
| ACCAL | 0022 | ACCAL | | | | | | | | | | | | | | | | 0000 |
| ACCAH | 0024 | ACCAH | | | | | | | | | | | | | | | | 0000 |
| ACCAU | 0026 | Sign Extension of ACCA<39> | | | | | | | | ACCAU | | | | | | | | 0000 |
| ACCBL | 0028 | ACCBL | | | | | | | | | | | | | | | | 0000 |
| ACCBH | 002A | ACCBH | | | | | | | | | | | | | | | | 0000 |
| ACCBU | 002C | Sign Extension of ACCB<39> | | | | | | | | ACCBU | | | | | | | | 0000 |
| PCL | 002E | PCL<15:1> | | | | | | | | | | | | | | | — | 0000 |
| PCH | 0030 | — | — | — | — | — | — | — | — | — | PCH<6:0> | | | | | | | 0000 |
| DSRPAG | 0032 | — | — | — | — | — | — | Extended Data Space (EDS) Read Page Register (DSRPAG<9:0>) | | | | | | | | | | 0001 |
| DSWPAG[1] | 0034 | — | — | — | — | — | — | — | Extended Data Space (EDS) Write Page Register (DSWPAG8:0>)[1] | | | | | | | | | 0001 |
| RCOUNT | 0036 | RCOUNT<15:0> | | | | | | | | | | | | | | | | 0000 |
| DCOUNT | 0038 | DO Loop Count Register (DCOUNT<15:0>) | | | | | | | | | | | | | | | | 0000 |
| DOSTARTL | 003A | DO Start Address Register Low (DOSTARTL<15:1>) | | | | | | | | | | | | | | | — | 0000 |
| DOSTARTH | 003C | — | — | — | — | — | — | — | — | — | — | DO Start Address Register High (DOSTARTH<5:0>) | | | | | | 0000 |

**Legend:** x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** The contents of this register should never be modified. The DSWPAG must always point to the first page.

## 4.6 Instruction Addressing Modes

The addressing modes shown in Table 4-38 form the basis of the addressing modes optimized to support the specific features of individual instructions. The addressing modes provided in the MAC class of instructions differ from those in the other instruction types.

### 4.6.1 FILE REGISTER INSTRUCTIONS

Most file register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (Near Data Space). Most file register instructions employ a Working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same file register or WREG (with the exception of the MUL instruction), which writes the result to a register or register pair. The MOV instruction allows additional flexibility and can access the entire Data Space.

### 4.6.2 MCU INSTRUCTIONS

The three-operand MCU instructions are of the form:

```
Operand 3 = Operand 1 <function> Operand 2
```

where Operand 1 is always a Working register (that is, the addressing mode can only be Register Direct), which is referred to as Wb. Operand 2 can be a W register fetched from data memory or a 5-bit literal. The result location can either be a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-Modified
- Register Indirect Pre-Modified
- 5-Bit or 10-Bit Literal

**Note:** Not all instructions support all the addressing modes given above. Individual instructions can support different subsets of these addressing modes.

**TABLE 4-38: FUNDAMENTAL ADDRESSING MODES SUPPORTED**

| Addressing Mode | Description |
|---|---|
| File Register Direct | The address of the file register is specified explicitly. |
| Register Direct | The contents of a register are accessed directly. |
| Register Indirect | The contents of Wn form the Effective Address (EA). |
| Register Indirect Post-Modified | The contents of Wn form the EA. Wn is post-modified (incremented or decremented) by a constant value. |
| Register Indirect Pre-Modified | Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA. |
| Register Indirect with Register Offset (Register Indexed) | The sum of Wn and Wb forms the EA. |
| Register Indirect with Literal Offset | The sum of Wn and a literal forms the EA. |

## 5.0 FLASH PROGRAM MEMORY

> **Note 1:** This data sheet summarizes the features of the dsPIC33EPXXGS50X family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Flash Programming"** (DS70005156) in the *"dsPIC33/PIC24 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 dsPIC33EPXXGS50X family devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire VDD range.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™) programming capability
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)
- Run-Time Self-Programming (RTSP)

ICSP allows for a dsPIC33EPXXGS50X family device to be serially programmed while in the end application circuit. This is done with a programming clock and programming data (PGECx/PGEDx) line, and three other lines for power (VDD), ground (VSS) and Master Clear (MCLR). This allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the Program Executive, to manage the programming process. Using an SPI data frame format, the Program Executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

RTSP is accomplished using `TBLRD` (Table Read) and `TBLWT` (Table Write) instructions. With RTSP, the user application can write program memory data with a single program memory word and erase program memory in blocks or 'pages' of 512 instructions (1536 bytes) at a time.

### 5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using bits<7:0> of the TBLPAG register and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 5-1. The `TBLRDL` and the `TBLWTL` instructions are used to read or write to bits<15:0> of program memory. `TBLRDL` and `TBLWTL` can access program memory in both Word and Byte modes. The `TBLRDH` and `TBLWTH` instructions are used to read or write to bits<23:16> of program memory. `TBLRDH` and `TBLWTH` can also access program memory in Word or Byte mode.

**FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS**

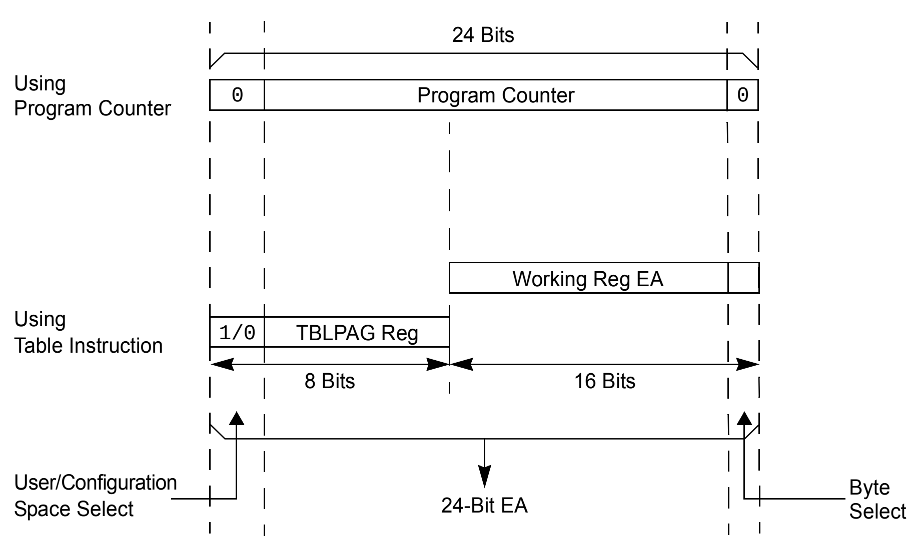
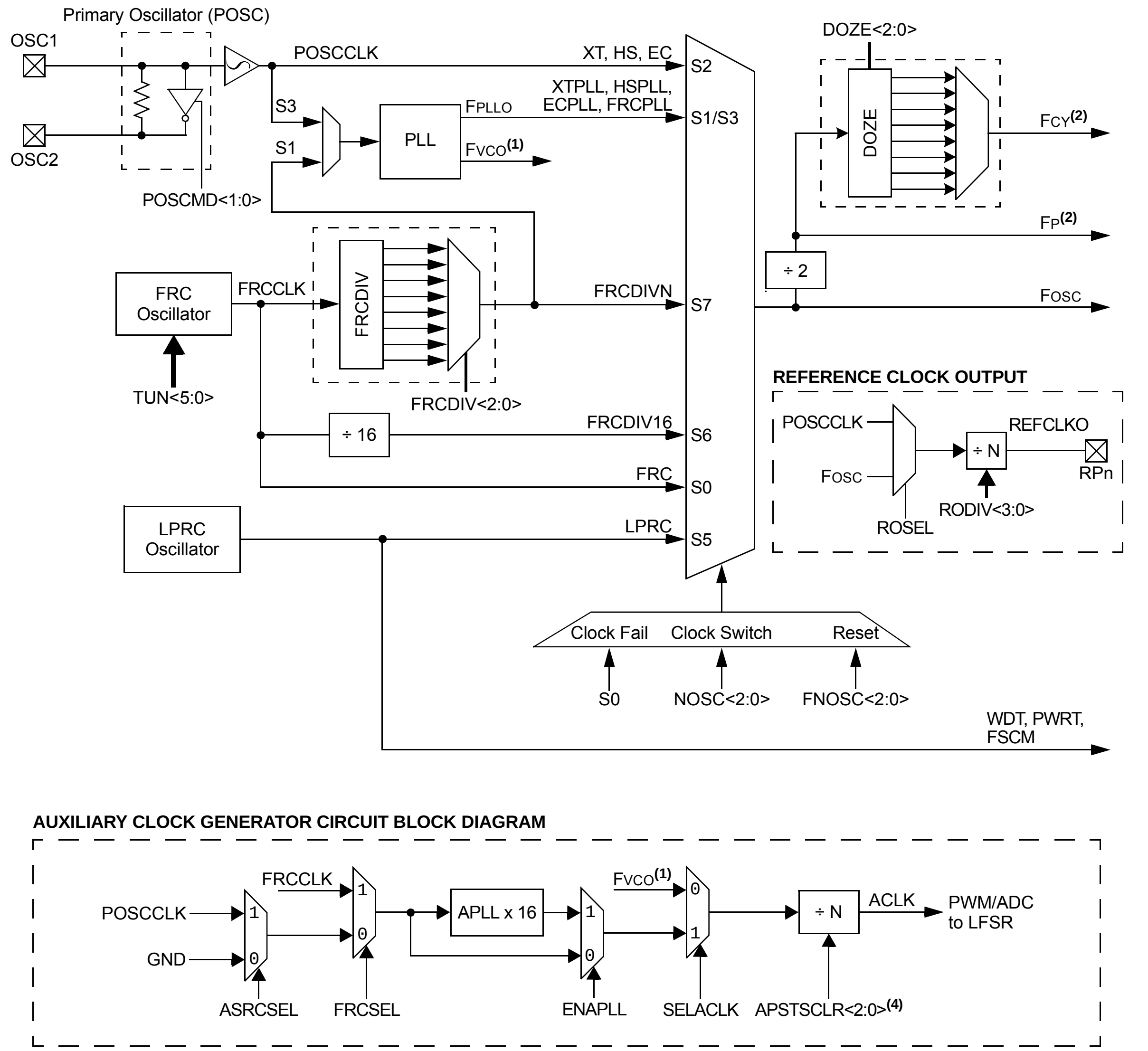


**FIGURE 8-1: OSCILLATOR SYSTEM DIAGRAM**




**Note 1:** See Figure 8-2 for the source of the FVCO signal.

**2:** FP refers to the clock source for all the peripherals, while FCY (or MIPS) refers to the clock source for the CPU. Throughout this document, FCY and FP are used interchangeably, except in the case of Doze mode. FP and FCY will be different when Doze mode is used in any ratio other than 1:1.

**3:** The auxiliary clock postscaler must be configured to divide-by-1 (APSTSCLR<2:0> = `111`) for proper operation of the PWM and ADC modules.

### REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER(1) (CONTINUED)

| | |
|---|---|
| bit 4 | **Unimplemented:** Read as '0' |
| bit 3 | **CF:** Clock Fail Detect bit(3)<br>1 = FSCM has detected a clock failure<br>0 = FSCM has not detected a clock failure |
| bit 2-1 | **Unimplemented:** Read as '0' |
| bit 0 | **OSWEN:** Oscillator Switch Enable bit<br>1 = Requests oscillator switch to the selection specified by the NOSC<2:0> bits<br>0 = Oscillator switch is complete |

**Note 1:** Writes to this register require an unlock sequence.

**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 transitional clock source between the two PLL modes.

**3:** This bit should only be cleared in software. Setting the bit in software (= 1) will have the same effect as an actual oscillator failure and will trigger an oscillator failure trap.

**REGISTER 9-5: PMD6: PERIPHERAL MODULE DISABLE CONTROL REGISTER 6**

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | PWM5MD | PWM4MD | PWM3MD | PWM2MD | PWM1MD |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-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 15-13 **Unimplemented:** Read as ‘0’

bit 12 **PWM5MD:** PWM5 Module Disable bit
1 = PWM5 module is disabled
0 = PWM5 module is enabled

bit 11 **PWM4MD:** PWM4 Module Disable bit
1 = PWM4 module is disabled
0 = PWM4 module is enabled

bit 10 **PWM3MD:** PWM3 Module Disable bit
1 = PWM3 module is disabled
0 = PWM3 module is enabled

bit 9 **PWM2MD:** PWM2 Module Disable bit
1 = PWM2 module is disabled
0 = PWM2 module is enabled

bit 8 **PWM1MD:** PWM1 Module Disable bit
1 = PWM1 module is disabled
0 = PWM1 module is enabled

bit 7-0 **Unimplemented:** Read as ‘0’

**REGISTER 10-12: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| SCK1INR7 | SCK1INR6 | SCK1INR5 | SCK1INR4 | SCK1INR3 | SCK1INR2 | SCK1INR1 | SCK1INR0 |
| 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 |
|---|---|---|---|---|---|---|---|
| SDI1R7 | SDI1R6 | SDI1R5 | SDI1R4 | SDI1R3 | SDI1R2 | SDI1R1 | SDI1R0 |
| 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 **SCK1INR<7:0>:** Assign SPI1 Clock Input (SCK1) to the Corresponding RPn Pin bits

10110101 = Input tied to RP181
10110100 = Input tied to RP180
•
•
•
00000001 = Input tied to RP1
00000000 = Input tied to VSS

bit 7-0 **SDI1R<7:0>:** Assign SPI1 Data Input (SDI1) to the Corresponding RPn Pin bits

10110101 = Input tied to RP181
10110100 = Input tied to RP180
•
•
•
00000001 = Input tied to RP1
00000000 = Input tied to VSS

## 11.2 Timer1 Control Register

**REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER**

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

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | TGATE | TCKPS1 | TCKPS0 | — | TSYNC(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:** Timer1 On bit(1)
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **TSIDL:** Timer1 Stop in Idle Mode bit
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:** Timer1 Gated Time Accumulation Enable bit
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>:** Timer1 Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit(1)
When TCS = 1:
1 = Synchronizes external clock input
0 = Does not synchronize external clock input
When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit(1)
1 = External clock is from pin, T1CK (on the rising edge)
0 = Internal clock (FP)

bit 0 **Unimplemented:** Read as ‘0’

**Note 1:** When Timer1 is enabled in External Synchronous Counter mode (TCS = 1, TSYNC = 1, TON = 1), any attempts by user software to write to the TMR1 register are ignored.

**NOTES:**

**REGISTER 15-5: STCON: PWMx SECONDARY MASTER TIME BASE CONTROL REGISTER**

| U-0 | U-0 | U-0 | R-0, HSC | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | SESTAT | SEIEN | EIPU(1) | SYNCPOL | SYNCOEN |
| 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 |
|---|---|---|---|---|---|---|---|
| SYNCEN | SYNCSRC2 | SYNCSRC1 | SYNCSRC0 | SEVTPS3 | SEVTPS2 | SEVTPS1 | SEVTPS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **SESTAT:** Special Event Interrupt Status bit
1 = Secondary special event interrupt is pending
0 = Secondary special event interrupt is not pending

bit 11 **SEIEN:** Special Event Interrupt Enable bit
1 = Secondary special event interrupt is enabled
0 = Secondary special event interrupt is disabled

bit 10 **EIPU:** Enable Immediate Period Updates bit(1)
1 = Active Secondary Period register is updated immediately
0 = Active Secondary Period register updates occur on PWMx cycle boundaries

bit 9 **SYNCPOL:** Synchronize Input and Output Polarity bit
1 = SYNCIx/SYNCO2 polarity is inverted (active-low)
0 = SYNCIx/SYNCO2 polarity is active-high

bit 8 **SYNCOEN:** Secondary Master Time Base Synchronization Enable bit
1 = SYNCO2 output is enabled
0 = SYNCO2 output is disabled

bit 7 **SYNCEN:** External Secondary Master Time Base Synchronization Enable bit
1 = External synchronization of secondary time base is enabled
0 = External synchronization of secondary time base is disabled

bit 6-4 **SYNCSRC<2:0>:** Secondary Time Base Sync Source Selection bits
111 = Reserved
101 = Reserved
100 = Reserved
011 = Reserved
010 = Reserved
001 = SYNCI2
000 = SYNCI1

bit 3-0 **SEVTPS<3:0>:** PWMx Secondary Special Event Trigger Output Postscaler Select bits
1111 = 1:16 Postscale
0001 = 1:2 Postscale
•
•
•
0000 = 1:1 Postscale

**Note 1:** This bit only applies to the secondary master time base period.

**REGISTER 15-24: LEBCONx: PWMx LEADING-EDGE BLANKING (LEB) CONTROL REGISTER (x = 1 to 5)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| PHR | PHF | PLR | PLF | FLTLEBEN | CLLEBEN | — | — |
| 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 |
|---|---|---|---|---|---|---|---|
| — | — | BCH(1) | BCL(1) | BPHH | BPHL | BPLH | BPLL |
| 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 **PHR:** PWMxH Rising Edge Trigger Enable bit
1 = Rising edge of PWMxH will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the rising edge of PWMxH

bit 14 **PHF:** PWMxH Falling Edge Trigger Enable bit
1 = Falling edge of PWMxH will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the falling edge of PWMxH

bit 13 **PLR:** PWMxL Rising Edge Trigger Enable bit
1 = Rising edge of PWMxL will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the rising edge of PWMxL

bit 12 **PLF:** PWMxL Falling Edge Trigger Enable bit
1 = Falling edge of PWMxL will trigger the Leading-Edge Blanking counter
0 = Leading-Edge Blanking ignores the falling edge of PWMxL

bit 11 **FLTLEBEN:** Fault Input Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to the selected Fault input
0 = Leading-Edge Blanking is not applied to the selected Fault input

bit 10 **CLLEBEN:** Current-Limit Leading-Edge Blanking Enable bit
1 = Leading-Edge Blanking is applied to the selected current-limit input
0 = Leading-Edge Blanking is not applied to the selected current-limit input

bit 9-6 **Unimplemented:** Read as ‘0’

bit 5 **BCH:** Blanking in Selected Blanking Signal High Enable bit(1)
1 = State blanking (of current-limit and/or Fault input signals) when the selected blanking signal is high
0 = No blanking when the selected blanking signal is high

bit 4 **BCL:** Blanking in Selected Blanking Signal Low Enable bit(1)
1 = State blanking (of current-limit and/or Fault input signals) when the selected blanking signal is low
0 = No blanking when the selected blanking signal is low

bit 3 **BPHH:** Blanking in PWMxH High Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxH output is high
0 = No blanking when the PWMxH output is high

bit 2 **BPHL:** Blanking in PWMxH Low Enable bit
1 = State blanking (of current-limit and/or Fault input signals) when the PWMxH output is low
0 = No blanking when the PWMxH output is low

**Note 1:** The blanking signal is selected via the BLANKSEL<3:0> bits in the AUXCONx register.

### REGISTER 21-1: PGAxCON: PGAx CONTROL REGISTER (CONTINUED)

bit 2-0 **GAIN<2:0>:** PGAx Gain Selection bits

`111` = Reserved
`110` = Gain of 64x
`101` = Gain of 32x
`100` = Gain of 16x
`011` = Gain of 8x
`010` = Gain of 4x
`001` = Reserved
`000` = Reserved

### REGISTER 21-2: PGAxCAL: PGAx CALIBRATION REGISTER

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|
| — | — | PGACAL<5:0> | | | | | |
| 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 |

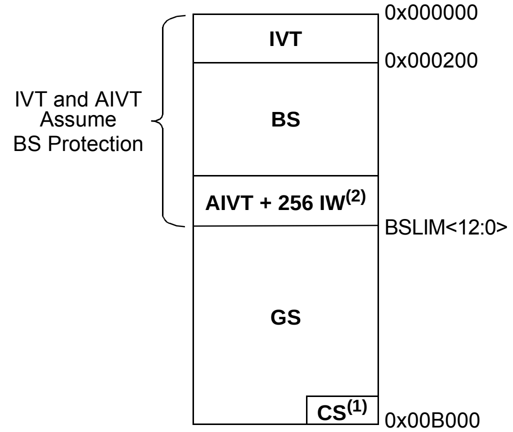
bit 15-6 **Unimplemented:** Read as ‘`0`’

bit 5-0 **PGACAL<5:0>:** PGAx Offset Calibration bits

The calibration values for PGA1 and PGA2 must be copied from Flash addresses, 0x800E48 and 0x800E4C, respectively, into these bits before the module is enabled. Refer to the calibration data address table (Table 23-3) in **Section 23.0 “Special Features”** for more information.

The different device security segments are shown in Figure 23-3. Here, all three segments are shown but are not required. If only basic code protection is required, then GS can be enabled independently or combined with CS, if desired.

**FIGURE 23-3: SECURITY SEGMENTS EXAMPLE FOR dsPIC33EP64GS50X DEVICES**




**Note 1:** If CS is write-protected, the last page (GS + CS) of program memory will be protected from an erase condition.

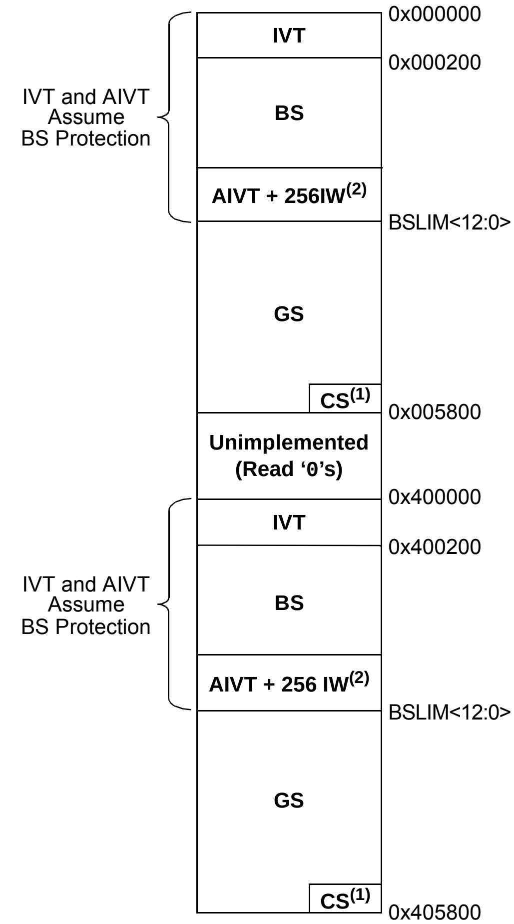
**2:** The last half (256 IW) of the last page of BS is unusable program memory.

dsPIC33EP64GS50X family devices can be operated in Dual Partition mode, where security is required for each partition. When operating in Dual Partition mode, the Active and Inactive Partitions both contain unique copies of the Reset vector, Interrupt Vector Tables (IVT and AIVT, if enabled) and the Flash Configuration Words. Both partitions have the three security segments described previously. Code may not be executed from the Inactive Partition, but it may be programmed by, and read from, the Active Partition, subject to defined code protection. Figure 23-4 shows the different security segments for a device operating in Dual Partition mode.

The device may also operate in a Protected Dual Partition mode or in Privileged Dual Partition mode. In Protected Dual Partition mode, Partition 1 is permanently erase/write-protected. This implementation allows for a “Factory Default” mode, which provides a fail-safe backup image to be stored in Partition 1. For example, a fail-safe bootloader can be placed in Partition 1, along with a fail-safe backup code image, which can be used or rewritten into Partition 2 in the event of a failed Flash update to Partition 2.

Privileged Dual Partition mode performs the same function as Protected Dual Partition mode, except additional constraints are applied in an effort to prevent code in the Boot Segment and General Segment from being used against each other.

**FIGURE 23-4: SECURITY SEGMENTS EXAMPLE FOR dsPIC33EP64GS50X DEVICES (DUAL PARTITION MODES)**




**Note 1:** If CS is write-protected, the last page (GS + CS) of program memory will be protected from an erase condition.

**2:** The last half (256 IW) of the last page of BS is unusable program memory.

## 25.6 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.

## 25.7 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.

## 25.8 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.

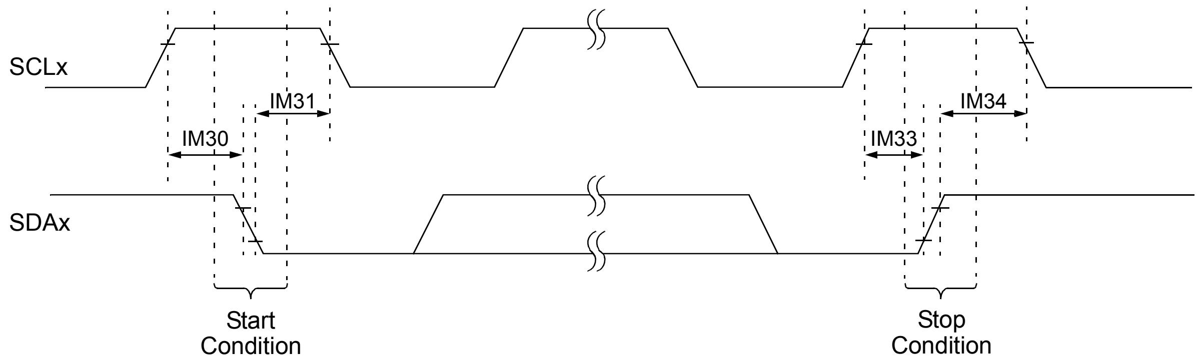
## 25.9 PICkit 3 In-Circuit Debugger/ Programmer

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™).

## 25.10 MPLAB PM3 Device Programmer

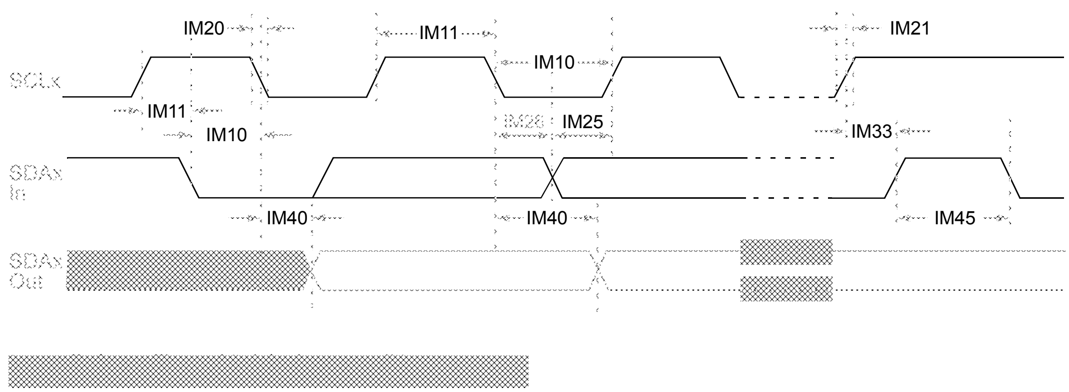
The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at $V_{DDMIN}$ and $V_{DDMAX}$ for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various 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.

**FIGURE 26-19: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)**




**Note:** Refer to Figure 26-1 for load conditions.

**FIGURE 26-20: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)**



## 27.0 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 27-1: VOH – 4x DRIVER PINS**

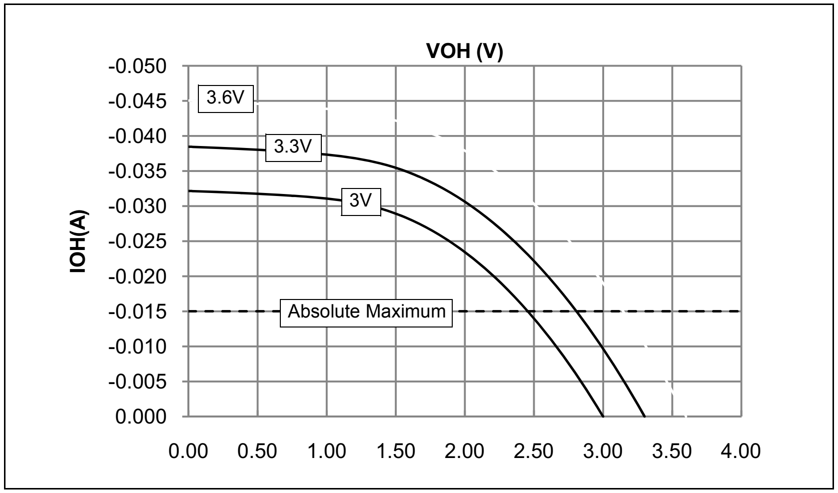



**FIGURE 27-2: VOH – 8x DRIVER PINS**

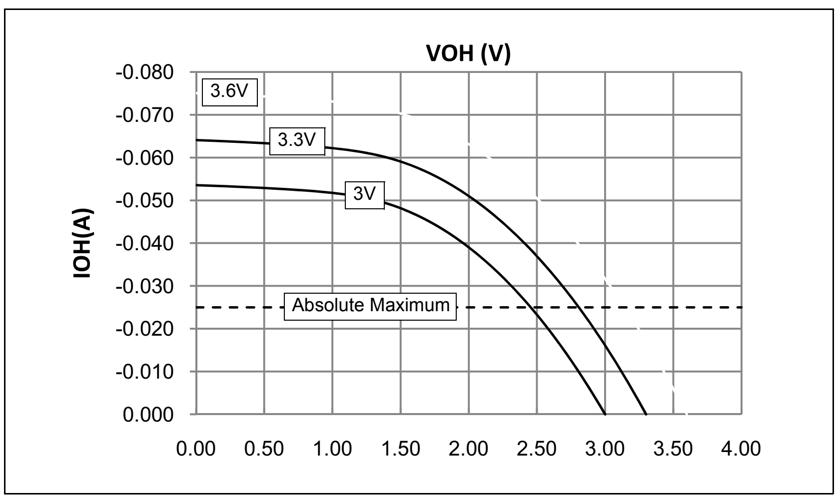



**FIGURE 27-3: VOL – 4x DRIVER PINS**

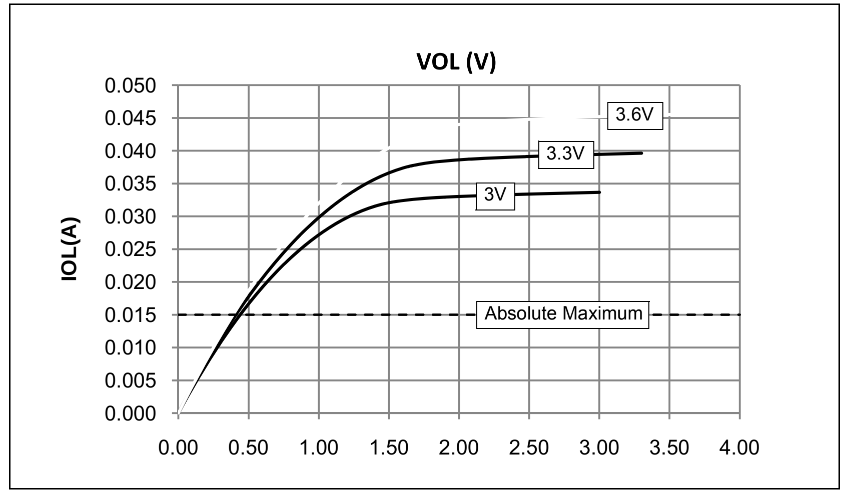
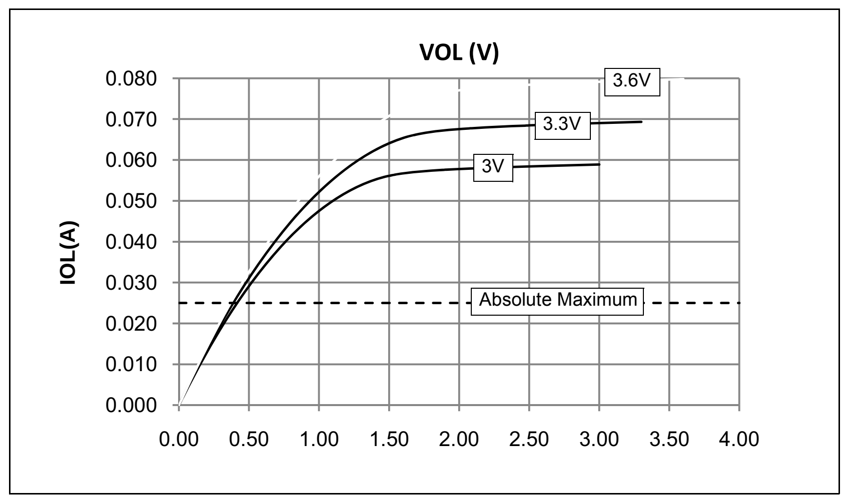


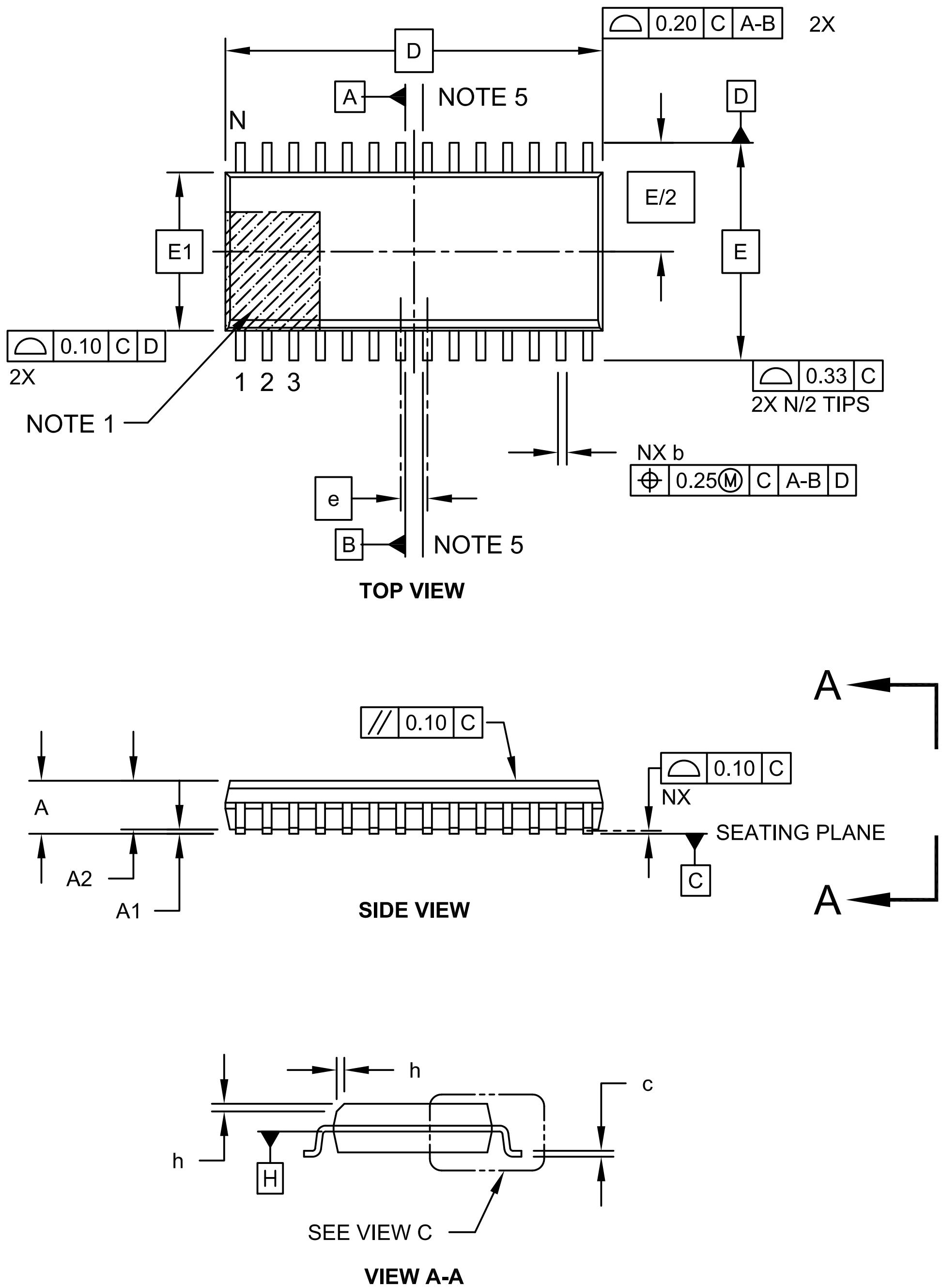

**FIGURE 27-4: VOL – 8x DRIVER PINS**



## 28.2 Package Details

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

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




Microchip Technology Drawing C04-052C Sheet 1 of 2

## Revision C (October 2015)

Updates Note 2 in Table 1-1.

Updates Figure 2-5.

Inserts new **Section 4.2 "Unique Device Identifier (UDID)"** and adds Table 4-1. Subsequent tables were renumbered accordingly. Updates Table 4-3 (which was Table 4-2), Table 4-5 (which was Table 4-4), Table 4-10 (which was Table 4-9), Table 4-11 (which was Table 4-10), Table 4-21 (which was Table 4-20), Table 4-32 (which was Table 4-31), Table 4-36 (which was Table 4-35) and Table 4-37 (which was Table 4-36). Updates **Section 4.8.1 "Bit-Reversed Addressing Implementation"** (which was Section 4.7.1).

Updates Register 9-1.

Updates Figure 12-2 and Register 12-2.

Updates Register 13-1.

Updates Note 1 in **Section 14.0 "Output Compare"**.

Updates Register 15-1, Register 15-6, Register 15-20 and Register 15-22.

Updates Figure 17-1.

Updates Register 18-2.

Updates Figure 19-2 and Figure 19-3. Updates Register 19-1, Register 19-2, Register 19-3, Register 19-4, Register 19-26 and Register 19-33. Adds Register 19-27.

Updates Figure 21-2.

Updates **Section 23.6.2 "Sleep and Idle Modes"**.

Updates Table 26-8, Table 26-11, Table 26-29. Adds new Table 26-42. Subsequent tables were renumbered accordingly. Updates Table 26-43 (which was Table 26-42), Table 26-46 (which was Table 26-45) and Table 26-48 (which was Table 26-47).

Updated diagrams in **Section 28.0 "Packaging Information"**.

Updates the Product Identification System section.

Other minor typographic corrections throughout the document.

## Revision D (May 2017)

Updates Pin 14 Function on page 3, updates Pin 11 Function on page 4, updates Pin 41 Function on page 5, updates Pin 41 Function on page 6, updates Pin 45 Function on page 7 and updates Pin 43 Function on page 8.

Updates Table 1-1, Table 4-8, Table 4-9, Table 4-10, Table 4-11, Table 4-12, Table 4-16, Table 26-4, Table 26-40, Table 26-43 and Table 26-45.

Updates Register 5-1, Register 8-4, Register 15-22, Register 19-5, Register 19-6, Register 19-26, Register 19-27, Register 19-28, Register 19-29 and Register 19-30.

Updates Figure 20-2, Figure 26-20 and Figure 26-22.

Adds 48-Lead Thin Quad Flatpack (Y8) - 7x7x1.0 mm Body TQFP drawings to **Section 28.0 "Packaging Information"** section.

Updates **Section 20.6 "Hysteresis"**

# INDEX