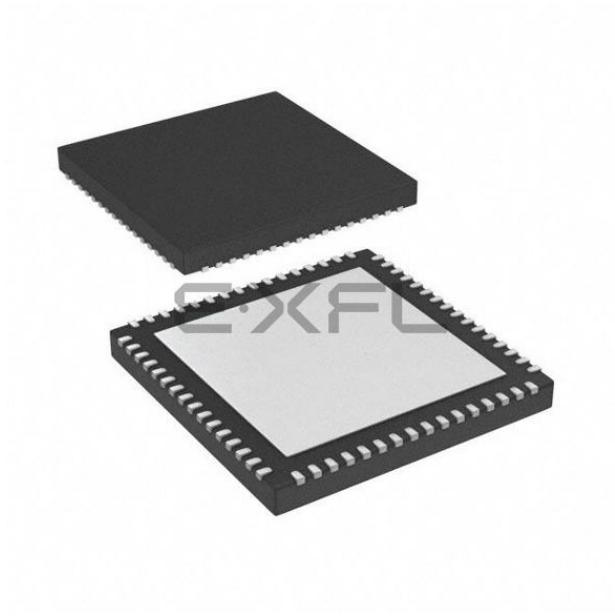


**Welcome to E-XFL.COM**

## 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 | PIC |
| Core Size | 16-Bit |
| Speed | 32MHz |
| Connectivity | I²C, IrDA, LINbus, PMP, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT |
| Number of I/O | 53 |
| Program Memory Size | 256KB (85.5K x 24) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 32K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 3.6V |
| Data Converters | A/D 16x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-VFQFN Exposed Pad |
| Supplier Device Package | 64-QFN (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256ga606-i-mr |

## 7.0 RESETS

| | |
|---|---|
| **Note:** | This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Reset"** (DS39712), which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM. |

The Reset module combines all Reset sources and controls the device Master Reset Signal, $\overline{\text{SYSRST}}$. The following is a list of device Reset sources:

- POR: Power-on Reset
- $\overline{\text{MCLR}}$: Master Clear Pin Reset
- SWR: RESET Instruction
- WDT: Watchdog Timer Reset
- BOR: Brown-out Reset
- CM: Configuration Mismatch Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode Reset
- UWR: Uninitialized W Register Reset

A simplified block diagram of the Reset module is shown in Figure 7-1.

Any active source of Reset will make the $\overline{\text{SYSRST}}$ signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state. Most registers are unaffected by a Reset; their status is unknown on POR and unchanged by all other Resets.

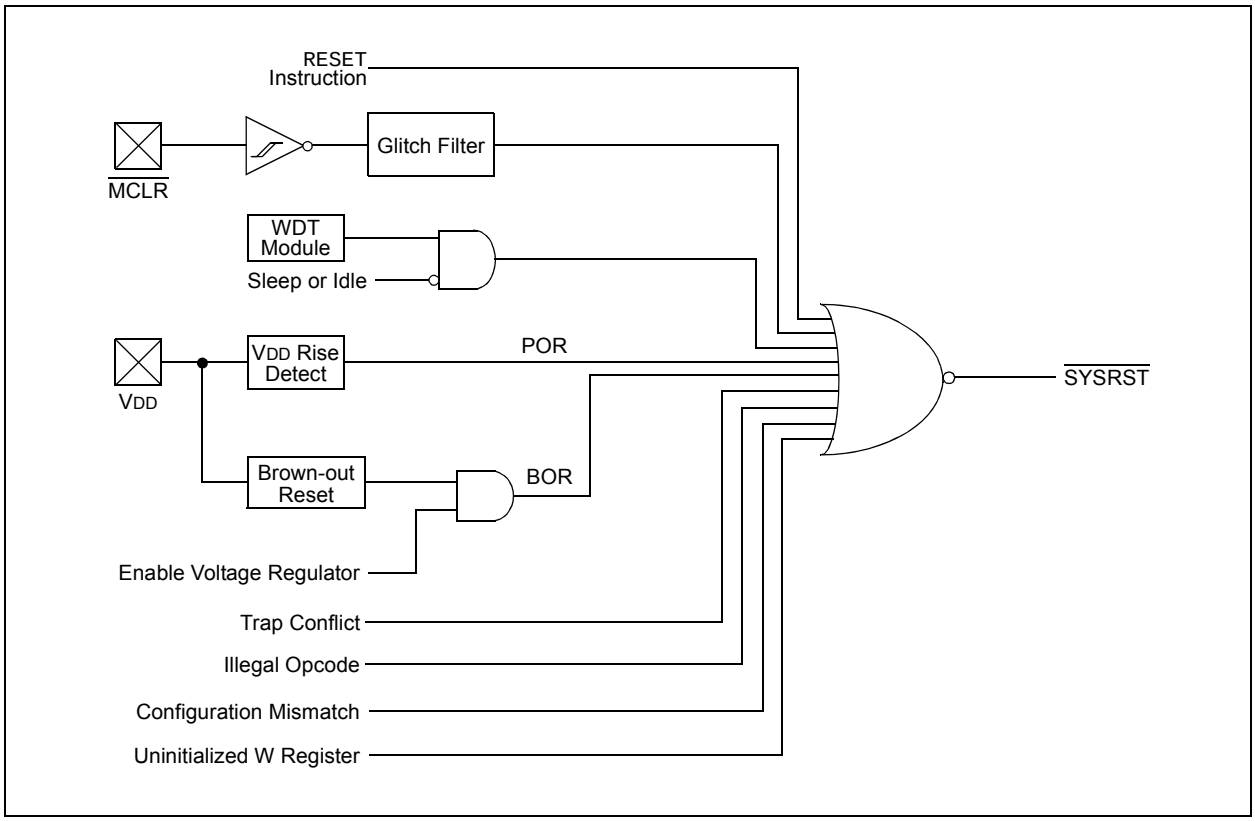
| | |
|---|---|
| **Note:** | Refer to the specific peripheral or CPU section of this manual for register Reset states. |

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 7-1). A POR will clear all bits, except for the BOR and POR (RCON<1:0>) bits, which are set. The user may set or clear any bit at any time during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software will not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer and device power-saving states. The function of these bits is discussed in other sections of this data sheet.

| | |
|---|---|
| **Note:** | The status bits in the RCON register should be cleared after they are read so that the next RCON register values after a device Reset will be meaningful. |

**FIGURE 7-1: RESET SYSTEM BLOCK DIAGRAM**



**REGISTER 10-5: PMD5: PERIPHERAL MODULE DISABLE REGISTER 5**

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

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | CCP7MD | CCP6MD | CCP5MD | CCP4MD | CCP3MD | CCP2MD | CCP1MD |
| bit 7 | | | | | | | bit 0 |

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

bit 15-7 **Unimplemented:** Read as '0'

bit 6 **CCP7MD:** SCCP7 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 5 **CCP6MD:** SCCP6 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 4 **CCP5MD:** SCCP5 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 3 **CCP4MD:** MCCP4 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 2 **CCP3MD:** MCCP3 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 1 **CCP2MD:** MCCP2 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

bit 0 **CCP1MD:** MCCP1 Module Disable bit
1 = Module is disabled
0 = Module power and clock sources are enabled

**REGISTER 11-40: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP9R5 | RP9R4 | RP9R3 | RP9R2 | RP9R1 | RP9R0 |
| 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 |
|---|---|---|---|---|---|---|---|
| — | — | RP8R5 | RP8R4 | RP8R3 | RP8R2 | RP8R1 | RP8R0 |
| 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 **RP9R<5:0>:** RP9 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP9 (see Table 11-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP8R<5:0>:** RP8 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP8 (see Table 11-4 for peripheral function numbers).

**REGISTER 11-41: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP11R5 | RP11R4 | RP11R3 | RP11R2 | RP11R1 | RP11R0 |
| 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 |
|---|---|---|---|---|---|---|---|
| — | — | RP10R5 | RP10R4 | RP10R3 | RP10R2 | RP10R1 | RP10R0 |
| 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 **RP11R<5:0>:** RP11 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP11 (see Table 11-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP10R<5:0>:** RP10 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP10 (see Table 11-4 for peripheral function numbers).

**REGISTER 14-2: ICxCON2: INPUT CAPTURE x CONTROL REGISTER 2**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | IC32 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0, HS | U-0 | R/W-0 | R/W-1 | R/W-1 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| ICTRIG | TRIGSTAT | — | SYNCSEL4 | SYNCSEL3 | SYNCSEL2 | SYNCSEL1 | SYNCSEL0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | HS = Hardware Settable 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 15-9 **Unimplemented:** Read as '0'

bit 8 **IC32:** Cascade Two Input Capture Modules Enable bit (32-bit operation)
1 = ICx and ICy operate in cascade as a 32-bit module (this bit must be set in both modules)
0 = ICx functions independently as a 16-bit module

bit 7 **ICTRIG:** Input Capture x Sync/Trigger Select bit
1 = Triggers ICx from the source designated by the SYNCSELx bits
0 = Synchronizes ICx with the source designated by the SYNCSELx bits

bit 6 **TRIGSTAT:** Timer Trigger Status bit
1 = Timer source has been triggered and is running (set in hardware, can be set in software)
0 = Timer source has not been triggered and is being held clear

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

**Note 1:** Use these inputs as Trigger sources only and never as Sync sources.
**2:** Never use an Input Capture x module as its own Trigger source by selecting this mode.

**NOTES:**

**REGISTER 17-3: SPIxCON2L: SPIx CONTROL REGISTER 2 LOW**

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

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | WLENGTH<4:0>(1,2) | | | | |
| 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 15-5 **Unimplemented:** Read as '0'

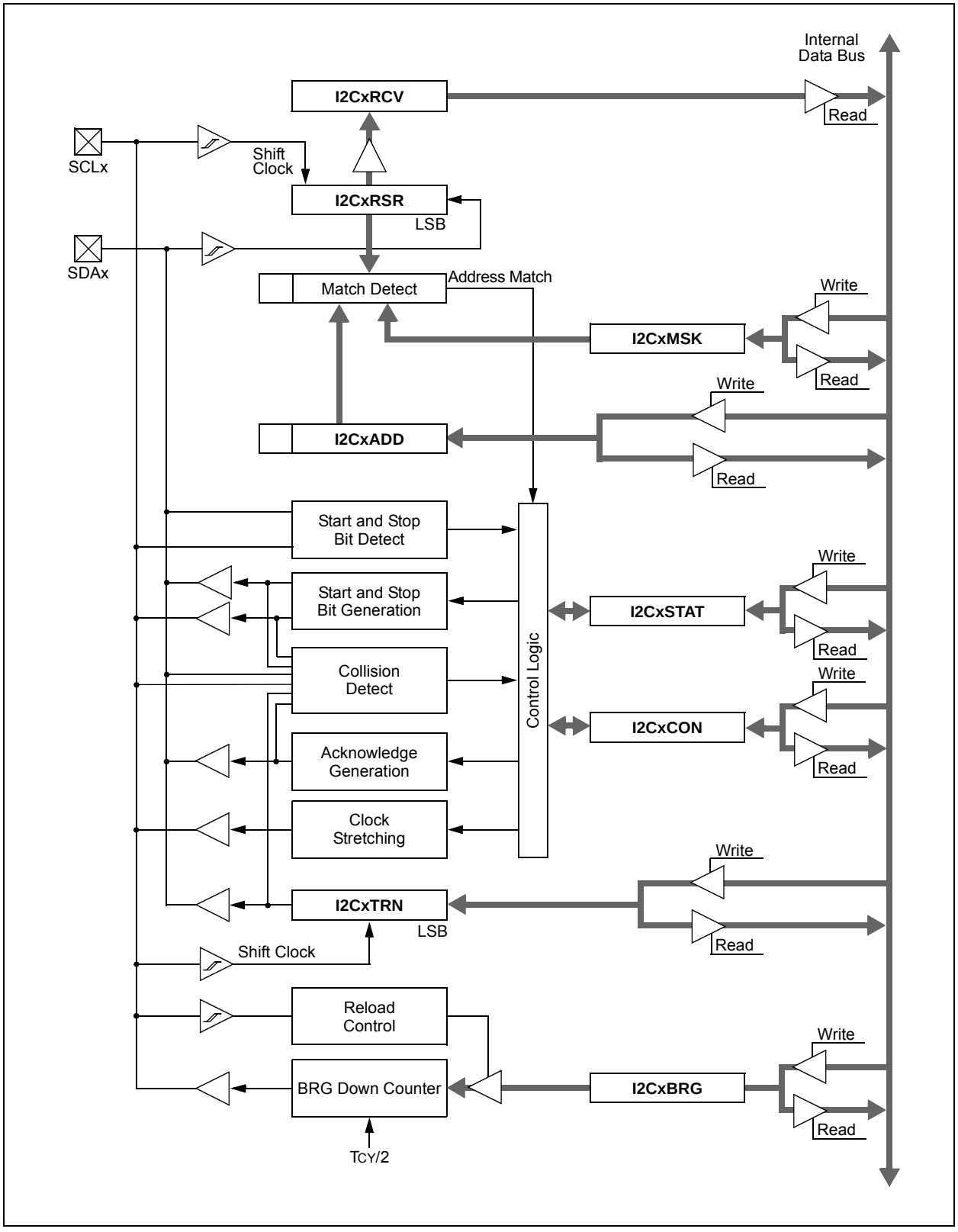
bit 4-0 **WLENGTH<4:0>:** Variable Word Length bits(1,2)

11111 = 32-bit data
11110 = 31-bit data
11101 = 30-bit data
11100 = 29-bit data
11011 = 28-bit data
11010 = 27-bit data
11001 = 26-bit data
11000 = 25-bit data
10111 = 24-bit data
10110 = 23-bit data
10101 = 22-bit data
10100 = 21-bit data
10011 = 20-bit data
10010 = 19-bit data
10001 = 18-bit data
10000 = 17-bit data
01111 = 16-bit data
01110 = 15-bit data
01101 = 14-bit data
01100 = 13-bit data
01011 = 12-bit data
01010 = 11-bit data
01001 = 10-bit data
01000 = 9-bit data
00111 = 8-bit data
00110 = 7-bit data
00101 = 6-bit data
00100 = 5-bit data
00011 = 4-bit data
00010 = 3-bit data
00001 = 2-bit data
00000 = See MODE<32,16> bits in SPIxCON1L<11:10>

**Note 1:** These bits are effective when AUDEN = 0 only.

**2:** Varying the length by changing these bits does not affect the depth of the TX/RX FIFO.

**FIGURE 18-1: I2Cx BLOCK DIAGRAM**



## 19.1 UARTx Baud Rate Generator (BRG)

The UARTx module includes a dedicated, 16-bit Baud Rate Generator. The UxBRG register controls the period of a free-running, 16-bit timer. Equation 19-1 shows the formula for computation of the baud rate when BRGH = 0.

**EQUATION 19-1: UARTx BAUD RATE WITH BRGH = 0(1,2)**

$$\text{Baud Rate} = \frac{F_{CY}}{16 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{16 \bullet \text{Baud Rate}} - 1$$

**Note 1:** FCY denotes the instruction cycle clock frequency (FOSC/2).

**2:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

Example 19-1 shows the calculation of the baud rate error for the following conditions:

- FCY = 4 MHz
- Desired Baud Rate = 9600

The maximum baud rate (BRGH = 0) possible is FCY/16 (for UxBRG = 0) and the minimum baud rate possible is FCY/(16 * 65536).

Equation 19-2 shows the formula for computation of the baud rate when BRGH = 1.

**EQUATION 19-2: UARTx BAUD RATE WITH BRGH = 1(1,2)**

$$\text{Baud Rate} = \frac{F_{CY}}{4 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{4 \bullet \text{Baud Rate}} - 1$$

**Note 1:** FCY denotes the instruction cycle clock frequency.

**2:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 1) possible is FCY/4 (for UxBRG = 0) and the minimum baud rate possible is FCY/(4 * 65536).

Writing a new value to the UxBRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

**EXAMPLE 19-1: BAUD RATE ERROR CALCULATION (BRGH = 0)(1)**

Desired Baud Rate = FCY/(16 (UxBRG + 1))

Solving for UxBRG Value:

UxBRG = ((FCY/Desired Baud Rate)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate) / Desired Baud Rate
= (9615 – 9600)/9600
= 0.16%

**Note 1:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

**REGISTER 20-1: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER PROTOTYPE, USB MODE (BD0STAT THROUGH BD63STAT)**

| R/W-x | R/W-x | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC |
|---|---|---|---|---|---|---|---|
| UOWN | DTS | PID3 | PID2 | PID1 | PID0 | BC9 | BC8 |
| bit 15 | | | | | | | bit 8 |

| R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC | R/W-x, HSC |
|---|---|---|---|---|---|---|---|
| BC7 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 |
| 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 **UOWN:** USB Own bit

1 = The USB module owns the BD and its corresponding buffer; the CPU must not modify the BD or the buffer

bit 14 **DTS:** Data Toggle Packet bit

1 = Data 1 packet
0 = Data 0 packet

bit 13-10 **PID<3:0>:** Packet Identifier bits (written by the USB module)

In Device mode:
Represents the PID of the received token during the last transfer.

In Host mode:
Represents the last returned PID or the transfer status indicator.

bit 9-0 **BC<9:0>:** Byte Count bits

This represents the number of bytes to be transmitted or the maximum number of bytes to be received during a transfer. Upon completion, the byte count is updated by the USB module with the actual number of bytes transmitted or received.

## 22.2 RTCC Module Registers

The RTCC module registers are organized into four categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers
- Timestamp Registers

### 22.2.1 REGISTER MAPPING

Previous RTCC implementations used a Register Pointer to access the RTCC Time and Date registers, as well as the Alarm Time and Date registers. These Registers are now mapped to memory and are individually addressable.

### 22.2.2 WRITE LOCK

To prevent spurious changes to the RTCC Control or RTCC Value registers, the WRLOCK bit (RTCCON1L<11>) must be cleared (‘0’). The POR default state is the WRLOCK bit is ‘0’ and is cleared on any device Reset (POR, BOR, $\overline{\text{MCLR}}$). It is recommended that the WRLOCK bit be set to ‘1’ after the RTCC Value registers are properly initialized, and after the RTCEN bit (RTCCON1L<15>) has been set.

Any attempt to write to the RTCEN bit, the RTCCON2L/H registers or the RTCC Value registers, will be ignored as long as WRLOCK is ‘1’. The RTCC Control, Alarm Value and Timestamp registers can be changed when WRLOCK is ‘1’.

Clearing the WRLOCK bit requires an unlock sequence after it has been written to a ‘1’, writing two bytes consecutively to the NVMKEY register. A sample assembly sequence is shown in Example 22-1. If WRLOCK is already cleared, it can be set to ‘1’ without using the unlock sequence.

| | |
|---|---|
| **Note:** | To avoid accidental writes to the timer, it is recommended that the WRLOCK bit (RTCCON1L<11>) is kept clear at any other time. For the WRLOCK bit to be set, there is only one instruction cycle time window allowed between the 55h/AA sequence and the setting of WRLOCK; therefore, it is recommended that code follow the procedure in Example 22-1. |

### 22.2.3 SELECTING RTCC CLOCK SOURCE

The clock source for the RTCC module can be selected using the CLKSEL<1:0> bits in the RTCCON2L register. When the bits are set to ‘00’, the Secondary Oscillator (SOSC) is used as the reference clock and when the bits are ‘01’, LPRC is used as the reference clock. When CLKSEL<1:0> = 10, the external power-line (50 Hz and 60 Hz) is used as the clock source. When CLKSEL<1:0> = 11, the system clock is used as the clock source.

**EXAMPLE 22-1: SETTING THE WRLOCK BIT**

```
DISI #6                    ;disable interrupts for 6 instructions
MOV  #NVKEY, W1
MOV  #0x55, W2             ; first unlock code
MOV  W2, [W1]              ; write first unlock code
MOV  #0xAA, W3             ; second unlock sequence
MOV  W3, [W1]              ; write second unlock sequence
BCLR RTCCON1L, #WRLOCK     ; clear the WRLOCK bit
```

**REGISTER 22-18: TSADATEH: RTCC TIMESTAMP A DATE REGISTER (HIGH)(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 |
|---|---|---|---|---|---|---|---|
| YRTEN3 | YRTEN2 | YRTEN1 | YRTEN0 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | MTHTEN | MTHONE3 | MTHONE2 | MTHONE1 | MTHONE0 |
| 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 **YRTEN<3:0>:** Binary Coded Decimal Value of Years ‘10’ Digit bits

bit 11-8 **YRONE<3:0>:** Binary Coded Decimal Value of Years ‘1’ Digit bits

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **MTHTEN:** Binary Coded Decimal Value of Months ‘10’ Digit bit
Contains a value from 0 to 1.

bit 3-0 **MTHONE<2:0>:** Binary Coded Decimal Value of Months ‘1’ Digit bits
Contains a value from 0 to 9.

**Note 1:** If TSAEN = 0, bits<15:0> can be used for persistence storage throughout a non-Power-on Reset ($\overline{\text{MCLR}}$, WDT, etc.).

## 25.2 Extended DMA Operations

In addition to the standard features available on all 12-bit A/D Converters, PIC24FJ1024GA610/GB610 family devices implement a limited extension of DMA functionality. This extension adds features that work with the device's DMA Controller to expand the A/D module's data storage abilities beyond the module's built-in buffer.

The Extended DMA functionality is controlled by the DMAEN bit (AD1CON1<11>); setting this bit enables the functionality. The DMABM bit (AD1CON1<12>) configures how the DMA feature operates.

### 25.2.1 EXTENDED BUFFER MODE

Extended Buffer mode (DMABM = 1) maps the A/D Data Buffer registers and data from all channels above 26 into a user-specified area of data RAM. This allows users to read the conversion results of channels above 26, which do not have their own memory-mapped A/D buffer locations, from data memory.

To accomplish this, the DMA must be configured in Peripheral Indirect Addressing mode and the DMA destination address must point to the beginning of the buffer. The DMA count must be set to generate an interrupt after the desired number of conversions.

In Extended Buffer mode, the A/D control bits will function similarly to non-DMA modes. The BUFREGEN bit will still select between FIFO mode and Channel-Aligned mode, but the number of words in the destination FIFO will be determined by the SMPI<4:0> bits in DMA mode. In FIFO mode, the BUFM bit will still split the output FIFO into two sets of 13 results (the SMPIx bits should be set accordingly), and the BUFS bit will still indicate which set of results is being written to and which can be read.

### 25.2.2 PIA MODE

When DMABM = 0, the A/D module is configured to function with the DMA Controller for Peripheral Indirect Addressing (PIA) mode operations. In this mode, the A/D module generates an 11-bit Indirect Address (IA). This is ORed with the destination address in the DMA Controller to define where the A/D conversion data will be stored.

In PIA mode, the buffer space is created as a series of contiguous smaller buffers, one per analog channel. The size of the channel buffer determines how many analog channels can be accommodated. The size of the buffer is selected by the DMABL<2:0> bits (AD1CON4<2:0>). The size options range from a single word per buffer to 128 words. Each channel is allocated a buffer of this size, regardless of whether or not the channel will actually have conversion data.

The IA is created by combining the base address within a channel buffer with three to five bits (depending on the buffer size) to identify the channel. The base address ranges from zero to seven bits wide, depending on the buffer size. The address is right-padded with a '0' in order to maintain address alignment in the Data Space. The concatenated channel and base address bits are then left-padded with zeros, as necessary, to complete the 11-bit IA.

The IA is configured to auto-increment which channel is written in each analog input's sub-buffer during write operations by using the SMPIx bits (AD1CON2<6:2>).

As with PIA operations for any DMA-enabled module, the base destination address in the DMADSTn register must be masked properly to accommodate the IA. Table 25-1 shows how complete addresses are formed. Note that the address masking varies for each buffer size option. Because of masking requirements, some address ranges may not be available for certain buffer sizes. Users should verify that the DMA base address is compatible with the buffer size selected.

Figure 25-2 shows how the parts of the address define the buffer locations in data memory. In this case, the module "allocates" 256 bytes of data RAM (1000h to 1100h) for 32 buffers of four words each. However, this is not a hard allocation and nothing prevents these locations from being used for other purposes. For example, in the current case, if Analog Channels 1, 3 and 8 are being sampled and converted, conversion data will only be written to the channel buffers, starting at 1008h, 1018h and 1040h. The holes in the PIA buffer space can be used for any other purpose. It is the user's responsibility to keep track of buffer locations and prevent data overwrites.

**REGISTER 25-6: AD1CHS: A/D SAMPLE SELECT 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 |
|---|---|---|---|---|---|---|---|
| CH0NB2 | CH0NB1 | CH0NB0 | CH0SB4 | CH0SB3 | CH0SB2 | CH0SB1 | CH0SB0 |
| 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 |
|---|---|---|---|---|---|---|---|
| CH0NA2 | CH0NA1 | CH0NA0 | CH0SA4 | CH0SA3 | CH0SA2 | CH0SA1 | CH0SA0 |
| 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 **CH0NB<2:0>:** Sample B Channel 0 Negative Input Select bits
1xx = Unimplemented
01x = Unimplemented
001 = Unimplemented
000 = AVSS

bit 12-8 **CH0SB<4:0>:** Sample B Channel 0 Positive Input Select bits
11110 = AVDD **(1)**
11101 = AVSS **(1)**
11100 = Band Gap Reference (VBG) **(1)**
11011 = Reserved
11010 = Reserved
11001 = No channels connected (used for CTMU)
11000 = No channels connected (used for CTMU temperature sensor)
10111 = AN23
10110 = AN22
10101 = AN21
10100 = AN20
10011 = AN19
10010 = AN18
10001 = AN17
10000 = AN16
01111 = AN15
01110 = AN14
01101 = AN13
01100 = AN12
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 7-5 **CH0NA<2:0>:** Sample A Channel 0 Negative Input Select bits
Same definitions as for CHONB<2:0>.

bit 4-0 **CH0SA<4:0>:** Sample A Channel 0 Positive Input Select bits
Same definitions as for CHOSB<4:0>.

**Note 1:** These input channels do not have corresponding memory-mapped result buffers.

**REGISTER 28-3: CTMUCON2L: CTMU CONTROL REGISTER 2 LOW**

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

| U-0 | U-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | IRSTEN | — | DSCHS2 | DSCHS1 | DSCHS0 |
| 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 15-5 **Unimplemented:** Read as ‘0’

bit 4 **IRSTEN:** CTMU Current Source Reset Enable bit
1 = Signal selected by DSCHS<2:0> bits or IDISSEN control bit will reset CTMU edge detect logic
0 = CTMU edge detect logic will not occur

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **DSCHS<2:0>:** Discharge Source Select bits
111 = CLC2 out
110 = CLC1 out
101 = Disabled
100 = A/D end of conversion
011 = MCCP3 auxiliary output
010 = MCCP2 auxiliary output
001 = MCCP1 auxiliary output
000 = Disabled

**REGISTER 30-7: FOSC CONFIGURATION REGISTER**

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 23 | | | | | | | bit 16 |

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| R/PO-1 | R/PO-1 | R/PO-1 | R/PO-1 | R/PO-1 | R/PO-1 | R/PO-1 | R/PO-1 |
|---|---|---|---|---|---|---|---|
| FCKSM1 | FCKSM0 | IOL1WAY | PLLSS(1) | SOSCSEL | OSCIOFCN | POSCMD1 | POSCMD0 |
| bit 7 | | | | | | | bit 0 |

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

bit 23-8 **Unimplemented:** Read as ‘1’

bit 7-6 **FCKSM<1:0>:** Clock Switching and Monitor Selection bits
1x = Clock switching and the Fail-Safe Clock Monitor are disabled
01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
00 = Clock switching and the Fail-Safe Clock Monitor are enabled

bit 5 **IOL1WAY:** Peripheral Pin Select Configuration bit
1 = The IOLOCK bit can be set only once (with unlock sequence).
0 = The IOLOCK bit can be set and cleared as needed (with unlock sequence)

bit 4 **PLLSS:** PLL Source Selection Configuration bit(1)
1 = PLL is fed by the Primary Oscillator (EC, XT or HS mode)
0 = PLL is fed by the on-chip Fast RC (FRC) Oscillator

bit 3 **SOSCSEL:** SOSC Selection Configuration bit
1 = Crystal (SOSCI/SOSCO) mode
0 = Digital (SOSCI) mode

bit 2 **OSCIOFCN:** CLKO Enable Configuration bit
1 = CLKO output signal is active on the OSCO pin (when the Primary Oscillator is disabled or configured for EC mode)
0 = CLKO output is disabled

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Configuration bits
11 = Primary Oscillator mode is disabled
10 = HS Oscillator mode is selected (10 MHz-32 MHz)
01 = XT Oscillator mode is selected (1.5 MHz-10 MHz)
00 = External Clock mode is selected

**Note 1:** When the primary clock source is greater than 8 MHz, this bit must be set to ‘0’ to prevent overclocking the PLL.

**REGISTER 30-8: FWDT CONFIGURATION REGISTER (CONTINUED)**

bit 3-0 **WDTPS<3:0>:** Watchdog Timer Postscale Select bits

1111 = 1:32,768
1110 = 1:16,384
1101 = 1:8,192
1100 = 1:4,096
1011 = 1:2,048
1010 = 1:1,024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

**NOTES:**

**FIGURE 33-5: INL vs. CODE (10-BIT MODE)**

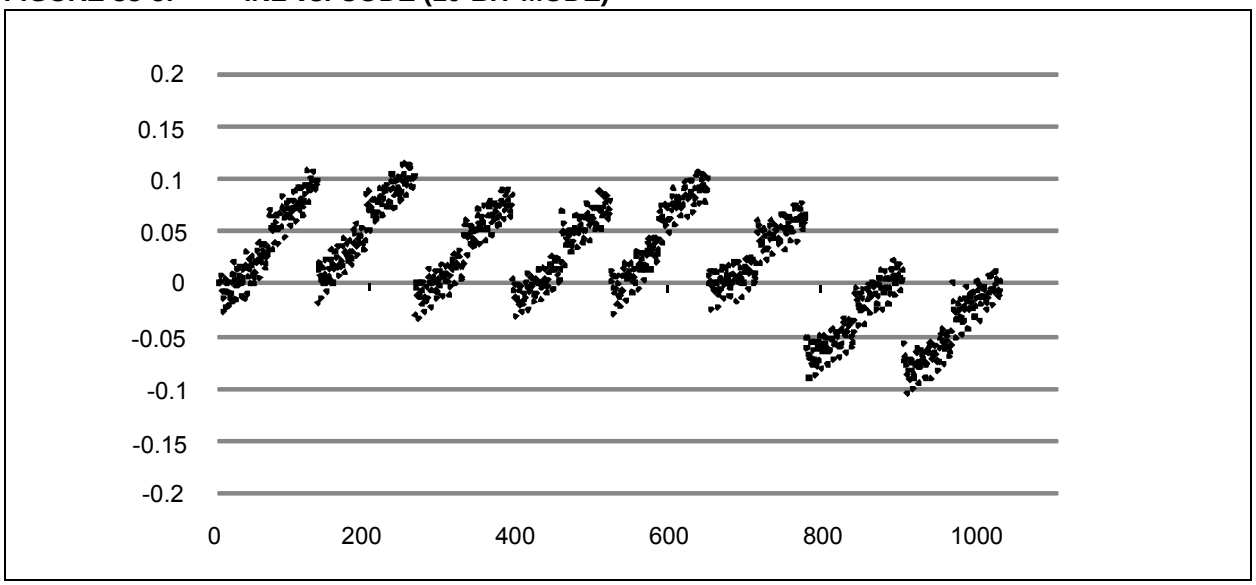
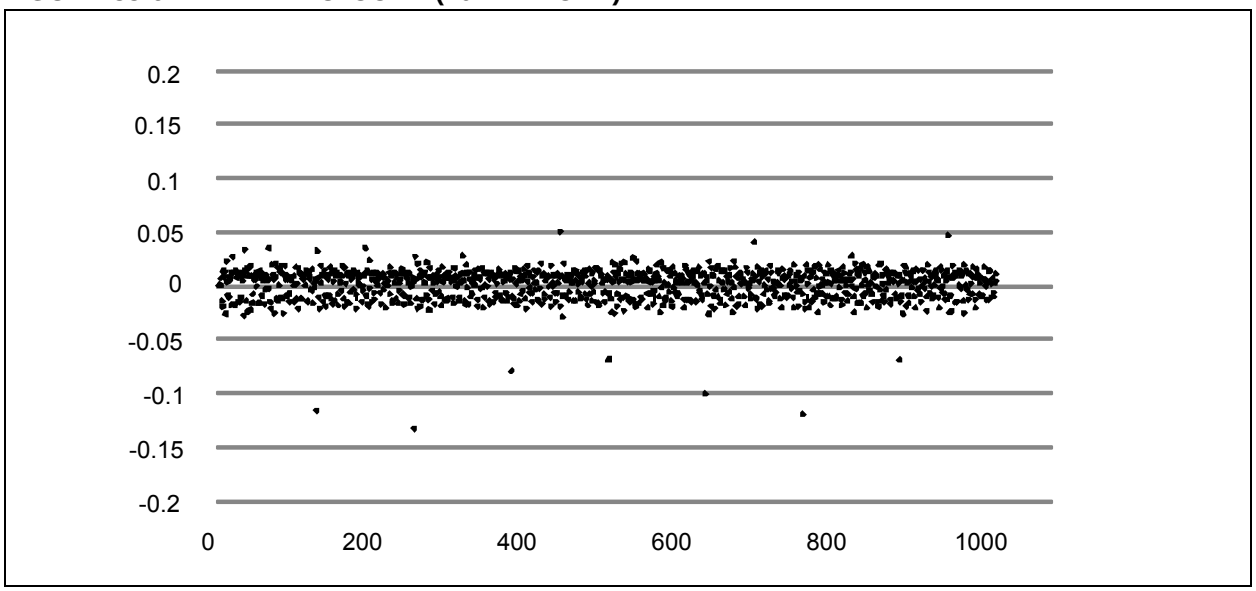
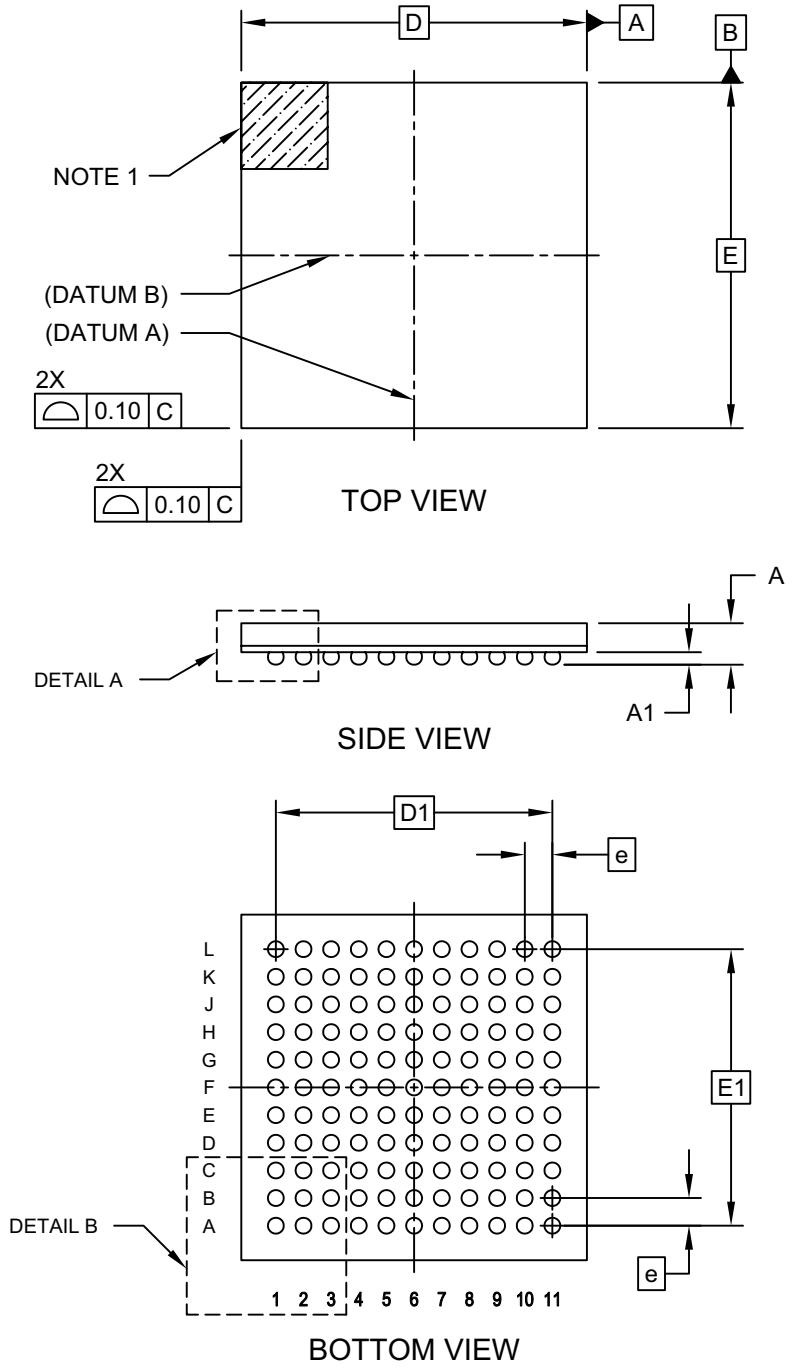



**FIGURE 33-6: DNL vs. CODE (10-BIT MODE)**



**121-Ball Plastic Thin Profile Fine Pitch Ball Grid Array (BG) - 10x10x1.10 mm Body [TFBGA]**

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




Microchip Technology Drawing C04-148 Rev F Sheet 1 of 2

**NOTES:**