Welcome to E-XFL.COM

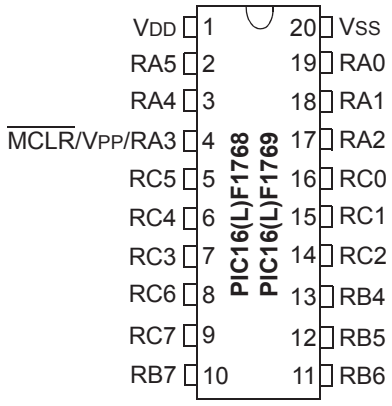
## What is "Embedded - Microcontrollers"?

"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"

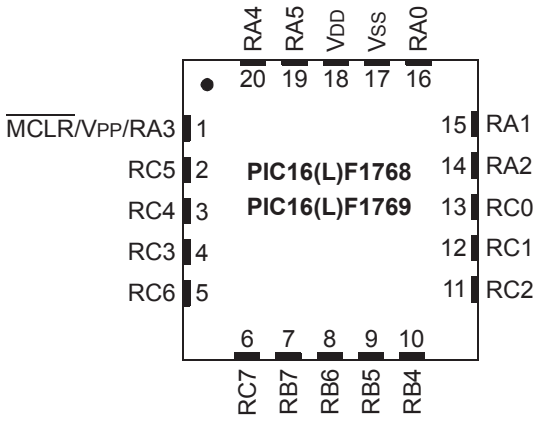
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 18 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 12x10b; D/A 2x5b, 2x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 20-DIP (0.300", 7.62mm) |
| Supplier Device Package | 20-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1768-i-p |

**FIGURE 3: 20-PIN PDIP, SOIC, SSOP**




**Note:** See Table 4 for location of all peripheral functions.

**FIGURE 4: 20-PIN QFN (4x4)**




**Note:** See Table 4 for location of all peripheral functions.

**TABLE 1-2: PIC16(L)F1764/5 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC4/T3G/PRG1R/CLCIN1/CK | RC4 | TTL/ST | CMOS | General purpose I/O. |
| | T3G(1) | TTL/ST | — | Timer3 gate input. |
| | PRG1R(1) | TTL/ST | — | Ramp generator set_rising input. |
| | CLCIN1(1) | TTL/ST | — | CLC Input 1. |
| | CK(1) | TTL/ST | — | EUSART clock input. |
| RC5/T3CKI/PRG1F/CCP1/RX | RC5 | TTL/ST | CMOS | General purpose I/O. |
| | T3CKI(1) | TTL/ST | — | Timer3 clock input. |
| | PRG1F(1) | TTL/ST | — | Ramp generator set_falling input. |
| | CCP1(1) | TTL/ST | — | CCP1 capture input. |
| | RX(1,3) | TTL/ST | — | EUSART receive input. |
| VDD | VDD | Power | — | Positive supply. |
| VSS | VSS | Power | — | Ground reference. |
| OUT(2) | C1OUT | | CMOS | Comparator 1 output. |
| | C2OUT | | CMOS | Comparator 2 output. |
| | CCP1 | | CMOS | Compare/PWM1 output. |
| | MD1OUT | | CMOS | Data Signal Modulator 1 output. |
| | PWM3 | | CMOS | PWM3 output. |
| | PWM5 | | CMOS | PWM5 output. |
| | COG1A | | CMOS | Complementary Output Generator Output A. |
| | COG1B | | CMOS | Complementary Output Generator Output B. |
| | COG1C | | CMOS | Complementary Output Generator Output C. |
| | COG1D | | CMOS | Complementary Output Generator Output D. |
| | SDA(3) | | OD | $I^2C$ data output. |
| | SCK | | CMOS | SPI clock output. |
| | SCL(3) | | OD | $I^2C$ clock output. |
| | SDO | | CMOS | SPI data output. |
| | TX | | CMOS | EUSART asynchronous TX data out. |
| | CK | | CMOS | EUSART synchronous clock out. |
| | DT(3) | | CMOS | EUSART synchronous data output. |
| | CLC1OUT | | CMOS | Configurable Logic Cell 1 output. |
| | CLC2OUT | | CMOS | Configurable Logic Cell 2 output. |
| | CLC3OUT | | CMOS | Configurable Logic Cell 3 output. |

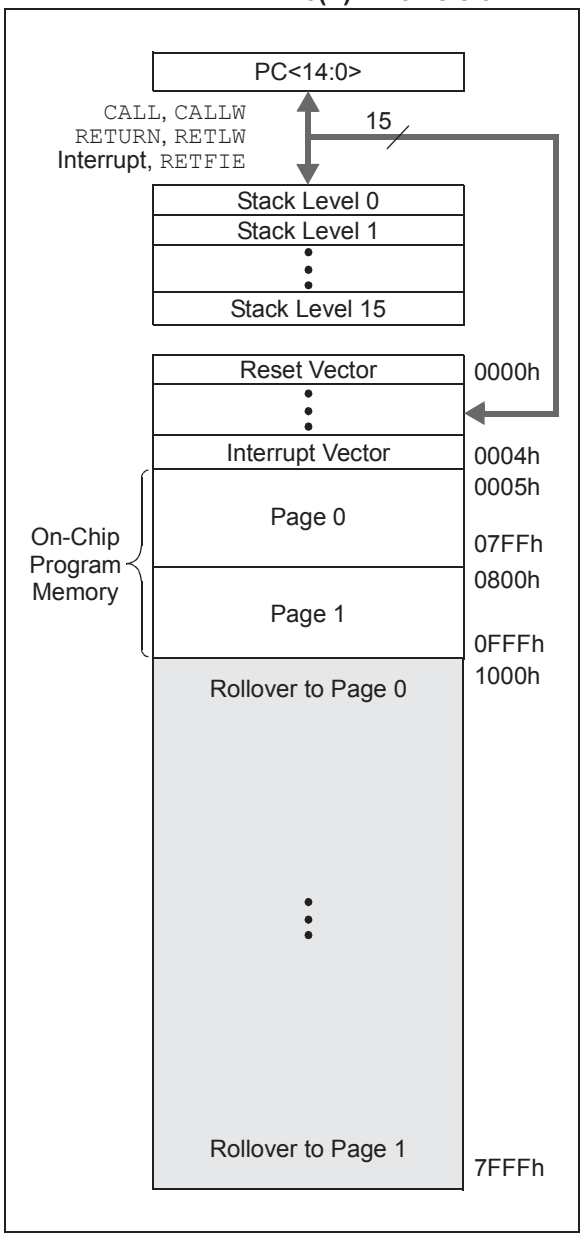
**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels $I^2C$ = Schmitt Trigger input with $I^2C$
HV = High Voltage XTAL = Crystal levels

**Note 1:** Default peripheral input. Alternate pins can be selected as the peripheral input with the PPS Input Selection registers.

**2:** All pin digital outputs default to PORT latch data. Alternate outputs can be selected as peripheral digital outputs with the PPS Output Selection registers.

**3:** These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

**FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1764/5/8/9**




### 3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of `RETLW` instructions. The second method is to set an FSRn to point to the program memory.

#### 3.2.1.1 `RETLW` Instruction

The `RETLW` instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

**EXAMPLE 3-1: `RETLW` INSTRUCTION**

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The `BRW` instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the `BRW` instruction is not available, so the older table read method must be used.

#### 3.2.1.2 Indirect Read with FSRn

The program memory can be accessed as data by setting bit 7 of the FSRnH register and reading the matching INDFn register. The `MOVIW` instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDFn registers. Instructions that access the program memory via the FSRn require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSRn.

The high directive will set bit 7 if a label points to a location in program memory.

**TABLE 3-16: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on All Other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 8** | | | | | | | | | | | |
| 40Ch — 40Dh | — | Unimplemented | | | | | | | | — | — |
| 40Eh | HIDRVC | — | — | HIDC<5:4> | | — | — | — | — | --00 ---- | --00 ---- |
| 40Fh — 412h | — | Unimplemented | | | | | | | | — | — |
| 413h | T4TMR | Holding Register for the 8-Bit TMR4 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 413h | T4PR | TMR4 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 415h | T4CON | ON | CKPS<2:0> | | | OUTPS<3:0> | | | | 0000 0000 | 0000 0000 |
| 416h | T4HLT | PSYNC | CKPOL | CKSYNC | MODE<4:0> | | | | | 0000 0000 | 0000 0000 |
| 417h | T4CLKCON | — | — | — | — | CS<3:0> | | | | ---- 0000 | ---- 0000 |
| 418h | T4RST | — | — | — | — | RSEL<3:0> | | | | ---- 0000 | ---- 0000 |
| 419h | — | Unimplemented | | | | | | | | — | — |
| 41Ah | T6TMR | Holding Register for the 8-Bit TMR4 Register | | | | | | | | 0000 0000 | 0000 0000 |
| 41Bh | T6PR | TMR4 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 41Ch | T6CON | ON | CKPS<2:0> | | | OUTPS<3:0> | | | | 0000 0000 | 0000 0000 |
| 41Dh | T6HLT | PSYNC | CKPOL | CKSYNC | MODE<4:0> | | | | | 0000 0000 | 0000 0000 |
| 41Eh | T6CLKCON | — | — | — | — | CS<3:0> | | | | ---- 0000 | ---- 0000 |
| 41Fh | T6RST | — | — | — | — | RSEL<3:0> | | | | ---- 0000 | ---- 0000 |
| **Bank 9** | | | | | | | | | | | |
| 48Ch to 492h | — | Unimplemented | | | | | | | | — | — |
| 493h | TMR3L | Holding Register for the Least Significant Byte of the 16-Bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 494h | TMR3H | Holding Register for the Most Significant Byte of the 16-Bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 495h | T3CON | CS<1:0> | | CKPS<1:0> | | OSCEN | $\overline{\text{SYNC}}$ | — | ON | 0000 00-0 | uuuu uu-u |
| 496h | T3GCON | GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | GSS<1:0> | | 0000 0x00 | uuuu uxuu |
| 497h to 499h | — | Unimplemented | | | | | | | | — | — |
| 49Ah | TMR5L | Holding Register for the Least Significant Byte of the 16-Bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 49Bh | TMR5H | Holding Register for the Most Significant Byte of the 16-Bit TMR1 Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 49Ch | T5CON | CS<1:0> | | CKPS<1:0> | | OSCEN | $\overline{\text{SYNC}}$ | — | ON | 0000 00-0 | uuuu uu-u |
| 49Dh | T5GCON | GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | GSS<1:0> | | 0000 0x00 | uuuu uxuu |
| 49Eh to 49Fh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown; u = unchanged; q = value depends on condition; - = unimplemented, read as '0'; r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1768/9 only.
**3:** PIC16(L)F1764/5 only.
**4:** Unimplemented on PIC16LF1764/5/8/9.

## 4.6 Device ID and Revision ID

The 14-bit Device ID word is located at 8006h and the 14-bit Revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 10.4 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

## 4.7 Register Definitions: Device and Revision

**REGISTER 4-3: DEVID: DEVICE ID REGISTER**

| R | R | R | R | R | R |
|---|---|---|---|---|---|
| DEV<13:8> | | | | | |
| bit 13 | | | | | bit 8 |

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | '1' = Bit is set | '0' = Bit is cleared |

bit 13-0 **DEV<13:0>**: Device ID bits

| Device | DEVID<13:0> Values |
|---|---|
| PIC16F1764 | 11 0000 1000 0000 (3080h) |
| PIC16F1765 | 11 0000 1000 0001 (3081h) |
| PIC16F1768 | 11 0000 1000 0100 (3084h) |
| PIC16F1769 | 11 0000 1000 0101 (3085h) |
| PIC16LF1764 | 11 0000 1000 0010 (3082h) |
| PIC16LF1765 | 11 0000 1000 0011 (3083h) |
| PIC16LF1768 | 11 0000 1000 0110 (3086h) |
| PIC16LF1769 | 11 0000 1000 0111 (3087h) |

## 5.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS<1:0> bits:

- Default system oscillator determined by the FOSCx bits in the Configuration Words
- Timer1 32 kHz crystal oscillator
- Internal Oscillator Block (INTOSC)

### 5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS<1:0>) bits of the OSCCON register select the system clock source that is used for the CPU and peripherals.

- When SCS<1:0> = 00, the system clock source is determined by the value of the FOSC<2:0> bits in the Configuration Words.
- When SCS<1:0> = 01, the system clock source is the secondary oscillator.
- When SCS<1:0> = 1x, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCSx bits of the OSCCON register are always cleared.

| | |
|---|---|
| **Note:** | Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source. |

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

### 5.3.2 OSCILLATOR START-UP TIMER STATUS (OSTS) BIT

The Oscillator Start-up Timer Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes. The OST does not reflect the status of the secondary oscillator.

### 5.3.3 SECONDARY OSCILLATOR

The secondary oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the SOSCO and SOSCI device pins.

The secondary oscillator is enabled using the OSCEN control bit in the T1CON register. See **Section 22.0 "Timer1/3/5 Module with Gate Control"** for more information about the Timer1 peripheral.

### 5.3.4 SECONDARY OSCILLATOR READY (SOSCR) BIT

The user must ensure that the secondary oscillator is ready to be used before it is selected as a system clock source. The Secondary Oscillator Ready (SOSCR) bit of the OSCSTAT register indicates whether the secondary oscillator is ready to be used. After the SOSCR bit is set, the SCSx bits can be configured to select the secondary oscillator.

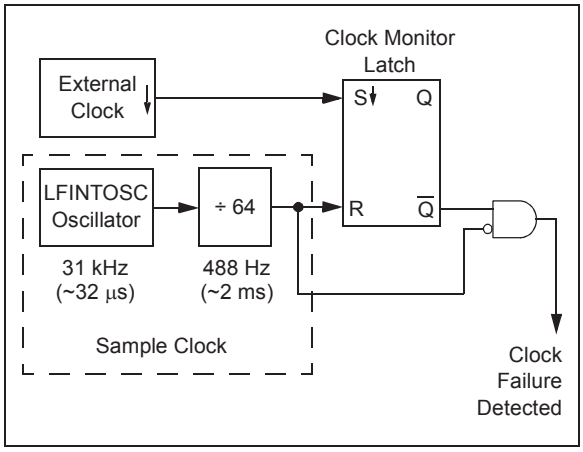
### 5.3.5 CLOCK SWITCH BEFORE SLEEP

When a clock switch from an old clock to a new clock is requested just prior to entering Sleep mode, it is necessary to confirm that the switch is complete before the sleep instruction is executed. Failure to do so may result in an incomplete switch and consequential loss of the system clock altogether. Clock switching is confirmed by monitoring the clock status bits in the OSCSTAT register. Switch confirmation can be accomplished by sensing that the ready bit for the new clock is set or the ready bit for the old clock is cleared. For example, when switching between the internal oscillator with the PLL and the internal oscillator without the PLL, monitor the PLLR bit. When PLLR is set, the switch to 32 MHz operation is complete. Conversely, when PLLR is cleared, the switch from 32 MHz operation to the selected internal clock is complete.

## 5.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (LP, XT, HS, EC, Secondary Oscillator and RC).

**FIGURE 5-9: FSCM BLOCK DIAGRAM**




### 5.5.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 5-9. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

### 5.5.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR2 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<3:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

### 5.5.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a `SLEEP` instruction or changing the SCSx bits of the OSCCON register. When the SCSx bits are changed, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSFIF flag will again become set by hardware.

### 5.5.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

| | |
|---|---|
| **Note:** | Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the status bits in the OSCSTAT register to verify the oscillator start-up and that the system clock switchover has successfully completed. |

## 6.12 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON registers are updated to indicate the cause of the Reset. Table 6-3 and Table 6-4 show the Reset conditions of these registers.

**TABLE 6-3: RESET STATUS BITS AND THEIR SIGNIFICANCE**

| STKOVF | STKUNF | $\overline{\text{RWDT}}$ | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Condition |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 | x | 1 | 1 | Power-on Reset |
| 0 | 0 | 1 | 1 | 1 | 0 | x | 0 | x | Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | 1 | 0 | x | x | 0 | Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | u | 1 | 1 | u | 0 | 1 | 1 | Brown-out Reset |
| u | u | 0 | u | u | u | u | 0 | u | WDT Reset |
| u | u | u | u | u | u | u | 0 | 0 | WDT Wake-up from Sleep |
| u | u | u | u | u | u | u | 1 | 0 | Interrupt Wake-up from Sleep |
| u | u | u | 0 | u | u | u | u | u | $\overline{\text{MCLR}}$ Reset during Normal Operation |
| u | u | u | 0 | u | u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |
| u | u | u | u | 0 | u | u | u | u | RESET Instruction Executed |
| 1 | u | u | u | u | u | u | u | u | Stack Overflow Reset (STVREN = 1) |
| u | 1 | u | u | u | u | u | u | u | Stack Underflow Reset (STVREN = 1) |

**TABLE 6-4: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 0000h | ---1 1000 | 00-- 110x |
| $\overline{\text{MCLR}}$ Reset during Normal Operation | 0000h | ---u uuuu | uu-- 0uuu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0000h | ---1 0uuu | uu-- 0uuu |
| WDT Reset | 0000h | ---0 uuuu | uu-- uuuu |
| WDT Wake-up from Sleep | PC + 1 | ---0 0uuu | uu-- uuuu |
| Brown-out Reset | 0000h | ---1 1uuu | 00-- 11u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | ---1 0uuu | uu-- uuuu |
| RESET Instruction Executed | 0000h | ---u uuuu | uu-- u0uu |
| Stack Overflow Reset (STVREN = 1) | 0000h | ---u uuuu | 1u-- uuuu |
| Stack Underflow Reset (STVREN = 1) | 0000h | ---u uuuu | u1-- uuuu |

**Legend:** u = unchanged; x = unknown; - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Interrupt Enable bit (GIE) is set, the return address is pushed onto the stack and the PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

**REGISTER 11-5: WPUA: WEAK PULL-UP PORTA REGISTER**

| U-0 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| — | — | WPUA<5:0>(1,2) | | | | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

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

bit 5-0 **WPUA<5:0>**: Weak Pull-up PORTA Register bits(1,2)
1 = Pull-up is enabled
0 = Pull-up is disabled

**Note 1:** The global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
**2:** The weak pull-up device is automatically disabled if the pin is configured as an output.

**REGISTER 11-6: ODCONA: PORTA OPEN-DRAIN CONTROL REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | ODA<5:4> | | — | ODA<2:0> | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

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

bit 5-4 **ODA<5:4>:** PORTA Open-Drain Enable bits
For RA<5:4> Pins:
1 = Port pins operate as open-drain drive (sink current only)
0 = Port pins operate as standard push-pull drive (source and sink current)

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

bit 2-0 **ODA<2:0>:** PORTA Open-Drain Enable bits
For RA<2:0> Pins:
1 = Port pins operate as open-drain drive (sink current only)
0 = Port pins operate as standard push-pull drive (source and sink current)

### 23.6.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

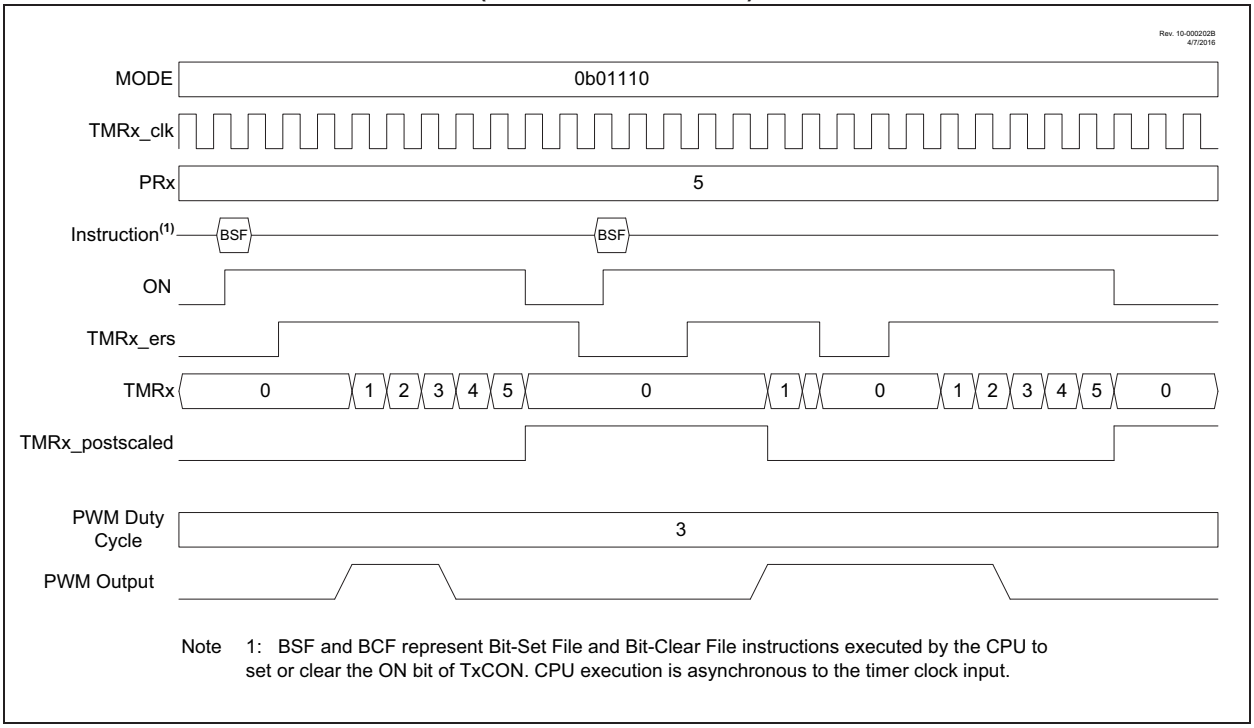
In Level-Triggered One-Shot mode, the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = 01110)
- High Reset level (MODE<4:0> = 01111)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control, a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation, the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse-width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

**FIGURE 23-11: LOW-LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE<4:0> = 01110)**




Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

**REGISTER 25-2: PWMxDCH: PWMx DUTY CYCLE REGISTER HIGH BITS**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| DC<9:2> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

bit 7-0 **DC<9:2>:** PWM Duty Cycle Most Significant bits
These bits are the MSbs of the PWM duty cycle. The two LSbs are found in the PWMxDCL register.

**REGISTER 25-3: PWMxDCL: PWMx DUTY CYCLE REGISTER LOW BITS**

| R/W-x/u | R/W-x/u | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| DC<1:0> | | — | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

bit 7-6 **DC<1:0>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in the PWMxDCH register.

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

**TABLE 25-4: SUMMARY OF REGISTERS ASSOCIATED WITH 10-BIT PWM**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| CCPTMRS | P4TSEL<1:0> | | P3TSEL<1:0> | | C2TSEL<1:0>(1) | | C1TSEL<1:0> | | 264 |
| PWMxCON | EN | — | OUT | POL | MODE<1:0> | | — | — | 265 |
| PWMxDCH | DC<9:2> | | | | | | | | 266 |
| PWMxDCL | DC<1:0> | | — | — | — | — | — | — | 266 |
| RxyPPS | — | — | — | RxyPPS<4:0> | | | | | 154 |
| TxCON | ON | CKPS<2:0> | | | OUTPS<3:0> | | | | 244 |
| TxCLKCON | — | — | — | — | CS<3:0> | | | | 243 |
| TxPR | TMRx Period Register | | | | | | | | 227 |
| TRISA | — | — | TRISA<5:4> | | —(1) | TRISA<2:0> | | | 136 |
| TRISB(2) | TRISB<7:4> | | | | — | — | — | — | 142 |
| TRISC | TRISC7(2) | TRISC6(2) | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 147 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the PWM.
**Note 1:** Unimplemented, read as '1'.
**2:** PIC16(L)F1768/9 only.

**FIGURE 26-4: STANDARD PWMx MODE TIMING DIAGRAM**

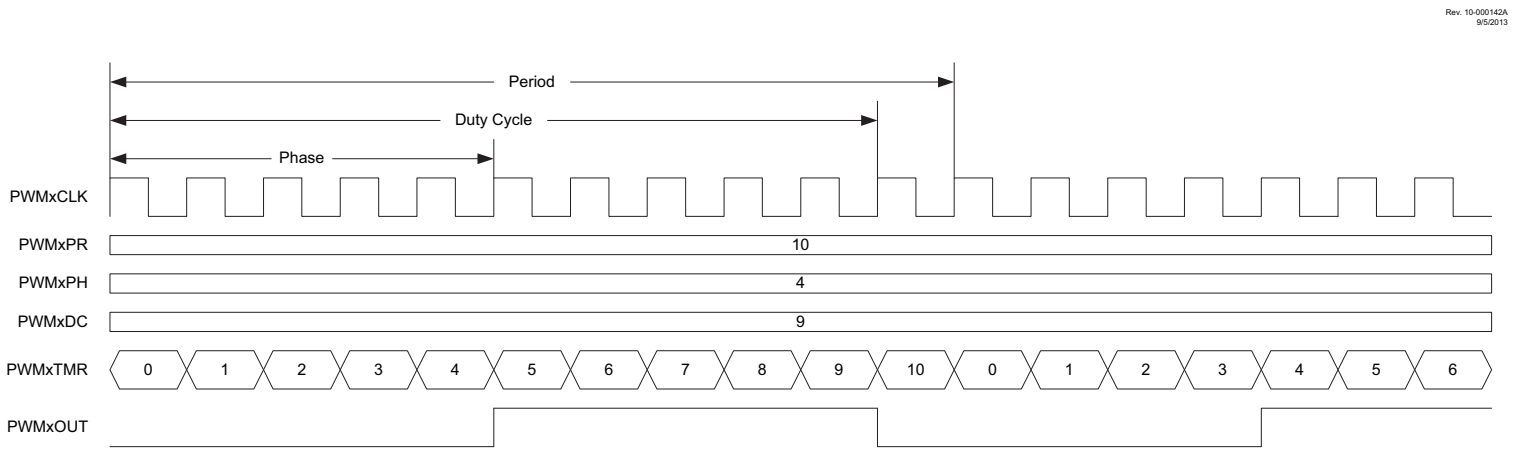



**FIGURE 26-5: SET ON MATCH PWMx MODE TIMING DIAGRAM**

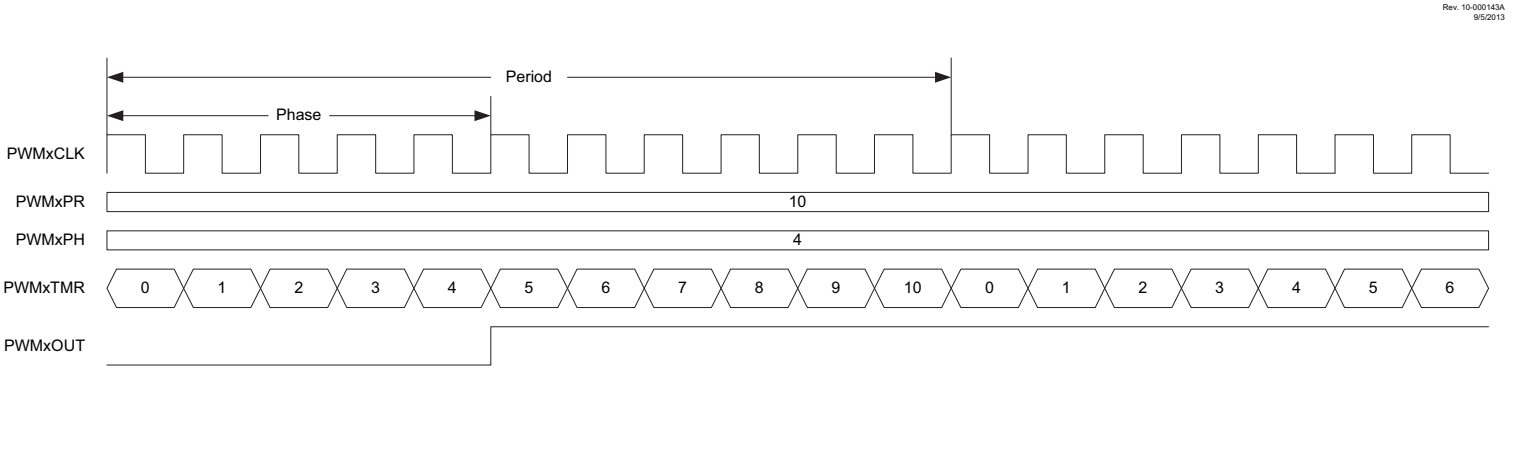


**FIGURE 26-13: OFFSET MATCH ON DECREMENTING TIMER TIMING DIAGRAM**

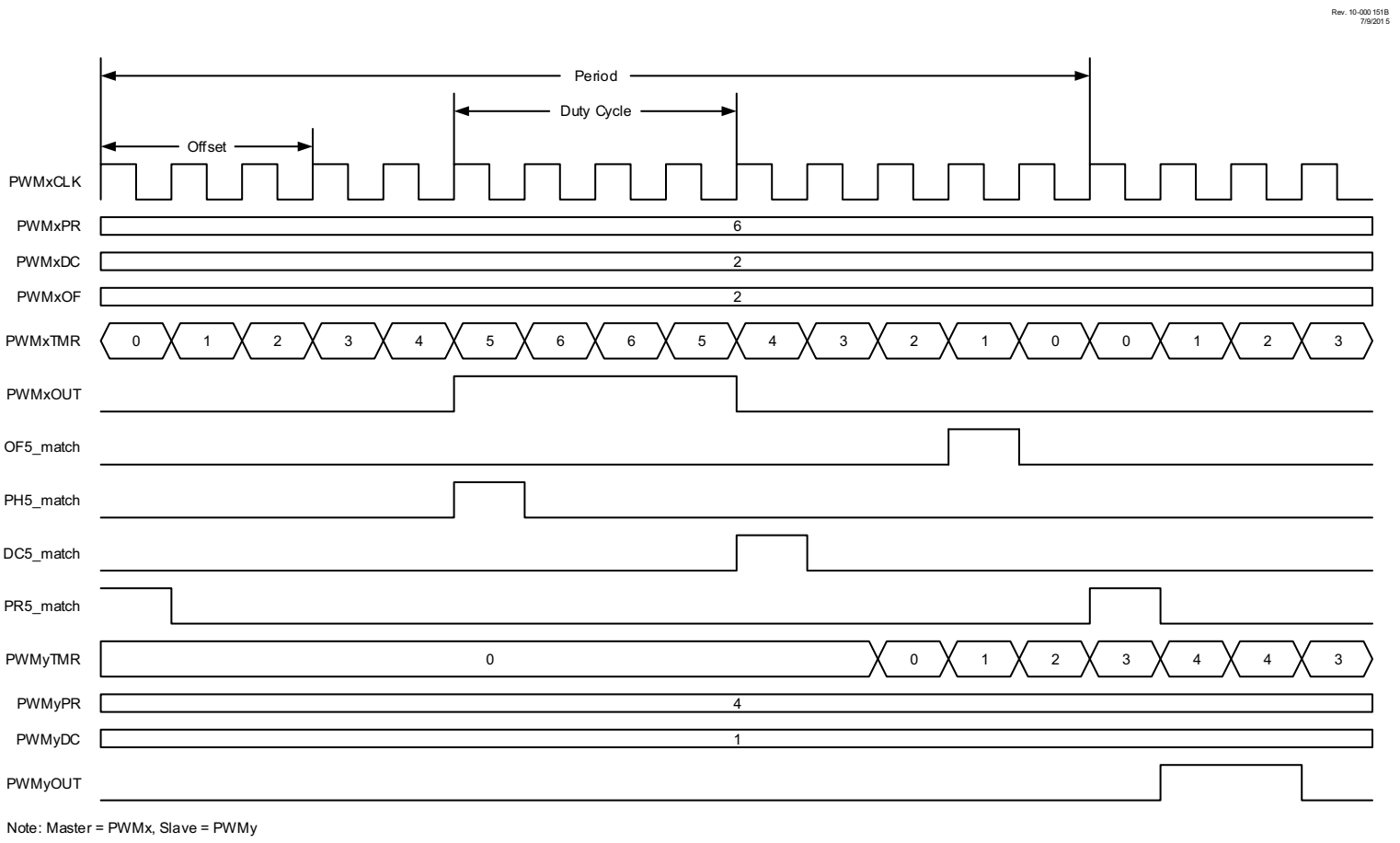


**FIGURE 27-7: COG (RISING/FALLING) INPUT BLOCK**

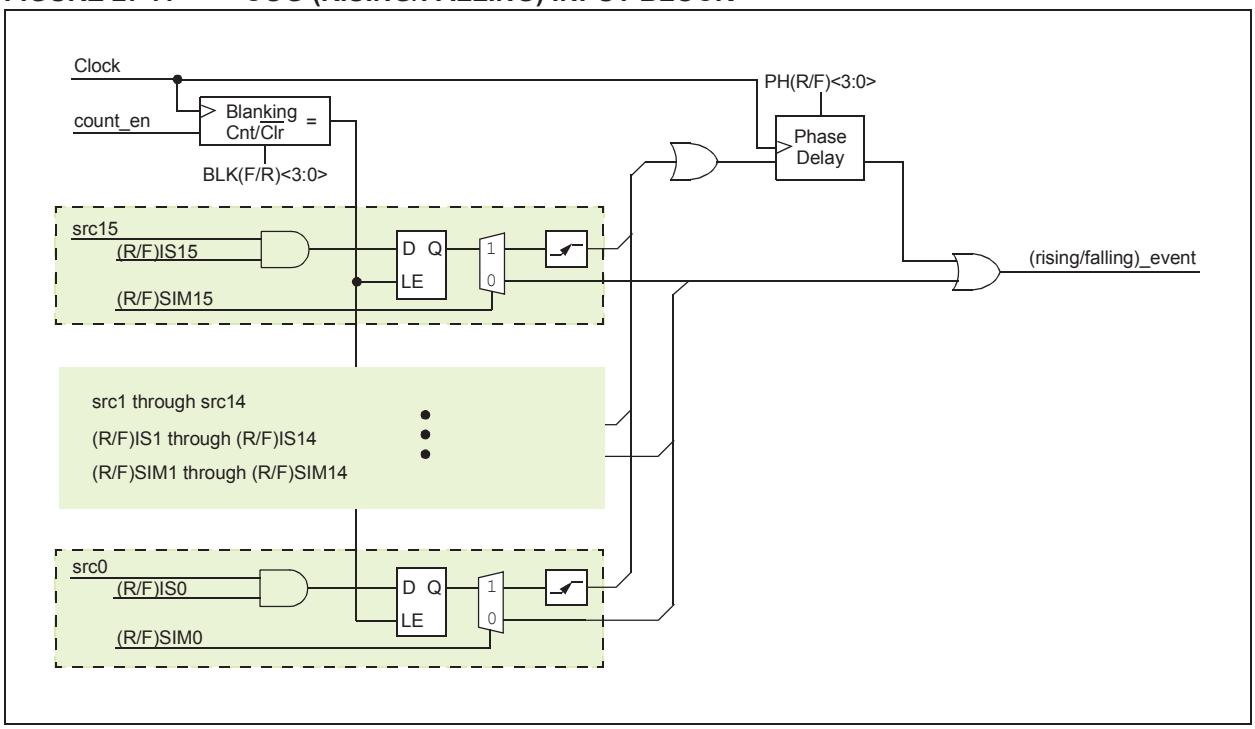



**FIGURE 27-8: COG (RISING/FALLING) DEAD-BAND BLOCK**

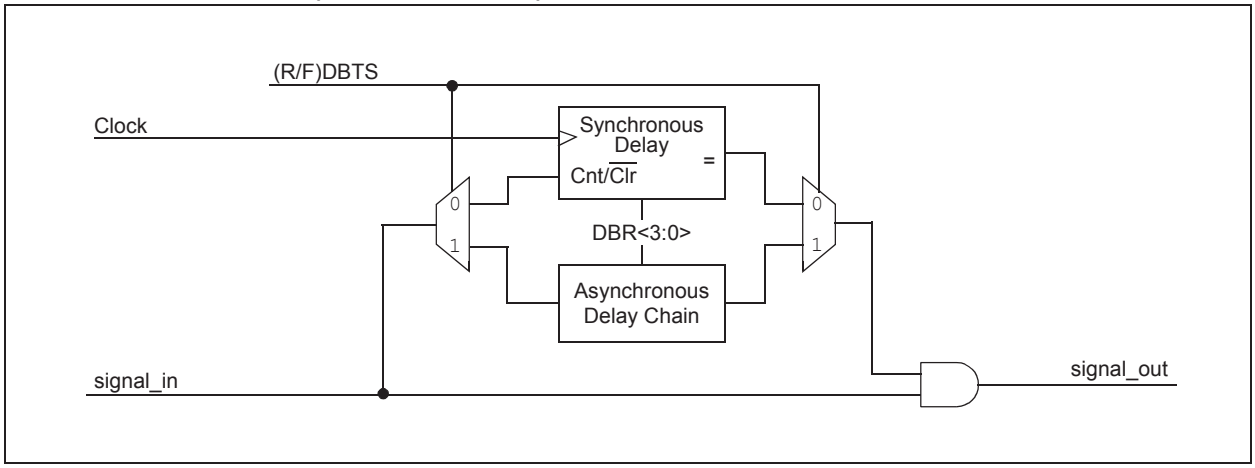
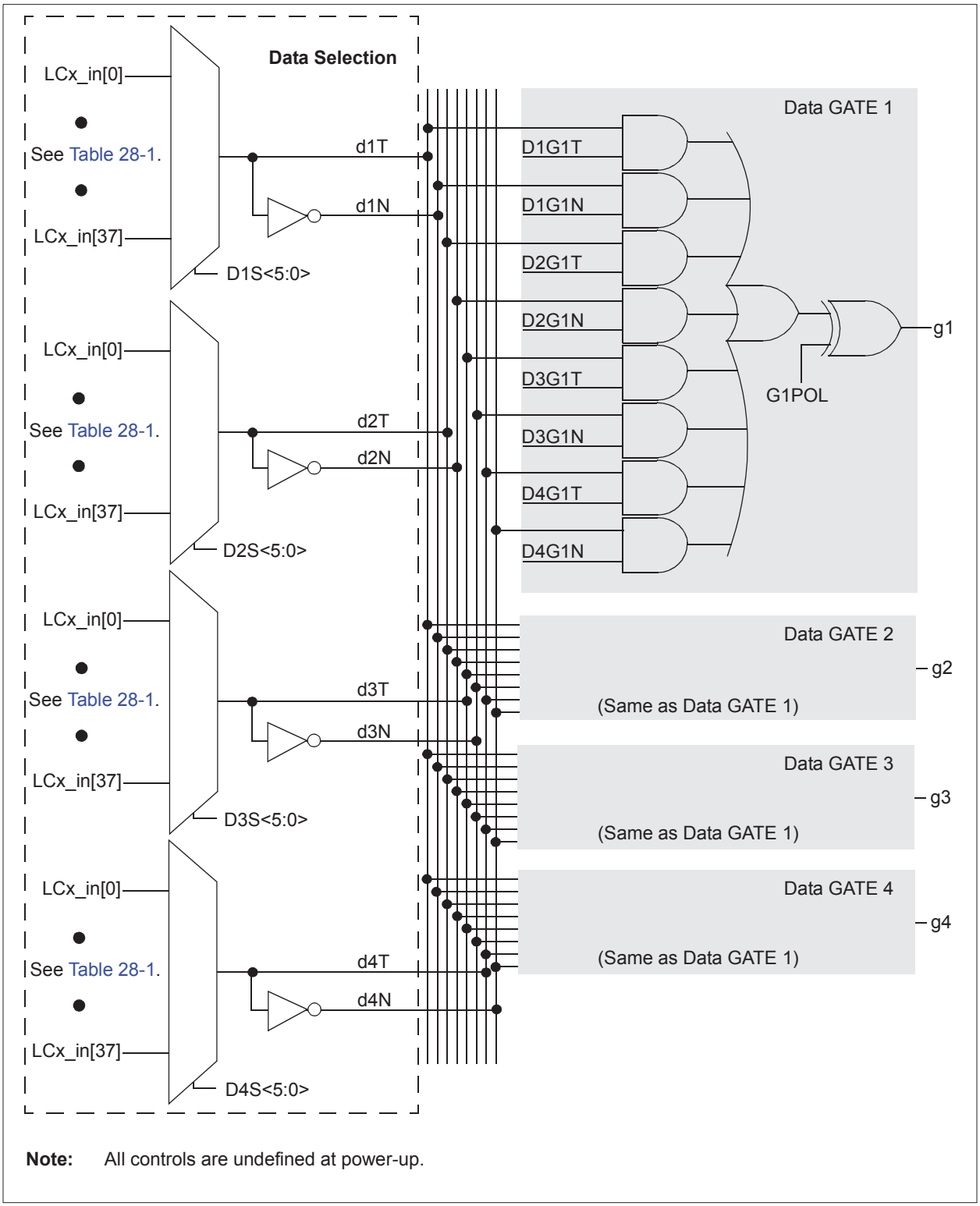


**FIGURE 28-2: INPUT DATA SELECTION AND GATING**




**Note:** All controls are undefined at power-up.

**REGISTER 29-3: OPAxNCHS: OP AMP x NEGATIVE CHANNEL SOURCE SELECT REGISTER**

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

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **NCH<3:0>:** Op Amp Inverting Input Channel Selection bits

1111 = Reserved; do not use
•
•
•
1010 = Reserved; do not use
1001 = Programmable Ramp Generator PRG2_out(1)
1000 = Programmable Ramp Generator PRG1_out
0111 = Reserved. Do not use.
0110 = FVR_Buffer2
0101 = DAC4_out(1)
0100 = DAC3_out
0011 = DAC2_out(1)
0010 = DAC1_out
0001 = OPAxIN1- pin(1)
0000 = OPAxIN0- pin

**Note 1:** PIC16(L)F1768/9 only

## 32.4 I²C MODE OPERATION

All MSSP I²C communication is byte-oriented and shifted out, MSb first. Six SFR registers and two interrupt flags interface the module with the PIC® microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

### 32.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an Acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

### 32.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips® I²C specification.

### 32.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRISx bits.

**Note 1:** Data is tied to output zero when an I²C mode is enabled.

**2:** Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

### 32.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

**TABLE 32-2: I²C BUS TERMS**

| TERM | Description |
|---|---|
| Transmitter | The device which shifts data out onto the bus. |
| Receiver | The device which shifts data in from the bus. |
| Master | The device that initiates a transfer, generates clock signals and terminates a transfer. |
| Slave | The device addressed by the master. |
| Multi-master | A bus with more than one device that can initiate data transfers. |
| Arbitration | Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted. |
| Synchronization | Procedure to synchronize the clocks of two or more devices on the bus. |
| Idle | No master is controlling the bus, and both SDA and SCL lines are high. |
| Active | Any time one or more master devices is controlling the bus. |
| Addressed Slave | Slave device that has received a matching address and is actively being clocked by a master. |
| Matching Address | Address byte that is clocked into a slave that matches the value stored in SSPxADD. |
| Write Request | Slave receives a matching address with R/$\overline{W}$ bit clear and is ready to clock in data. |
| Read Request | Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the slave. This data is the next and all following bytes until a Restart or Stop. |
| Clock Stretching | When a device on the bus hold SCL low to stall communication. |
| Bus Collision | Any time the SDA line is sampled low by the module while it is outputting and expected high state. |

## 32.8 Register Definitions: MSSP Control

### REGISTER 32-1: SSP1STAT: MSSP STATUS REGISTER

| R/W-0/0 | R/W-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | |
| u = Bit is unchanged | x = Bit is unknown | U = Unimplemented bit, read as '0' |
| '1' = Bit is set | '0' = Bit is cleared | -n/n = Value at POR and BOR/Value at all other Resets |

bit 7 **SMP:** SPI Data Input Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

In $I^2C$ Master or Slave mode:
1 = Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)

In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

In $I^2C$ mode only:
1 = Enables input logic so that thresholds are compliant with SMBus specification
0 = Disables SMBus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit ($I^2C$ mode only; this bit is cleared when the MSSP module is disabled, SSPEN is cleared)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit ($I^2C$ mode only; this bit is cleared when the MSSP module is disabled, SSPEN is cleared)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ bit information ($I^2C$ mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{ACK}$ bit.

In $I^2C$ Slave mode:
1 = Read
0 = Write

In $I^2C$ Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

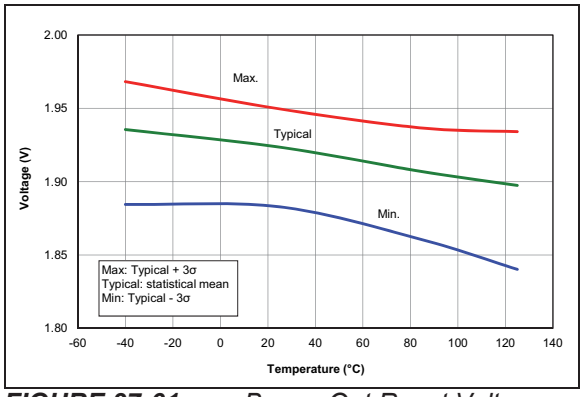



***FIGURE 37-61:*** *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16LF1764/5/8/9 Only.*

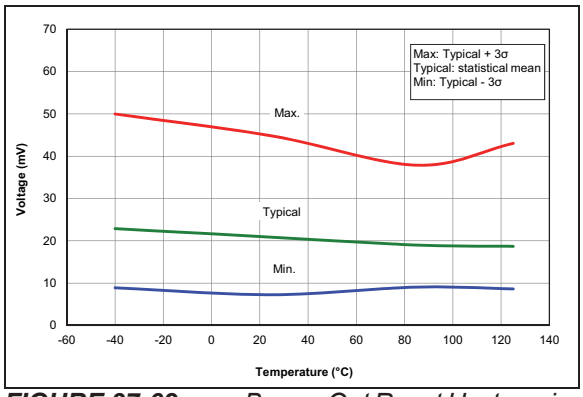



***FIGURE 37-62:*** *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16LF1764/5/8/9 Only.*

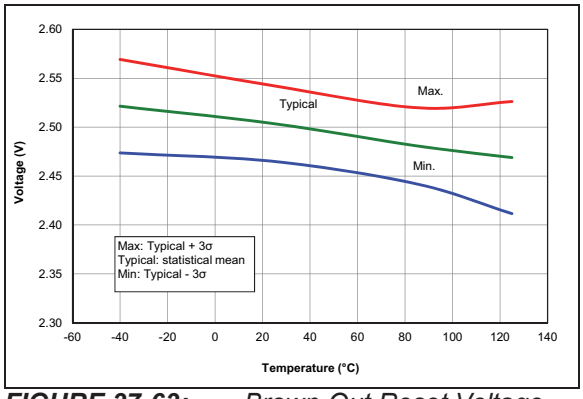



***FIGURE 37-63:*** *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16F1764/5/8/9 Only.*

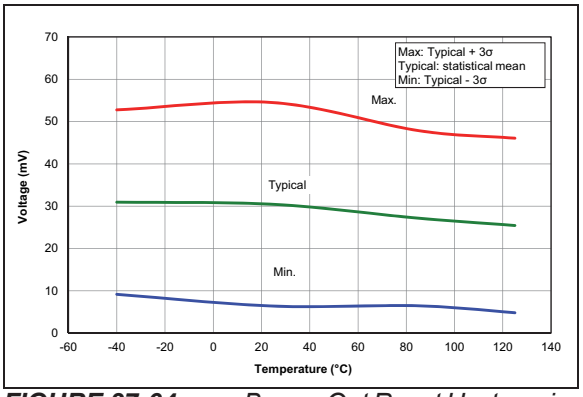



***FIGURE 37-64:*** *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16F1764/5/8/9 Only.*

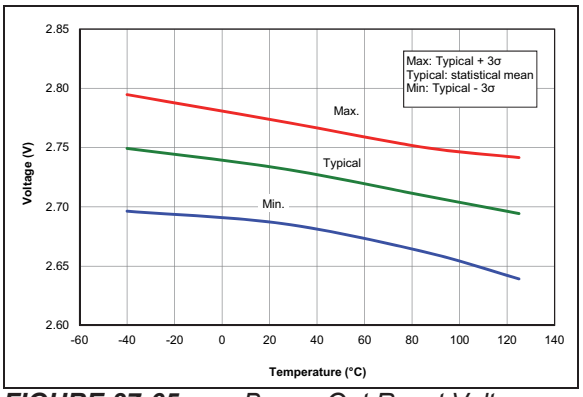



***FIGURE 37-65:*** *Brown-Out Reset Voltage, High Trip Point (BORV = 0).*

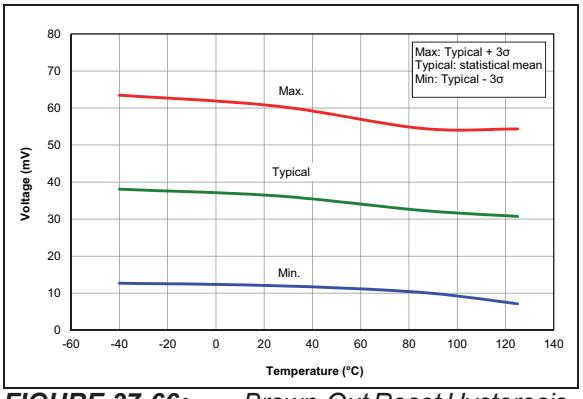



***FIGURE 37-66:*** *Brown-Out Reset Hysteresis, High Trip Point (BORV = 0).*

**Note:** Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

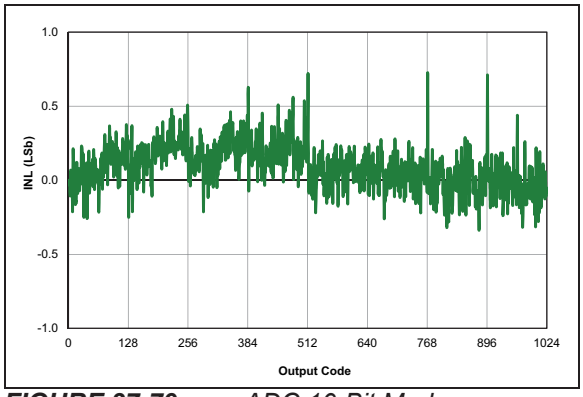



***FIGURE 37-79:*** *ADC 10-Bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

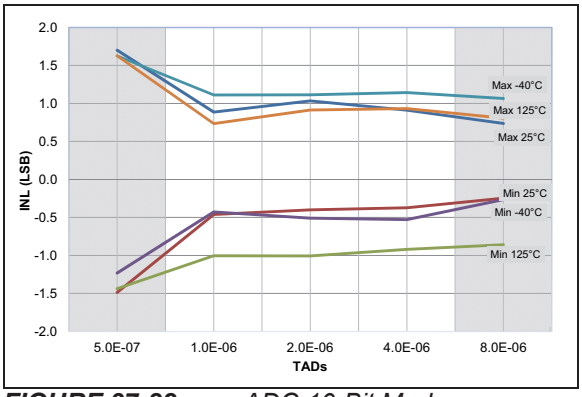



***FIGURE 37-82:*** *ADC 10-Bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.*

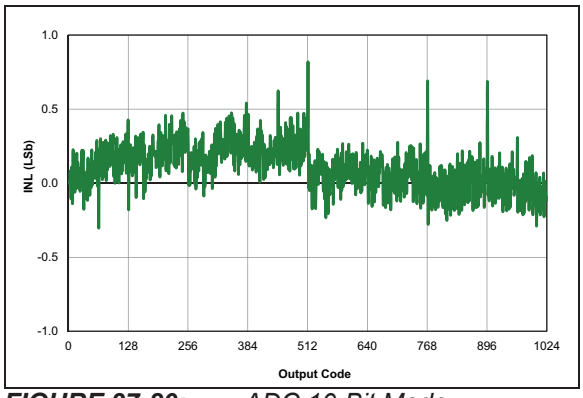



***FIGURE 37-80:*** *ADC 10-Bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*

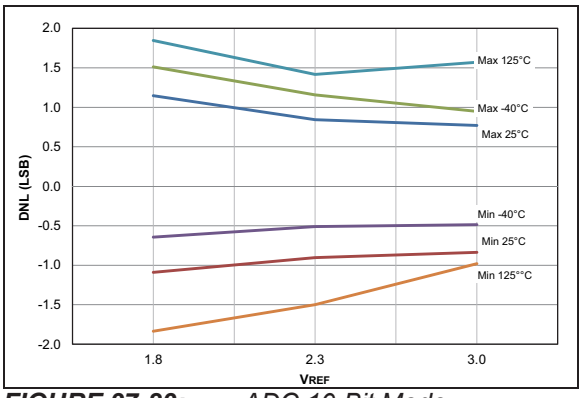



***FIGURE 37-83:*** *ADC 10-Bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS.*

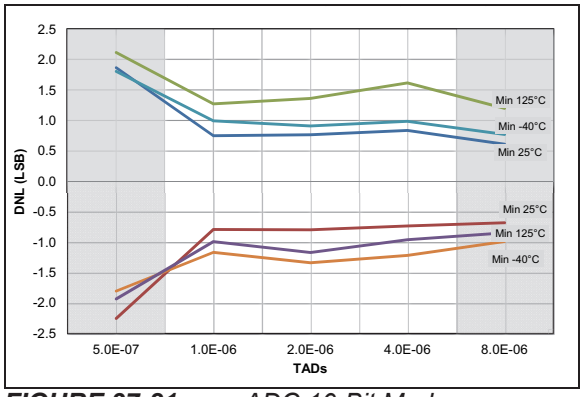



***FIGURE 37-81:*** *ADC 10-Bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.*

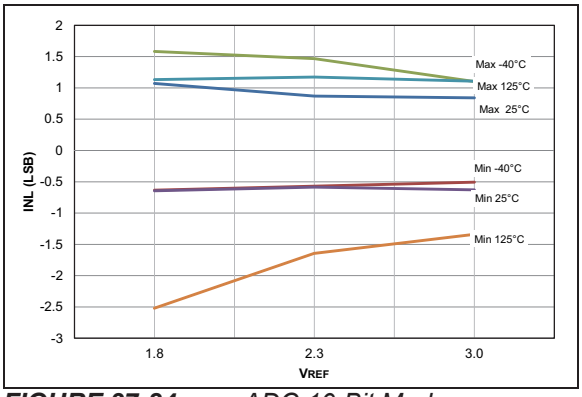



***FIGURE 37-84:*** *ADC 10-Bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS.*