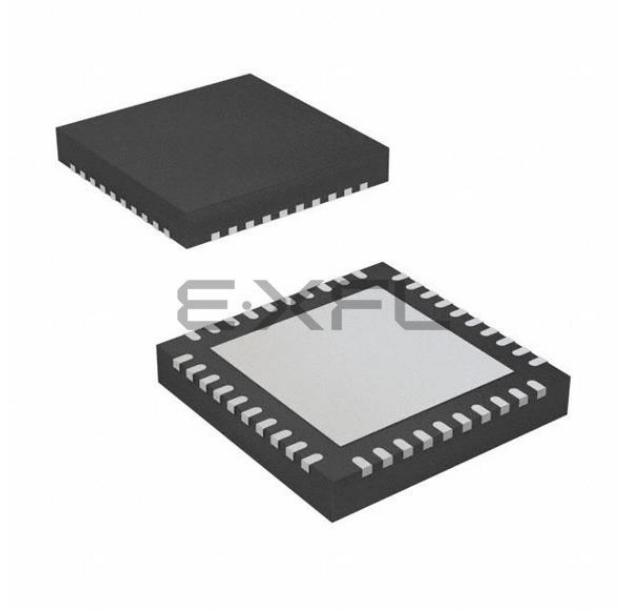


**Welcome to E-XFL.COM**

## 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, PSMC, PWM, WDT |
| Number of I/O | 35 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 14x12b; D/A 1x8b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 40-UFQFN Exposed Pad |
| Supplier Device Package | 40-UQFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1787t-i-mv |

### PIC16(L)F178X Family Types

| Device | Data Sheet Index | Program Memory Flash (words) | Data EEPROM (bytes) | Data SRAM (bytes) | I/O's(2) | 12-bit ADC (ch) | Comparators | Operational Amplifiers | DAC (8/5-bit) | Timers (8/16-bit) | Programmable Switch Mode Controllers (PSMC) | CCP | EUSART | MSSP ($I^2C$™/SPI) | Debug(1) | XLP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC16(L)F1782 | (1) | 2048 | 256 | 256 | 25 | 11 | 3 | 2 | 1/0 | 2/1 | 2 | 2 | 1 | 1 | I | Y |
| PIC16(L)F1783 | (1) | 4096 | 256 | 512 | 25 | 11 | 3 | 2 | 1/0 | 2/1 | 2 | 2 | 1 | 1 | I | Y |
| PIC16(L)F1784 | (2) | 4096 | 256 | 512 | 36 | 15 | 4 | 3 | 1/0 | 2/1 | 3 | 3 | 1 | 1 | I | Y |
| PIC16(L)F1786 | (2) | 8192 | 256 | 1024 | 25 | 11 | 4 | 2 | 1/0 | 2/1 | 3 | 3 | 1 | 1 | I | Y |
| PIC16(L)F1787 | (2) | 8192 | 256 | 1024 | 36 | 15 | 4 | 3 | 1/0 | 2/1 | 3 | 3 | 1 | 1 | I | Y |
| PIC16(L)F1788 | (3) | 16384 | 256 | 2048 | 25 | 11 | 4 | 2 | 1/3 | 2/1 | 4 | 3 | 1 | 1 | I | Y |
| PIC16(L)F1789 | (3) | 16384 | 256 | 2048 | 36 | 15 | 4 | 3 | 1/3 | 2/1 | 4 | 3 | 1 | 1 | I | Y |

**Note 1:** I - Debugging, Integrated on Chip; H - Debugging, available using Debug Header.

**2:** One pin is input-only.

**Data Sheet Index:** (Unshaded devices are described in this document.)

**1:** DS40001579 PIC16(L)F1782/3 Data Sheet, 28-Pin Flash, 8-bit Advanced Analog MCUs.

**2:** DS40001637 PIC16(L)F1784/6/7 Data Sheet, 28/40/44-Pin Flash, 8-bit Advanced Analog MCUs.

**3:** DS40001675 PIC16(L)F1788/9 Data Sheet, 28/40/44-Pin Flash, 8-bit Advanced Analog MCUs.

**Note:** For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.

**TABLE 1: 28-PIN ALLOCATION TABLE (PIC16(L)F1786) (Continued)**

| I/O | 28-Pin SPDIP, SOIC, SSOP | 28-Pin QFN, | ADC | Reference | Comparator | Operation Amplifiers | 8-bit DAC | Timers | PSMC | CCP | EUSART | MSSP | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RB7 | 28 | 25 | — | — | — | — | DAC1OUT2 | — | — | — | RX(1) DT(1) | SCK(1) SCL(1) | IOC | Y | ICSPDAT |
| RC0 | 11 | 8 | — | — | — | — | — | T1OSO T1CKI | PSMC1A | — | — | — | IOC | Y | — |
| RC1 | 12 | 9 | — | — | — | — | — | T1OSI | PSMC1B | CCP2 | — | — | IOC | Y | — |
| RC2 | 13 | 10 | — | — | — | — | — | — | PSMC1C PSMC3B | CCP1 | — | — | IOC | Y | — |
| RC3 | 14 | 11 | — | — | — | — | — | — | PSMC1D | — | — | SCK SCL | IOC | Y | — |
| RC4 | 15 | 12 | — | — | — | — | — | — | PSMC1E | — | — | SDI SDA | IOC | Y | — |
| RC5 | 16 | 13 | — | — | — | — | — | — | PSMC1F PSMC3A | — | — | SDO | IOC | Y | — |
| RC6 | 17 | 14 | — | — | — | — | — | — | PSMC2A | CCP3 | TX CK | — | IOC | Y | — |
| RC7 | 18 | 15 | — | — | C4OUT | — | — | — | PSMC2B | — | RX DT | — | IOC | Y | — |
| RE3 | 1 | 26 | — | — | — | — | — | — | — | — | — | — | IOC | Y | $\overline{\text{MCLR}}$ $V_{PP}$ |
| $V_{DD}$ | 20 | 17 | — | — | — | — | — | — | — | — | — | — | — | — | $V_{DD}$ |
| $V_{SS}$ | 8, 19 | 5, 16 | — | — | — | — | — | — | — | — | — | — | — | — | $V_{SS}$ |

**Note 1:** Alternate pin function selected with the APFCON1 (Register 13-1) and APFCON2 (Register 13-2) registers.

**TABLE 3-12: 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 7** | | | | | | | | | | | |
| 38Ch | INLVLA | Input Type Control for PORTA | | | | | | | | 0000 0000 | 0000 0000 |
| 38Dh | INLVLB | Input Type Control for PORTB | | | | | | | | 0000 0000 | 0000 0000 |
| 38Eh | INLVLC | Input Type Control for PORTC | | | | | | | | 1111 1111 | 1111 1111 |
| 38Fh | INLVLD(3) | Input Type Control for PORTD | | | | | | | | 1111 1111 | 1111 1111 |
| 390h | INLVLE | — | — | — | — | INLVLE3 | INLVLE2(3) | INLVLE1(3) | INLVLE0(3) | ---- 1111 | ---- 1111 |
| 391h | IOCAP | IOCAP<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 392h | IOCAN | IOCAN<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 393h | IOCAF | IOCAF<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 394h | IOCBP | IOCBP<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 395h | IOCBN | IOCBN<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 396h | IOCBF | IOCBF<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 397h | IOCCP | IOCCP<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 398h | IOCCN | IOCCN<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 399h | IOCCF | IOCCF<7:0> | | | | | | | | 0000 0000 | 0000 0000 |
| 39Ah — 39Ch | — | Unimplemented | | | | | | | | — | — |
| 39Dh | IOCEP | — | — | — | — | IOCEP3 | — | — | — | ---- 0--- | ---- 0--- |
| 39Eh | IOCEN | — | — | — | — | IOCEN3 | — | — | — | ---- 0--- | ---- 0--- |
| 39Fh | IOCEF | — | — | — | — | IOCEF3 | — | — | — | ---- 0--- | ---- 0--- |
| **Bank 8-9** | | | | | | | | | | | |
| 40Ch or 41Fh and 48Ch or 49Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 10** | | | | | | | | | | | |
| 50Ch — 510h | — | Unimplemented | | | | | | | | — | — |
| 511h | OPA1CON | OPA1EN | OPA1SP | — | — | — | — | OPA1PCH<1:0> | | 00-- --00 | 00-- --00 |
| 512h | — | Unimplemented | | | | | | | | — | — |
| 513h | OPA2CON | OPA2EN | OPA2SP | — | — | — | — | OPA2PCH<1:0> | | 00-- --00 | 00-- --00 |
| 514h | — | Unimplemented | | | | | | | | — | — |
| 515h | OPA3CON(3) | OPA3EN | OPA3SP | — | — | — | — | OPA3PCH<1:0> | | 00-- --00 | 00-- --00 |
| 51Ah | CLKRCON | CLKREN | CLKROE | CLKRSLR | CLKRDC<1:0> | | CLKRDIV<2:0> | | | 0011 0000 | 0011 0000 |
| 51Bh — 51Fh | — | 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:** These registers can be addressed from any bank.
**2:** Unimplemented, read as '1'.
**3:** PIC16(L)F1784/7 only.
**4:** PIC16F1784/6/7 only.

## 5.13 Power Control (PCON) Register

The Power Control (PCON) register contains flag bits to differentiate between a:

- Power-on Reset ($\overline{POR}$)
- Brown-out Reset ($\overline{BOR}$)
- Reset Instruction Reset ($\overline{RI}$)
- $\overline{MCLR}$ Reset ($\overline{RMCLR}$)
- Watchdog Timer Reset ($\overline{RWDT}$)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

The PCON register bits are shown in Register 5-2.

## 5.14 Register Definitions: Power Control

**REGISTER 5-2: PCON: POWER CONTROL REGISTER**

| R/W/HS-0/q | R/W/HS-0/q | U-0 | R/W/HC-1/q | R/W/HC-1/q | R/W/HC-1/q | R/W/HC-q/u | R/W/HC-q/u |
|---|---|---|---|---|---|---|---|
| STKOVF | STKUNF | — | $\overline{RWDT}$ | $\overline{RMCLR}$ | $\overline{RI}$ | $\overline{POR}$ | $\overline{BOR}$ |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| HC = Bit is cleared by hardware | | HS = Bit is set by hardware |
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -m/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **STKOVF:** Stack Overflow Flag bit
1 = A Stack Overflow occurred
0 = A Stack Overflow has not occurred or cleared by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit
1 = A Stack Underflow occurred
0 = A Stack Underflow has not occurred or cleared by firmware

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **$\overline{RWDT}$:** Watchdog Timer Reset Flag bit
1 = A Watchdog Timer Reset has not occurred or set to ‘1’ by firmware
0 = A Watchdog Timer Reset has occurred (cleared by hardware)

bit 3 **$\overline{RMCLR}$:** $\overline{MCLR}$ Reset Flag bit
1 = A $\overline{MCLR}$ Reset has not occurred or set to ‘1’ by firmware
0 = A $\overline{MCLR}$ Reset has occurred (cleared by hardware)

bit 2 **$\overline{RI}$:** RESET Instruction Flag bit
1 = A RESET instruction has not been executed or set to ‘1’ by firmware
0 = A RESET instruction has been executed (cleared by hardware)

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

## 8.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 or PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 8.5 "Automatic Context Saving"."**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

| | |
|---|---|
| **Note 1:** | Individual interrupt flag bits are set, regardless of the state of any other enable bits. |
| **2:** | All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again. |

## 8.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 8-2 and Figure 8.3 for more details.

## 11.6 Register Definitions: Watchdog Control

**REGISTER 11-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER**

| U-0 | U-0 | R/W-0/0 | R/W-1/1 | R/W-0/0 | R/W-1/1 | R/W-1/1 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | WDTPS<4:0> | | | | | SWDTEN |
| bit 7 | | | | | | | bit 0 |

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

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

bit 5-1 **WDTPS<4:0>:** Watchdog Timer Period Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)

•

•

•

10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 ($2^{23}$) (Interval 256s nominal)
10001 = 1:4194304 ($2^{22}$) (Interval 128s nominal)
10000 = 1:2097152 ($2^{21}$) (Interval 64s nominal)
01111 = 1:1048576 ($2^{20}$) (Interval 32s nominal)
01110 = 1:524288 ($2^{19}$) (Interval 16s nominal)
01101 = 1:262144 ($2^{18}$) (Interval 8s nominal)
01100 = 1:131072 ($2^{17}$) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

**Note 1:** Times are approximate. WDT time is based on 31 kHz LFINTOSC.

## 13.8 Register Definitions: PORTC

**REGISTER 13-19: PORTC: PORTC REGISTER**

| 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 |
|---|---|---|---|---|---|---|---|
| RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **RC<7:0>**: PORTC General Purpose I/O Pin bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

**REGISTER 13-20: TRISC: PORTC TRI-STATE REGISTER**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **TRISC<7:0>:** PORTC Tri-State Control bits
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

**REGISTER 13-21: LATC: PORTC DATA LATCH REGISTER**

| 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 |
|---|---|---|---|---|---|---|---|
| LATC7 | LATC6 | LATC5 | LATC4 | LATC3 | LATC2 | LATC1 | LATC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **LATC<7:0>**: PORTC Output Latch Value bits(1)

**Note 1:** Writes to PORTC are actually written to corresponding LATC register. Reads from PORTC register is return of actual I/O pin values.

## 14.6 Register Definitions: Interrupt-on-Change Control

**REGISTER 14-1: IOCxP: INTERRUPT-ON-CHANGE POSITIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCxP7 | IOCxP6 | IOCxP5 | IOCxP4 | IOCxP3 | IOCxP2 | IOCxP1 | IOCxP0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **IOCxP<7:0>:** Interrupt-on-Change Positive Edge Enable bits(1)

1 = Interrupt-on-Change enabled on the pin for a positive going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

**Note 1:** For IOCEP register, bit 3 (IOCEP3) is the only implemented bit in the register.

**REGISTER 14-2: IOCxN: INTERRUPT-ON-CHANGE NEGATIVE EDGE REGISTER**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| IOCxN7 | IOCxN6 | IOCxN5 | IOCxN4 | IOCxN3 | IOCxN2 | IOCxN1 | IOCxN0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-0 **IOCxN<7:0>:** Interrupt-on-Change Negative Edge Enable bits(1)

1 = Interrupt-on-Change enabled on the pin for a negative going edge. Associated Status bit and interrupt flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

**Note 1:** For IOCEN register, bit 3 (IOCEN3) is the only implemented bit in the register.

## 15.4 Register Definitions: FVR Control

**REGISTER 15-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0/0 | R-q/q | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| FVREN | FVRRDY(1) | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 CDAFVR<1:0>: Comparator and DAC Fixed Voltage Reference Selection bit
11 = Comparator and DAC Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)
10 = Comparator and DAC Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
01 = Comparator and DAC Fixed Voltage Reference Peripheral output is 1x (1.024V)
00 = Comparator and DAC Fixed Voltage Reference Peripheral output is off.

bit 1-0 **ADFVR<1:0>:** ADC Fixed Voltage Reference Selection bit
11 = ADC Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)
10 = ADC Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
01 = ADC Fixed Voltage Reference Peripheral output is 1x (1.024V)
00 = ADC Fixed Voltage Reference Peripheral output is off.

**Note 1:** FVRRDY is always '1' on "F" devices only.
**2:** Fixed Voltage Reference output cannot exceed $V_{DD}$.
**3:** See **Section 16.0 "Temperature Indicator Module"** for additional information.

**TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 162 |

**Legend:** Shaded cells are not used with the Fixed Voltage Reference.

### 24.1.1 PERIOD EVENT

The period event determines the frequency of the active pulse. Period event sources include any combination of the following:

- PSMCxTMR counter match
- PSMC input pin
- sync_C1OUT
- sync_C2OUT
- sync_C3OUT
- sync_C4OUT

Period event sources are selected with the PSMC Period Source (PSMCxPRS) register (Register 24-14).

**Section 24.2.1.2 "16-bit Period Register"** contains details on configuring the PSMCxTMR counter match for synchronous period events.

All period events cause the PSMCxTMR counter to reset on the counting clock edge immediately following the period event. The PSMCxTMR counter resumes counting from zero on the counting clock edge after the period event Reset.

During a period, the rising event and falling event are each permitted to occur only once. Subsequent rising or falling events that may occur within the period are suppressed, thereby preventing output chatter from spurious inputs.

### 24.1.2 RISING EDGE EVENT

The rising edge event determines the start of the active drive period. The rising edge event is also referred to as the phase because two synchronized PSMC peripherals may have different rising edge events relative to the period start, thereby creating a phase relationship between the two PSMC peripheral outputs.

Depending on the PSMC mode, one or more of the PSMC outputs will change in immediate response to the rising edge event. Rising edge event sources include any combination of the following:

- Synchronous:
  - PSMCxTMR time base counter match
- Asynchronous:
  - PSMC input pin
  - sync_C1OUT
  - sync_C2OUT
  - sync_C3OUT
  - sync_C4OUT

Rising edge event sources are selected with the PSMC Phase Source (PSMCxPHS) register (Register 24-12).

For configuring the PSMCxTMR time base counter match for synchronous rising edge events, see **Section 24.2.1.3 "16-bit Phase Register"**.

The first rising edge event in a cycle period is the only one permitted to cause action. All subsequent rising edge events in the same period are suppressed to prevent the PSMC output from chattering in the presence of spurious event inputs. A rising edge event is also suppressed when it occurs after a falling edge event in the same period.

The rising edge event also triggers the start of two other timers when needed: falling edge blanking and dead-band period. For more detail refer to **Section 24.2.8 "Input Blanking"** and **Section 24.4 "Dead-Band Control"**.

When the rising edge event is delayed from the period start, the amount of delay subtracts from the total amount of time available for the drive duty cycle. For example, if the rising edge event is delayed by 10% of the period time, the maximum duty cycle for that period is 90%. A 100% duty cycle is still possible in this example, but duty cycles from 90% to 100% are not possible.

### 24.1.3 FALLING EDGE EVENT

The falling edge event determines the end of the active drive period. The falling edge event is also referred to as the duty cycle because varying the falling edge event, while keeping the rising edge event and period events fixed, varies the active drive duty cycle.

Depending on the PSMC mode, one or more of the PSMC outputs will change in immediate response to the falling edge event. Falling edge event sources include any combination of the following:

- Synchronous:
  - PSMCxTMR time base counter match
- Asynchronous:
  - PSMC input pin
  - sync_C1OUT
  - sync_C2OUT
  - sync_C3OUT
  - sync_C4OUT

Falling edge event sources are selected with PSMC Duty Cycle Source (PSMCxDCS) register (Register 24-13).

For configuring the PSMCxTMR time base counter match for synchronous falling edge events, see **Section 24.2.1.4 "16-bit Duty Cycle Register"**.

The first falling edge event in a cycle period is the only one permitted to cause action. All subsequent falling edge events in the same period are suppressed to prevent the PSMC output from chattering in the presence of spurious event inputs.

A falling edge event suppresses any subsequent rising edges that may occur in the same period. In other words, if an asynchronous falling event input should come late and occur early in the period, following that for which it was intended, the rising edge in that period will be suppressed. This will have a similar effect as pulse skipping.

The falling edge event also triggers the start of two other timers: rising edge blanking and dead-band period. For more detail refer to **Section 24.2.8 "Input Blanking"** and **Section 24.4 "Dead-Band Control"**.

## 24.5 Output Steering

Output steering allows for PWM signals generated by the PSMC module to be placed on different pins under software control. Synchronized steering will hold steering changes until the first period event after the PSMCxLD bit is set. Unsynchronized steering changes will take place immediately.

Output steering is available in the following modes:

- 3-phase PWM
- Single PWM
- Complementary PWM

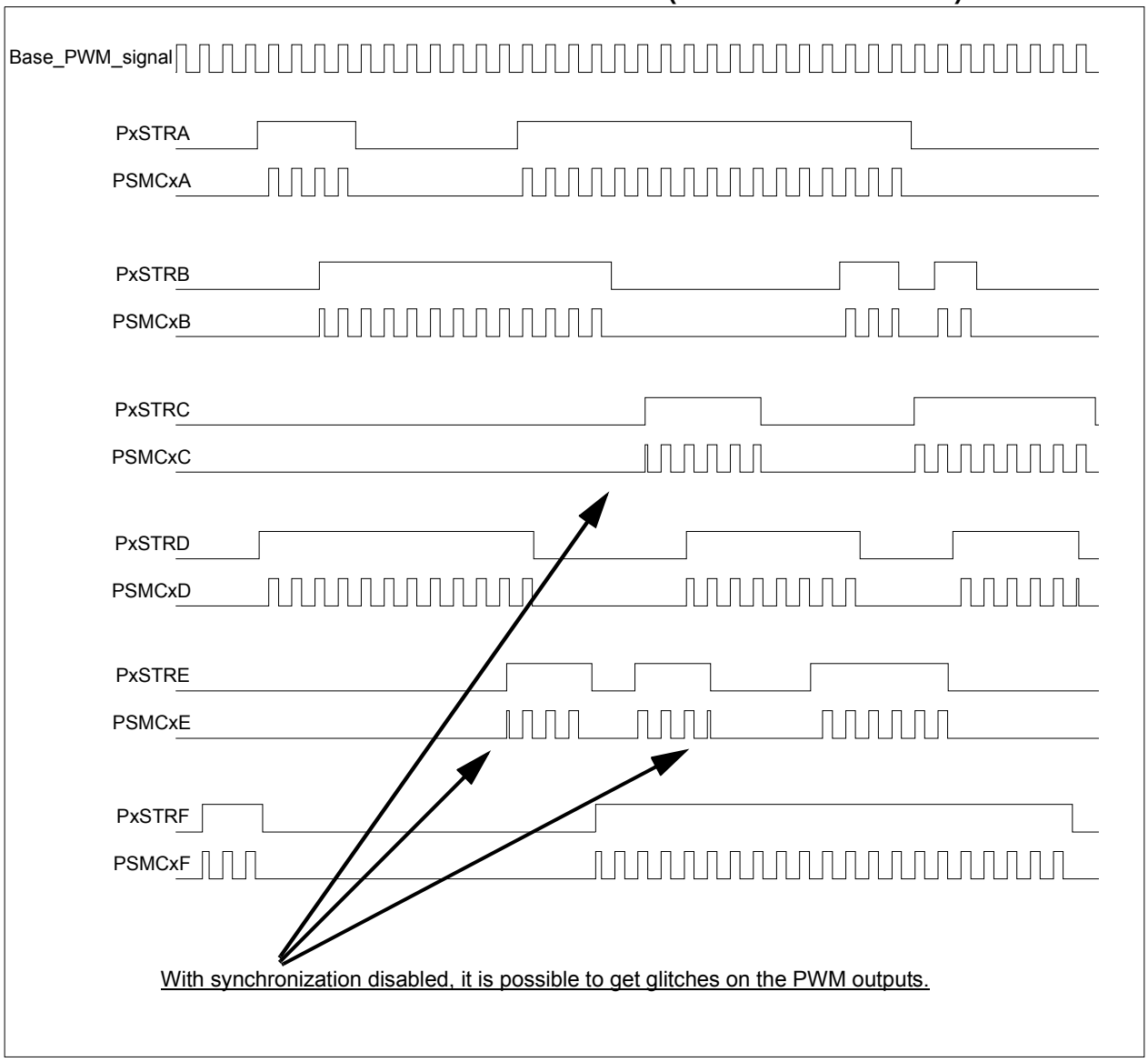
### 24.5.1 3-PHASE STEERING

3-phase steering is available in the 3-Phase Modulation mode only. For more details on 3-phase steering refer to **Section 24.3.12 "3-Phase PWM"**.

### 24.5.2 SINGLE PWM STEERING

In Single PWM Steering mode, the single PWM signal can be routed to any combination of the PSMC output pins. Examples of unsynchronized single PWM steering are shown in Figure 24-16.

**FIGURE 24-16: SINGLE PWM STEERING WAVEFORM (NO SYNCHRONIZATION)**



## 26.4 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.

### 26.4.1 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 8th 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.

### 26.4.2 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.

### 26.4.3 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 TRIS bits.

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

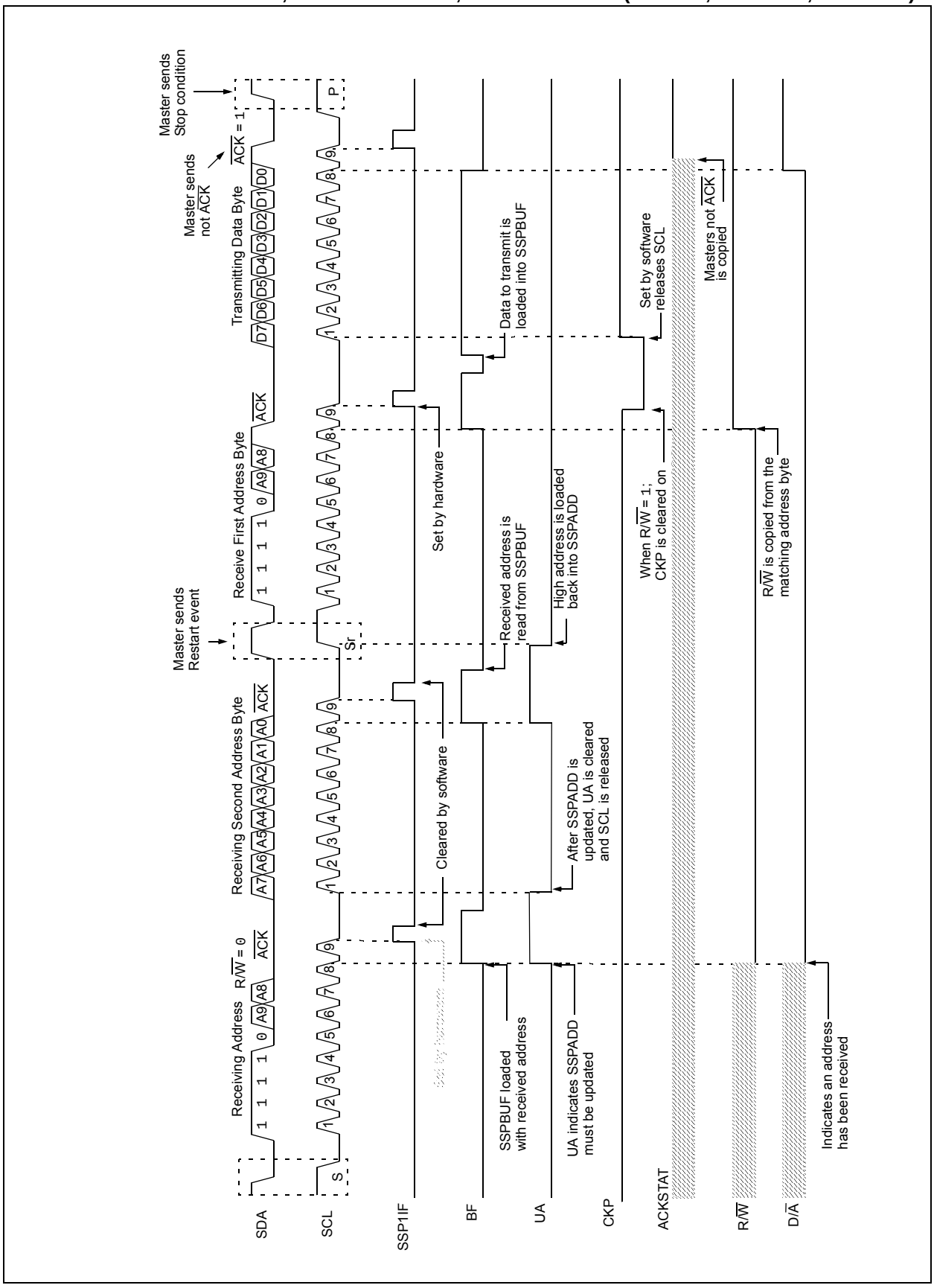
### 26.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPCON3 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 26-2: 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 are 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 SSPADD. |
| 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 out-putting and expected high state. |

**FIGURE 26-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)**



### 26.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSP1IF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 26-27).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSP1IF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

#### 26.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPSTAT register is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

#### 26.6.6.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

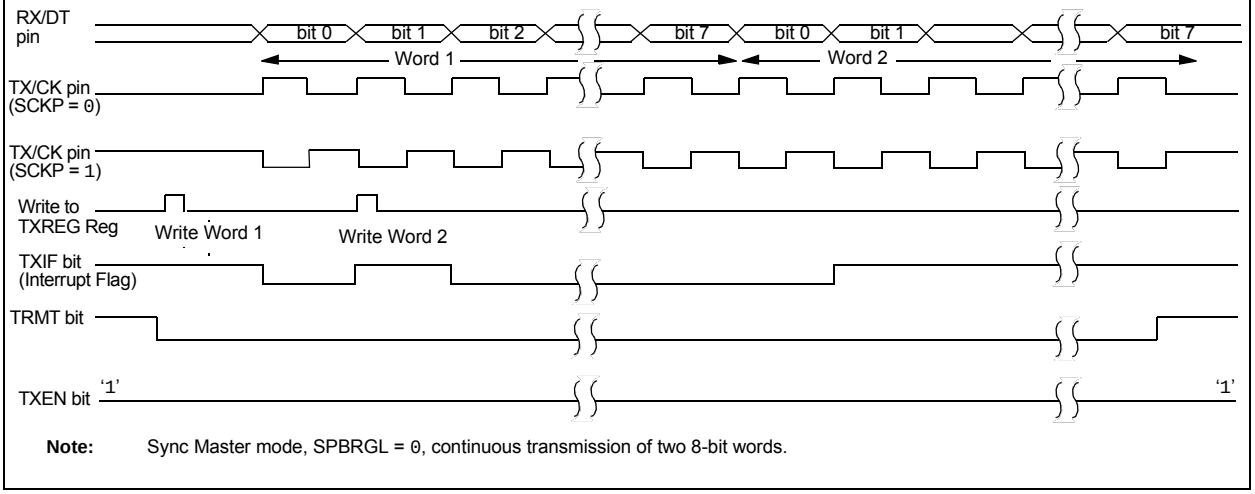
#### 26.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

#### 26.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSP1IF is set by hardware on completion of the Start.
3. SSP1IF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all 8 bits are transmitted. Transmission begins as soon as SSPBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSP1IF bit.
9. The user loads the SSPBUF with eight bits of data.
10. Data is shifted out the SDA pin until all 8 bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

**FIGURE 27-10: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode, SPBRGL = 0, continuous transmission of two 8-bit words.

**FIGURE 27-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

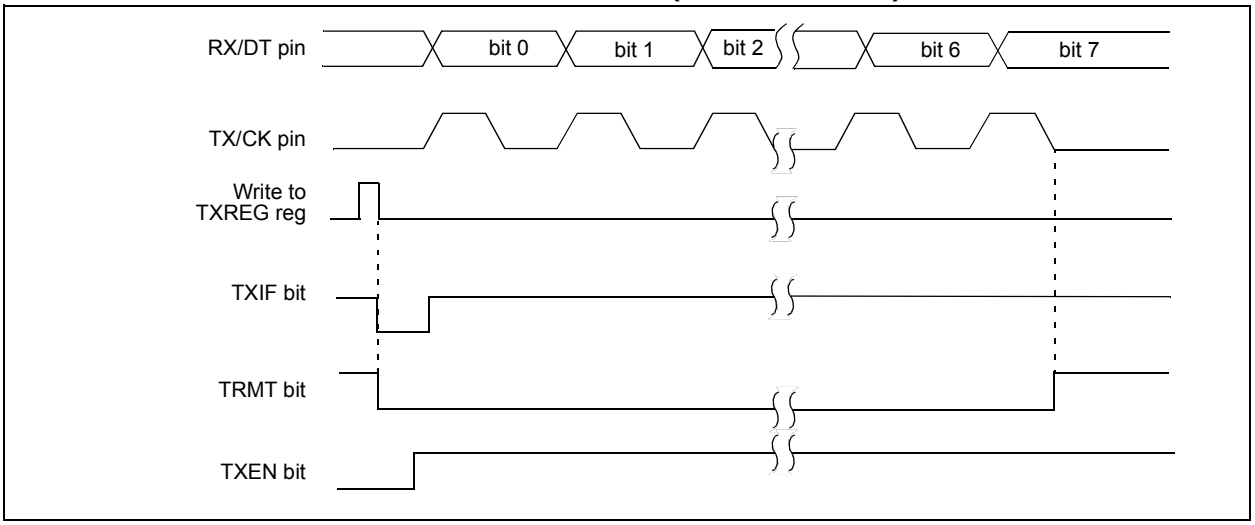



**TABLE 27-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| APFCON1 | C2OUTSEL | CC1PSEL | SDOSEL | SCKSEL | SDISEL | TXSEL | RXSEL | CCP2SEL | 127 |
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 347 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 93 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 94 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 98 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 346 |
| SPBRGL | BRG<7:0> | | | | | | | | 348 |
| SPBRGH | BRG<15:8> | | | | | | | | 348 |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 142 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 337* |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 345 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master transmission.

* Page provides register information.

**TABLE 30-9: CLKOUT AND I/O TIMING PARAMETERS**

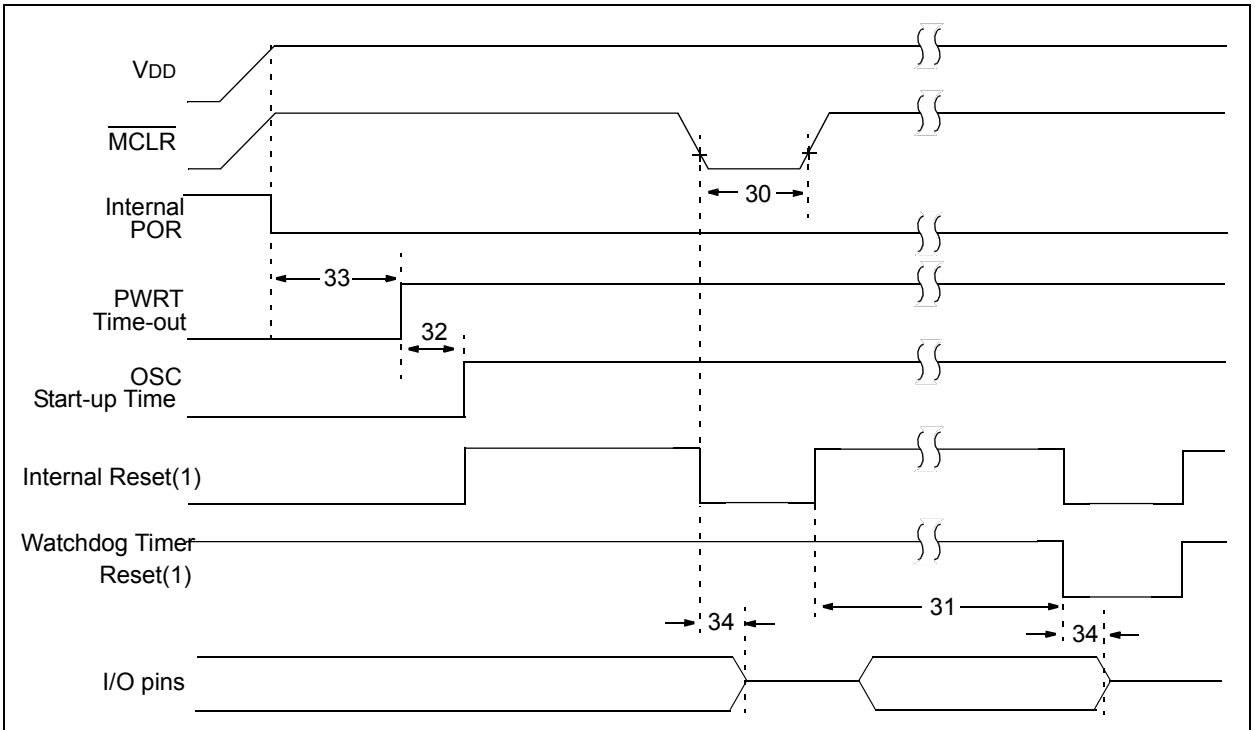
| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11 | TosH2ckL | FOSC↑ to CLKOUT↓ (1) | — | — | 70 | ns | VDD = 3.3-5.0V |
| OS12 | TosH2ckH | FOSC↑ to CLKOUT↑ (1) | — | — | 72 | ns | VDD = 3.3-5.0V |
| OS13 | TckL2ioV | CLKOUT↓ to Port out valid(1) | — | — | 20 | ns | |
| OS14 | TioV2ckH | Port input valid before CLKOUT↑(1) | TOSC + 200 ns | — | — | ns | |
| OS15 | TosH2ioV | Fosc↑ (Q1 cycle) to Port out valid | — | 50 | 70* | ns | VDD = 3.3-5.0V |
| OS16 | TosH2ioI | Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 50 | — | — | ns | VDD = 3.3-5.0V |
| OS17 | TioV2osH | Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time) | 20 | — | — | ns | |
| OS18* | TioR | Port output rise time | —<br>— | 40<br>15 | 72<br>32 | ns | VDD = 1.8V<br>VDD = 3.3-5.0V |
| OS19* | TioF | Port output fall time | —<br>— | 28<br>15 | 55<br>30 | ns | VDD = 1.8V<br>VDD = 3.3-5.0V |
| OS20* | Tinp | INT pin input high or low time | 25 | — | — | ns | |
| OS21* | Tioc | Interrupt-on-change new input level time | 25 | — | — | ns | |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

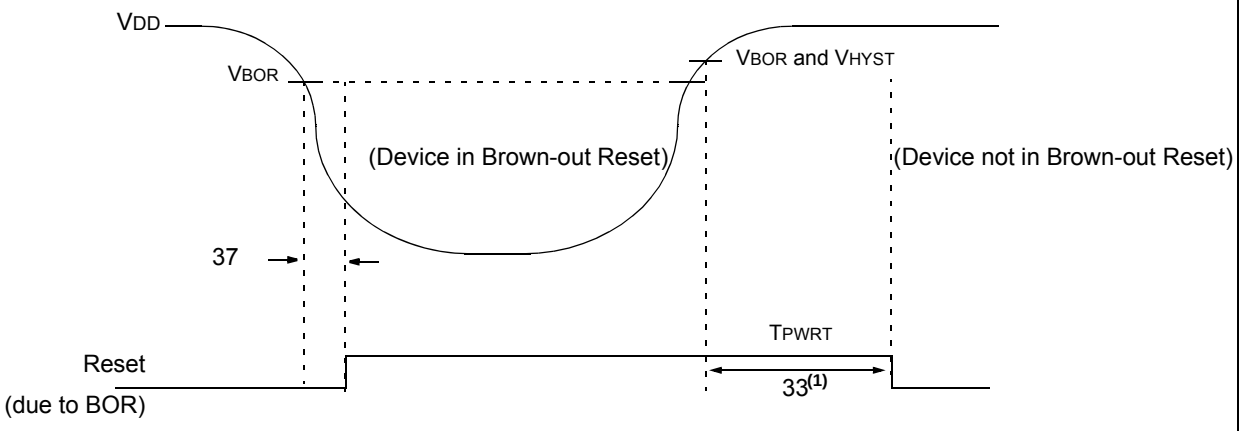
**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

**FIGURE 30-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note 1:** Asserted low.

**FIGURE 30-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS**




**Note 1:** The delay, (TPWRT) releasing Reset, only occurs when the Power-up Timer is enabled, ($\overline{PWRTE}$ = 0).

**Note:** Unless otherwise noted, $V_{IN}$ = 5V, $F_{OSC}$ = 300 kHz, $C_{IN}$ = 0.1 µF, $T_A$ = 25°C.

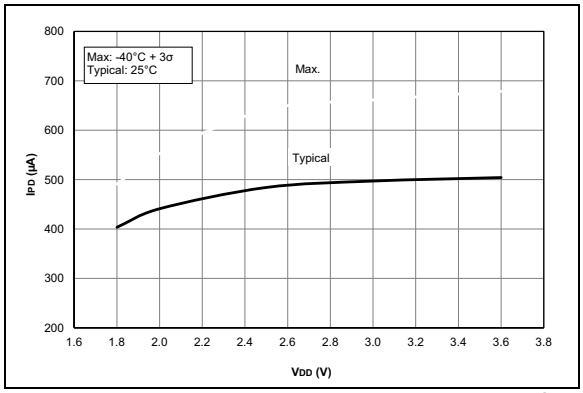



***FIGURE 31-49:*** *IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1784/6/7 Only.*

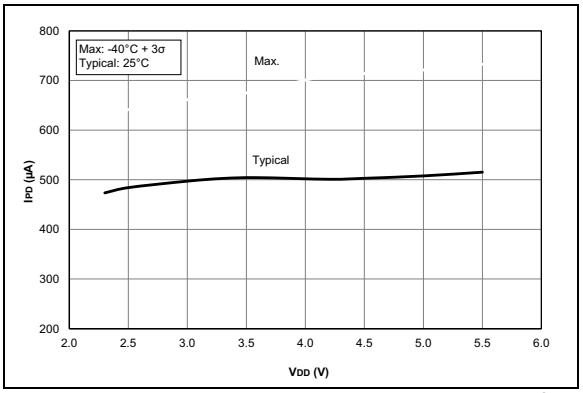



***FIGURE 31-50:*** *IPD, Comparator, NP Mode (CxSP = 1), PIC16F1784/6/7 Only.*

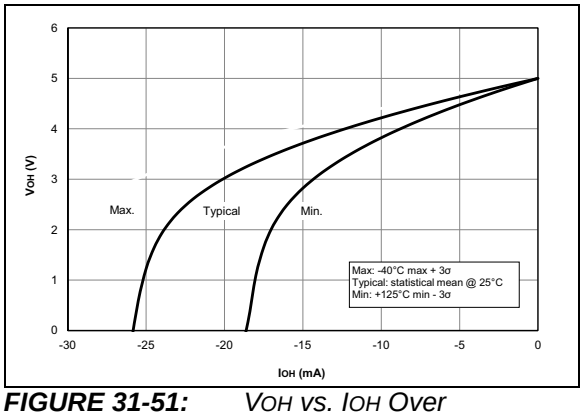



***FIGURE 31-51:*** *VOH vs. IOH Over Temperature, VDD = 5.0V, PIC16F1784/6/7 Only.*

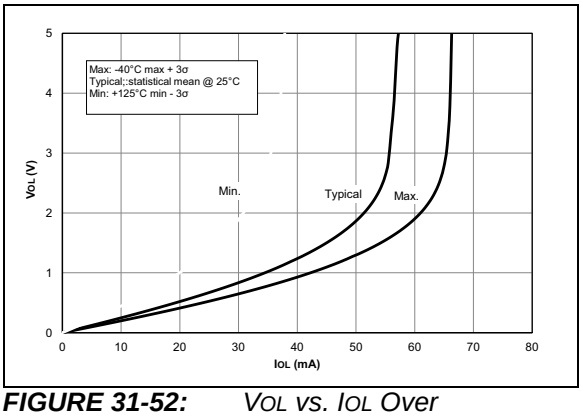



***FIGURE 31-52:*** *VOL vs. IOL Over Temperature, VDD = 5.0V, PIC16F1784/6/7 Only.*

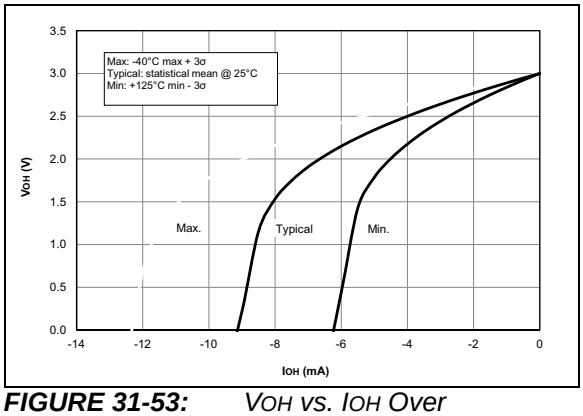



***FIGURE 31-53:*** *VOH vs. IOH Over Temperature, VDD = 3.0V.*

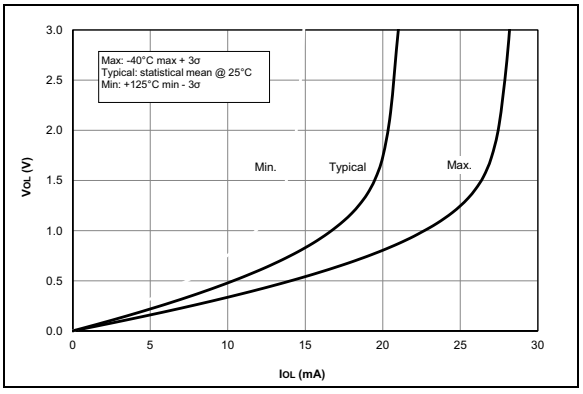



***FIGURE 31-54:*** *VOL vs. IOL Over Temperature, VDD = 3.0V.*

**Note:** Unless otherwise noted, $V_{IN}$ = 5V, $F_{OSC}$ = 300 kHz, $C_{IN}$ = 0.1 µF, $T_A$ = 25°C.

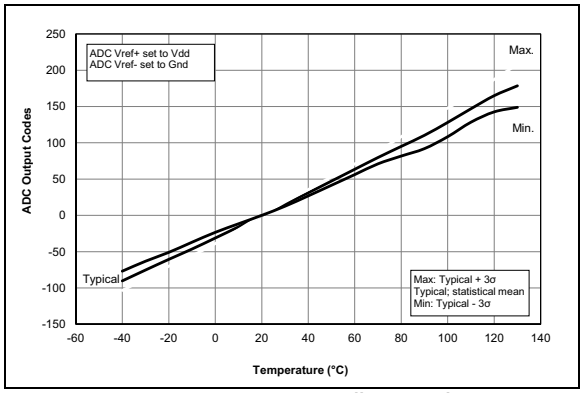



***FIGURE 31-103:*** *Temp. Indicator Slope Normalized to 20°C, High Range, $V_{DD}$ = 3.6V, PIC16F1784/6/7 Only.*

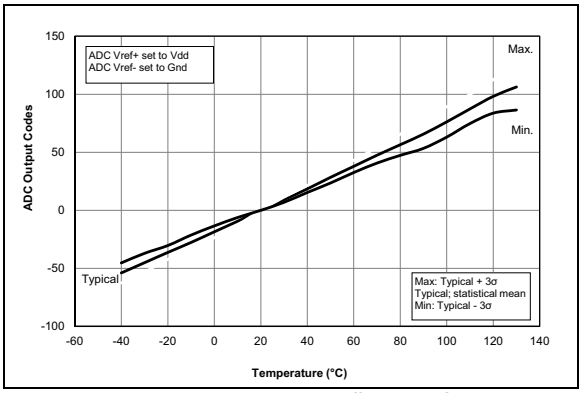



***FIGURE 31-104:*** *Temp. Indicator Slope Normalized to 20°C, Low Range, $V_{DD}$ = 3.0V, PIC16F1784/6/7 Only.*

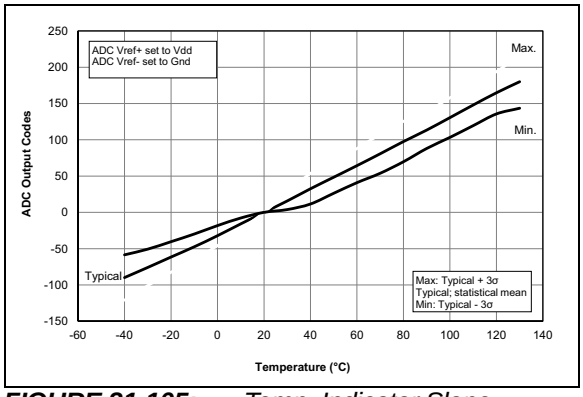



***FIGURE 31-105:*** *Temp. Indicator Slope Normalized to 20°C, Low Range, $V_{DD}$ = 1.8V, PIC16LF1784/6/7 Only.*

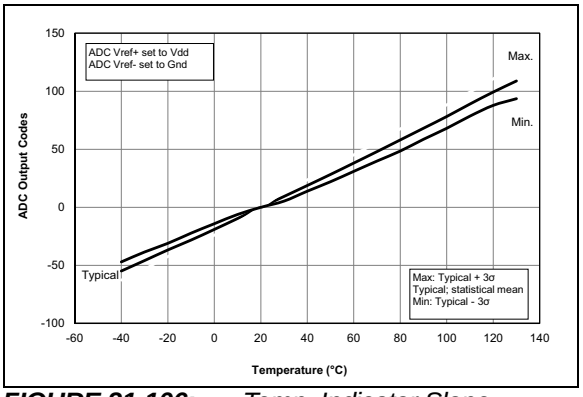



***FIGURE 31-106:*** *Temp. Indicator Slope Normalized to 20°C, Low Range, $V_{DD}$ = 3.0V, PIC16LF1784/6/7 Only.*

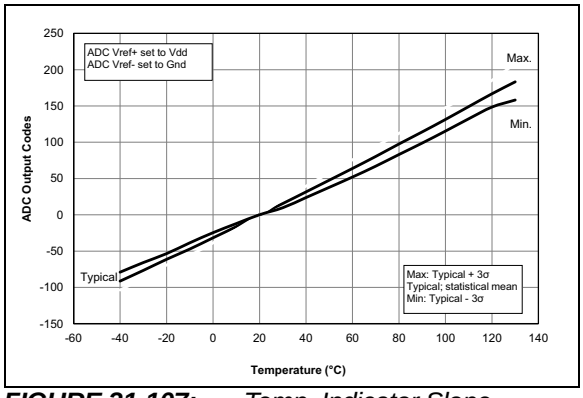



***FIGURE 31-107:*** *Temp. Indicator Slope Normalized to 20°C, High Range, $V_{DD}$ = 3.6V, PIC16LF1784/6/7 Only.*

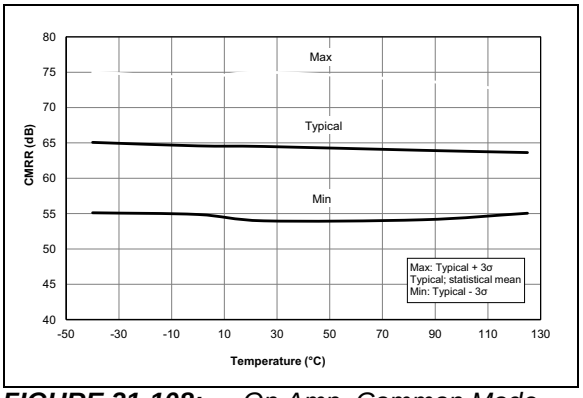



***FIGURE 31-108:*** *Op Amp, Common Mode Rejection Ratio (CMRR), $V_{DD}$ = 3.0V.*

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

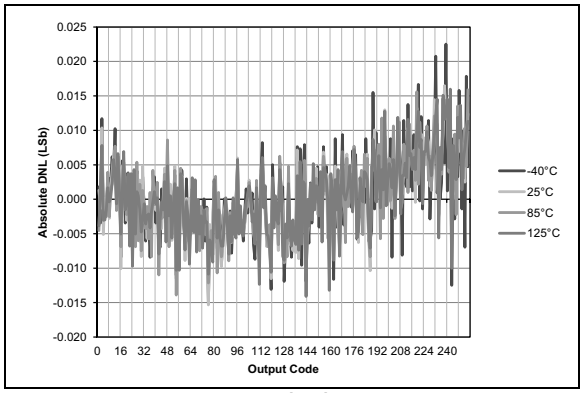



***FIGURE 31-126:*** *Typical DAC DNL Error, VDD = 3.0V, VREF = External 3V.*

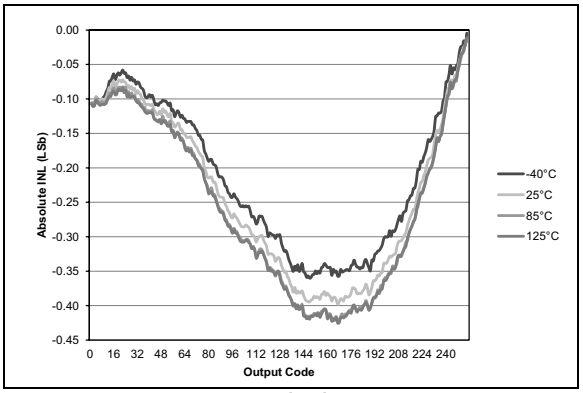



***FIGURE 31-127:*** *Typical DAC INL Error, VDD = 3.0V, VREF = External 3V.*

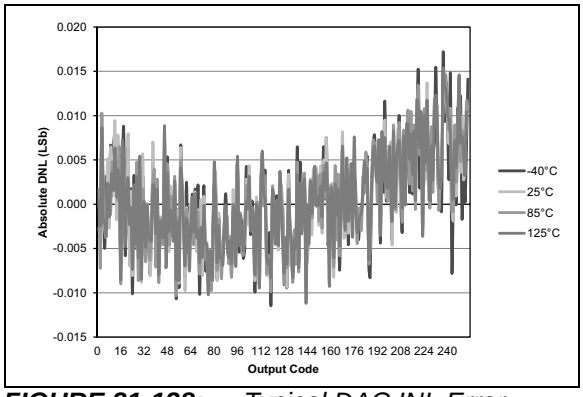



***FIGURE 31-128:*** *Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC16F1784/6/7 Only.*

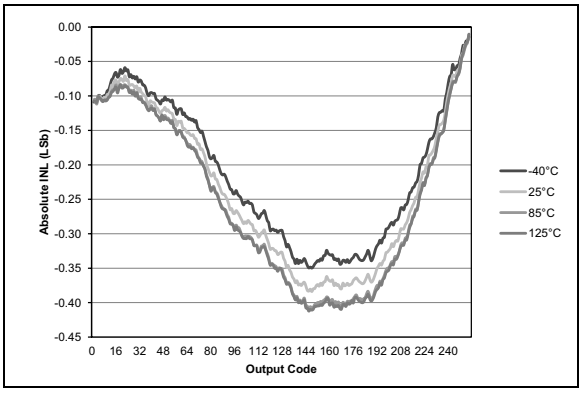



***FIGURE 31-129:*** *Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC16F1784/6/7 Only.*

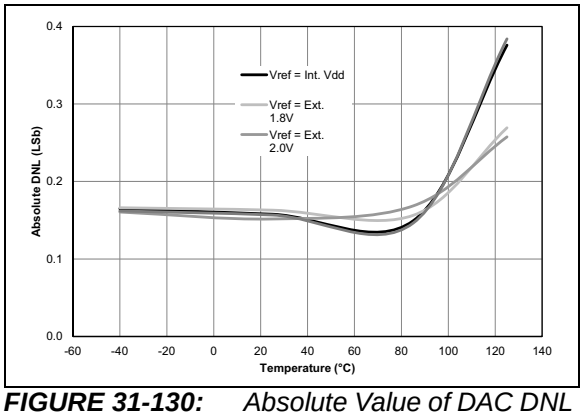



***FIGURE 31-130:*** *Absolute Value of DAC DNL Error, VDD = 3.0V, VREF = VDD.*

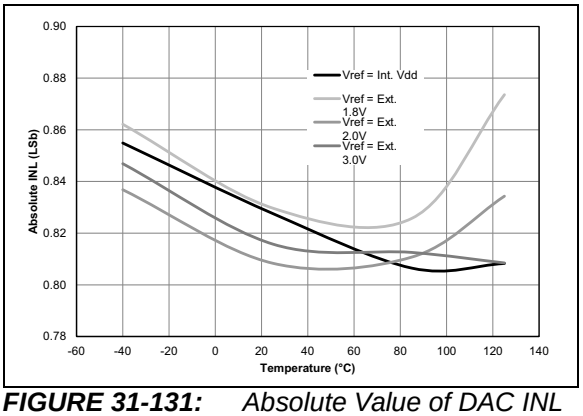



***FIGURE 31-131:*** *Absolute Value of DAC INL Error, VDD = 3.0V.*