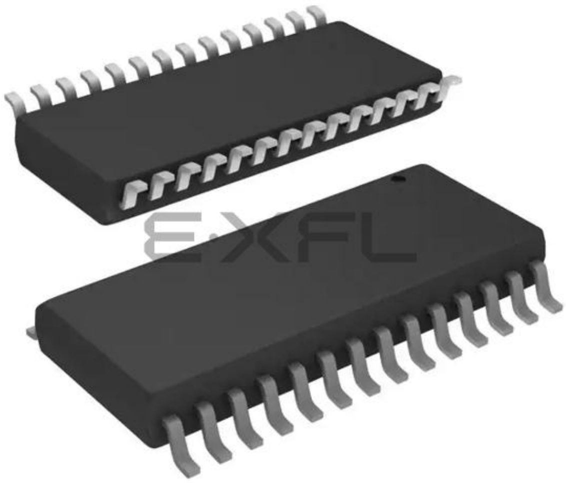




## What is "Embedded - Microcontrollers"?

"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 | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 32KB (16K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 24x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f25k42t-i-so |

**REGISTER 9-4: OSCSTAT: OSCILLATOR STATUS REGISTER 1**

| R-q/q | R-q/q | R-q/q | R-q/q | R-q/q | R-q/q | U-0 | R-q/q |
|---|---|---|---|---|---|---|---|
| EXTOR | HFOR | MFOR | LFOR | SOR | ADOR | — | PLLR |
| 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 = Reset value is determined by hardware |

bit 7 **EXTOR:** EXTOSC (external) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 6 **HFOR:** HFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 5 **MFOR:** MFINTOSC Oscillator Ready
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 4 **LFOR:** LFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 3 **SOR:** Secondary (Timer1) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 2 **ADOR:** ADC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **PLLR:** PLL is Ready bit
1 = The PLL is ready to be used
0 = The PLL is not enabled, the required input source is not ready, or the PLL is not locked.

**REGISTER 9-6: OSCTUNE: HFINTOSC TUNING REGISTER**

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

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

bit 5-0 **TUN<5:0>:** HFINTOSC Frequency Tuning bits

`01 1111` = Maximum frequency
•
•
•
`00 0000` = Center frequency. Oscillator module is running at the calibrated frequency (default value).
•
•
•
`10 0000` = Minimum frequency

**TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| OSCCON1 | — | NOSC<2:0> | | | NDIV<3:0> | | | | 109 |
| OSCCON2 | — | COSC<2:0> | | | CDIV<3:0> | | | | 110 |
| OSCCON3 | CSWHOLD | SOSCPWR | — | ORDY | NOSCR | — | — | — | 110 |
| OSCSTAT | EXTOR | HFOR | MFOR | LFOR | SOR | ADOR | — | PLLR | 111 |
| OSCTUNE | — | — | TUN<5:0> | | | | | | 113 |
| OSCFRQ | — | — | — | — | FRQ<3:0> | | | | 112 |
| OSCEN | EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | — | — | 114 |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

**TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES**

| Name | Bits | Bit -/7 | Bit -/6 | Bit 13/5 | Bit 12/4 | Bit 11/3 | Bit 10/2 | Bit 9/1 | Bit 8/0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG1 | 13:8 | — | — | FCMEN | — | CSWEN | — | — | $\overline{\text{CLKOUTEN}}$ | 71 |
| | 7:0 | — | RSTOSC<2:0> | | | — | FEXTOSC<2:0> | | | |

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

**REGISTER 11-29: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | U-0 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| CLC1IP | CWG1IP | NCO1IP | — | CCP1IP | TMR2IP | TMR1GIP | TMR1IP |
| 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 **CLC1IP:** CLC1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CWG1IP:** CWG1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **NCO1IP:** NCO1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

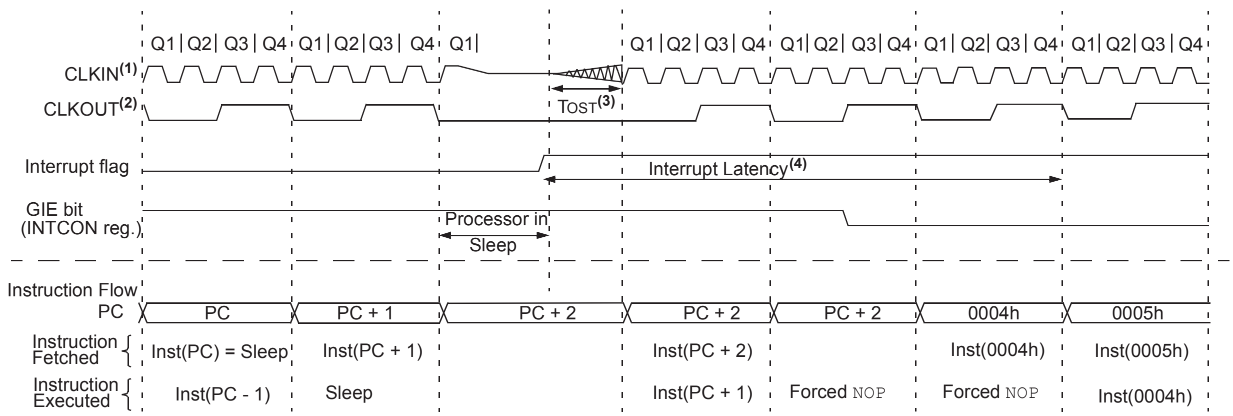
bit 2 **TMR2IP:** TMR2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR1GIP:** TMR1 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Interrupt Priority bit
1 = High priority
0 = Low priority

**FIGURE 12-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** External clock. High, Medium, Low mode assumed.
**2:** CLKOUT is shown here for timing reference.
**3:** TOST = 1024 TOSC. This delay does not apply to EC and INTOSC Oscillator modes.
**4:** GIE = `1` assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = `0`, execution will continue in-line.

### 12.2.3 LOW-POWER SLEEP MODE

The PIC18F2X/4XK42 device family contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

The PIC18F2xK42 devices allows the user to optimize the operating current in Sleep, depending on the application requirements.

Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register.

#### 12.2.3.1 Sleep Current vs. Wake-up Time

In the default operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking-up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The Normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

**REGISTER 13-5: WDTTMR: WDT TIMER REGISTER (READ-ONLY)**

| R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| WDTTMR<4:0> | | | | | STATE | PSCNT<17:16> | |
| 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 7-3 **WDTTMR<4:0>:** Watchdog Window Value bits

| WINDOW | WDT Window State | | Open Percent |
|---|---|---|---|
| | Closed | Open | |
| 111 | N/A | 00000-11111 | 100 |
| 110 | 00000-00011 | 00100-11111 | 87.5 |
| 101 | 00000-00111 | 01000-11111 | 75 |
| 100 | 00000-01011 | 01100-11111 | 62.5 |
| 011 | 00000-01111 | 10000-11111 | 50 |
| 010 | 00000-10011 | 10100-11111 | 37.5 |
| 001 | 00000-10111 | 11000-11111 | 25 |
| 000 | 00000-11011 | 11100-11111 | 12.5 |

bit 2 **STATE:** WDT Armed Status bit
1 = WDT is armed
0 = WDT is not armed

bit 1-0 **PSCNT<17:16>:** Prescale Select Upper Byte bits(1)

**Note 1:** The 18-bit WDT prescale value, PSCNT<17:0> includes the WDTPSL, WDTPSH and the lower bits of the WDTTMR registers. PSCNT<17:0> is intended for debug operations and should not be read during normal operation.

### REGISTER 16-6: CRCACCL: CRC ACCUMULATOR LOW BYTE 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 |
|---|---|---|---|---|---|---|---|
| ACC<7:0> | | | | | | | |
| 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 | |

bit 7-0 **ACC<7:0>**: CRC Accumulator Register bits

### REGISTER 16-7: CRCSHIFTH: CRC SHIFT HIGH BYTE REGISTER

| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SHIFT<15:8> | | | | | | | |
| 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 **SHIFT<15:8>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

### REGISTER 16-8: CRCSHIFTL: CRC SHIFT LOW BYTE REGISTER

| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SHIFT<7:0> | | | | | | | |
| 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 | |

bit 7-0 **SHIFT<7:0>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

## 17.5 DMA Message Transfers

Once the Enable bit is set to start DMA message transfers, the Source/Destination pointer and counter registers are initialized to the conditions shown in Table 17-3.

**TABLE 17-3: DMA INITIAL CONDITIONS**

| Register | Value loaded |
|---|---|
| DMAxSPTR<21:0> | DMAxSSA<21:0> |
| DMAxSCNT<11:0> | DMAxSSZ<11:0> |
| DMAxDPTR<15:0> | DMAxDSA<15:0> |
| DMAxDCNT<11:0> | DMAxDSZ<11:0> |

During the DMA Operation after each transaction, Table 17-4, and Table 17-5 indicate how the Source/ Destination pointer and counter registers are modified

**TABLE 17-4: DMA SOURCE POINTER/COUNTER DURING OPERATION**

| Register | Modified Source Counter/Pointer Value |
|---|---|
| DMAxSCNT<11:0> != 1 | DMAxSCNT = DMAxSCNT -1 |
| | SMODE = 00: DMAxSPTR = DMAxSPTR |
| | SMODE = 01: DMAxSPTR = DMAxSPTR + 1 |
| | SMODE = 10: DMAxSPTR = DMAxSPTR - 1 |
| DMAxSCNT<11:0> == 1 | DMAxSCNT = DMAxSSZ |
| | DMAxSPTR = DMAxSSA |

**TABLE 17-5: DMA DESTINATION POINTER/COUNTER DURING OPERATION**

| Register | Modified Destination Counter/Pointer Value |
|---|---|
| DMAxDCNT<11:0> != 1 | DMAxDCNT = DMAxDCNT -1 |
| | DMODE = 00: DMAxDPTR = DMAxDPTR |
| | DMODE = 01: DMAxDPTR = DMAxDPTR + 1 |
| | DMODE = 10: DMAxDPTR = DMAxDPTR - 1 |
| DMAxDCNT<11:0> == 1 | DMAxDCNT = DMAxDSZ |
| | DMAxDPTR = DMAxDSA |

The following sections discuss how to initiate and terminate DMA transfers.

### 17.5.1 STARTING DMA MESSAGE TRANSFERS

The DMA can initiate data transactions by either of the following two conditions:

1. User software control
2. Hardware trigger, SIRQ

#### 17.5.1.1 User Software Control

Software starts or stops DMA transaction by setting/ clearing the DGO bit. The DGO bit is also used to indicate whether a DMA hardware trigger has been received and a message is in progress.

> **Note 1:** Software start can only occur if the EN bit (DMAxCON1) is set.
>
> **2:** If the CPU writes to the DGO bit while it is already set, there is no effect on the system, the DMA will continue to operate normally.

## 26.2 Register Definitions: PWM Control

Long bit name prefixes for the PWM peripherals are shown in Table 26-3. Refer to **Section 1.4.2.2 "Long Bit Names"** for more information.

**TABLE 26-3:**

| Peripheral | Bit Name Prefix |
|---|---|
| PWM5 | PWM5 |
| PWM6 | PWM6 |
| PWM7 | PWM7 |
| PWM8 | PWM8 |

**REGISTER 26-1: PWMxCON: PWM CONTROL REGISTER**

| R/W-0/0 | U-0 | R-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| EN | — | OUT | POL | — | — | — | — |
| 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 **EN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

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

bit 5 **OUT:** PWM Module Output Level When Bit is Read

bit 4 **POL:** PWM Output Polarity Select bit
1 = PWM output is inverted
0 = PWM output is normal

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

### REGISTER 26-3: PWMxDCH: PWM DUTY CYCLE 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 = 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 **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 PWMxDCL Register.

### REGISTER 26-4: PWMxDCL: PWM DUTY CYCLE 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 = 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 7-6 **DC<1:0>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in PWMxDCH Register.

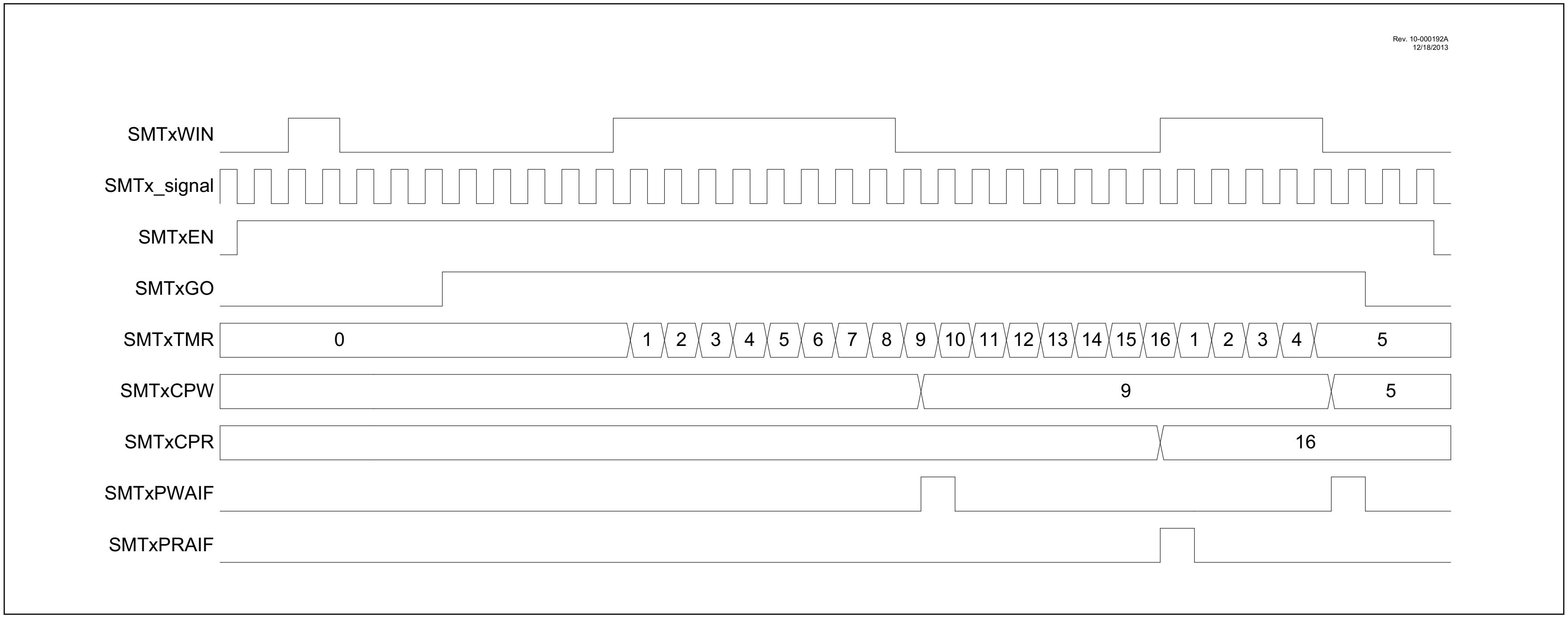
bit 5-0 **Unimplemented:** Read as ‘0’

### TABLE 26-4: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| PWMxCON | EN | — | OUT | POL | — | — | — | — | 361 |
| PWMxDCH | DC<9:2> | | | | | | | | 363 |
| PWMxDCL | DC<1:0> | | — | — | — | — | — | — | 363 |

**Legend:** – = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the PWM.

**FIGURE 27-21: WINDOWED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM**






Preliminary

**REGISTER 27-2: SMT1CON1: SMT CONTROL REGISTER 1**

| R/W/HC-0/0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| GO | REPEAT | — | — | MODE<3:0> | | | |
| 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 | -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 |

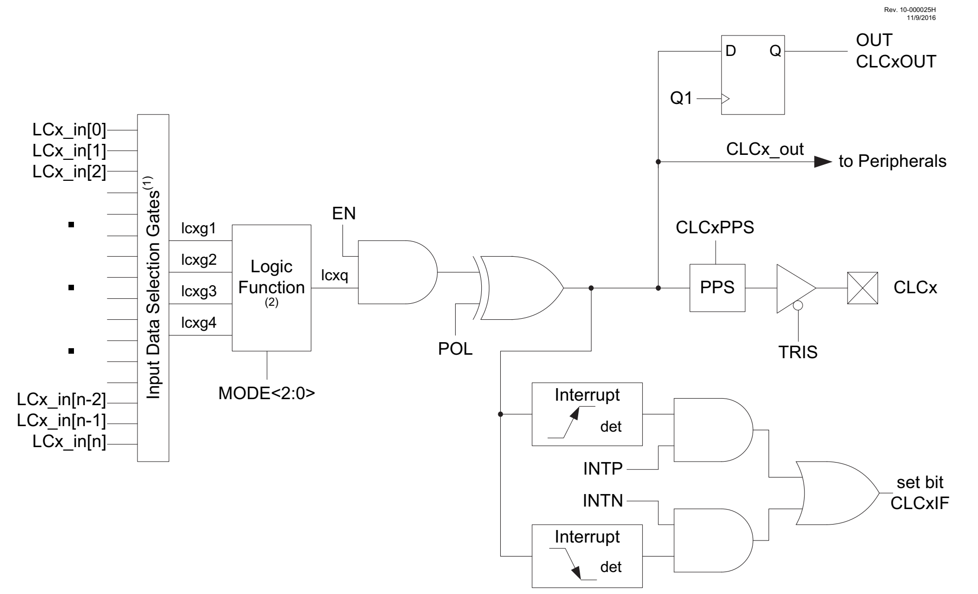
bit 7 **GO:** GO Data Acquisition bit
1 = Incrementing, acquiring data is enabled
0 = Incrementing, acquiring data is disabled

bit 6 **REPEAT:** SMT Repeat Acquisition Enable bit
1 = Repeat Data Acquisition mode is enabled
0 = Single Acquisition mode is enabled

bit 5-4 **Unimplemented**: Read as ‘0’

bit 3-0 **MODE<3:0>** SMT Operation Mode Select bits
1111 = Reserved
•
•
•
1011 = Reserved
1010 = Windowed counter
1001 = Gated counter
1000 = Counter
0111 = Capture
0110 = Time of flight
0101 = Gated windowed measure
0100 = Windowed measure
0011 = High and low time measurement
0010 = Period and Duty-Cycle Acquisition
0001 = Gated Timer
0000 = Timer

**FIGURE 29-1: CLCx SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** See Figure 29-2: Input Data Selection and Gating

**2:** See Figure 29-3: Programmable Logic Functions.

## 29.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

### 29.1.1 DATA SELECTION

There are 32 signals available as inputs to the configurable logic. Four 32-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of Figure 29-2. Data inputs in the figure are identified by a generic numbered input name.

Table 29-1 correlates the generic input name to the actual signal for each CLC module. The column labeled ‘DyS<4:0> Value’ indicates the MUX selection code for the selected data input. DyS is an abbreviation for the MUX select input codes: D1S<4:0> through D4S<4:0>.

Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 29-14 through Register 29-17).

| **Note:** | Data selections are undefined at power-up. |
|---|---|

## 33.3 Asynchronous Address Mode

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems.

When Asynchronous Address mode is enabled, all data is transmitted and received as 9-bit characters. The 9th bit determines whether the character is an address or data. When the 9th bit is set, the eight Least Significant bits are the address. When the 9th bit is clear, the Least Significant bits are data. In either case, the 9th bit is stored in PERIF when the byte is written to the receive FIFO. When PERIE is also set, the RXIF will be suppressed, thereby suspending DMA transfers allowing software to process the received address.

An address character will enable all receivers that match the address and disable all other receivers. Once a receiver is enabled, all non-address characters will be received until an address character is received that does not match.

### 33.3.1 ADDRESS MODE TRANSMIT

The UART transmitter is enabled for asynchronous address operation by configuring the following control bits:

- TXEN = `1`
- MODE<3:0> = `0100`
- UxBRGH:L = desired baud rate
- RxyPPS = code for desired output pin
- ON = `1`

Addresses are sent by writing to the UxP1L register. This transmits the written byte with the 9th bit set, which indicates that the byte is an address.

Data is sent by writing to the UxTXB register. This transmits the written byte with the 9th bit cleared, which indicates that the byte is data.

To send data to a particular device on the transmission bus, first transmit the address of the intended device. All subsequent data will be accepted only by that device until an address of another device is transmitted.

Writes to UxP1L take precedence over writes to UxTXB. When both the UxP1L and UxTXB registers are written while the TSR is busy, the next byte to be transmitted will be from UxP1L.

To ensure that all data intended for one device is sent before the address is changed, wait until the TXMTIF bit is high before writing UxP1L with the new address.

### 33.3.2 ADDRESS MODE RECEIVE

The UART receiver is enabled for asynchronous address operation by configuring the following control bits:

- RXEN = `1`
- MODE<3:0> = `0100`
- UxBRGH:L = desired baud rate
- RXPPS = code for desired input pin
- Input pin ANSEL bit = `0`
- UxP2L = receiver address
- UxP3L = address mask
- ON = `1`

In Address mode, no data will be transferred to the input FIFO until a valid address is received. This is the default state. Any of the following conditions will cause the UART to revert to the default state:

- ON = `0`
- RXEN = `0`
- Received address does not match

When a character with the 9th bit set is received, the Least Significant eight bits of that character will be qualified by the values in the UxP2L and UxP3L registers.

The byte is XOR'd with UxP2L then AND'd with UxP3L. A match occurs when the result is 0h, in which case, the unaltered received character is stored in the receive FIFO, thereby setting the UxRXIF interrupt bit. The 9th bit is stored in the corresponding PERIF bit, identifying this byte as an address.

An address match also enables the receiver for all data such that all subsequent characters without the 9th bit set will be stored in the receive FIFO.

When the 9th bit is set and a match does not occur, the character is not stored in the receive FIFO and all subsequent data is ignored.

The UxP3L register mask allows a range of addresses to be accepted. Software can then determine the sub-address of the range by processing the received address character.

## 34.2 SPI REGISTERS

- SPI Interrupt Flag Register (SPIxINTF)
- SPI Interrupt Enable Register (SPIxINTE)
- SPI Byte Count High and Low Registers (SPIxTCTH/L)
- SPI Bit Count Register (SPIxTWIDTH)
- SPI Baud Rate Register (SPIxBAUD)
- SPI Control Register 0 (SPIxCON0)
- SPI Control Register 1 (SPIxCON1)
- SPI Control Register 2 (SPIxCON2)
- SPI FIFO Status Register (SPIxSTATUS)
- SPI Receiver Buffer Register (SPIxRB)
- SPI Transmit Buffer Register (SPIxTB)
- SPI Clock Select Register (SPIxCLKSEL)

SPIxCON0, SPIxCON1, and SPIxCON2 are control registers for the SPI module.

SPIxSTATUS contains several Status bits that indicate the status of both the SPI module and the receive and transmit FIFOs.

SPIxBAUD and SPIxCLKSEL control the baud rate generator of the SPI module when in Master mode. The SPIxCLKSEL selects the clock source that is used. The SPIxBAUD configures the clock divider used on that clock. More information on the baud rate generator is available in **Section 34.5.6 "Master Mode SPI Clock Configuration"**."

SPIxTxB and SPIxRxB are the transmit and receive buffer registers used to send and receive data on the SPI bus. They both offer indirect access to shift registers that are used for shifting the data in and out. Both registers access the two-byte FIFOs, allowing for multiple transmissions/receptions to be stored between software transfers the data.

The SPIxTCTH:L register pair either count or control the number of bits or bytes in a data transfer. When BMODE = `1`, the SPIxTCT value signifies bytes and the SPIxTWIDTH value signifies the number of bits in a byte. When BMODE = `0`, the SPIxTCT value is concatenated with the SPIxTWIDTH register to signify bits. In Master Receive-only mode (TXR = `0` and RXR = `1`), the data transfer is initiated by writing SPIxTCT with the desired bit or byte value to transfer. In Master Transmit mode (TXR = `1`), the data transfer is initiated by writing the SPIxTxB register, in which case the SPIxTCT is a down counter for the bits or bytes transferred.

The SPIxINTF and SPIxINTE are the flags and enables, respectively, for SPI-specific interrupts. They are tied to the SPIxIF flag and SPIxIE enable in the PIR and PIE registers, which is triggered when any interrupt contained in the SPIxINTF/SPIxINTE registers is triggered. The PIR/PIE registers also contain SPIxTXIF/SPIxTXIE bits, which are the interrupt flag and enable for the SPI Transmit Interrupt, as well as the SPIxRXIF/SPIxRXIE bits, which are the interrupt flag and enable for the SPI Receive Interrupt.

**REGISTER 34-9: SPIxCON2: SPI CONFIGURATION REGISTER 2**

| R-0/0 | R-0/0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| BUSY | SSFLT | — | — | — | SSET | TXR(1) | RXR(1) |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |

bit 7 **BUSY**: SPI Module Busy Status bit
1 = Data exchange is busy
0 = Data exchange is not taking place

bit 6 **SSFLT**: SS(in) Fault Status bit
If SSET = 0:
1 = SS(in) ended the transaction unexpectedly, and the data byte being received was lost
0 = SS(in) ended normally
If SSET = 1:
This bit is unchanged.

bit 5-3 **Unimplemented**: Read as ‘0’

bit 2 **SSET**: Slave Select Enable bit
Master mode:
1 = SS(out) is driven to the active state continuously
0 = SS(out) is driven to the active state while the transmit counter is not zero
Slave mode:
1 = SS(in) is ignored and data is clocked on all SCK(in) (as though SS = TRUE at all times)
0 = SS(in) enables/disables data input and tri-states SDO if the TRIS bit associated with the SDO pin is set (see Table 30-2 for details)

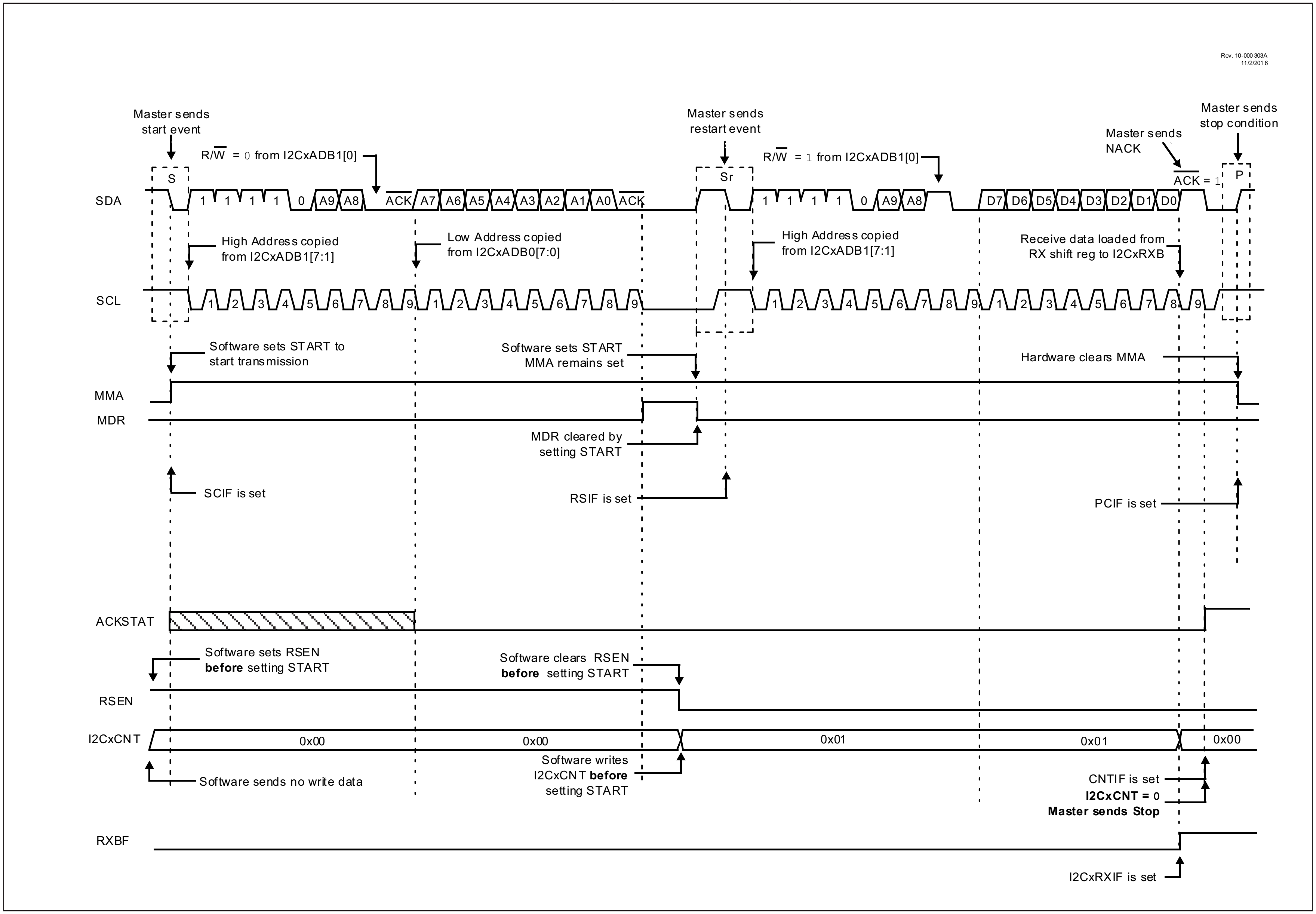
bit 1 **TXR**: Transmit Data-Required Control bit(1)
1 = TxFIFO data is required for a transfer
0 = TxFIFO data is not required for a transfer

bit 0 **RXR**: Receive FIFO Space-Required Control bit(1)
1 = Data transfers are suspended if the RxFIFO is full
0 = Received data is not stored in the FIFO

**Note 1:** See Table 34-1 as well as **Section 34.5 “Master mode”** and **Section 34.6 “Slave Mode”** for more details pertaining to TXR and RXR function.

**2:** This register should not be written to while a transfer is in progress (BUSY bit of SPIxCON2 is set).

**FIGURE 35-22: I²C MASTER, 10-BIT ADDRESS, RECEPTION (USING RSTEN BIT)**



### 37.2.1 CALIBRATION

#### 37.2.1.1 Single-Point Calibration

Single-point calibration is performed by application software using Equation 37-1 and the assumed Mt. A reading of VTSENSE at a known temperature is taken, and the theoretical temperature is calculated by temporarily setting TOFFSET = 0. Then TOFFSET is computed as the difference of the actual and calculated temperatures. Finally, TOFFSET is stored in nonvolatile memory within the device, and is applied to future readings to gain a more accurate measurement.

#### 37.2.1.2 Higher-Order Calibration

If the application requires more precise temperature measurement, additional calibrations steps will be necessary. For these applications, two-point or three-point calibration is recommended.

**Note 1:** The TOFFSET value may be determined by the user with a temperature test.

**2:** Although the measurement range is -40°C to +125 °C, due to the variations in offset error, the single-point uncalibrated calculated TSENSE value may indicate a temperature from -140°C to +225°C, before the calibration offset is applied.

**3:** The User must take into consideration self-heating of the device at different clock frequencies and output pin loading. For package related thermal characteristics information, refer to Table 46-6.

### 37.2.2 TEMPERATURE RESOLUTION

The resolution of the ADC reading, Ma (°C/count), depends on both the ADC resolution N and the reference voltage used for conversion, as shown in Equation 37-2. It is recommended to use the smallest VREF value, such as 2.048 FVR reference voltage, instead of VDD.

**Note:** Refer to Table 46-17 for FVR reference voltage accuracy.

**EQUATION 37-2: TEMPERATURE RESOLUTION (°C/LSb)**

$$Ma = \frac{V_{REF}}{2^N} \times Mt$$

$$Ma = \frac{\frac{V_{REF}}{2^N}}{Mv}$$

Where:

Mv = sensor voltage sensitivity (V/°C)

VREF = Reference voltage of the ADC module (in Volts)

N = Resolution of the ADC

The typical Mv value for a single diode is approximately -1.267 to -1.32 mV/C.

The typical Mv value for a stack of two diodes (Low Range setting) is approximately -2.533 mV/C.

The typical Mv value for a stack of three diodes (High range setting) is approximately -3.8 mV/C.

## 37.3 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait a certain time for the ADC value to settle, after the ADC input multiplexer is connected to the temperature indicator output, before the conversion is performed.

### 38.6.5 BURST AVERAGE MODE

The Burst Average mode (ADMD = `011`) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the CNT value is greater than or equal to RPT, even if Continuous Sampling mode (see **Section 38.6.8 "Continuous Sampling mode"**) is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

### 38.6.6 LOW-PASS FILTER MODE

The Low-pass Filter mode (ADMD = `100`) acts similarly to the Average mode in how it handles samples (accumulates samples until CNT value greater than or equal to RPT, then triggers threshold comparison), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. (see Table 38-2 for a more detailed description of the mathematical operation). In this mode, the ADCRS bits determine the cut-off frequency of the low-pass filter (as demonstrated by Table 38-3).

### 38.6.7 THRESHOLD COMPARISON

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error is calculated based on a difference calculation which is selected by the ADCALC<2:0> bits in the ADCON3 register. The value can be one of the following calculations (see Register 38-4 for more details):
  - The first derivative of single measurements
  - The CVD result in CVD mode
  - The current result vs. a setpoint
  - The current result vs. the filtered/average result
  - The first derivative of the filtered/average value
  - Filtered/average value vs. a setpoint
- The result of the calculation (ERR) is compared to the upper and lower thresholds, UTH<ADUTHH:ADUTHL> and LTH<ADLTHH:ADLTHL> registers, to set the ADUTHR and ADLTHR flag bits. The threshold logic is selected by ADTMD<2:0> bits in the ADCON3 register. The threshold trigger option can be one of the following:
  - Never interrupt
  - Error is less than lower threshold
  - Error is greater than or equal to lower threshold
  - Error is between thresholds (inclusive)
  - Error is outside of thresholds
  - Error is less than or equal to upper threshold
  - Error is greater than upper threshold
  - Always interrupt regardless of threshold test results
  - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

> **Note 1:** The threshold tests are signed operations.
>
> **2:** If ADAOV is set, a threshold interrupt is signaled.

### 38.6.8 CONTINUOUS SAMPLING MODE

Setting the CONT bit in the ADCON0 register automatically retriggers a new conversion cycle after updating the ADACC register. The GO bit remains set and re-triggering occurs automatically.

If ADSOI = `1`, a threshold interrupt condition will clear GO and the conversions will stop.

### 38.6.9 DOUBLE SAMPLE CONVERSION

Double sampling is enabled by setting the ADDSEN bit of the ADCON1 register. When this bit is set, two conversions are required before the module will calculate threshold error (each conversion must still be triggered separately). The first conversion will set the ADMATH bit of the ADSTAT register and update ADACC, but will not calculate ERR or trigger ADTIF. When the second conversion completes, the first value is transferred to PREV (depending on the setting of ADPSIS) and the value of the second conversion is placed into ADRES. Only upon the completion of the second conversion is ERR calculated and ADTIF triggered (depending on the value of ADCALC).

## 38.7 Register Definitions: ADC Control

**REGISTER 38-1: ADCON0: ADC CONTROL REGISTER 0**

| R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | U-0 | R/W-0/0 | U-0 | R/W/HC-0 |
|---|---|---|---|---|---|---|---|
| ON | CONT | — | CS | — | FM | — | GO |
| 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 | HC = Bit is cleared by hardware |

bit 7 **ON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled

bit 6 **CONT:** ADC Continuous Operation Enable bit
1 = GO is retriggered upon completion of each conversion trigger until ADTIF is set (if ADSOI is set) or until GO is cleared (regardless of the value of ADSOI)
0 = ADC is cleared upon completion of each conversion trigger

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **CS:** ADC Clock Selection bit
1 = Clock supplied from FRC dedicated oscillator
0 = Clock supplied by FOSC, divided according to ADCLK register

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **FM:** ADC results Format/alignment Selection
1 = ADRES and PREV data are right-justified
0 = ADRES and PREV data are left-justified, zero-filled

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **GO:** ADC Conversion Status bit(1)
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the CONT bit
0 = ADC conversion completed/not in progress

**Note 1:** This bit requires ON bit to be set.
**2:** If cleared by software while a conversion is in progress, the results of the conversion up to this point will be transfered to ADRES and the state machine will be reset, but the ADIF interrupt flag bit will not be set; filter and threshold operations will not be performed.