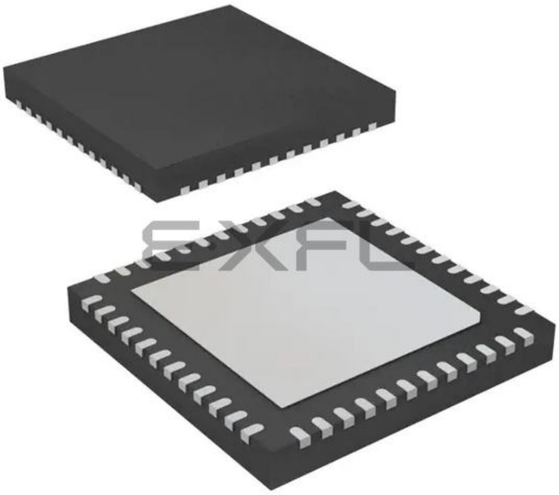
# Microchip Technology - PIC18F55K42-E/MV Datasheet



**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 | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LVD, POR, PWM, WDT |
| Number of I/O | 44 |
| 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 43x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-UFQFN Exposed Pad |
| Supplier Device Package | 48-UQFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f55k42-e-mv |

**TABLE 2: 40/44-PIN ALLOCATION TABLE FOR PIC18(L)F4XK42**

| I/O | 40-Pin PDIP | 44-Pin TQFP | 40-Pin UQFN | 44-Pin QFN | ADC | Voltage Reference | DAC | Comparators | Zero Cross Detect | I$^2$C | SPI | UART | DSM | Timers/SMT | CCP and PWM | CWG | CLC | NCO | Clock Reference (CLKR) | Interrupt-on-Change | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 2 | 19 | 17 | 19 | ANA0 | — | — | C1IN0-<br>C2IN0- | — | — | — | — | — | — | — | — | CLCIN0[1] | — | — | IOCA0 | — |
| RA1 | 3 | 20 | 18 | 20 | ANA1 | — | — | C1IN1-<br>C2IN1- | — | — | — | — | — | — | — | — | CLCIN1[1] | — | — | IOCA1 | — |
| RA2 | 4 | 21 | 19 | 21 | ANA2 | VREF- | DAC1OUT1 | C1IN0+<br>C2IN0+ | — | — | — | — | — | — | — | — | — | — | — | IOCA2 | — |
| RA3 | 5 | 22 | 20 | 22 | ANA3 | VREF+ | — | C1IN1+ | — | — | — | — | MDCARL[1] | — | — | — | — | — | — | IOCA3 | — |
| RA4 | 6 | 23 | 21 | 23 | ANA4 | — | — | — | — | — | — | — | MDCARH[1] | T0CKI[1] | — | — | — | — | — | IOCA4 | — |
| RA5 | 7 | 24 | 22 | 24 | ANA5 | — | — | — | — | — | SS1[1] | — | MDSRC[1] | — | — | — | — | — | — | IOCA5 | — |
| RA6 | 14 | 31 | 29 | 33 | ANA6 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA6 | OSC2<br>CLKOUT |
| RA7 | 13 | 30 | 28 | 32 | ANA7 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA7 | OSC1<br>CLKIN |
| RB0 | 33 | 8 | 8 | 9 | ANB0 | — | — | C2IN1+ | ZCD | — | — | — | — | — | CCP4[1] | CWG1IN[1] | — | — | — | INT0[1]<br>IOCB0 | — |
| RB1 | 34 | 9 | 9 | 10 | ANB1 | — | — | C1IN3-<br>C2IN3- | — | SCL2[3,4] | — | — | — | — | — | CWG2IN[1] | — | — | — | INT1[1]<br>IOCB1 | — |
| RB2 | 35 | 10 | 10 | 11 | ANB2 | — | — | — | — | SDA2[3,4] | — | — | — | — | — | CWG3IN[1] | — | — | — | INT2[1]<br>IOCB2 | — |
| RB3 | 36 | 11 | 11 | 12 | ANB3 | — | — | C1IN2-<br>C2IN2- | — | — | — | — | — | — | — | — | — | — | — | IOCB3 | — |
| RB4 | 37 | 14 | 12 | 14 | ANB4<br>ADCACT[1] | — | — | — | — | — | — | — | — | T5G[1] | — | — | — | — | — | IOCB4 | — |
| RB5 | 38 | 15 | 13 | 15 | ANB5 | — | — | — | — | — | — | — | — | T1G[1] | CCP3[1] | — | — | — | — | IOCB5 | — |
| RB6 | 39 | 16 | 14 | 16 | ANB6 | — | — | — | — | — | — | CTS2[1] | — | - | — | — | CLCIN2[1] | — | — | IOCB6 | ICSPCLK |
| RB7 | 40 | 17 | 15 | 17 | ANB7 | — | DAC1OUT2 | — | — | — | — | RX2[1] | — | T6IN[1] | — | — | CLCIN3[1] | — | — | IOCB7 | ICSPDAT |
| RC0 | 15 | 32 | 30 | 34 | ANC0 | — | — | — | — | — | — | — | — | T1CKI[1]<br>T3CKI[1]<br>T3G[1]<br>SMTWIN1[1] | — | — | — | — | — | IOCC0 | SOSCO |
| RC1 | 16 | 35 | 31 | 35 | ANC1 | — | — | — | — | — | — | — | — | SMTSIG1[1] | CCP2[1] | — | — | — | — | IOCC1 | SOSCI |
| RC2 | 17 | 36 | 32 | 36 | ANC2 | — | — | — | — | — | — | — | — | T5CKI[1] | CCP1[1] | — | — | — | — | IOCC2 | — |

**Note 1:** This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

**2:** All output signals shown in this row are PPS remappable.

**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

**4:** These pins can be configured for I$^2$C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4/RD0/RD1 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I$^2$C specific or SMBus input buffer thresholds.

© 2016-2017 Microchip Technology Inc.

## 2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 7.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-3. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

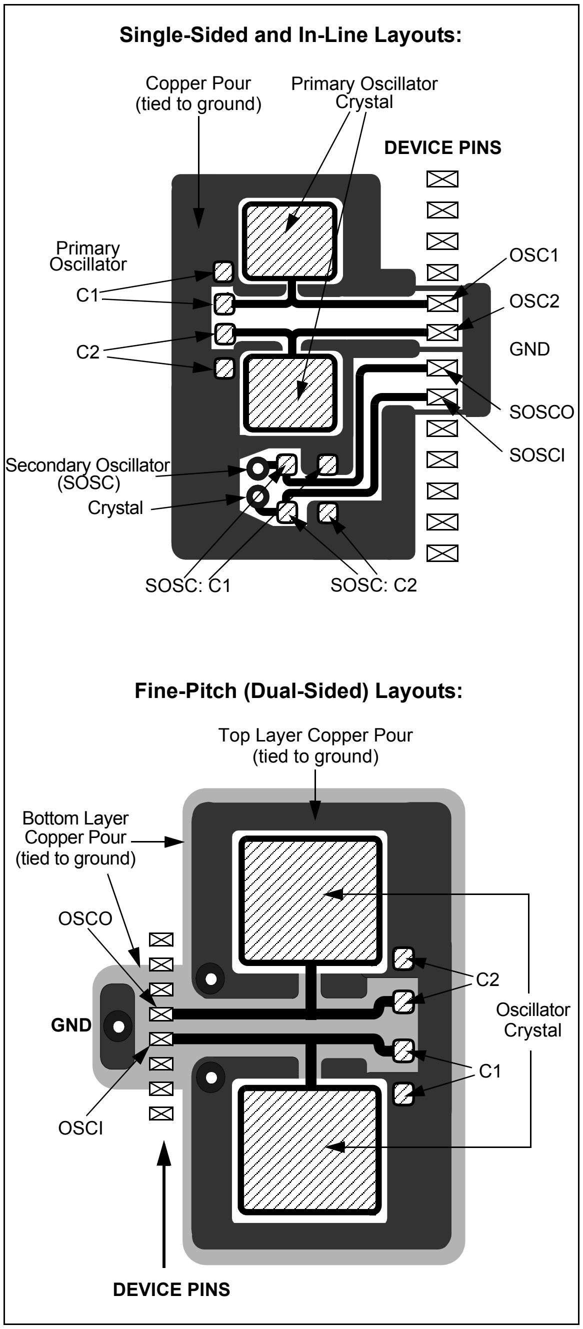
For additional information and design guidance on oscillator circuits, refer to these Microchip application notes, available at the corporate website (www.microchip.com):

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- AN849, *"Basic PICmicro® Oscillator Design"*
- AN943, *"Practical PICmicro® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

## 2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

**FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**



### REGISTER 5-9: CONFIGURATION WORD 5L (30 0008h)

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | $\overline{\text{CP}}$ |
| bit 7 | | | | | | | bit 0 |

**Legend:**

| | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘1’ | |
| -n = Value for blank device | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 7-1 **Unimplemented:** Read as ‘1’

bit 0 **$\overline{\text{CP}}$:** User Program Flash Memory and Data EEPROM Code Protection bit
1 = User Program Flash Memory and Data EEPROM code protection is disabled
0 = User Program Flash Memory and Data EEPROM code protection is enabled

### REGISTER 5-10: CONFIGURATION WORD 5H (30 0009h)

| U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 | U-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

**Legend:**

| | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘1’ | |
| -n = Value for blank device | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 7-0 **Unimplemented:** Read as ‘1’

### TABLE 5-2: SUMMARY OF CONFIGURATION WORDS

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default/ Unprogrammed Value |
|---|---|---|---|---|---|---|---|---|---|---|
| 30 0000h | CONFIG1L | — | RSTOSC<2:0> | | | — | FEXTOSC<2:0> | | | 1111 1111 |
| 30 0001h | CONFIG1H | — | — | FCMEN | — | CSWEN | — | PR1WAY | $\overline{\text{CLKOUTEN}}$ | 1111 1111 |
| 30 0002h | CONFIG2L | BOREN<1:0> | | $\overline{\text{LPBOREN}}$ | IVT1WAY | MVECEN | PWRTS<1:0> | | MCLRE | 1111 1111 |
| 30 0003h | CONFIG2H | $\overline{\text{XINST}}$ | — | $\overline{\text{DEBUG}}$ | STVREN | PPS1WAY | $\overline{\text{ZCD}}$ | BORV<1:0> | | 1111 1111 |
| 30 0004h | CONFIG3L | — | WDTE<1:0> | | WDTCPS<4:0> | | | | | 1111 1111 |
| 30 0005h | CONFIG3H | — | — | WDTCCS<2:0> | | | WDTCWS<2:0> | | | 1111 1111 |
| 30 0006h | CONFIG4L | $\overline{\text{WRTAPP}}$ | — | — | $\overline{\text{SAFEN}}$ | $\overline{\text{BBEN}}$ | BBSIZE<2:0> | | | 1111 1111 |
| 30 0007h | CONFIG4H | — | — | LVP | — | $\overline{\text{WRTSAF}}$ | $\overline{\text{WRTD}}$ | $\overline{\text{WRTC}}$ | $\overline{\text{WRTB}}$ | 1111 1111 |
| 30 0008h | CONFIG5L | — | — | — | — | — | — | — | $\overline{\text{CP}}$ | 1111 1111 |
| 30 0009h | CONFIG5H | — | — | — | — | — | — | — | — | 1111 1111 |

## 7.5 Register Definitions: Oscillator Control

### REGISTER 7-1: OSCCON1: OSCILLATOR CONTROL REGISTER 1

| U-0 | R/W-f/f | R/W-f/f | R/W-f/f | R/W-q/q | R/W-q/q | R/W-q/q | R/W-q/q |
|---|---|---|---|---|---|---|---|
| — | NOSC<2:0> | | | NDIV<3:0> | | | |
| 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 | f = determined by Configuration bit setting |
| | | q = Reset value is determined by hardware |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **NOSC<2:0>:** New Oscillator Source Request bits(1,2,3)
The setting requests a source oscillator and PLL combination per Table 7-1.
POR value = RSTOSC (Register 5-1).

bit 3-0 **NDIV<3:0>:** New Divider Selection Request bits(2,3)
The setting determines the new postscaler division ratio per Table 7-1.

**Note 1:** The default value (f/f) is determined by the RSTOSC Configuration bits. See Table 7-2 below.
**2:** If NOSC is written with a reserved value (Table 7-1), the operation is ignored and neither NOSC nor NDIV is written.
**3:** When CSWEN = 0, this register is read-only and cannot be changed from the POR value.

### TABLE 7-2: DEFAULT OSCILLATOR SETTINGS

| RSTOSC | SFR Reset Values | | | Initial Fosc Frequency |
|---|---|---|---|---|
| | NOSC/COSC | CDIV | OSCFRQ | |
| 111 | 111 | 1:1 | 4 MHz | EXTOSC per FEXTOSC |
| 110 | 110 | 4:1 | | FOSC = 1 MHz (4 MHz/4) |
| 101 | 101 | 1:1 | | LFINTOSC |
| 100 | 100 | 1:1 | | SOSC |
| 011 | Reserved | | | |
| 010 | 010 | 1:1 | 4 MHz | EXTOSC + 4xPLL(1) |
| 001 | Reserved | | | |
| 000 | 110 | 1:1 | 64 MHz | FOSC = 64 MHz |

**Note 1:** EXTOSC must meet the PLL specifications (Table 44-11).

© 2017 Microchip Technology Inc.

### REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REGISTER 2[1]

| R-0/0 | R-0/0 | R-0/0 | R-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| I2C1RXIF[2] | SPI1IF[3] | SPI1TXIF[4] | SPI1RXIF[4] | DMA1AIF | DMA1ORIF | DMA1DCNTIF | DMA1SCNTIF |
| 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 | HS = Hardware set |

bit 7 **I2C1RXIF:** I$^2$C1 Receive Interrupt Flag bit[2]
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 6 **SPI1IF:** SPI1 Interrupt Flag bit[3]
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 5 **SPI1TXIF:** SPI1 Transmit Interrupt Flag bit[4]
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 4 **SPI1RXIF:** SPI1 Receive Interrupt Flag bit[4]
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 3 **DMA1AIF:** DMA1 Abort Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 2 **DMA1ORIF:** DMA1 Overrun Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 1 **DMA1DCNTIF:** DMA1 Destination Count Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 0 **DMA1SCNTIF:** DMA1 Source Count Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

**Note 1:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

**2:** I2CxTXIF and I2CxRXIF are read-only bits. To clear the interrupt condition, the CLRBF bit in I2CxSTAT1 register must be set.

**3:** SPIxIF is a read-only bit. To clear the interrupt condition, all bits in the SPIxINTF register must be cleared.

**4:** SPIxTXIF and SPIxRXIF are read-only bits and cannot be set/cleared by the software.

### REGISTER 9-18: PIE4: PERIPHERAL INTERRUPT ENABLE REGISTER 4

| R/W-0/0 | R/W-0/0 | R/W-0/0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| CLC1IE | CWG1IE | NCO1IE | — | CCP1IE | TMR2IE | TMR1GIE | TMR1IE |
| 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 **CLC1IE:** CLC1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **CWG1IE:** CWG1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **NCO1IE:** NCO1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **Unimplemented:** Read as ‘0’

bit 3 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **TMR2IE:** TMR2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR1GIE:** TMR1 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **TMR1IE:** TMR1 Interrupt Enable bit
1 = Enabled
0 = Disabled

### 13.3.8 ERASING THE DATA EEPROM MEMORY

Data EEPROM Memory can be erased by writing 0xFF to all locations in the Data EEPROM Memory that needs to be erased.

**EXAMPLE 13-7: DATA EEPROM REFRESH ROUTINE**

```
        CLRF     NVMADRL                ; Start at address 0
        BCF      NVMCON1, CFGS          ; Set for memory
        BCF      NVMCON1, EEPGD         ; Set for Data EEPROM
        BCF      INTCON0, GIE           ; Disable interrupts
        BSF      NVMCON1, WREN          ; Enable writes
Loop                                    ; Loop to refresh array
        BSF      NVMCON1, RD            ; Read current address
        MOVLW    55h                    ;
        MOVWF    NVMCON2                ; Write 55h
        MOVLW    0AAh                   ;
        MOVWF    NVMCOM2                ; Write 0AAh
        BSF      NVMCON1, WR            ; Set WR bit to begin write
        BTFSC    NVMCON1, WR            ; Wait for write to complete
        BRA      $-2
        INCFSZ   NVMADRL, F             ; Increment address
        BRA      LOOP                   ; Not zero, do it again

        BCF      NVMCON1, WREN          ; Disable writes
        BSF      INTCON0, GIE           ; Enable interrupts
```

### 15.8.4 OVERRUN INTERRUPT

When the DMA receives a trigger to start a new message before the current message is completed, then the DMAxORIF Overrun interrupt flag is set.

This condition indicates that the DMA is being requested before its current transaction is finished. This implies that the active DMA may not be able to keep up with the demands from the peripheral module being serviced, which may result in data loss.

The DMAxORIF flag being set does not cause the current DMA transfer to terminate.

The Overrun interrupt is only available for trigger sources that are edge based and not available for sources that are level-based. Therefore a level-based interrupt source does not trigger a DMA overrun error due to the potential latency issues in the system.

An example of an interrupt that could use the overrun interrupt would be a timer overflow (or period match) interrupt. This event only happens every time the timer rolls over and is not dependent on any other system conditions.

An example of an interrupt that does not allow the overrun interrupt would be the UARTTX buffer. The UART will continue to assert the interrupt until the DMA is able to process the MSG. Due to latency issues, the DMA may not be able to service an empty buffer immediately, but the UART continues to assert its transmit interrupt until it is serviced. If overrun was allowed in this case, the overrun would occur almost immediately as the module samples the interrupt sources every instruction cycle.

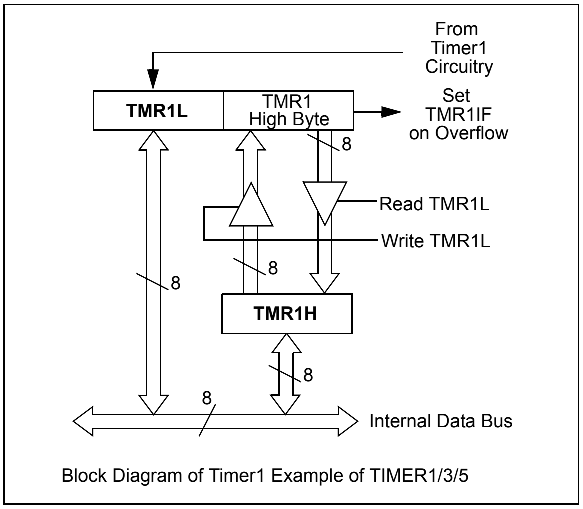
## 15.9 DMA Setup and Operation

The following steps illustrate how to configure the DMA for data transfer:

1. Program the appropriate Source and Destination addresses for the transaction into the DMAxSSA and DMAxDSA registers
2. Select the source memory region that is being addressed by DMAxSSA register, using the SMR<1:0> bits.
3. Program the SMODE and DMODE bits to select the addressing mode.
4. Program the Source size DMAxSSZ and Destination size DMAxDSZ registers with the number of bytes to be transferred. It is recommended for proper operation that the size registers be a multiple of each other.
5. If the user desires to disable data transfers once the message has completed, then the SSTP and DSTP bits in DMAxCON0 register need to be set. (see **Section 15.5.3.2 "Source/Destination Stop"**).
6. If using hardware triggers for data transfer, setup the hardware trigger interrupt sources for the starting and aborting DMA transfers (DMAxSIRQ and DMAxAIRQ), and set the corresponding interrupt request enable bits (SIRQEN and AIRQEN).
7. Select the priority level for the DMA (see **Section 3.1 "System Arbitration"**) and lock the priorities (see **Section 3.1.1 "Priority Lock"**)
8. Enable the DMA (DMAxCON1bits. EN = `1`)
9. If using software control for data transfer, set the DGO bit, else this bit will be set by the hardware trigger.

Once the DMA is set up, the following flow chart describes the sequence of operation when the DMA uses hardware triggers and utilizes the unused CPU cycles (bubble) for DMA transfers.

FIGURE 21-2: TIMER1/3/5 16-BIT READ/WRITE MODE BLOCK DIAGRAM



## 21.6 Timer1/3/5 Gate

Timer1/3/5 can be configured to count freely or the count can be enabled and disabled using Timer1/3/5 gate circuitry. This is also referred to as Timer1/3/5 gate enable.

Timer1/3/5 gate can also be driven by multiple selectable sources.

### 21.6.1 TIMER1/3/5 GATE ENABLE

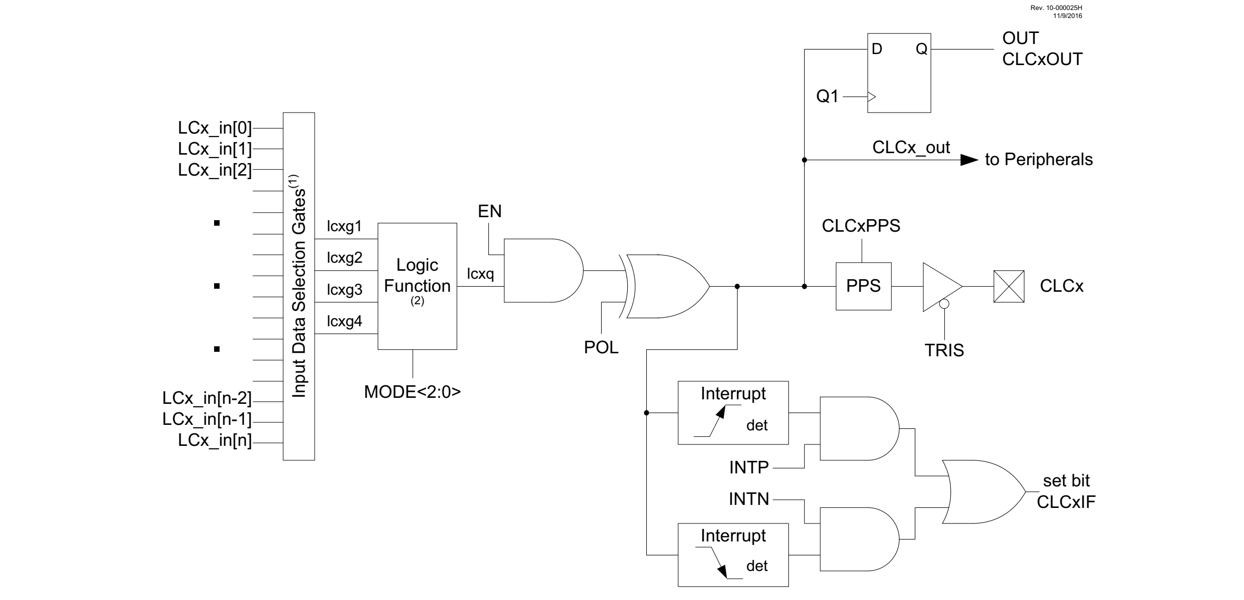
The Timer1/3/5 Gate Enable mode is enabled by setting the TMRxGE bit of the TxGCON register. The polarity of the Timer1/3/5 Gate Enable mode is configured using the TxGPOL bit of the TxGCON register.

When Timer1/3/5 Gate Enable mode is enabled, Timer1/3/5 will increment on the rising edge of the Timer1/3/5 clock source. When Timer1/3/5 Gate signal is inactive, the timer will not increment and hold the current count. See Figure 21-4 for timing details.

TABLE 21-2: TIMER1/3/5 GATE ENABLE SELECTIONS

| TMRxCLK | TxGPOL | TxG | Timer1/3/5 Operation |
|---|---|---|---|
| ↑ | 1 | 1 | Counts |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 0 | 0 | Counts |

**FIGURE 27-1: CLCx SIMPLIFIED BLOCK DIAGRAM**



**Note 1:** See Figure 27-2: Input Data Selection and Gating

**2:** See Figure 27-3: Programmable Logic Functions.

## 27.1 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.

### 27.1.1 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 27-2. Data inputs in the figure are identified by a generic numbered input name.

Table 27-1 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 27-3 through Register 27-6).

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

## 31.13 Checksum (UART1 only)

This section does not apply to the LIN mode, which handles checksums automatically.

The transmit and receive checksum adders are enabled when the C0EN bit in the UxCON2 register is set. When enabled, the adders accumulate every byte that is transmitted or received. The accumulated sum includes the carry of the addition. Software is responsible for clearing the checksum registers before a transaction and performing the check at the end of the transaction.

The following is an example of how the checksum registers could be used in the asynchronous modes.

### 31.13.1 TRANSMIT CHECKSUM METHOD

1. Clear the UxTXCHK register.
2. Set the C0EN bit.
3. Send all bytes of the transaction output.
4. Invert UxTXCHK and send the result as the last byte of the transaction.

### 31.13.2 RECEIVE CHECKSUM METHOD

1. Clear the UxRXCHK register.
2. Set the C0EN bit.
3. Receive all bytes in the transaction including the checksum byte.
4. Set MSb of UxRXCHK if 7-bit mode is selected.
5. Add 1 to UxRXCHK.
6. If the result is ‘0’, the checksum passes, otherwise it fails.

The CERIF checksum interrupt flag is not active in any mode other than LIN.

## 31.14 Collision Detection

External forces that interfere with the transmit line are detected in all modes of operation with collision detection. Collision detection is always active when RXEN and TXEN are both set.

When the receive input is connected to the transmit output through either the same I/O pin or external circuitry, a character will be received for every character transmitted. The collision detection circuit provides a warning when the word received does not match the word transmitted.

The TXCIF flag in the UxERRIR register is used to signal collisions. This signal is only useful when the TX output is looped back to the RX input and everything that is transmitted is expected to be received. If more than one transmitter is active at the same time, it can be assumed that the TX word will not match the RX word. The TXCIF detects this mismatch and flags an interrupt. The TXCIF bit will also be set in DALI mode transmissions when the received bit is missing the expected mid-bit transition.

Collision detection is always active, regardless of whether or not the RX input is connected to the TX output. It is up to the user to disable the TXCIE bit when collision interrupts are not required.

The software overhead of unloading the receive buffer of transmitted data is avoided by setting the RUNOVF bit in UxCON2 and ignoring the receive interrupt and letting the receive buffer overflow. When the transmission is complete, prepare for receiving data by flushing the receive buffer (see Section 31.11.2, FIFO Reset) and clearing the RXFOIF overflow flag in the UxERRIR register.

## 31.15 RX/TX Activity Timeout

The UART works in conjunction with the HLT timers to monitor activity on the RX and TX lines. Use this feature to determine when there has been no activity on the receive or transmit lines for a user specified period of time.

To use this feature, set the HLT to the desired timeout period by a combination of the HLT clock source, timer prescale value, and timer period registers. Configure the HLT to reset on the UART TX or RX line and start the HLT at the same time the UART is started. UART activity will keep resetting the HLT to prevent a full HLT period from elapsing. When there has been no activity on the selected TX or RX line for longer than the HLT period then an HLT interrupt will occur signaling the timeout event.

For example, the following register settings will configure HLT2 for a 5 ms timeout of no activity on U1RX:

- T2PR = 0x9C (156 prescale periods)
- T2CLKCON = 0x05 (500 kHz internal oscillator)
- T2HLT = 0x04 (free running, reset on rising edge)
- T2RST = 0x15 (reset on U1RX)
- T2CON = 0xC0 (Timer2 on with 1:16 prescale)

## 32.4 Transfer Counter

In all master modes, the transfer counter can be used to determine how many data transfers the SPI will send/receive. The transfer counter is comprised of the SPIxTCTH/L set of registers, and is also partially controlled by the SPIxTWIDTH register. The Transfer Counter has two primary modes, determined by the BMODE bit of the SPIxCON0 register. Each mode uses the SPIxTCTH/L and SPIxTWIDTH registers to determine the number and size of the transfers. In both modes, when the transfer counter reaches zero, the TCZIF interrupt flag is set.

> **Note:** When BMODE=1 in all master modes (and at all times in slave modes), the Transfer Counter will still decrement as transfers occur and can be used to count the number of messages sent/received, as well as to control SS(out) and to trigger TCZIF. Also when BMODE = 1, the SPIxTWIDTH register can be used in Master and Slave modes to determine the size of messages sent and received by the SPI, even if the Transfer Counter is not being actively used to control the number of messages being sent/received by the SPI module.

### 32.4.1 TOTAL BIT COUNT MODE (BMODE = 0)

In this mode, SPIxTCTH/L and SPIxTWIDTH are concatenated to determine the total number of bits to be transferred. These bits will be loaded from/into the transmit/receive FIFOs in 8-bit increments and the transfer counter will be decremented by eight until the total number of remaining bits is less than eight. If there are any remaining bits (SPIxTWIDTH ≠ 0), the transmit FIFO will send out one final message with any extra bits greater than the remainder ignored. The SPIxTWIDTH is the remaining bit count but the value does not change as it does for the SPIxTCT value. Similarly, the receiver will load a final byte into the receiver FIFO, and pad the extra bits with zeros. The LSBF bit of SPIxCON0 determines whether the Most Significant or Least Significant bits of this final byte are ignored/padded. For example, when LSBF = 0 and the final transfer contains only two bits then if the last byte sent was 5Fh then the RXB of the receiver will contain 40h which are the two MSbits of the final byte padded with zeros in the LSbits.

In this mode, the SPI master will only transmit messages when the SPIxTCT value is greater than zero, regardless of TXR and RXR settings. In Master Transmit mode, the transfer starts with the data write to the SPIxTXB register or the count value written to the SPIxTCTL register, which ever occurs last. In Master Receive-only mode, the transfer clocks start when the SPIxTCTL value is written. Transfer clocks are suspended when the receive FIFO is full and resume as the FIFO is read.

### 32.4.2 VARIABLE TRANSFER SIZE MODE (BMODE = 1)

In this mode, SPIxTWIDTH specifies the width of every individual piece of the data transfer in bits. SPIxTCTH/SPIxTCTL specifies the number of transfers of this bit length. If SPIxTWIDTH = 0, each piece is a full byte of data. If SPIxTWIDTH ≠ 0, then only the specified number of bits from the transmit FIFO are shifted out, with the unused bits ignored. Received data is padded with zeros in the unused bit areas when transfered into the receive FIFO. The LSBF bit of SPIxCON0 determines whether the Most Significant or Least Significant bits of the transfers are ignored/padded. In this mode, the transfer counter being zero only stops messages from being sent/received when in “Receive only” mode.

> **Note:** With BMODE = 1, it is possible for the transfer counter (SPIxTCTH/L) to decrement below zero, although when in “Receive only” Master mode, transfer clocks will cease when the transfer counter reaches zero.

### REGISTER 33-11: I2CxPIE: I2CxIE INTERRUPT AND HOLD ENABLE REGISTER

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| CNTIE | ACKTIE | — | WRIE | ADRIE | PCIE | RSCIE | SCIE |
| 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 | HS = Hardware set HC = Hardware clear |

bit 7 **CNTIE:** Byte Count Interrupt Enable bit
1 = When CNTIF is set
0 = Byte count interrupts are disabled

bit 6 **ACKTIE:** Acknowledge Interrupt and Hold Enable bit
1 = When ACKTIF is set
If ACK is generated, CSTR is also set.
If NACK is generated, CSTR is unchanged
0 = Acknowledge holding and interrupt is disabled

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **WRIE:** Data Write Interrupt and Hold Enable bit
1 = When WRIF is set; CSTR is set
0 = Data Write holding and interrupt is disabled

bit 3 **ADRIE:** Address Interrupt and Hold Enable bit
1 = When ADRIF is set; CSTR is set
0 = Address holding and interrupt is disabled

bit 2 **PCIE:** Stop Condition Interrupt Enable
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled

bit 1 **RSCIE:** Restart Condition Interrupt Enable
1 = Enable interrupt on detection of Restart condition
0 = Start detection interrupts are disabled

bit 0 **SCIE:** Start Condition Interrupt Enable
1 = Enable interrupt on detection of Start condition
0 = Start detection interrupts are disabled

**Note 1:** Enabled interrupt flags are OR’d to produce the PIRx<I2CxIF> bit.

### 36.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIRx register. The ADC Interrupt Enable is the ADIE bit in the PIEx register. The ADIF bit must be cleared in software.

> **Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
>
> **2:** The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake up from Sleep and resume in-line code execution, the ADIE bit of the PIEx register and the GIE bits of the INTCON0 register must both be set. If all these bits are set, the execution will switch to the Interrupt Service Routine.

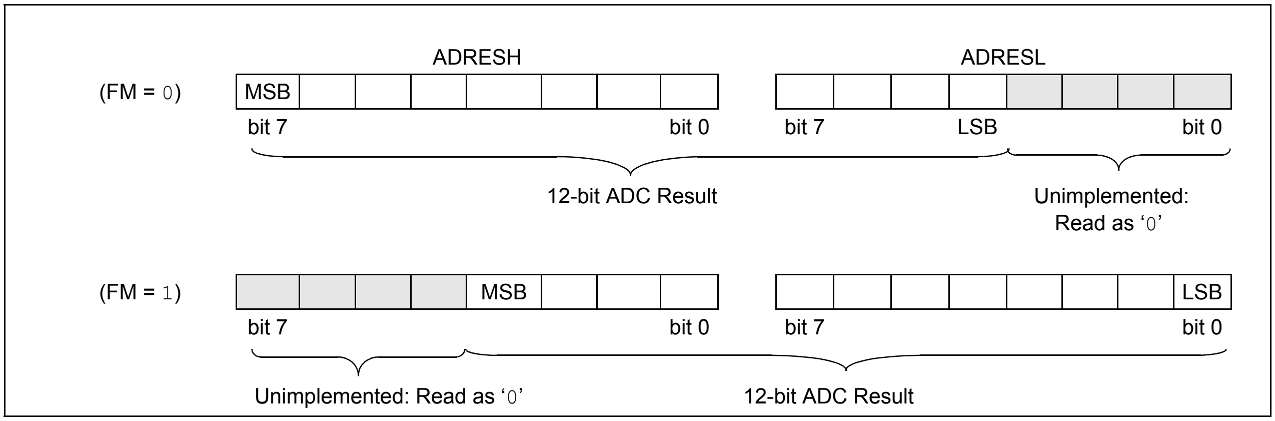
### 36.1.6 RESULT FORMATTING

The 12-bit ADC conversion result can be supplied in two formats, left justified or right justified. The FM bits of the ADCON0 register controls the output format.

Figure 36-3 shows the two output formats.

Writes to the ADRES register pair are always right justified regardless of the selected format mode. Therefore, data read after writing to ADRES when ADFRM0 = 0 will be shifted left four places.

**FIGURE 36-3: 12-BIT ADC CONVERSION RESULT FORMAT**



## 36.4 ADC Charge Pump

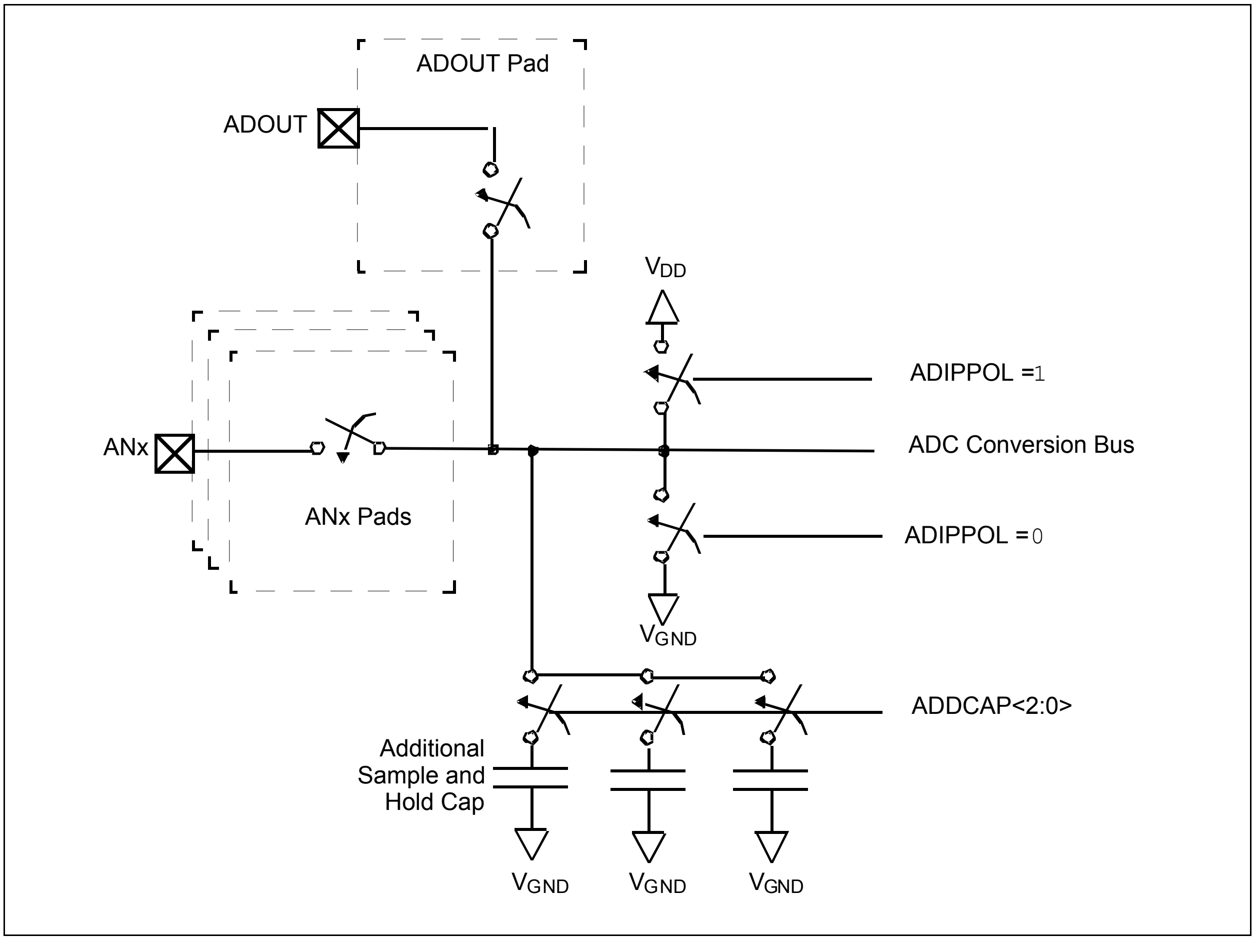
The ADC module has a dedicated charge pump which can be controlled through the ADCP register (Register 36-36). The primary purpose of the charge pump is to supply a constant voltage to the gates of transistor devices in the A/D converter, signal and reference input pass-gates, to prevent degradation of transistor performance at low operating voltage.

The charge pump can be enabled by setting the CPON bit in the ADC register. Once enabled, the pump will undergo a start-up time to stabilize the charge pump output. Once the output stabilizes and is ready for use, the CPRDY bit of the ADCP register will be set.

## 36.5 Capacitive Voltage Divider (CVD) Features

The ADC module contains several features that allow the user to perform a relative capacitance measurement on any ADC channel using the internal ADC sample and hold capacitance as a reference. This relative capacitance measurement can be used to implement capacitive touch or proximity sensing applications. Figure 36-6 shows the basic block diagram of the CVD portion of the ADC module.

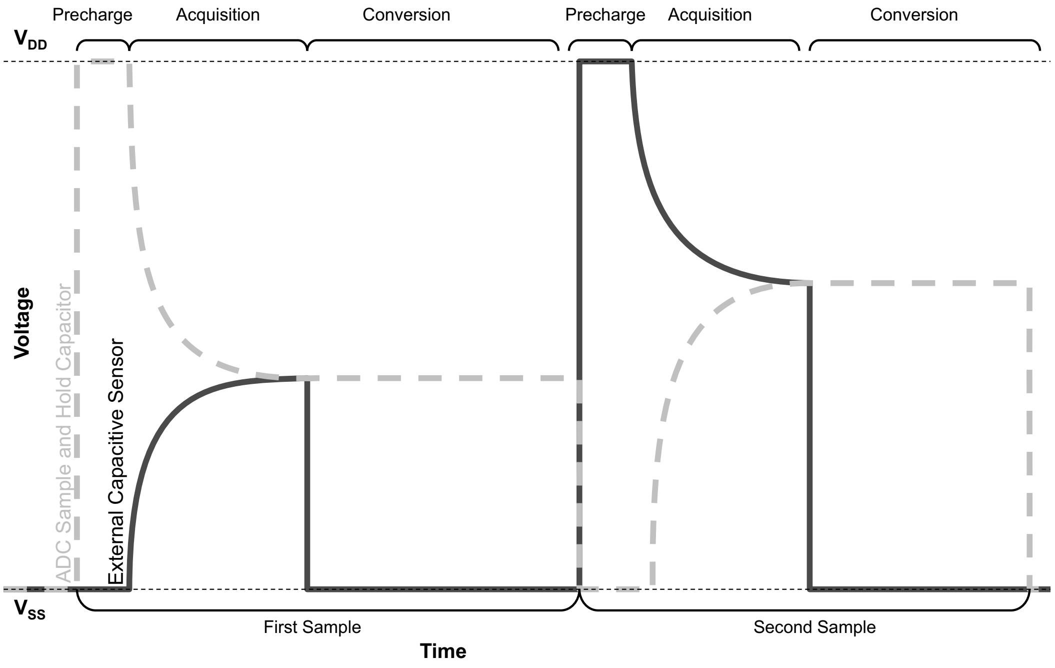
**FIGURE 36-6: HARDWARE CAPACITIVE VOLTAGE DIVIDER BLOCK DIAGRAM**



### 36.5.1 CVD OPERATION

A CVD operation begins with the ADC's internal sample and hold capacitor ($C_{HOLD}$) being disconnected from the path which connects it to the external capacitive sensor node. While disconnected, $C_{HOLD}$ is precharged to VDD or VSS, while the path to the sensor node is precharged to the level opposite that of CHOLD. When the precharge phase is complete, the VDD/VSS precharge paths for the two nodes are shut off and CHOLD and the path to the external sensor node are re-connected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged CHOLD and sensor nodes, which results in a final voltage level setting on CHOLD, which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on CHOLD. This process is then repeated with inverted precharge levels for both the CHOLD and external sensor nodes. Figure 36-7 shows the waveform for two inverted CVD measurements, which is known as differential CVD measurement.

**FIGURE 36-7: DIFFERENTIAL CVD MEASUREMENT WAVEFORM**



### REGISTER 36-8: ADPCH: ADC POSITIVE CHANNEL SELECTION 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 |
|---|---|---|---|---|---|---|---|
| — | — | ADPCH<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 **ADPCH<5:0>**: ADC Positive Input Channel Selection bits

111111 = FVR buffer 2[2]
111110 = FVR buffer 1[2]
111101 = DAC1 output[1]
111100 = Temperature indicator[3]
111011 = Vss (Analog Ground)
111010 = Reserved. No channel connected.
•
•
•
110000 = Reserved. No channel connected.
101111 = ANF7[4]
101110 = ANF6[4]
101101 = ANF5[4]
101100 = ANF4[4]
101011 = ANF3[4]
101010 = ANF2[4]
101001 = ANF1[4]
101000 = ANF0[4]
100111 = Reserved. No channel connected.
•
•
100011 = Reserved. No channel connected.
100010 = ANE2[5]
100001 = ANE1[5]
100000 = ANE0[5]
011111 = AND7[5]
011110 = AND6[5]
011101 = AND5[5]
011100 = AND4[5]
011011 = AND3[5]
011010 = AND2[5]
011001 = AND1[5]
011000 = AND0[5]

010111 = ANC7
010110 = ANC6
010101 = ANC5
010100 = ANC4
010011 = ANC3
010010 = ANC2
010001 = ANC1
010000 = ANC0
001111 = ANB7
001110 = ANB6
001101 = ANB5
001100 = ANB4
001011 = ANB3
001010 = ANB2
001001 = ANB1
001000 = ANB0
000111 = ANA7
000110 = ANA6
000101 = ANA5
000100 = ANA4
000011 = ANA3
000010 = ANA2
000001 = ANA1
000000 = ANA0

**Note 1:** See **Section 37.0 “5-Bit Digital-to-Analog Converter (DAC) Module”** for more information.
**2:** See **Section 34.0 “Fixed Voltage Reference (FVR)”** for more information.
**3:** See **Section 35.0 “Temperature Indicator Module”** for more information.
**4:** Reserved on PIC18(L)F26/27/45/46/47K42 parts.
**5:** Reserved on PIC18(L)F26/27K42 parts.

### REGISTER 36-18: ADRESH: ADC RESULT REGISTER HIGH, FM = 0

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |
| ADRES<11:4> | | | | | | | |
| 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 **ADRES<11:4>**: ADC Result Register bits
Upper eight bits of 12-bit conversion result.

### REGISTER 36-19: ADRESL: ADC RESULT REGISTER LOW, FM = 0

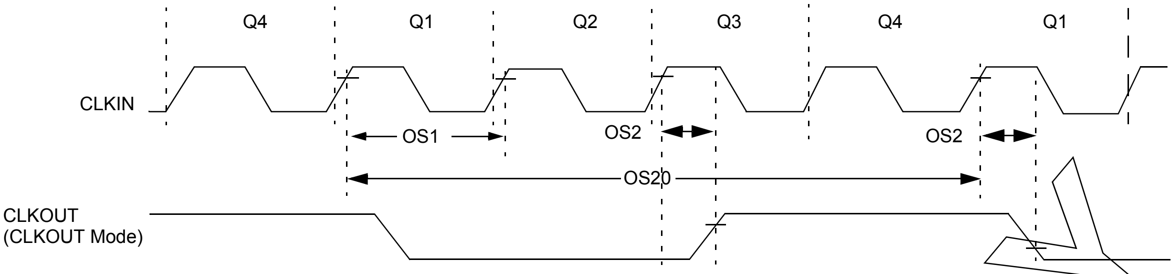
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | U-0 | U-0 | U-0 | U-0 |
| ADRES<3:0> | | | | — | — | — | — |
| bit 7 | | | | | | | bit 0 |

| | | |
|---|---|---|
| **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 7-4 **ADRES<3:0>**: ADC Result Register bits. Lower four bits of 12-bit conversion result.

bit 3-0 **Reserved**

## FIGURE 44-5: CLOCK TIMING



**Note 1:** See Table 44-9.

## TABLE 44-9: EXTERNAL CLOCK/OSCILLATOR TIMING REQUIREMENTS

**Standard Operating Conditions (unless otherwise stated)**

| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| **ECL Oscillator** | | | | | | | |
| OS1 | $F_{ECL}$ | Clock Frequency | — | — | 500 | kHz | |
| OS2 | $T_{ECL\_DC}$ | Clock Duty Cycle | 40 | — | 60 | % | |
| **ECM Oscillator** | | | | | | | |
| OS3 | $F_{ECM}$ | Clock Frequency | — | — | 8 | MHz | |
| OS4 | $T_{ECM\_DC}$ | Clock Duty Cycle | 40 | — | 60 | % | |
| **ECH Oscillator** | | | | | | | |
| OS5 | $F_{ECH}$ | Clock Frequency | — | — | 64 | MHz | |
| OS6 | $T_{ECH\_DC}$ | Clock Duty Cycle | 40 | — | 60 | % | |
| **LP Oscillator** | | | | | | | |
| OS7 | $F_{LP}$ | Clock Frequency | — | — | 100 | kHz | **Note 4** |
| **XT Oscillator** | | | | | | | |
| OS8 | $F_{XT}$ | Clock Frequency | — | — | 4 | MHz | **Note 4** |
| **HS Oscillator** | | | | | | | |
| OS9 | $F_{HS}$ | Clock Frequency | — | — | 20 | MHz | **Note 4** |
| **Secondary Oscillator** | | | | | | | |
| OS10 | $F_{SEC}$ | Clock Frequency | 32.4 | 32.768 | 33.1 | kHz | |
| **System Oscillator** | | | | | | | |
| OS20 | $F_{OSC}$ | System Clock Frequency | — | — | 64 | MHz | **(Note 2, Note 3)** |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

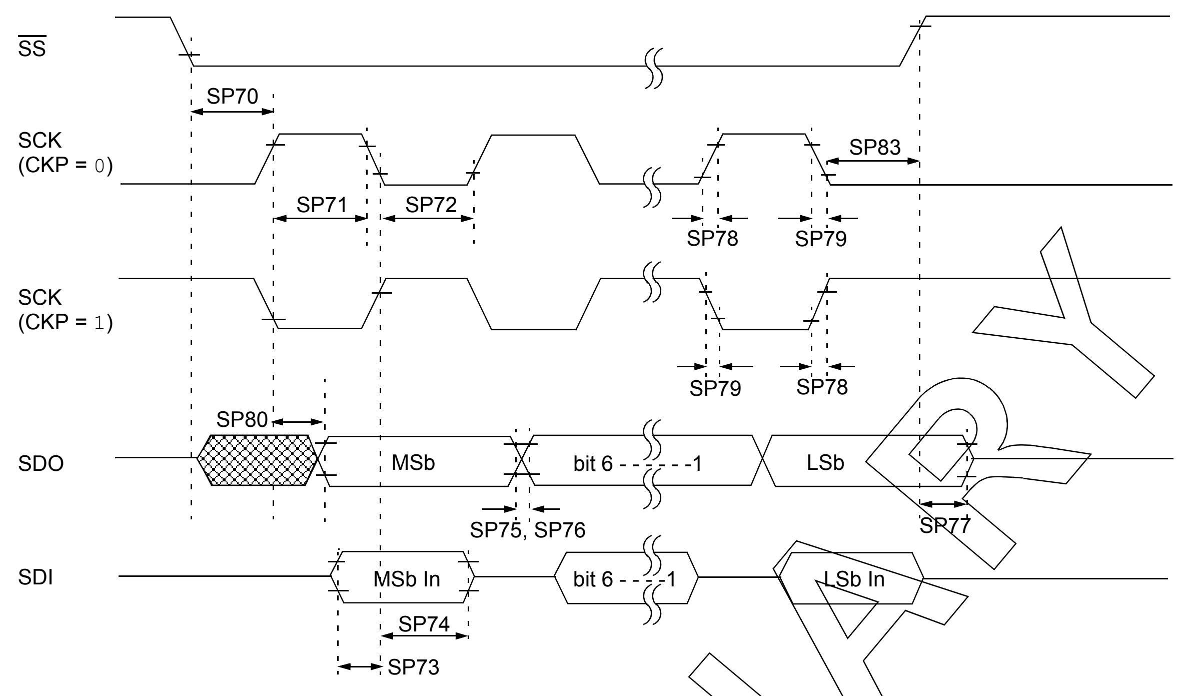
**2:** The system clock frequency (FOSC) is selected by the "main clock switch controls" as described in **Section 10.0 "Power-Saving Operation Modes"**.

**3:** The system clock frequency (FOSC) must meet the voltage requirements defined in the **Section 44.2 "Standard Operating Conditions"**.

**4:** LP, XT and HS oscillator modes require an appropriate crystal or resonator to be connected to the device. For clocking the device with the external square wave, one of the EC mode selections must be used.

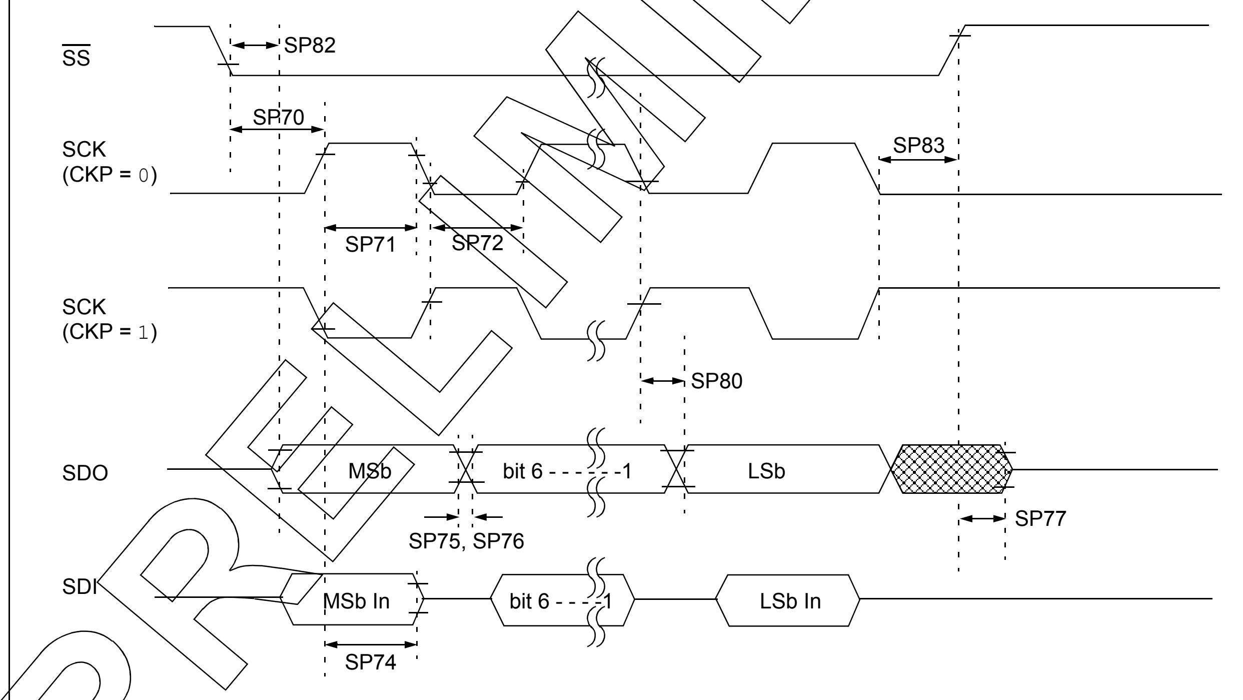
PRELIMINARY

**FIGURE 44-16: SPI SLAVE MODE TIMING (CKE = 0)**



**Note:** Refer to Figure 44-4 for load conditions.

**FIGURE 44-17: SPI SLAVE MODE TIMING (CKE = 1)**



**Note:** Refer to Figure 44-4 for load conditions.

PRELIMINARY