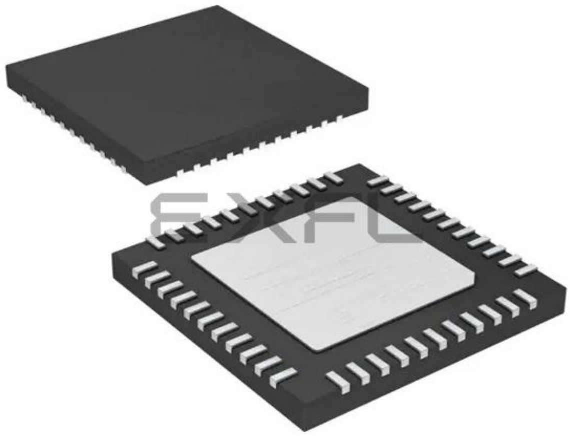
# Microchip Technology - PIC18F46K42T-I/ML Datasheet



## 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 | 36 |
| Program Memory Size | 64KB (32K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 35x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-VQFN Exposed Pad |
| Supplier Device Package | 44-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k42t-i-ml |

## Table of Contents

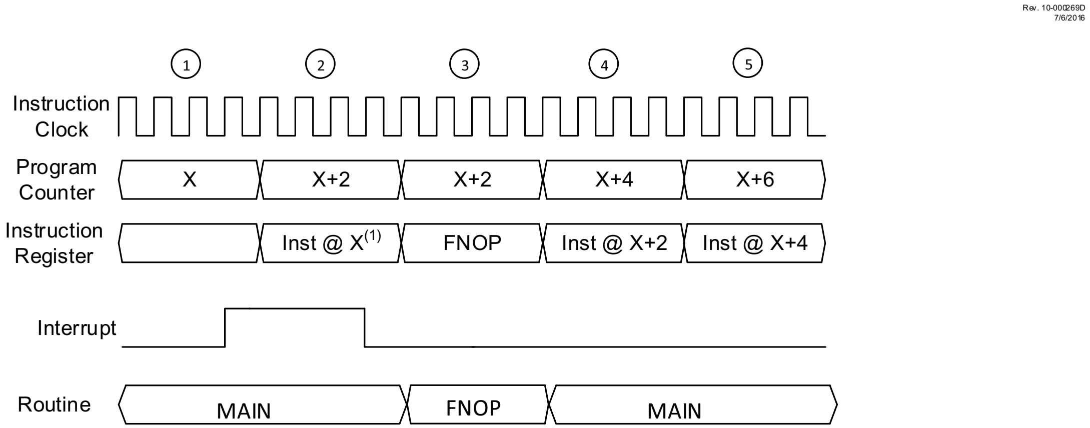
### 9.7.1 ABORTING INTERRUPTS

If the last instruction before the interrupt controller vectors to the ISR from main routine clears the GIE, PIE or PIR bit associated with the interrupt, the controller executes one force `NOP` cycle before it returns to the main routine.

Figure 9-10 illustrates the sequence of events when a peripheral interrupt is asserted and then cleared on the last executed instruction cycle.

If the GIE, PIE or PIR bit associated with the interrupt is cleared prior to vectoring to the ISR, then the controller continues executing the main routine.

**FIGURE 9-10: INTERRUPT TIMING DIAGRAM - ABORTING INTERRUPTS**



**Note 1:** Inst @ X clears the interrupt flag, Example BCF INTCON0, GIE.

Example 12-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 12-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES<3:0>).

### EQUATION 12-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned}
\mathrm{RES3{:}RES0} &= \mathrm{ARG1H{:}ARG1L} \bullet \mathrm{ARG2H{:}ARG2L} \\
&= (\mathrm{ARG1H} \bullet \mathrm{ARG2H} \bullet 2^{16}) + \\
&\quad (\mathrm{ARG1H} \bullet \mathrm{ARG2L} \bullet 2^{8}) + \\
&\quad (\mathrm{ARG1L} \bullet \mathrm{ARG2H} \bullet 2^{8}) + \\
&\quad (\mathrm{ARG1L} \bullet \mathrm{ARG2L})
\end{aligned}$$

### EXAMPLE 12-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 12-4 shows the sequence to do a 16 x 16 signed multiply. Equation 12-2 shows the algorithm used. The 32-bit result is stored in four registers (RES<3:0>). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

### EQUATION 12-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$\begin{aligned}
\mathrm{RES3{:}RES0} &= \mathrm{ARG1H{:}ARG1L} \bullet \mathrm{ARG2H{:}ARG2L} \\
&= (\mathrm{ARG1H} \bullet \mathrm{ARG2H} \bullet 2^{16}) + \\
&\quad (\mathrm{ARG1H} \bullet \mathrm{ARG2L} \bullet 2^{8}) + \\
&\quad (\mathrm{ARG1L} \bullet \mathrm{ARG2H} \bullet 2^{8}) + \\
&\quad (\mathrm{ARG1L} \bullet \mathrm{ARG2L}) + \\
&\quad (-1 \bullet \mathrm{ARG2H}{<}7{>} \bullet \mathrm{ARG1H{:}ARG1L} \bullet 2^{16}) + \\
&\quad (-1 \bullet \mathrm{ARG1H}{<}7{>} \bullet \mathrm{ARG2H{:}ARG2L} \bullet 2^{16})
\end{aligned}$$

### EXAMPLE 12-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1       ; no, check ARG1
    MOVF    ARG1L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG1H, W        ;
    SUBWFB  RES3
;
SIGN_ARG1
    BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
    BRA     CONT_CODE       ; no, done
    MOVF    ARG2L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG2H, W        ;
    SUBWFB  RES3
;
CONT_CODE
    :
```

### REGISTER 14-11: SCANCON0: SCANNER ACCESS CONTROL REGISTER 0

| R/W-0/0 | R/W-0/0 | R/W/HC-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| EN | TRIGEN | SGO | — | — | MREG | BURSTMD | BUSY |
| 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 **EN:** Scanner Enable bit(1)
1 = Scanner is enabled
0 = Scanner is disabled

bit 6 **TRIGEN:** Scanner Trigger Enable bit(2)
1 = Scanner trigger is enabled
0 = Scanner trigger is disabled
Refer Table 14-1.

bit 5 **SGO:** Scanner GO bit(3, 4)
1 = When the CRC is ready, the Memory region set by the MREG bit will be accessed and data is passed to the CRC peripheral.
0 = Scanner operations will not occur

bit 4-3 **Unimplemented:** Read as ‘0’

bit 2 **MREG:** Scanner Memory Region Select bit(2)
1 = Scanner address points to Data EEPROM
0 = Scanner address points to Program Flash Memory

bit 1 **BURSTMD:** Scanner Burst Mode bit
1 = Memory access request to the CPU Arbiter is always true
0 = Memory access request to the CPU Arbiter is dependent on the CRC request and Trigger
Refer Table 14-1.

bit 0 **BUSY:** Scanner Busy Indicator bit
1 = Scanner cycle is in process
0 = Scanner cycle is compete (or never started)

**Note 1:** Setting EN = 1 (SCANCON0 register) does not affect any other register content.
**2:** Scanner trigger selection can be set using the SCANTRIG register.
**3:** This bit can be cleared in software. It is cleared in hardware when LADR>HADR (and a data cycle is not occurring) or when CRCGO = 0 (CRCCON0 register).
**4:** CRCEN and CRCGO bits (CRCCON0 register) must be set before setting the SGO bit.

## 17.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 17-1.

The peripheral input is selected with the peripheral xxxPPS register (Register 17-1), and the peripheral output is selected with the PORT RxyPPS register (Register 17-2). For example, to select PORTC<7> as the UART1 RX input, set U1RXPPS to `0b1 0111`, and to select PORTC<6> as the UART1 TX output set RC6PPS to `0b01 0011`.

### 17.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has analog functions associated, the ANSEL bit for that pin must be cleared to enable the digital input buffer.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 17-1.

> **Note:** The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, INT0PPS.

### 17.2 PPS Outputs

Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- UART

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 17-2.

> **Note:** The notation “Rxy” is a place holder for the pin identifier. For example, RA0PPS.

## 20.3 Programmable Prescaler

A software programmable prescaler is available for exclusive use with Timer0. There are 16 prescaler options for Timer0 ranging in powers of two from 1:1 to 1:32768. The prescaler values are selected using the CKPS<3:0> bits of the T0CON1 register.

The prescaler is not directly readable or writable. Clearing the prescaler register can be done by writing to the TMR0L register or to the T0CON0/T0CON1 register or by any Reset.

## 20.4 Programmable Postscaler

A software programmable postscaler (output divider) is available for exclusive use with Timer0. There are 16 postscaler options for Timer0 ranging from 1:1 to 1:16. The postscaler values are selected using the OUTPS bits of the T0CON0 register.

The postscaler is not directly readable or writable. Clearing the postscaler register can be done by writing to the TMR0L register or to the T0CON0/T0CON1 register or by any Reset.

## 20.5 Operation During Sleep

When operating synchronously, Timer0 will halt. When operating asynchronously, Timer0 will continue to increment and wake the device from Sleep (if Timer0 interrupts are enabled) provided that the input clock source is active.

## 20.6 Timer0 Interrupts

The Timer0 interrupt flag bit (TMR0IF) is set when either of the following conditions occur:

- 8-bit TMR0L matches the TMR0H value
- 16-bit TMR0 rolls over from ‘FFFFh’

When the postscaler bits (OUTPS) are set to 1:1 operation (no division), the T0IF flag bit will be set with every TMR0 match or rollover. In general, the TMR0IF flag bit will be set every OUTPS +1 matches or rollovers.

If Timer0 interrupts are enabled (TMR0IE bit of the PIE3 register = ‘1’), the CPU will be interrupted and the device may wake from Sleep (see **Section 20.2 “Clock Source Selection”** for more details).

## 20.7 Timer0 Output

The Timer0 output can be routed to any I/O pin via the RxyPPS output selection register (see **Section 17.0 “Peripheral Pin Select (PPS) Module”** for additional information). The Timer0 output can also be used by other peripherals, such as the auto-conversion trigger of the Analog-to-Digital Converter. Finally, the Timer0 output can be monitored through software via the Timer0 output bit (OUT) of the T0CON0 register (Register 20-1).

TMR0_out will be a pulse of one postscaled clock period when a match occurs between TMR0L and PR0 (Period register for TMR0) in 8-bit mode, or when TMR0 rolls over in 16-bit mode. The Timer0 output is a 50% duty cycle that toggles on each TMR0_out rising clock edge.

### REGISTER 21-2: TxGCON: TIMERx GATE CONTROL REGISTER

| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R-x | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | — | — |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **GE:** Timerx Gate Enable bit

If TMRxON = 1:
1 = Timerx counting is controlled by the Timerx gate function
0 = Timerx is always counting
If TMRxON = 0:
This bit is ignored

bit 6 **GPOL:** Timerx Gate Polarity bit

1 = Timerx gate is active-high (Timerx counts when gate is high)
0 = Timerx gate is active-low (Timerx counts when gate is low)

bit 5 **GTM:** Timerx Gate Toggle Mode bit

1 = Timerx Gate Toggle mode is enabled
0 = Timerx Gate Toggle mode is disabled and Toggle flip-flop is cleared
Timerx Gate Flip Flop Toggles on every rising edge

bit 4 **GSPM:** Timerx Gate Single Pulse Mode bit

1 = Timerx Gate Single Pulse mode is enabled and is controlling Timerx gate)
0 = Timerx Gate Single Pulse mode is disabled

bit 3 **GGO/$\overline{\text{DONE}}$:** Timerx Gate Single Pulse Acquisition Status bit

1 = Timerx Gate Single Pulse Acquisition is ready, waiting for an edge
0 = Timerx Gate Single Pulse Acquisition has completed or has not been started.
This bit is automatically cleared when TxGSPM is cleared.

bit 2 **GVAL:** Timerx Gate Current State bit

Indicates the current state of the Timerx gate that could be provided to TMRxH:TMRxL
Unaffected by Timerx Gate Enable (TMRxGE)

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

## 22.0 TIMER2/4/6 MODULE

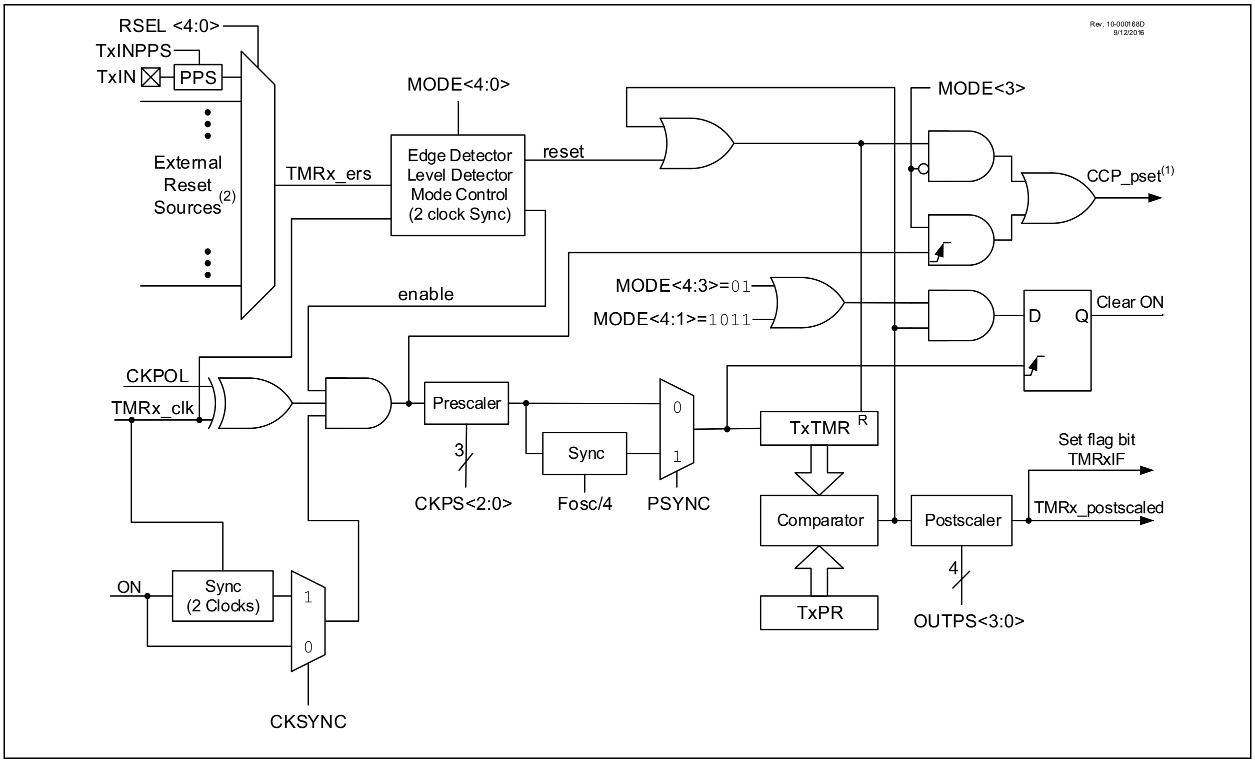
The Timer2/4/6 modules are 8-bit timers that can operate as free-running period counters or in conjunction with external signals that control start, run, freeze, and reset operation in One-Shot and Monostable modes of operation. Sophisticated waveform control such as pulse density modulation are possible by combining the operation of these timers with other internal peripherals such as the comparators and CCP modules. Features of the timer include:

- 8-bit timer register
- 8-bit period register
- Selectable external hardware timer resets
- Programmable prescaler (1:1 to 1:128)
- Programmable postscaler (1:1 to 1:16)
- Selectable synchronous/asynchronous operation
- Alternate clock sources
- Interrupt on period
- Three modes of operation:
  - Free Running Period
  - One-Shot
  - Monostable

See Figure 22-1 for a block diagram of Timer2. See Figure 22-2 for the clock source block diagram.

> **Note:** Three identical Timer2 modules are implemented on this device. The timers are named Timer2, Timer4, and Timer6. All references to Timer2 apply as well to Timer4 and Timer6. All references to T2PR apply as well to T4PR and T6PR.

**FIGURE 22-1: TIMER2 BLOCK DIAGRAM**



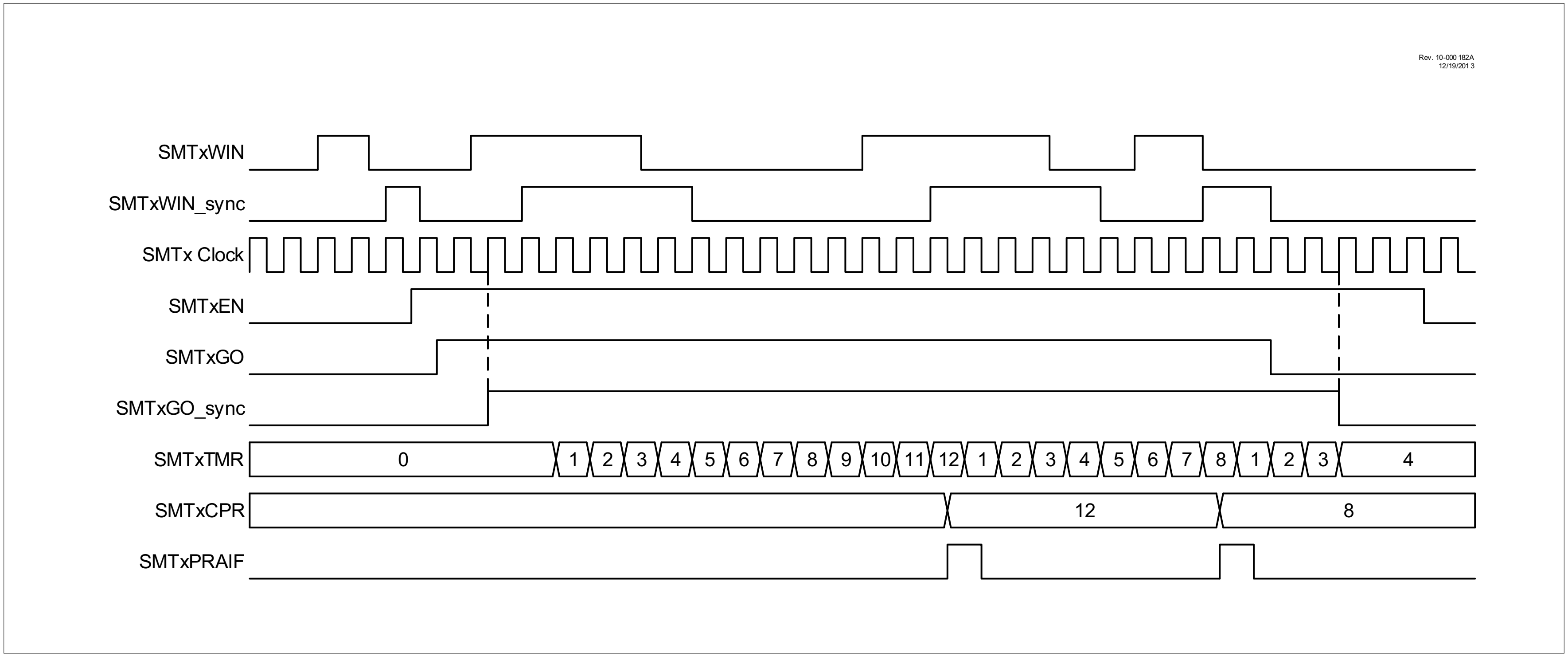
### 25.6.4 HIGH AND LOW MEASURE MODE

This mode measures the high and low pulse time of the SMTSIGx relative to the SMT clock. It begins incrementing the SMT1TMR on a rising edge on the SMTSIGx input, then updates the SMT1CPW register with the value and resets the SMT1TMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMT1CPR register with its current value and once again resets the SMT1TMR value and begins incrementing again. See Figure 25-8 and Figure 25-9.

### 25.6.5 WINDOWED MEASURE MODE

This mode measures the window duration of the SMTWINx input of the SMT. It begins incrementing the timer on a rising edge of the SMTWINx input and updates the SMT1CPR register with the value of the timer and resets the timer on a second rising edge. See Figure 25-10 and Figure 25-11.

### FIGURE 25-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



## 25.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

### 25.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMT1CPW and SMT1CPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMT1CPW interrupt is controlled by SMT1PWAIF and SMT1PWAIE bits in the respective PIR and PIE registers. The SMT1CPR interrupt is controlled by the SMT1PRAIF and SMT1PRAIE bits, also located in the respective PIR and PIE registers.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMT1CLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

### 25.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 25.1.2 "Period Match interrupt"**, the SMT will also interrupt upon SMT1TMR, matching SMT1PR with its period match limit functionality described in **Section 25.3 "Halt Operation"**. The period match interrupt is controlled by SMT1IF and SMT1IE, located in the respective PIR and PIE registers.

## 28.8 NCO Control Registers

### REGISTER 28-1: NCO1CON: NCO CONTROL REGISTER

| R/W-0/0 | U-0 | R-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| EN | — | OUT | POL | — | — | — | PFM |
| 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:** NCO1 Enable bit
1 = NCO1 module is enabled
0 = NCO1 module is disabled

bit 6 **Unimplemented**: Read as ‘0’

bit 5 **OUT:** NCO1 Output bit
Displays the current output value of the NCO1 module.

bit 4 **POL:** NCO1 Polarity
1 = NCO1 output signal is inverted
0 = NCO1 output signal is not inverted

bit 3-1 **Unimplemented**: Read as ‘0’

bit 0 **PFM:** NCO1 Pulse Frequency Mode bit
1 = NCO1 operates in Pulse Frequency mode
0 = NCO1 operates in Fixed Duty Cycle mode, divide by 2

### REGISTER 31-10: UxRXB: UART RECEIVE REGISTER

| R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 | R-0/0 |
|---|---|---|---|---|---|---|---|
| RXB<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 **RXB<7:0>:** Top of Receive Buffer

### REGISTER 31-11: UxTXB: UART TRANSMIT 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 |
|---|---|---|---|---|---|---|---|
| TXB<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 | |

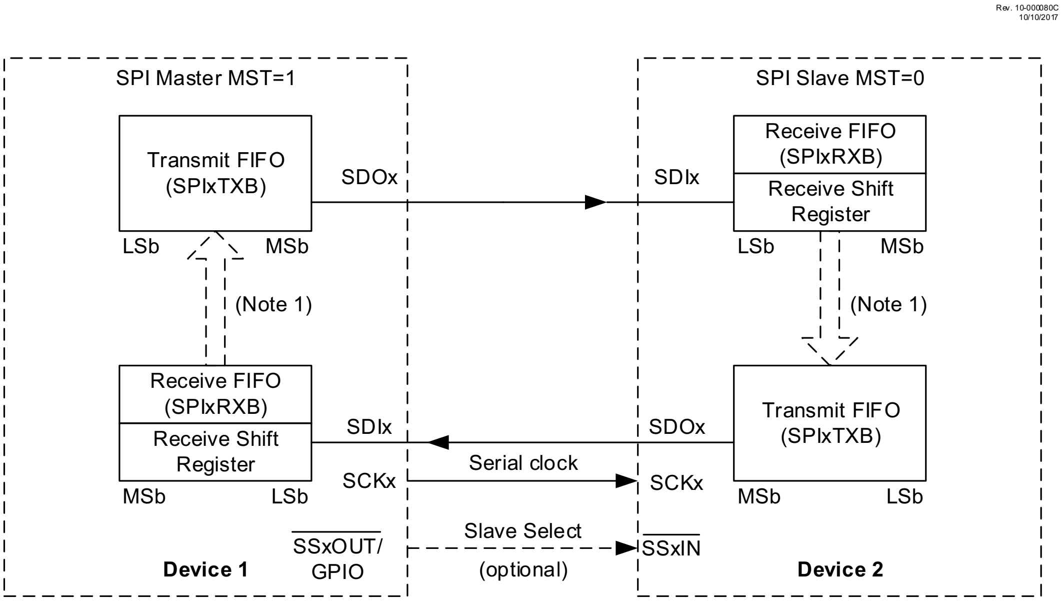
bit 7-0 **TXB<7:0>:** Bottom of Transmit Buffer

**TABLE 31-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE UART**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|
| UxCON0 | BRGS | ABDEN | TXEN | RXEN | MODE<3:0> | | | | 498 |
| UxCON1 | ON | — | — | WUE | RXBIMD | — | BRKOVR | SENDB | 499 |
| UxCON2 | RUNOVF | RXPOL | STP<1:0> | | C0EN | TXPOL | FLO<1:0> | | 500 |
| UxERRIR | TXMTIF | PERIF | ABDOVF | CERIF | FERIF | RXBKIF | RXFOIF | TXCIF | 501 |
| UxERRIE | TXMTIE | PERIE | ABDOVE | CERIE | FERIE; | RXBKIE | RXFOIE | TXCIE | 502 |
| UxUIR | WUIF | ABDIF | — | — | — | ABDIE | — | — | 503 |
| UxFIFO | TXWRE | STPMD | TXBE | TXBF | RXIDL | XON | RXBE | RXBF | 504 |
| UxBRGL | BRG<7:0> | | | | | | | | 505 |
| UxBRGH | BRG<15:8> | | | | | | | | 505 |
| UxRXB | RXB<7:0> | | | | | | | | 506 |
| UxTXB | TXB<7:0> | | | | | | | | 506 |
| UxP1H | — | — | — | — | — | — | — | P1<8> | 507 |
| UxP1L | P1<7:0> | | | | | | | | 507 |
| UxP2H | — | — | — | — | — | — | — | P2<8> | 508 |
| UxP2L | P2<7:0> | | | | | | | | 508 |
| UxP3H | — | — | — | — | — | — | — | P3<8> | 509 |
| UxP3L | P3<7:0> | | | | | | | | 509 |
| UxTXCHK | TXCHK<7:0> | | | | | | | | 510 |
| UxRXCHK | RXCHK<7:0> | | | | | | | | 510 |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are unused by the UART module.

**FIGURE 32-2: SPI MASTER/SLAVE CONNECTION WITH FIFOs**



**Note 1:** In some modes, if the Transmit FIFO is empty, the most recently received byte of data will be transmitted.

**2:** This diagram assumes that the LSBF bit is cleared (communications are MSb-first). If LSBF is set, the communications will be LSb-first.

bit 3 **MDR:** Master Data Request *(Master pause)*

1 = Master state mechine pauses until data is read/written to proceed (SCL is output held low)
0 = Master clocking of data is enabled.

MMA = 1 & RXBF = 1
*pause_for_rx* - Set by hardware on 7th falling SCL edge
- User must read from I2CRXB to release SCL

MMA = 1 & TXBE = 1 & I2CCNT!= 0
*pause_for_tx* - Set by hardware on 8th falling SCL edge
- User must write to I2CTXB to release SCL

ADB = 1
- I2CCNT is ignored for the high and low address in 10-bit mode

*pause_for_restart* - Set by hardware on 9th falling SCL edge
RSEN = 1 & MMA = 1 && I2CCNT = 0 || ACKSTAT = 1
- User must set START or write to I2CTXB to release SCL and shift Restart onto bus

bit 2-0 **MODE<2:0>:** I$^2$C Mode Select bits

111 = I$^2$C Muti-Master mode (SMBus 2.0 Host), (5)
Works as both mode<2:0> = 001 and mode<2:0> = 100
110 = I$^2$C Muti-Master mode (SMBus 2.0 Host), (5)
Works as both mode<2:0> = 000 and mode<2:0> = 100
101 = I$^2$C Master mode, 10-bit address
100 = I$^2$C Master mode, 7-bit address
011 = I$^2$C Slave mode, one 10-bit address with masking
010 = I$^2$C Slave mode, two 10-bit address
001 = I$^2$C Slave mode, two 7-bit address with masking
000 = I$^2$C Slave mode, four 7-bit address

**Note 1:** SDA and SCL pins must be configured for open-drain with internal or external pull-up
**2:** SDA and SCL pins must be selected as both input and output in PPS
**3:** CSTR can be set by more than one hardware source, all sources must be addressed by user software before the SCL line is released. CSTR is a module status bit, and does not show the true bus state.
**4:** SMA is set on the same SCL edge as CSTR for a matching received address
**5:** In this mode, ADRIE should be set, this allows an interrupt to clear the BCLIF condition and allow the ACK of matching address.
**6:** In 10-bit Slave mode, when ADB = 1, CSTR will set when the high address has not been read out of I2CxRXB before the low address is shifted in.

### 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.

### TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|---|
| 3D6Ch | I2C1CNT | CNT | | | | | | | | 586 |
| 3D6Bh | I2C1TXB | TXB | | | | | | | | |
| 3D6Ah | I2C1RXB | RXB | | | | | | | | |
| 3D69h - 3D67h | — | Unimplemented | | | | | | | | |
| 3D66h | I2C2BTO | BTO | | | | | | | | 582 |
| 3D65h | I2C2CLK | CLK | | | | | | | | 581 |
| 3D64h | I2C2PIE | CNTIE | ACKTIE | — | WRIE | ADRIE | PCIE | RSCIE | SCIE | 588 |
| 3D63h | I2C2PIR | CNTIF | ACKTIF | — | WRIF | ADRIF | PCIF | RSCIF | SCIF | 587 |
| 3D62h | I2C2STAT1 | TXWE | — | — | — | RXRE | CLRBF | — | RXBF | 584 |
| 3D61h | I2C2STAT0 | BFRE | — | MMA | — | D | — | — | — | 583 |
| 3D60h | I2C2ERR | — | BTOIF | BCLIF | NACKIF | — | BTOIE | BCLIE | NACKIE | 585 |
| 3D5Fh | I2C2CON2 | ACNT | GCEN | FME | ABD | SDAHT | | BFRET | | 580 |
| 3D5Eh | I2C2CON1 | ACKCNT | ACKDT | ACKSTAT | ACKT | — | RXO | TXU | CSD | 579 |
| 3D5Dh | I2C2CON0 | EN | RSEN | S | CSTR | MDR | MODE | | | 577 |
| 3D5Ch | I2C2ADR3 | ADR | | | | | | | — | 592 |
| 3D5Bh | I2C2ADR2 | ADR | | | | | | | | 591 |
| 3D5Ah | I2C2ADR1 | ADR | | | | | | | — | 590 |
| 3D59h | I2C2ADR0 | ADR | | | | | | | | 589 |
| 3D58h | I2C2ADB1 | ADB | | | | | | | | 594 |
| 3D57h | I2C2ADB0 | ADB | | | | | | | | 593 |
| 3D56h | I2C2CNT | CNT | | | | | | | | 586 |
| 3D55h | I2C2TXB | TXB | | | | | | | | |
| 3D54h | I2C2RXB | RXB | | | | | | | | |
| 3D53h - 3D1Dh | — | Unimplemented | | | | | | | | |
| 3D1Ch | SPI1CLK | CLKSEL | | | | | | | | 542 |
| 3D1Bh | SPI1INTE | SRMTIE | TCZIE | SOSIE | EOSIE | — | RXOIE | TXUIE | — | 536 |
| 3D1Ah | SPI1INTF | SRMTIF | TCZIF | SOSIF | EOSIF | — | RXOIF | TXUIF | — | 535 |
| 3D19h | SPI1BAUD | BAUD | | | | | | | | 538 |
| 3D18h | SPI1TWIDTH | — | — | — | — | — | TWIDTH | | | 537 |
| 3D17h | SPI1STATUS | TXWE | — | TXBE | — | RXRE | CLRBF | — | RXBF | 541 |
| 3D16h | SPI1CON2 | BUSY | SSFLT | — | — | — | SSET | TXR | RXR | 540 |
| 3D15h | SPI1CON1 | SMP | CKE | CKP | FST | — | SSP | SDIP | SDOP | 539 |
| 3D14h | SPI1CON0 | EN | — | — | — | — | LSBF | MST | BMODE | 538 |
| 3D13h | SPI1TCNTH | — | — | — | — | — | TCNTH | | | 537 |
| 3D12h | SPI1TCNTL | TCNTL | | | | | | | | 536 |
| 3D11h | SPI1TXB | TXB | | | | | | | | 542 |
| 3D10h | SPI1RXB | RXB | | | | | | | | 541 |
| 3D0Fh - 3CFFh | — | Unimplemented | | | | | | | | |
| 3CFEh | MD1CARH | — | — | — | CH | | | | | 471 |
| 3CFDh | MD1CARL | — | — | — | CL | | | | | 471 |
| 3CFCh | MD1SRC | — | — | — | MS | | | | | 472 |
| 3CFBh | MD1CON1 | — | — | CHPOL | CHSYNC | — | — | CLPOL | CLSYNC | 470 |
| 3CFAh | MD1CON0 | EN | — | OUT | OPOL | — | — | — | BIT | 469 |
| 3CF9h - 3CE7h | — | Unimplemented | | | | | | | | |

**Legend:** x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

**Note 1:** Unimplemented in LF devices.

**2:** Unimplemented in PIC18(L)F26/27K42.

**3:** Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.

**4:** Unimplemented in PIC18(L)F45/55K42.

## 44.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

| | |
|---|---|
| Operating Voltage: | VDDMIN ≤ VDD ≤ VDDMAX |
| Operating Temperature: | TA_MIN ≤ TA ≤ TA_MAX |

**VDD — Operating Supply Voltage**[1]

PIC18(L)F26/27/45/46/47/55/56/57K42

| | |
|---|---|
| VDDMIN (Fosc ≤ 16 MHz) | +1.8V |
| VDDMIN (Fosc ≤ 32 MHz) | +2.5V |
| VDDMIN (Fosc ≤ 64 MHz) | +2.7V |
| VDDMAX | +3.6V |

PIC18F26/45/46/55/56K42

| | |
|---|---|
| VDDMIN (Fosc ≤ 16 MHz) | +2.3V |
| VDDMIN (Fosc ≤ 32 MHz) | +2.5V |
| VDDMIN (Fosc ≤ 64 MHz) | +3.0V |
| VDDMAX | +5.5V |

**TA — Operating Ambient Temperature Range**

Industrial Temperature

| | |
|---|---|
| TA_MIN | -40°C |
| TA_MAX | +85°C |

Extended Temperature

| | |
|---|---|
| TA_MIN | -40°C |
| TA_MAX | +125°C |

**Note 1:** See Parameter Supply Voltage, DS Characteristics: Supply Voltage.