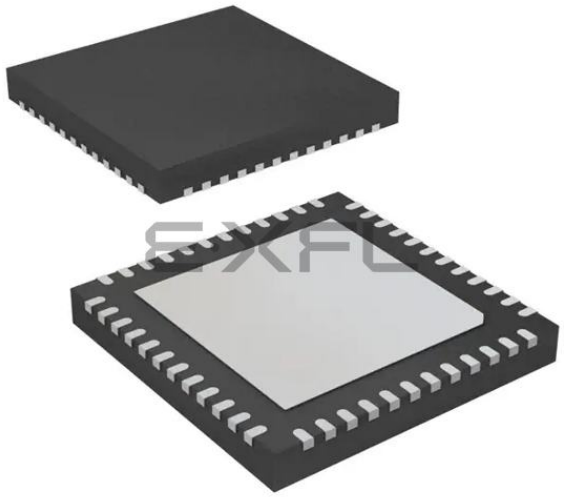


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LCD, POR, PWM, WDT |
| Number of I/O | 43 |
| Program Memory Size | 28KB (16K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 39x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°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/pic16lf19186-i-mv |

**TABLE 1-3: PIC16(L)F19175/76 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC7/DT1(3)/RX1(1)/IOCC7/ANC7/SEG23/VLCD1 | RC7 | TTL/ST | CMOS/OD | General purpose I/O. |
| | DT1(3) | — | — | EUSART synchronous data output |
| | RX1(1) | — | — | EUSART receive input. |
| | IOCC7 | TTL/ST | — | Interrupt-on-change input. |
| | ANC7 | AN | — | ADC Channel input. |
| | SEG23 | — | AN | LCD Analog output. |
| | VLCD1 | AN | — | LCD analog input |
| RD0/AND0/SEG24 | RD0 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND0 | AN | — | ADC Channel input. |
| | SEG24 | AN | — | LCD Analog output. |
| RD1/AND1/SEG25 | RD1 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND1 | AN | — | ADC Channel input. |
| | SEG25 | — | AN | LCD Analog output. |
| RD2/AND2/COM5 | RD2 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND2 | AN | — | ADC Channel input. |
| | SEG26 | — | AN | LCD Analog output. |
| | COM5 | — | AN | LCD Driver Common Outputs. |
| RD3/AND3/SEG27/COM4 | RD3 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND3 | AN | — | ADC Channel input. |
| | SEG27 | — | AN | LCD Analog output. |
| | COM4 | — | AN | LCD Driver Common Outputs. |
| RD4/AND4/SEG28 | RD4 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND4 | AN | — | ADC Channel input. |
| | SEG28 | — | AN | LCD Analog output. |
| RD5/AND5/SEG29 | RD5 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND5 | AN | — | ADC Channel input. |
| | SEG29 | AN | — | LCD Analog output. |
| RD6/AND6/SEG30 | RD6 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND6 | AN | — | ADC Channel input. |
| | SEG30 | — | AN | LCD Analog output. |
| RD7/AND7/SEG31 | RD7 | TTL/ST | CMOS/OD | General purpose I/O. |
| | AND7 | AN | — | ADC Channel input. |
| | SEG31 | AN | — | LCD Analog output. |
| RE0/ANE0/SEG32 | RE0 | TTL/ST | CMOS/OD | General purpose I/O. |
| | ANE0 | AN | — | ADC Channel input. |
| | SEG32 | — | AN | LCD Analog output. |
| RE1/ANE2/SEG33/COM6 | RE1 | TTL/ST | CMOS/OD | General purpose I/O. |
| | ANE2 | AN | — | ADC Channel input. |
| | SEG33 | — | AN | LCD analog input |
| | COM6 | — | AN | LCD Driver Common Outputs. |

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

**Note 1:** 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. Refer to Table 14-2 for details on which PORT pins may be used for this signal.

**2:** All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 14-3.

**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 are configured for $I^2C$ logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 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 $I^2C$ specific or SMBus input buffer thresholds.

**TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19155/56/75/76/85/86 (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on: MCLR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 19** | | | | | | | | | | | |
| CPU CORE REGISTERS; see Table 4-3 for specifics | | | | | | | | | | | |
| 98Ch | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 98Dh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 98Eh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 98Fh | CMOUT | — | — | — | — | — | — | MC2OUT | MC1OUT | 0000 0000 | 0000 0000 |
| 990h | CM1CON0 | ON | OUT | — | POL | — | — | HYS | SYNC | 00-0 --00 | 00-0 --00 |
| 991h | CM1CON1 | — | — | — | — | — | — | INTP | INTN | 0000 0000 | 0000 0000 |
| 992h | CM1NSEL | — | — | — | — | — | NCH<2:0> | | | 0000 0000 | 0000 0000 |
| 992h | | — | — | — | — | — | NCH2 | NCH1 | NCH0 | 0000 0000 | 0000 0000 |
| 993h | CM1PSEL | — | — | — | — | PCH<3:0> | | | | 0000 0000 | 0000 0000 |
| 993h | | — | — | — | — | — | PCH2 | PCH1 | PCH0 | 0000 0000 | 0000 0000 |
| 994h | CM2CON0 | ON | OUT | — | POL | — | — | HYS | SYNC | 00-0 --00 | 00-0 --00 |
| 995h | CM2CON1 | — | — | — | — | — | — | INTP | INTN | 0000 0000 | 0000 0000 |
| 996h | CM2NSEL | — | — | — | — | — | NCH<2:0> | | | 0000 0000 | 0000 0000 |
| 996h | | — | — | — | — | — | NCH2 | NCH1 | NCH0 | 0000 0000 | 0000 0000 |
| 997h | CM2PSEL | — | — | — | — | — | PCH<2:0> | | | 0000 0000 | 0000 0000 |
| 997h | | — | — | — | — | — | PCH2 | PCH1 | PCH0 | 0000 0000 | 0000 0000 |
| 998h | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 999h | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Ah | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Bh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Ch | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Dh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Eh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |
| 99Fh | — | Unimplemented | | | | | | | | ---- ---- | ---- ---- |

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

**Note 1:** Unimplemented data memory locations, read as '0'.

## 5.0 DEVICE CONFIGURATION

Device configuration consists of the Configuration Words, User ID, Device ID, Device Information Area (DIA), (see **Section 6.0 "Device Information Area"**), and the Device Configuration Information (DCI) regions, (see **Section 7.0 "Device Configuration Information"**).

### 5.1 Configuration Words

The devices have several Configuration Words starting at address 8007h. The Configuration bits establish configuration values prior to the execution of any software; Configuration bits enable or disable device-specific features.

In terms of programming, these important Configuration bits should be considered:

**1. LVP: Low-Voltage Programming Enable bit**

- 1 = ON – Low-Voltage Programming is enabled. $\overline{MCLR}$/$V_{PP}$ pin function is $\overline{MCLR}$. MCLRE Configuration bit is ignored.
- 0 = OFF – HV on $\overline{MCLR}$/$V_{PP}$ must be used for programming.

**2. CP: User Nonvolatile Memory (NVM) Program Memory Code Protection bit**

- 1 = OFF – User NVM code protection disabled
- 0 = ON – User NVM code protection enabled

**REGISTER 9-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3**

| R/W/HC-0/0 | R/W-0/0 | U-0 | R-0/0 | R-0/0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| CSWHOLD | SOSCPWR | — | ORDY | NOSCR | — | — | — |
| 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 **CSWHOLD:** Clock Switch Hold bit
- 1 = Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready
- 0 = Clock switch may proceed when the oscillator selected by NOSC is ready; if this bit is clear at the time that NOSCR becomes '1', the switch will occur

bit 6 **SOSCPWR:** Secondary Oscillator Power Mode Select bit
- 1 = Secondary oscillator operating in High-power mode
- 0 = Secondary oscillator operating in Low-power mode

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

bit 4 **ORDY:** Oscillator Ready bit (read-only)
- 1 = OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC
- 0 = A clock switch is in progress

bit 3 **NOSCR:** New Oscillator is Ready bit (read-only)
- 1 = A clock switch is in progress and the oscillator selected by NOSC indicates a "ready" condition
- 0 = A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready

bit 2-0 **Unimplemented:** Read as '0'

### 13.4.2 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- PFM Row Erase
- Load of PFM write latches
- Write of PFM write latches to PFM memory
- Write of PFM write latches to User IDs

The unlock sequence consists of the following steps and must be completed in order:

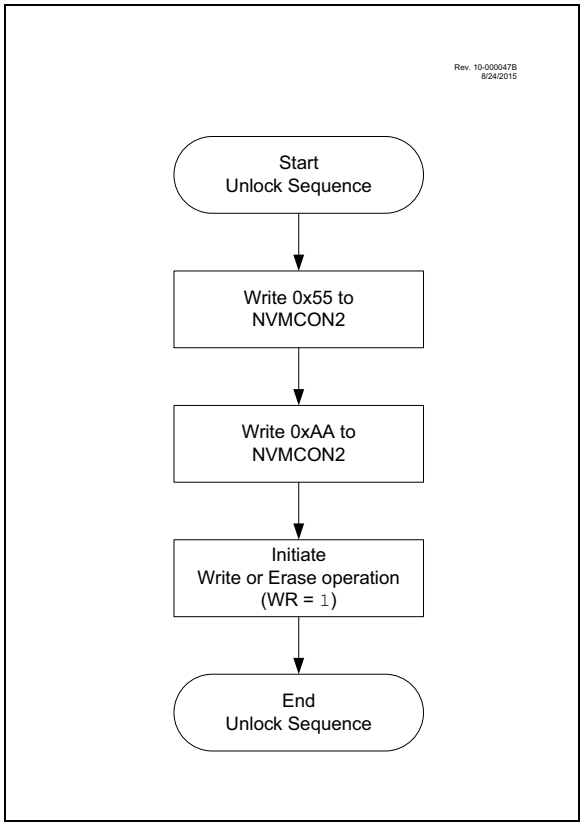
- Write 55h to NVMCON2
- Write AAh to NMVCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

| **Note:** | The two NOP instructions after setting the WR bit that were required in previous devices are not required for PIC16(L)F19155/56/75/76/85/86 devices. See Figure 13-2. |
|---|---|

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

**FIGURE 13-2: NVM UNLOCK SEQUENCE FLOWCHART**




**EXAMPLE 13-2: NVM UNLOCK SEQUENCE**

```
BCF      INTCON, GIE      ; Recommended so sequence is not interrupted
BANKSEL  NVMCON1          ;
BSF      NVMCON1, WREN    ; Enable write/erase
MOVLW    55h              ; Load 55h
MOVWF    NVMCON2          ; Step 1: Load 55h into NVMCON2
MOVLW    AAh              ; Step 2: Load W with AAh
MOVWF    NVMCON2          ; Step 3: Load AAH into NVMCON2
BSF      NVMCON1, WR      ; Step 4: Set WR bit to begin write/erase
BSF      INTCON, GIE      ; Re-enable interrupts
```

**Note 1:** Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.

**2:** Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

**EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY (PFM)**

```
; This write routine assumes the following:
; 1. 32 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

        BANKSEL     NVMADRH
        MOVF        ADDRH,W
        MOVWF       NVMADRH             ; Load initial address
        MOVF        ADDRL,W
        MOVWF       NVMADRL
        MOVLW       LOW DATA_ADDR       ; Load initial data address
        MOVWF       FSR0L
        MOVLW       HIGH DATA_ADDR
        MOVWF       FSR0H
        BCF         NVMCON1,NVMREGS     ; Set Program Flash Memory as write location
        BSF         NVMCON1,WREN        ; Enable writes
        BSF         NVMCON1,LWLO        ; Load only write latches

LOOP
        MOVIW       FSR0++
        MOVWF       NVMDATL             ; Load first data byte
        MOVIW       FSR0++
        MOVWF       NVMDATH             ; Load second data byte

        MOVF        NVMADRL,W
        XORLW       0x1F                ; Check if lower bits of address are 00000
        ANDLW       0x1F                ; and if on last of 32 addresses
        BTFSC       STATUS,Z            ; Last of 32 words?
        GOTO        START_WRITE         ; If so, go write latches into memory

        CALL        UNLOCK_SEQ          ; If not, go load latch
        INCF        NVMADRL,F           ; Increment address
        GOTO        LOOP

START_WRITE
        BCF         NVMCON1,LWLO        ; Latch writes complete, now write memory
        CALL        UNLOCK_SEQ          ; Perform required unlock sequence
        BCF         NVMCON1,WREN        ; Disable writes

UNLOCK_SEQ
        MOVLW       55h
        BCF         INTCON,GIE          ; Disable interrupts
        MOVWF       NVMCON2             ; Begin unlock sequence
        MOVLW       AAh
        MOVWF       NVMCON2
        BSF         NVMCON1,WR
        BSF         INTCON,GIE          ; Unlock sequence complete, re-enable interrupts
        return
```

### 19.4.2 PRECHARGE CONTROL

The Precharge stage is an optional period of time that brings the external channel and internal sample and hold capacitor to known voltage levels. Precharge is enabled by writing a non-zero value to the PRE register. This stage is initiated when an ADC conversion begins, either from setting the GO bit, a special event trigger, or a conversion restart from the computation functionality. If the PRE register is cleared when an ADC conversion begins, this stage is skipped.

During the precharge time, CHOLD is disconnected from the outer portion of the sample path that leads to the external capacitive sensor and is connected to either VDD or VSS, depending on the value of the ADPPOL bit of ADCON1. At the same time, the port pin logic of the selected analog channel is overridden to drive a digital high or low out, in order to precharge the outer portion of the ADC's sample path, which includes the external sensor. The output polarity of this override is also determined by the ADPPOL bit of ADCON1. The amount of time that this charging receives is controlled by the PRE register.

| | |
|---|---|
| **Note 1:** | The external charging overrides the TRIS setting of the respective I/O pin. |
| **2:** | If there is a device attached to this pin, Precharge should not be used. |

### 19.4.3 ACQUISITION CONTROL

The Acquisition stage is an optional time for the voltage on the internal sample and hold capacitor to charge or discharge from the selected analog channel. This acquisition time is controlled by the ADACQ register. If PRE = 0, acquisition starts at the beginning of conversion. When PRE = 1, the acquisition stage begins when precharge ends.

At the start of the acquisition stage, the port pin logic of the selected analog channel is overridden to turn off the digital high/low output drivers so they do not affect the final result of the charge averaging. Also, the selected ADC channel is connected to CHOLD. This allows charge averaging to proceed between the precharged channel and the CHOLD capacitor.

| | |
|---|---|
| **Note:** | When PRE! = 0, acquisition time cannot be '0'. In this case, setting ADACQ to '0' will set a maximum acquisition time (8191 ADC clock cycles). When precharge is disabled, setting ADACQ to '0' will disable hardware acquisition time control. |

### 19.4.4 GUARD RING OUTPUTS

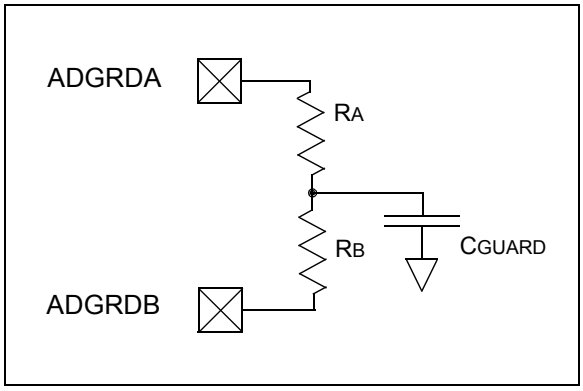
Figure 19-8 shows a typical guard ring circuit. CGUARD represents the capacitance of the guard ring trace placed on the PCB board. The user selects values for RA and RB that will create a voltage profile on CGUARD, which will match the selected acquisition channel.

The purpose of the guard ring is to generate a signal in phase with the CVD sensing signal to minimize the effects of the parasitic capacitance on sensing electrodes. It also can be used as a mutual drive for mutual capacitive sensing. For more information about active guard and mutual drive, see Application Note AN1478, "*mTouch™ Sensing Solution Acquisition Methods Capacitive Voltage Divider*" (DS01478).

The ADC has two guard ring drive outputs, ADGRDA and ADGRDB. These outputs can be routed through PPS controls to I/O pins (see **Section 15.0 "Peripheral Pin Select (PPS) Module"** for details) and the polarity of these outputs are controlled by the ADGPOL and ADIPEN bits of ADCON1.

At the start of the first precharge stage, both outputs are set to match the ADGPOL bit of ADCON1. Once the acquisition stage begins, ADGRDA changes polarity, while ADGRDB remains unchanged. When performing a double sample conversion, setting the ADIPEN bit of ADCON1 causes both guard ring outputs to transition to the opposite polarity of ADGPOL at the start of the second precharge stage, and ADGRDA toggles again for the second acquisition. For more information on the timing of the guard ring output, refer to Figure 19-8 and Figure 19-9.

**FIGURE 19-8: GUARD RING CIRCUIT**



**REGISTER 19-18: 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 19-19: 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 **Unimplemented:** Read as ‘0’

## 28.2 SMT Operation

The core of the module is the 24-bit counter, SMTxTMR combined with a complex data acquisition front-end. Depending on the mode of operation selected, the SMT can perform a variety of measurements summarized in Table .

### 28.2.1 CLOCK SOURCES

Clock sources available to the SMT include:

- FOSC
- FOSC/4
- HFINTOSC
- MFINTOSC (500 kHz and 31.25 kHz)
- LFINTOSC
- SOSC

The SMT clock source is selected by configuring the CSEL<2:0> bits in the SMTxCLK register. The clock source can also be prescaled using the PS<1:0> bits of the SMTxCON0 register. The prescaled clock source is used to clock both the counter and any synchronization logic used by the module.

### 28.2.2 PERIOD MATCH INTERRUPT

Similar to other timers, the SMT triggers an interrupt when SMTxTMR rolls over to '0'. This happens when SMTxTMR = SMTxPR, regardless of mode. Hence, in any mode that relies on an external signal or a window to reset the timer, proper operation requires that SMTxPR be set to a period larger than that of the expected signal or window.

## 28.3 Basic Timer Function Registers

The SMTxTMR time base and the SMTxCPW/SMTxPR/SMTxCPR buffer registers serve several functions and can be manually updated using software.

### 28.3.1 TIME BASE

The SMTxTMR is the 24-bit counter that is the center of the SMT. It is used as the basic counter/timer for measurement in each of the modes of the SMT. It can be reset to a value of 24'h00_0000 by setting the RST bit of the SMTxSTAT register. It can be written to and read from software, but it is not guarded for atomic access, therefore reads and writes to the SMTxTMR should only be made when the GO = 0, or the software should have other measures to ensure integrity of SMTxTMR reads/writes.

### 28.3.2 PULSE-WIDTH LATCH REGISTERS

The SMTxCPW registers are the 24-bit SMT pulse width latch. They are used to latch in the value of the SMTxTMR when triggered by various signals, which are determined by the mode the SMT is currently in.

The SMTxCPW registers can also be updated with the current value of the SMTxTMR value by setting the CPWUP bit of the SMTxSTAT register.

### 28.3.3 PERIOD LATCH REGISTERS

The SMTxCPR registers are the 24-bit SMT period latch. They are used to latch in values of the SMTxTMR when triggered by various other signals, which are determined by the mode the SMT is currently in.

The SMTxCPR registers can also be updated with the current value of the SMTxTMR value by setting the CPRUP bit in the SMTxSTAT register.

## 28.4 Halt Operation

The counter can be prevented from rolling-over using the STP bit in the SMTxCON0 register. When halting is enabled, the period match interrupt persists until the SMTxTMR is reset (either by a manual Reset, **Section 28.3.1 "Time Base"**) or by clearing the SMTxGO bit of the SMTxCON1 register and writing the SMTxTMR values in software.

## 28.5 Polarity Control

The three input signals for the SMT have polarity control to determine whether or not they are active high/positive edge or active low/negative edge signals.

The following bits apply to Polarity Control:

- WSEL bit (Window Polarity)
- SSEL bit (Signal Polarity)
- CSEL bit (Clock Polarity)

These bits are located in the SMTxCON0 register.

## 28.6 Status Information

The SMT provides input status information for the user without requiring the need to deal with the polarity of the incoming signals.

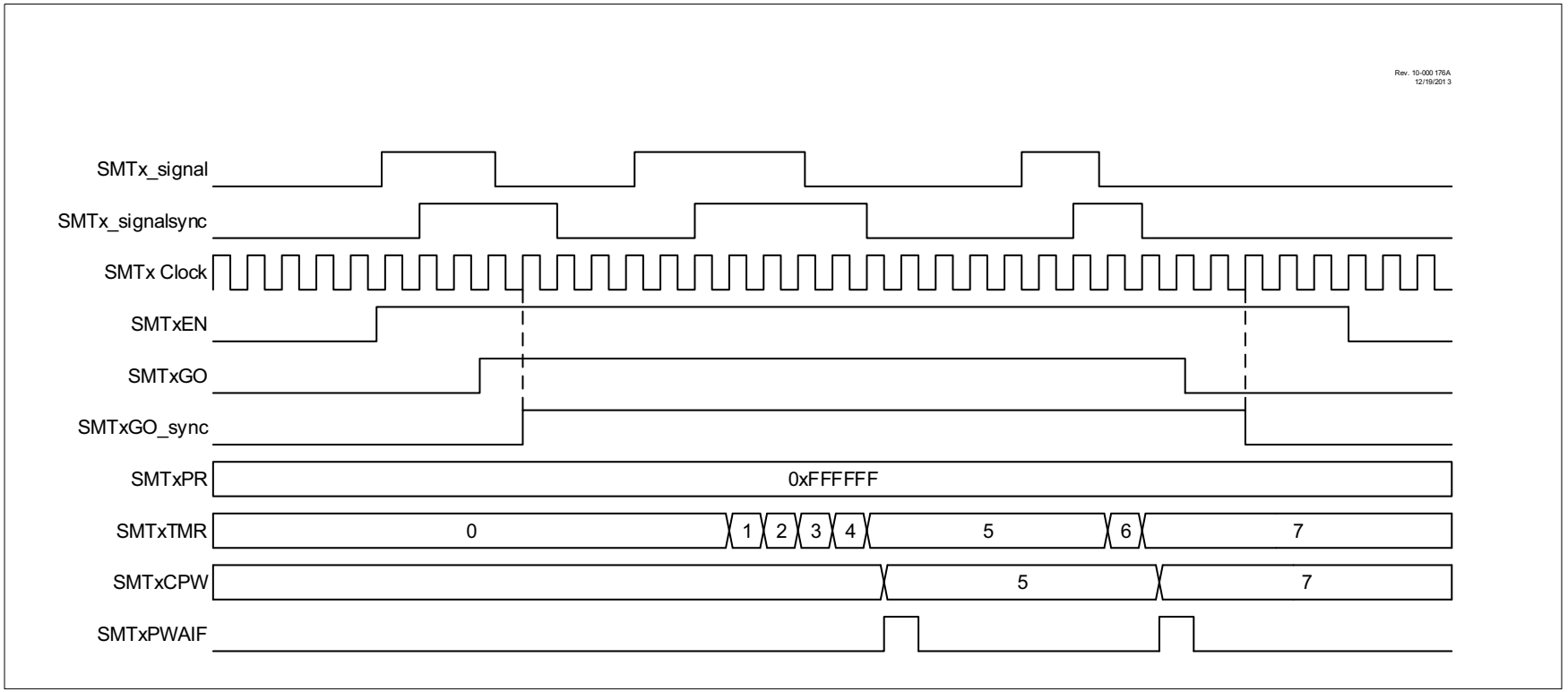
### 28.6.1 WINDOW STATUS

Window status is determined by the WS bit of the SMTxSTAT register. This bit is only used in Windowed Measure, Gated Counter and Gated Window Measure modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

### 28.6.2 SIGNAL STATUS

Signal status is determined by the AS bit of the SMTxSTAT register. This bit is used in all modes except Window Measure, Time of Flight and Capture modes, and is only valid when TS = 1, and will be delayed in time by synchronizer delays in non-Counter modes.

**FIGURE 28-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM**



## 31.0 COMPLEMENTARY WAVEFORM GENERATOR (CWG) MODULE

The Complementary Waveform Generator (CWG) produces half-bridge, full-bridge, and steering of PWM waveforms. It is backwards compatible with previous ECCP functions.

The CWG has the following features:

- Six Operating modes:
  - Synchronous Steering mode
  - Asynchronous Steering mode
  - Full-Bridge mode, Forward
  - Full-Bridge mode, Reverse
  - Half-Bridge mode
  - Push-Pull mode
- Output Polarity Control
- Output Steering:
  - Synchronized to rising event
  - Immediate effect
- Independent 6-Bit Rising and Falling Event Dead-Band Timers:
  - Clocked dead band
  - Independent rising and falling dead-band enables
- Auto-Shutdown Control with:
  - Selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control

The CWG modules available are shown in Table 31-1.

**TABLE 31-1: AVAILABLE CWG MODULES**

| Device | CWG1 |
|---|---|
| PIC16(L)F19155/56/75/76/85/86 | ● |

### 31.1 Fundamental Operation

The CWG module can operate in six different modes, as specified by MODE of the CWG1CON0 register:

- Half-Bridge mode (Figure 31-9)
- Push-Pull mode (Figure 31-2)
  - Full-Bridge mode, Forward (Figure 31-3)
  - Full-Bridge mode, Reverse (Figure 31-3)
- Steering mode (Figure 31-10)
- Synchronous Steering mode (Figure 31-11)

It may be necessary to guard against the possibility of circuit faults or a feedback event arriving too late or not at all. In this case, the active drive must be terminated before the Fault condition causes damage. Thus, all output modes support auto-shutdown, which is covered in **Section 31.10 “Auto-Shutdown”**.

#### 31.1.1 HALF-BRIDGE MODE

In Half-Bridge mode, two output signals are generated as true and inverted versions of the input as illustrated in Figure 31-9. A non-overlap (dead-band) time is inserted between the two outputs as described in **Section 31.5 “Dead-Band Control”**.

The unused outputs CWG1C and CWG1D drive similar signals, with polarity independently controlled by the POLC and POLD bits of the CWG1CON1 register, respectively.

## 33.5 I²C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected by the SSPM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operate the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

### 33.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 33-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 33-5) affects the address matching process. See **Section 33.5.9 "SSP Mask Register"** for more information.

#### 33.5.1.1 I²C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

#### 33.5.1.2 I²C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb's of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCL is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

### 33.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPxSTAT register is set, or bit SSPOV of the SSPxCON1 register is set. The BOEN bit of the SSPxCON3 register modifies this operation. For more information see Register 33-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSPxIF, must be cleared by software.

When the SEN bit of the SSPxCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPxCON1 register.

#### 33.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I²C slave in 7-bit Addressing mode. Figure 33-14 and Figure 33-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I²C communication.

1. Start bit detected.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSPxIF bit.
10. Software clears SSPxIF.
11. Software reads the received byte from SSPxBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPxSTAT, and the bus goes idle.

#### 34.1.2.3 Receive Interrupts

The RXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RXxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RXxIF interrupts are enabled by setting all of the following bits:

- RXxIE, Interrupt Enable bit of the PIE3 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RXxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

#### 34.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCxSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCxREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

| | |
|---|---|
| **Note:** | If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit. |

#### 34.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCxSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

#### 34.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

#### 34.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCxSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RXxIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

## 34.6 Register Definitions: EUSART Control

**REGISTER 34-1: TXxSTA: TRANSMIT STATUS AND CONTROL REGISTER**

| R/W-/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R-1/1 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN(1) | SYNC | SENDB | BRGH | TRMT | TX9D |
| 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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Unused in this mode – value ignored
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send SYNCH BREAK on next transmission – Start bit, followed by 12 '0' bits, followed by Stop bit; cleared by hardware upon completion
0 = SYNCH BREAK transmission disabled or completed
Synchronous mode:
Unused in this mode – value ignored

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode – value ignored

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.

**SLEEP** **Enter Sleep mode**

Syntax: [ *label* ] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
See **Section 11.2 "Sleep Mode"** for more information.

**SUBLW** **Subtract W from literal**

Syntax: [ *label* ] SUBLW k

Operands: 0 ≤ k ≤ 255

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

| | |
|---|---|
| C = 0 | W > k |
| C = 1 | W ≤ k |
| DC = 0 | W<3:0> > k<3:0> |
| DC = 1 | W<3:0> ≤ k<3:0> |

**SUBWF** **Subtract W from f**

Syntax: [ *label* ] SUBWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f.

| | |
|---|---|
| C = 0 | W > f |
| C = 1 | W ≤ f |
| DC = 0 | W<3:0> > f<3:0> |
| DC = 1 | W<3:0> ≤ f<3:0> |

**SUBWFB** **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

**SWAPF** **Swap Nibbles in f**

Syntax: [ *label* ] SWAPF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

**TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|---|
| 49Eh | — | Unimplemented | | | | | | | | |
| 49Fh | — | Unimplemented | | | | | | | | |
| 50Ch | — | Unimplemented | | | | | | | | |
| 50Dh | — | Unimplemented | | | | | | | | |
| 50Eh | — | Unimplemented | | | | | | | | |
| 50Fh | — | Unimplemented | | | | | | | | |
| 510h | — | Unimplemented | | | | | | | | |
| 511h | — | Unimplemented | | | | | | | | |
| 512h | — | Unimplemented | | | | | | | | |
| 513h | — | Unimplemented | | | | | | | | |
| 514h | — | Unimplemented | | | | | | | | |
| 515h | — | Unimplemented | | | | | | | | |
| 516h | — | Unimplemented | | | | | | | | |
| 517h | — | Unimplemented | | | | | | | | |
| 518h | — | Unimplemented | | | | | | | | |
| 519h | — | Unimplemented | | | | | | | | |
| 51Ah | — | Unimplemented | | | | | | | | |

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

**Note 1:** Unimplemented data memory locations, read as '0'.

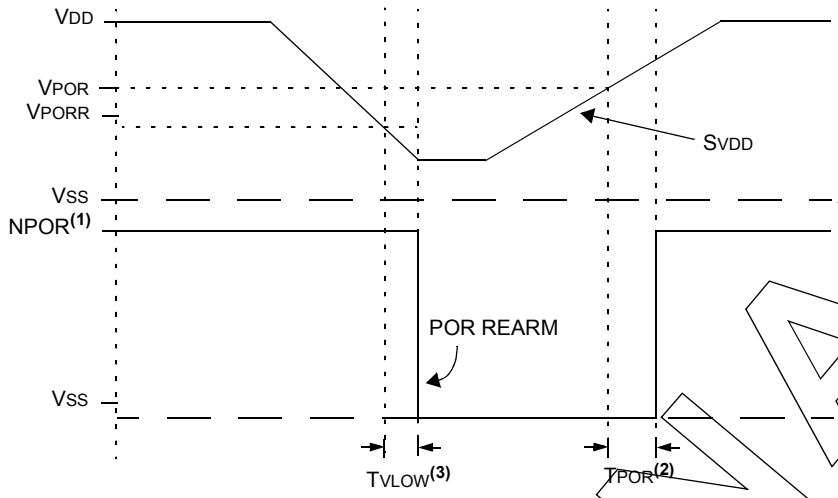
**TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on page |
|---|---|---|---|---|---|---|---|---|---|---|
| 891h | OSCEN | EXTOEN | HFOEN | MFOEN | LFOEN | SOSCEN | ADOEN | — | — | 156 |
| 892h | OSCTUNE | — | — | HFTUN<5:0> | | | | | | 157 |
| 893h | OSCFRQ | — | — | — | — | — | HFFRQ<2:0> | | | 157 |
| 894h | ACTCON | ACTEN | ACTUD | — | — | ACTLOCK | — | ACTORS | — | 158 |
| 895h | — | Unimplemented | | | | | | | | |
| 896h | — | Unimplemented | | | | | | | | |
| 897h | — | Unimplemented | | | | | | | | |
| 898h | — | Unimplemented | | | | | | | | |
| 899h | — | Unimplemented | | | | | | | | |
| 89Ah | — | Unimplemented | | | | | | | | |
| 89Bh | — | Unimplemented | | | | | | | | |
| 89Ch | — | Unimplemented | | | | | | | | |
| 89Dh | — | Unimplemented | | | | | | | | |
| 89Eh | — | Unimplemented | | | | | | | | |
| 89Fh | — | Unimplemented | | | | | | | | |
| 90Ch | FVRCON | FVREN | FVRRDY | TSEN | TSRNG | CDAFVR<1:0> | | ADFVR<1:0> | | 285 |
| 90Dh | — | Unimplemented | | | | | | | | |
| 90Eh | DAC1CON0 | EN | — | OE1 | OE2 | DAC1PSS<1:0> | | — | — | 332 |
| 90Fh | DAC1CON1 | — | — | — | DAC1R<4:0> | | | | | 332 |
| 90Fh | | — | — | — | DAC1R4 | DAC1R3 | DAC1R2 | DAC1R1 | DAC1R0 | 332 |
| 910h | — | Unimplemented | | | | | | | | |
| 911h | — | Unimplemented | | | | | | | | |
| 912h | — | Unimplemented | | | | | | | | |
| 913h | — | Unimplemented | | | | | | | | |
| 914h | — | Unimplemented | | | | | | | | |
| 915h | — | Unimplemented | | | | | | | | |
| 916h | — | Unimplemented | | | | | | | | |
| 917h | — | Unimplemented | | | | | | | | |
| 918h | — | Unimplemented | | | | | | | | |
| 919h | — | Unimplemented | | | | | | | | |
| 91Ah | — | Unimplemented | | | | | | | | |
| 91Bh | — | Unimplemented | | | | | | | | |
| 91Ch | — | Unimplemented | | | | | | | | |
| 91Dh | — | Unimplemented | | | | | | | | |
| 91Eh | — | Unimplemented | | | | | | | | |
| 91Fh | ZCDCON | ZCDSEN | — | ZCDOUT | ZCDPOL | — | — | ZCDINTP | ZCDINTN | 349 |
| 98Ch | — | Unimplemented | | | | | | | | |
| 98Dh | — | Unimplemented | | | | | | | | |
| 98Eh | — | Unimplemented | | | | | | | | |
| 98Fh | CMOUT | — | — | — | — | — | — | MC2OUT | MC1OUT | 343 |
| 990h | CM1CON0 | ON | OUT | — | POL | — | — | HYS | SYNC | 340 |
| 991h | CM1CON1 | — | — | — | — | — | — | INTP | INTN | 341 |
| 992h | CM1NSEL | — | — | — | — | — | NCH<2:0> | | | 342 |
| 992h | | — | — | — | — | — | NCH2 | NCH1 | NCH0 | 342 |
| 993h | CM1PSEL | — | — | — | — | PCH<3:0> | | | | 342 |
| 993h | | — | — | — | — | — | PCH2 | PCH1 | PCH0 | 342 |

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

**Note 1:** Unimplemented data memory locations, read as '0'.

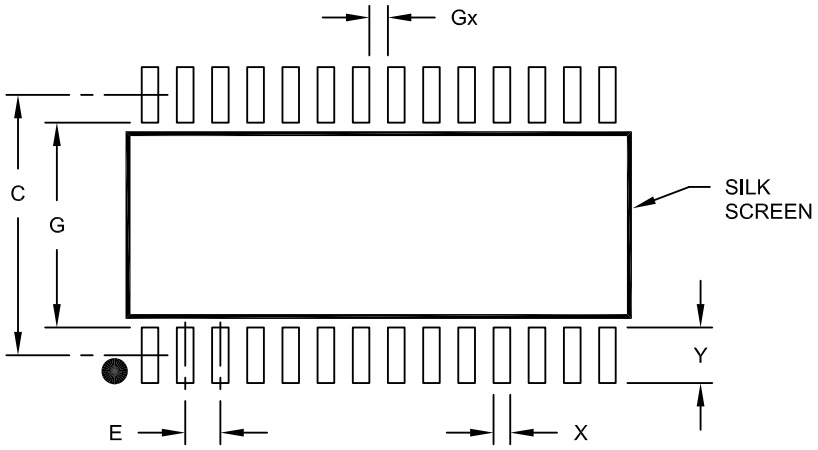
**FIGURE 39-3: POR AND POR REARM WITH SLOW RISING VDD**




**Note 1:** When NPOR is low, the device is held in Reset.
**2:** TPOR 1 μs typical.
**3:** TVLOW 2.7 μs typical.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 1.27 BSC | |
| Contact Pad Spacing | C | | 9.40 | |
| Contact Pad Width (X28) | X | | | 0.60 |
| Contact Pad Length (X28) | Y | | | 2.00 |
| Distance Between Pads | Gx | 0.67 | | |
| Distance Between Pads | G | 7.40 | | |

Notes:

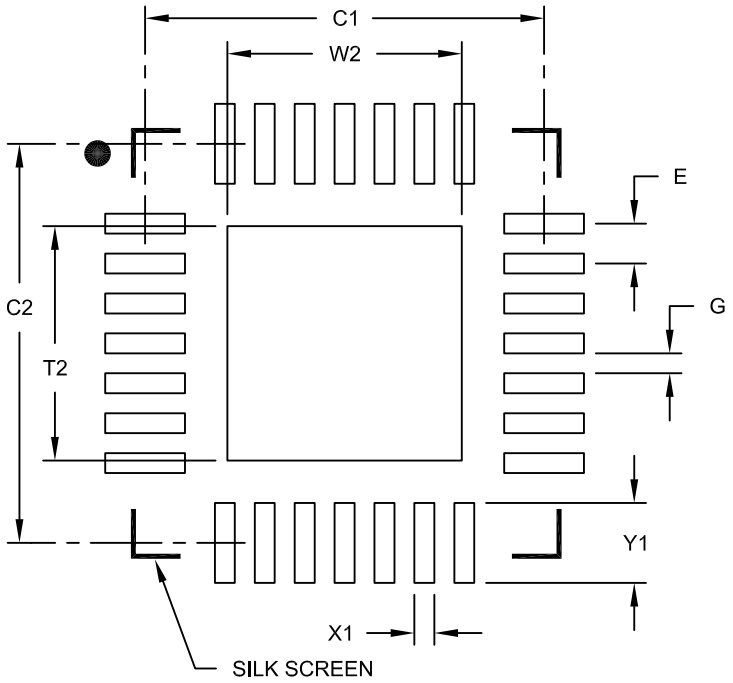
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

28-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) - 4x4 mm Body [UQFN]
With 0.40 mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.40 BSC | |
| Optional Center Pad Width | W2 | | | 2.35 |
| Optional Center Pad Length | T2 | | | 2.35 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Spacing | C2 | | 4.00 | |
| Contact Pad Width (X28) | X1 | | | 0.20 |
| Contact Pad Length (X28) | Y1 | | | 0.80 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2152A