**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 | 40MHz |
| Connectivity | EBI/EMI, I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, LVD, POR, PWM, WDT |
| Number of I/O | 69 |
| Program Memory Size | 64KB (32K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 3.8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 16x10b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 80-TQFP |
| Supplier Device Package | 80-TQFP (12x12) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf8621-i-pt |

**NOTES:**

## 4.0 MEMORY ORGANIZATION

There are three memory blocks in PIC18F6525/6621/8525/8621 devices. They are:

- Program Memory
- Data RAM
- Data EEPROM

Data and program memory use separate busses which allow for concurrent access of these blocks. Additional detailed information for Flash program memory and data EEPROM is provided in **Section 5.0 "Flash Program Memory"** and **Section 7.0 "Data EEPROM Memory"**, respectively.

In addition to on-chip Flash, the PIC18F8525/8621 devices are also capable of accessing external program memory through an external memory bus. Depending on the selected operating mode (discussed in **Section 4.1.1 "PIC18F6525/6621/8525/8621 Program Memory Modes"**), the controllers may access either internal or external program memory exclusively, or both internal and external memory in selected blocks. Additional information on the external memory interface is provided in **Section 6.0 "External Memory Interface"**.

### 4.1 Program Memory Organization

A 21-bit program counter is capable of addressing the 2-Mbyte program memory space. Accessing a location between the physically implemented memory and the 2-Mbyte address will cause a read of all '0's (a NOP instruction).

The PIC18F6525 and PIC18F8525 each have 48 Kbytes of on-chip Flash memory, while the PIC18F6621 and PIC18F8621 have 64 Kbytes of Flash. This means that PIC18FX525 devices can store internally up to 24,576 single-word instructions and PIC18FX621 devices can store up to 32,768 single-word instructions.

The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

Figure 4-1 shows the program memory map for PIC18FX525 devices, while Figure 4-2 shows the program memory map for PIC18FX621 devices.

#### 4.1.1 PIC18F6525/6621/8525/8621 PROGRAM MEMORY MODES

PIC18F8525/8621 devices differ significantly from their PIC18 predecessors in their utilization of program memory. In addition to available on-chip Flash program memory, these controllers can also address up to 2 Mbytes of external program memory through the external memory interface. There are four distinct operating modes available to the controllers:

- Microprocessor (MP)
- Microprocessor with Boot Block (MPBB)
- Extended Microcontroller (EMC)
- Microcontroller (MC)

The Program Memory mode is determined by setting the two Least Significant bits of the CONFIG3L Configuration Byte register as shown in Register 4-1 (see **Section 24.1 "Configuration Bits"** for additional details on the device configuration bits).

The Program Memory modes operate as follows:

- The **Microprocessor Mode** permits access only to external program memory; the contents of the on-chip Flash memory are ignored. The 21-bit program counter permits access to a 2-Mbyte linear program memory space.
- The **Microprocessor with Boot Block Mode** accesses on-chip Flash memory from addresses 000000h to 0007FFh. Above this, external program memory is accessed all the way up to the 2-Mbyte limit. Program execution automatically switches between the two memories as required.
- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the physical limit of the on-chip Flash (BFFFh for the PIC18FX525, FFFFh for the PIC18FX621) causes a read of all '0's (a NOP instruction).
  The Microcontroller mode is also the only operating mode available to PIC18F6525/6621 devices.
- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip Flash memory; above this, the device accesses external program memory up to the 2-Mbyte program space limit. As with Boot Block mode, execution automatically switches between the two memories as required.

In all modes, the microcontroller has complete access to data RAM and EEPROM.

Figure 4-3 compares the memory maps of the different program memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 4-1.

### 4.8.2 TABLE READS/TABLE WRITES

A better method of storing data in program memory allows 2 bytes of data to be stored in each instruction location.

Look-up table data may be stored 2 bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from, or written to program memory. Data is transferred to/from program memory, one byte at a time.

A description of the table read/table write operation is shown in **Section 5.0 “Flash Program Memory”**.

## 4.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. Figure 4-7 shows the data memory organization for the PIC18F6525/6621/8525/8621 devices.

The data memory map is divided into 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper 4 bits for the BSR are not implemented.

The data memory contains Special Function Registers (SFR) and General Purpose Registers (GPR). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratch pad operations in the user’s application. The SFRs start at the last location of Bank 15 (0FFFh) and extend downwards. Any remaining space beyond the SFRs in the bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as ‘0’s.

The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of a File Select Register (FSRn) and a corresponding Indirect File Operand (INDFn). Each FSR holds a 12-bit address value that can be used to access any location in the data memory map without banking.

The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction that moves a value from one register to another.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM. **Section 4.10 “Access Bank”** provides a detailed description of the Access RAM.

### 4.9.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly. Indirect addressing operates using a File Select Register and corresponding Indirect File Operand. The operation of indirect addressing is shown in **Section 4.12 “Indirect Addressing, INDF and FSR Registers”**.

Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

Data RAM is available for use as General Purpose Registers by all instructions. The top section of Bank 15 (F60h to FFFh) contains SFRs. All other banks of data memory contain GPRs, starting with Bank 0.

### 4.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 4-2 and Table 4-3.

The SFRs can be classified into two sets: those associated with the “core” function and those related to the peripheral functions. Those registers related to the “core” are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature. The SFRs are typically distributed among the peripherals whose functions they control.

The unused SFR locations are unimplemented and read as ‘0’s. The addresses for the SFRs are listed in Table 4-2.

**REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | INTEDG3 | TMR0IP | INT3IP | RBIP |
| bit 7 | | | | | | | bit 0 |

bit 7 **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2**: External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3**: External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RBIP**: RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

| **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 |

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

**TABLE 10-9: PORTE FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RE0/AD8/$\overline{RD}$/P2D | bit 0 | ST/TTL(1) | Input/output port pin, address/data bit 8, read control for Parallel Slave Port or Enhanced PWM 2 output P2D<br>For $\overline{RD}$ (PSP Control mode):<br>1 = Not a read operation<br>0 = Read operation, reads PORTD register (if chip selected) |
| RE1/AD9/$\overline{WR}$/P2C | bit 1 | ST/TTL(1) | Input/output port pin, address/data bit 9, write control for Parallel Slave Port or Enhanced PWM 2 output P2C<br>For $\overline{WR}$ (PSP Control mode):<br>1 = Not a write operation<br>0 = Write operation, writes PORTD register (if chip selected) |
| RE2/AD10/$\overline{CS}$/P2B | bit 2 | ST/TTL(1) | Input/output port pin, address/data bit 10, chip select control for Parallel Slave Port or Enhanced PWM 2 output P2B<br>For $\overline{CS}$ (PSP Control mode):<br>1 = Device is not selected<br>0 = Device is selected |
| RE3/AD11/P3C(2) | bit 3 | ST/TTL(1) | Input/output port pin, address/data bit 11 or Enhanced PWM 3 output P3C. |
| RE4/AD12/P3B(2) | bit 4 | ST/TTL(1) | Input/output port pin, address/data bit 12 or Enhanced PWM 3 output P3B. |
| RE5/AD13/P1C(2) | bit 5 | ST/TTL(1) | Input/output port pin, address/data bit 13 or Enhanced PWM 1 output P1C. |
| RE6/AD14/P1B(2) | bit 6 | ST/TTL(1) | Input/output port pin, address/data bit 14 or Enhanced PWM 1 output P1B. |
| RE7/AD15/ ECCP2(3)/P2A(3) | bit 7 | ST/TTL(1) | Input/output port pin, address/data bit 15, Enhanced Capture 2 input/ Compare 2 output/PWM 2 output or Enhanced PWM 2 output P2A. |

**Legend:** ST = Schmitt Trigger input, TTL = TTL input

**Note 1:** Input buffers are Schmitt Triggers when in I/O or CCP/ECCP modes and TTL buffers when in System Bus or PSP Control modes.

**2:** Valid for all PIC18F6525/6621 devices and PIC18F8525/8621 devices when ECCPMX is set. Alternate assignments for P1B/P1C/P3B/P3C are RH7, RH6, RH5 and RH4, respectively.

**3:** Valid for all PIC18F6525/6621 devices and PIC18F8525/8621 devices in Microcontroller mode when CCP2MX is not set. RC1 is the default assignment for ECCP2/P2A for all devices in Microcontroller mode when CCP2MX is set; RB3 is the alternate assignment for PIC18F8525/8621 devices in operating modes except Microcontroller mode when CCP2MX is not set.

**TABLE 10-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| TRISE | PORTE Data Direction Control Register | | | | | | | | 1111 1111 | 1111 1111 |
| PORTE | Read PORTE pin/Write PORTE Data Latch | | | | | | | | xxxx xxxx | uuuu uuuu |
| LATE | Read PORTE Data Latch/Write PORTE Data Latch | | | | | | | | xxxx xxxx | uuuu uuuu |
| MEMCON(1) | EBDIS | — | WAIT1 | WAIT0 | — | — | WM1 | WM0 | 0-00 --00 | 0000 --00 |
| PSPCON(2) | IBF | OBF | IBOV | PSPMODE | — | — | — | — | 0000 ---- | 0000 ---- |

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used by PORTE.

**Note 1:** This register is unused on PIC18F6525/6621 devices and reads as '0'.

**2:** Enabled only in Microcontroller mode for PIC18F8525/8621 devices.

## 10.6 PORTF, LATF and TRISF Registers

PORTF is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISF. Setting a TRISF bit (= 1) will make the corresponding PORTF pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISF bit (= 0) will make the corresponding PORTF pin an output (i.e., put the contents of the output latch on the selected pin).

Read-modify-write operations on the LATF register, read and write the latched output value for PORTF.

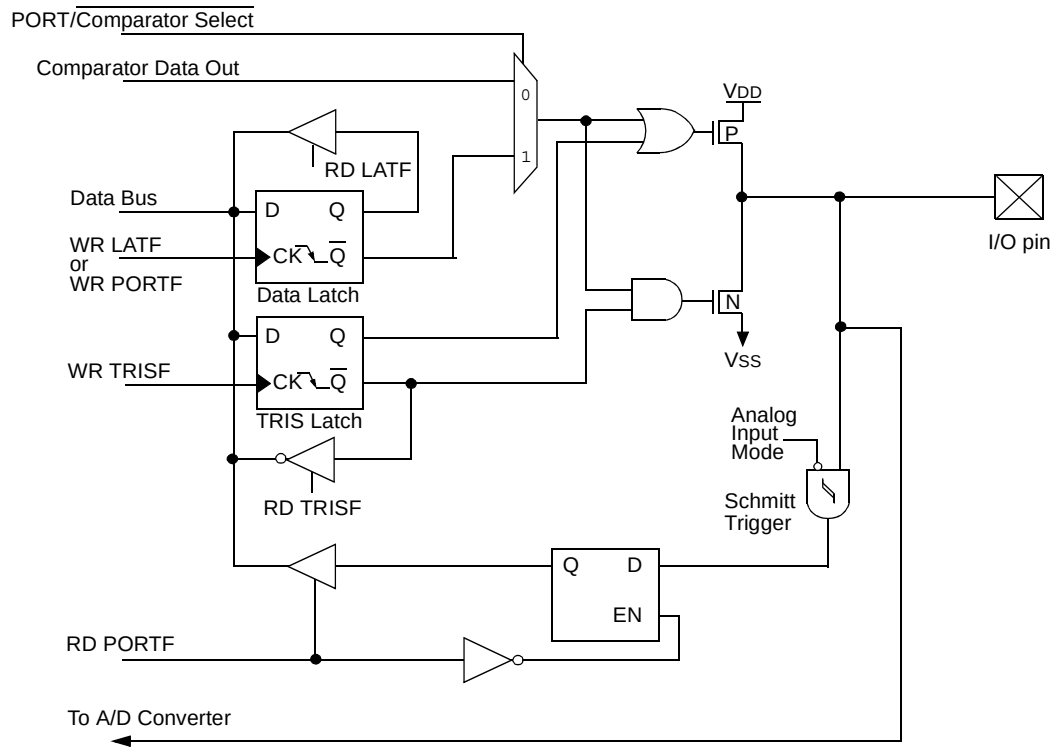
PORTF is multiplexed with several analog peripheral functions, including the A/D converter inputs and comparator inputs, outputs and voltage reference.

| | |
|---|---|
| **Note 1:** | On a Power-on Reset, the RF6:RF0 pins are configured as inputs and read as '0'. |
| **2:** | To configure PORTF as digital I/O, turn off comparators and set ADCON1 value. |

**EXAMPLE 10-6: INITIALIZING PORTF**

```
CLRF    PORTF   ; Initialize PORTF by
                ; clearing output
                ; data latches
CLRF    LATF    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0x07    ;
MOVWF   CMCON   ; Turn off comparators
MOVLW   0x0F    ;
MOVWF   ADCON1  ; Set PORTF as digital I/O
MOVLW   0xCF    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISF   ; Set RF3:RF0 as inputs
                ; RF5:RF4 as outputs
                ; RF7:RF6 as inputs
```

**FIGURE 10-13: PORTF RF1/AN6/C2OUT, RF2/AN7/C1OUT PINS BLOCK DIAGRAM**




**Note 1:** I/O pins have diode protection to VDD and VSS.

## 10.10 Parallel Slave Port

PORTD also operates as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (PSPCON<4>) is set. It is asynchronously readable and writable by the external world through $\overline{RD}$ control input pin, RE0/$\overline{RD}$ and $\overline{WR}$ control input pin, RE1/$\overline{WR}$.

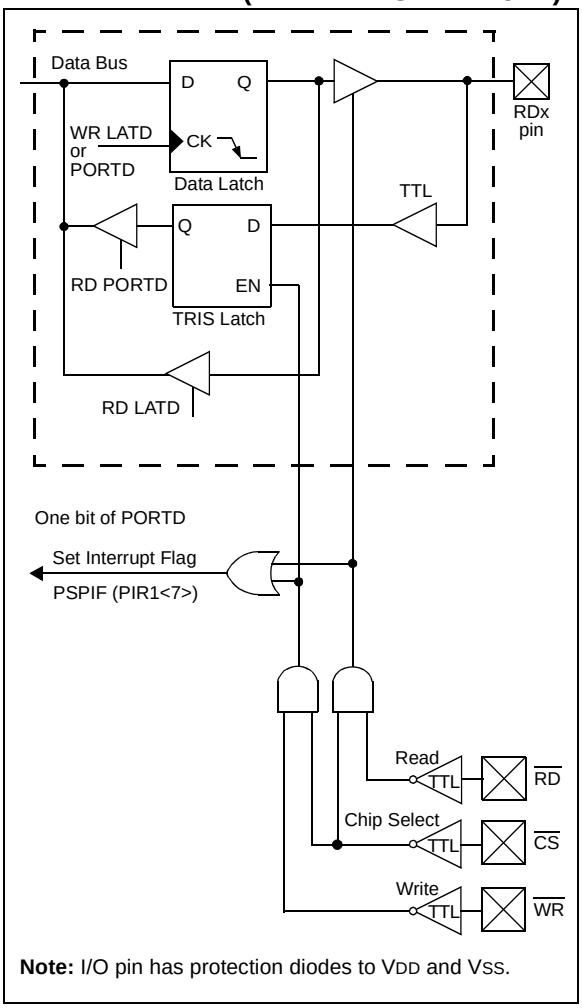
| **Note:** | For PIC18F8525/8621 devices, the Parallel Slave Port is available only in Microcontroller mode. |
|---|---|

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$ to be the $\overline{RD}$ input, RE1/$\overline{WR}$ to be the $\overline{WR}$ input and RE2/$\overline{CS}$ to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). The A/D port configuration bits, PCFG2:PCFG0 (ADCON1<2:0>), must be set, which will configure pins RE2:RE0 as digital I/O.

A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low.

The PORTE I/O pins become control inputs for the microprocessor port when bit PSPMODE (PSPCON<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs) and the ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

**FIGURE 10-24: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**




**Note:** I/O pin has protection diodes to $V_{DD}$ and $V_{SS}$.

## 12.1 Timer1 Operation

Timer1 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

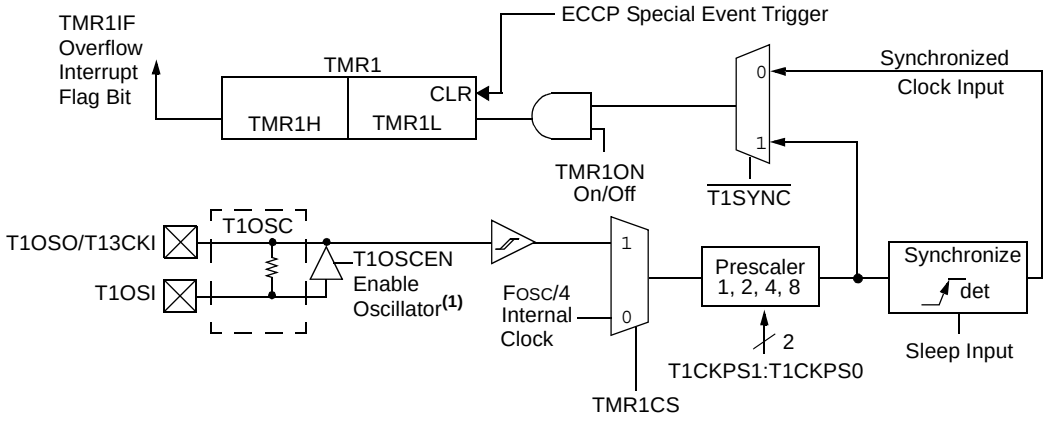
The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

When TMR1CS = 0, Timer1 increments every instruction cycle. When TMR1CS = 1, Timer1 increments on every rising edge of the external clock input or the Timer1 oscillator, if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. That is, the TRISC<1:0> value is ignored and the pins are read as '0'.

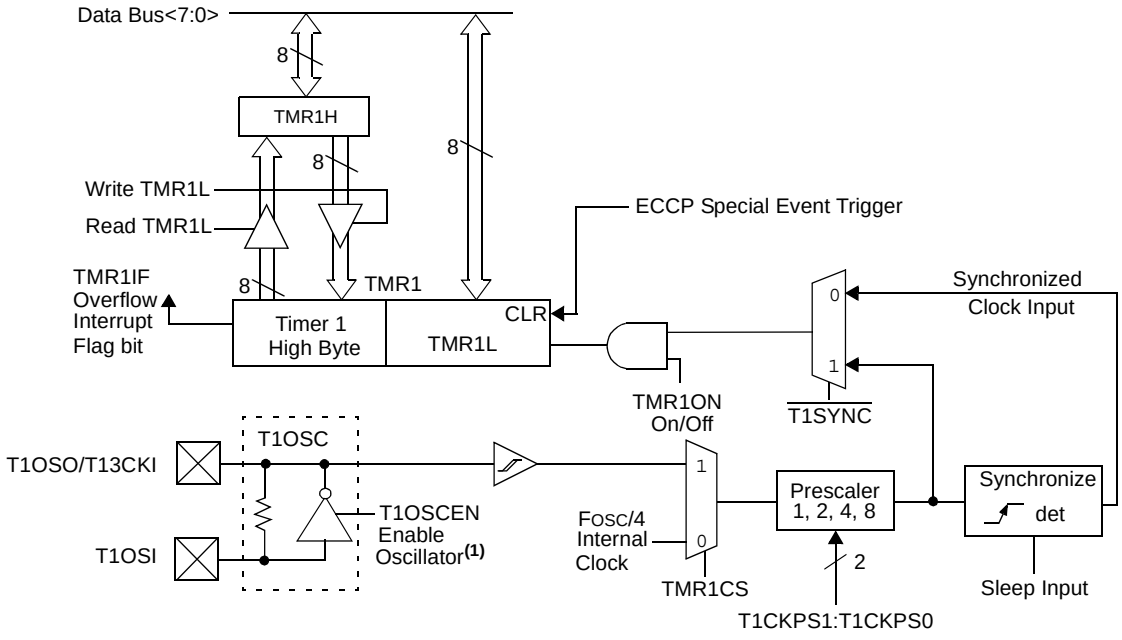
Timer1 also has an internal "Reset input". This Reset can be generated by the ECCP1 or ECCP2 special event trigger. This is discussed in detail in **Section 12.4 "Resetting Timer1 Using an ECCP Special Trigger Output"**.

**FIGURE 12-1: TIMER1 BLOCK DIAGRAM**




**Note 1:** When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

**FIGURE 12-2: TIMER1 BLOCK DIAGRAM: 16-BIT READ/WRITE MODE**




**Note 1:** When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

## 14.1 Timer3 Operation

Timer3 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

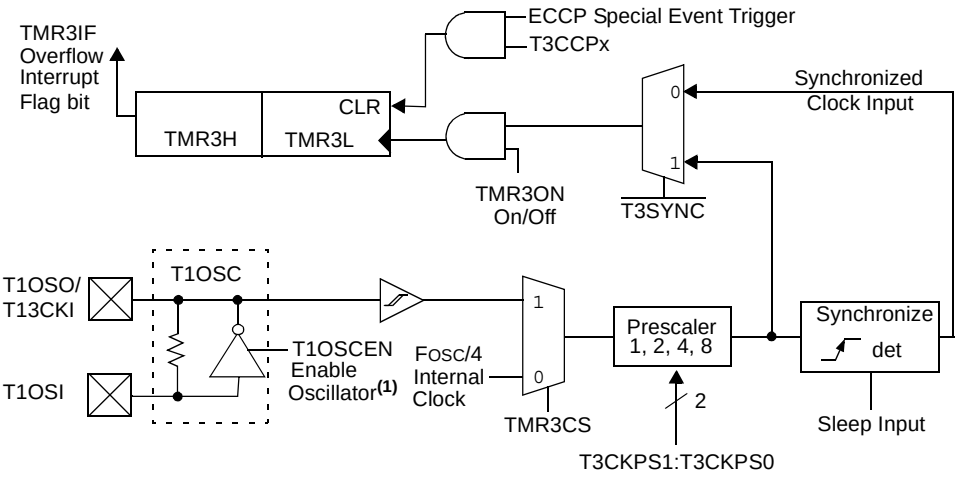
The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>).

When TMR3CS = 0, Timer3 increments every instruction cycle. When TMR3CS = 1, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. That is, the TRISC<1:0> value is ignored and the pins are read as '0'.

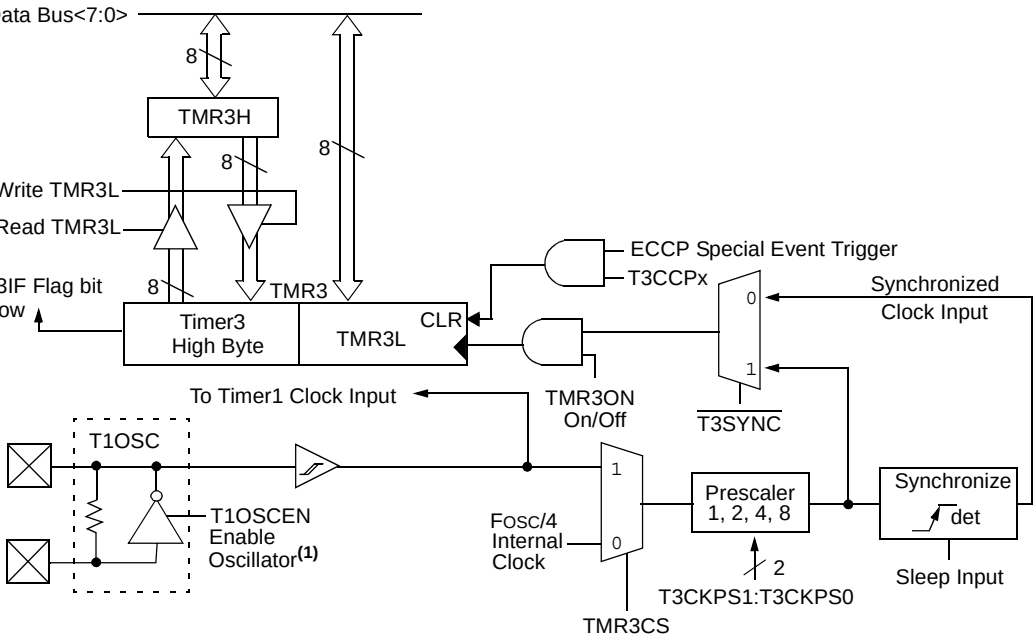
Timer3 also has an internal "Reset input". This Reset can be generated by the ECCP module (**Section 14.0 "Timer3 Module"**).

**FIGURE 14-1: TIMER3 BLOCK DIAGRAM**




**Note 1:** When enable bit T1OSCEN is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

**FIGURE 14-2: TIMER3 BLOCK DIAGRAM CONFIGURED IN 16-BIT READ/WRITE MODE**




**Note 1:** When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

## 15.0 TIMER4 MODULE

The Timer4 module timer has the following features:

- 8-bit timer (TMR4 register)
- 8-bit period register (PR4)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR4 match of PR4

Timer4 has a control register shown in Register 15-1. Timer4 can be shut off by clearing control bit, TMR4ON (T4CON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4 are also controlled by this register. Figure 15-1 is a simplified block diagram of the Timer4 module.

### 15.1 Timer4 Operation

Timer4 can be used as the PWM time base for the PWM mode of the CCP module. The TMR4 register is readable and writable and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T4CKPS1:T4CKPS0 (T4CON<1:0>). The match output of TMR4 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR4 interrupt, latched in flag bit TMR4IF (PIR3<3>).

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR4 register
- a write to the T4CON register
- any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset, or Brown-out Reset)

TMR4 is not cleared when T4CON is written.

**REGISTER 15-1: T4CON: TIMER4 CONTROL REGISTER**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | T4OUTPS3 | T4OUTPS2 | T4OUTPS1 | T4OUTPS0 | TMR4ON | T4CKPS1 | T4CKPS0 |
| bit 7 | | | | | | | bit 0 |

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

bit 6-3 **T4OUTPS3:T4OUTPS0**: Timer4 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR4ON**: Timer4 On bit
1 = Timer4 is on
0 = Timer4 is off

bit 1-0 **T4CKPS1:T4CKPS0**: Timer4 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

| **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 |

**TABLE 16-2: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 0000 000u |
| RCON | IPEN | — | — | $\overline{RI}$ | $\overline{TO}$ | $\overline{PD}$ | $\overline{POR}$ | $\overline{BOR}$ | 0--1 11qq | 0--q qquu |
| PIR1 | PSPIF(1) | ADIF | RC1IF | TX1IF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| PIE1 | PSPIE(1) | ADIE | RC1IE | TX1IE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| IPR1 | PSPIP(1) | ADIP | RC1IP | TX1IP | SSPIP | CCP1IP | TMR2IP | TMR1IP | 1111 1111 | 1111 1111 |
| PIR2 | — | CMIF | — | EEIF | BCLIF | LVDIF | TMR3IF | CCP2IF | -0-0 0000 | ---0 0000 |
| PIE2 | — | CMIE | — | EEIE | BCLIE | LVDIE | TMR3IE | CCP2IE | -0-0 0000 | ---0 0000 |
| IPR2 | — | CMIP | — | EEIP | BCLIP | LVDIP | TMR3IP | CCP2IP | -1-1 1111 | ---1 1111 |
| PIR3 | — | — | RC2IF | TX2IF | TMR4IF | CCP5IF | CCP4IF | CCP3IF | --00 0000 | --00 0000 |
| PIE3 | — | — | RC2IE | TX2IE | TMR4IE | CCP5IE | CCP4IE | CCP3IE | --00 0000 | --00 0000 |
| IPR3 | — | — | RC2IP | TX2IP | TMR4IP | CCP5IP | CCP4IP | CCP3IP | --11 1111 | --11 1111 |
| TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| TRISE | PORTE Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| TRISG | — | — | — | PORTG Data Direction Register | | | | | ---1 1111 | ---1 1111 |
| TMR1L | Timer1 Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| TMR1H | Timer1 Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| T1CON | RD16 | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | 0-00 0000 | u-uu uuuu |
| TMR3H | Timer3 Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| TMR3L | Timer3 Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| T3CON | RD16 | T3CCP2 | T3CKPS1 | T3CKPS0 | T3CCP1 | $\overline{T3SYNC}$ | TMR3CS | TMR3ON | 0000 0000 | uuuu uuuu |
| CCPR1L | Enhanced Capture/Compare/PWM Register 1 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCPR1H | Enhanced Capture/Compare/PWM Register 1 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCP1CON | P1M1 | P1M0 | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | 0000 0000 | 0000 0000 |
| CCPR2L | Enhanced Capture/Compare/PWM Register 2 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCPR2H | Enhanced Capture/Compare/PWM Register 2 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCP2CON | P2M1 | P2M0 | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | 0000 0000 | 0000 0000 |
| CCPR3L | Enhanced Capture/Compare/PWM Register 3 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCPR3H | Enhanced Capture/Compare/PWM Register 3 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCP3CON | P3M1 | P3M0 | DC3B1 | DC3B0 | CCP3M3 | CCP3M2 | CCP3M1 | CCP3M0 | 0000 0000 | 0000 0000 |
| CCPR4L | Capture/Compare/PWM Register 4 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCPR4H | Capture/Compare/PWM Register 4 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCP4CON | — | — | DC4B1 | DC4B0 | CCP4M3 | CCP4M2 | CCP4M1 | CCP4M0 | --00 0000 | --00 0000 |
| CCPR5L | Capture/Compare/PWM Register 5 Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCPR5H | Capture/Compare/PWM Register 5 High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| CCP5CON | — | — | DC5B1 | DC5B0 | CCP5M3 | CCP5M2 | CCP5M1 | CCP5M0 | --00 0000 | --00 0000 |

**Legend:** x = unknown, u = unchanged, — = unimplemented, read as '0'.
Shaded cells are not used by Capture and Compare, Timer1 or Timer3.

**Note 1:** Enabled only in Microcontroller mode for PIC18F8525/8621 devices.

### 18.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 18-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

**EXAMPLE 18-1: LOADING THE SSPBUF (SSPSR) REGISTER**

```
LOOP    BTFSS   SSPSTAT, BF         ;Has data been received (transmit complete)?
        BRA     LOOP                ;No
        MOVF    SSPBUF, W           ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA              ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W           ;W reg = contents of TXDATA
        MOVWF   SSPBUF              ;New data to xmit
```

### 18.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

| | |
|---|---|
| **Note 1:** | If RSEN is programmed while any other event is in progress, it will not take effect. |
| **2:** | A bus collision during the Repeated Start condition occurs if:<br>• SDA is sampled low when SCL goes from low-to-high.<br>• SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'. |

Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

#### 18.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

| | |
|---|---|
| **Note:** | Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete. |

**FIGURE 18-20: REPEATED START CONDITION WAVEFORM**

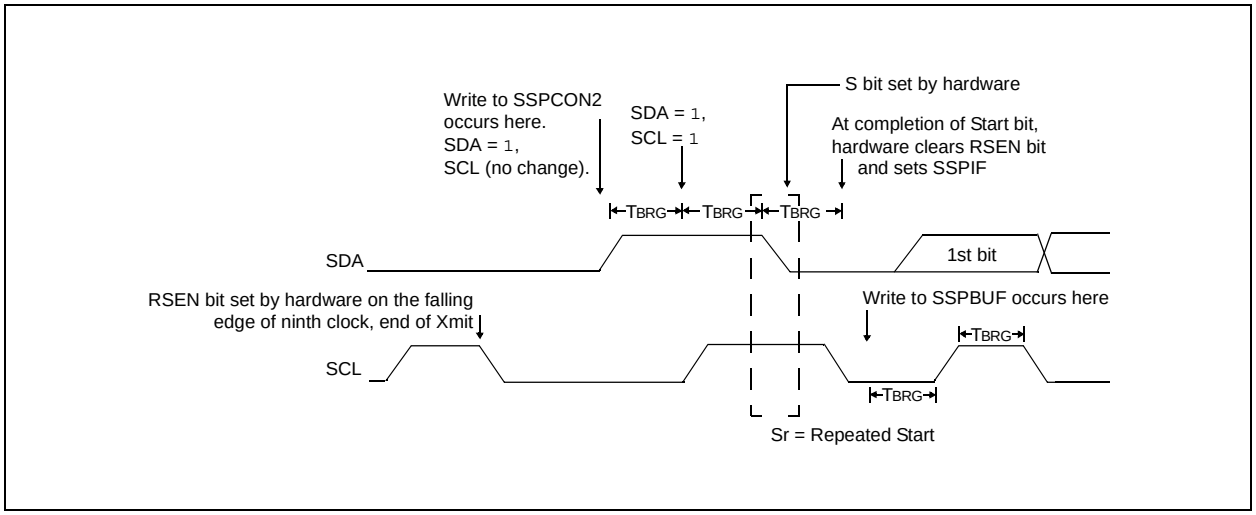


**FIGURE 18-22: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)**

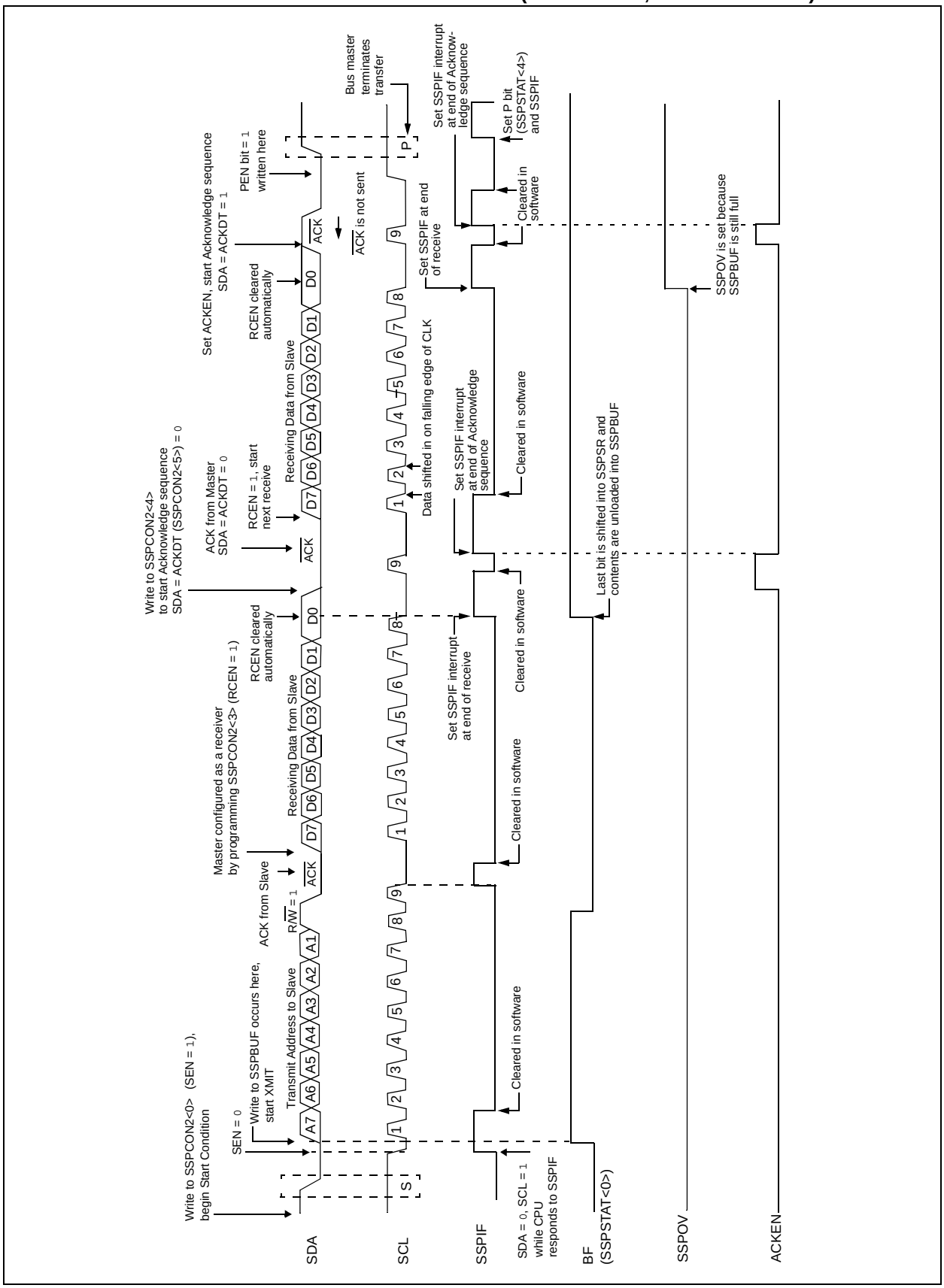


**FIGURE 18-27: BUS COLLISION DURING START CONDITION (SCL = 0)**

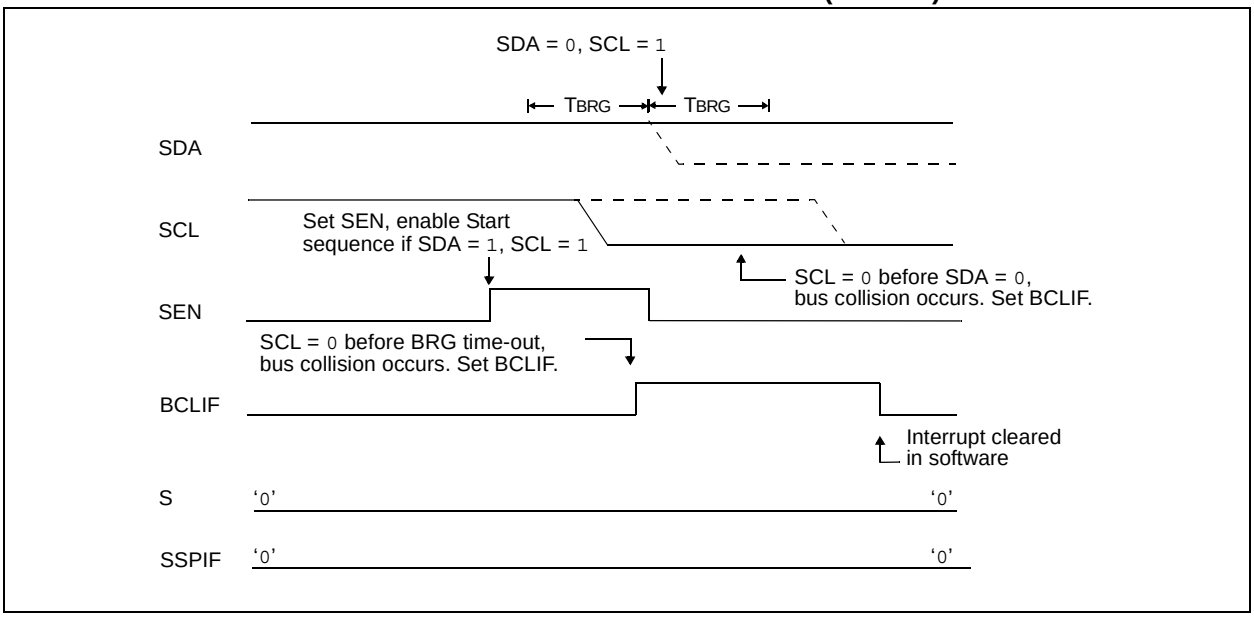
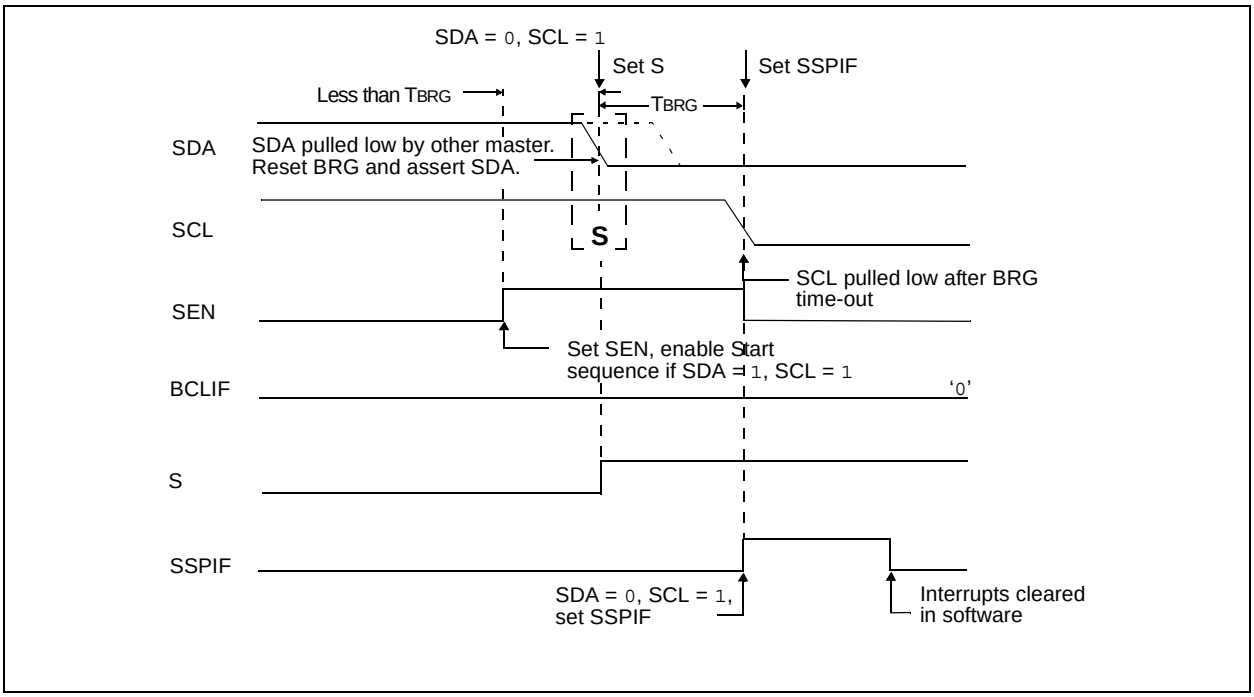



**FIGURE 18-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



## 23.1 Control Register

The Low-Voltage Detect Control register (Register 23-1) controls the operation of the Low-Voltage Detect circuitry.

**REGISTER 23-1: LVDCON: LOW-VOLTAGE DETECT CONTROL REGISTER**

| U-0 | U-0 | R-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | IRVST | LVDEN | LVDL3 | LVDL2 | LVDL1 | LVDL0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit

1 = Indicates that the Low-Voltage Detect logic will generate the interrupt flag at the specified voltage range

0 = Indicates that the Low-Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the LVD interrupt should not be enabled

bit 4 **LVDEN:** Low-Voltage Detect Power Enable bit

1 = Enables LVD, powers up LVD circuit

0 = Disables LVD, powers down LVD circuit

bit 3-0 **LVDL3:LVDL0:** Low-Voltage Detection Limit bits

1111 = External analog input is used (input comes from the LVDIN pin)
1110 = 4.45V-4.83V
1101 = 4.16V-4.5V
1100 = 3.96V-4.3V
1011 = 3.76V-3.92V
1010 = 3.57V-3.87V
1001 = 3.47V-3.75V
1000 = 3.27V-3.55V
0111 = 2.98V-3.22V
0110 = 2.77V-3.01V
0101 = 2.67V-2.89V
0100 = 2.48V-2.68V
0011 = 2.37V-2.57V
0010 = 2.18V-2.36V
0001 = 1.98V-2.14V
0000 = Reserved

**Note:** LVDL3:LVDL0 modes, which result in a trip point below the valid operating voltage of the device, are not tested.

| **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 |

**MULLW** **Multiply Literal with W**

Syntax: [ *label* ] MULLW k

Operands: $0 \le k \le 255$

Operation: (W) x k → PRODH:PRODL

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 1101 | kkkk | kkkk |

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged.
None of the Status flags are affected. Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'k' | Process Data | Write registers PRODH: PRODL |

Example: `MULLW 0xC4`

Before Instruction
W = 0xE2
PRODH = ?
PRODL = ?

After Instruction
W = 0xE2
PRODH = 0xAD
PRODL = 0x08

**MULWF** **Multiply W with f**

Syntax: [ *label* ] MULWF f [,a]

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: (W) x (f) → PRODH:PRODL

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 001a | ffff | ffff |

Description: An unsigned multiplication is carried out between the contents of W and the register file location 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte.
Both W and 'f' are unchanged.
None of the Status flags are affected. Note that neither overflow nor carry is possible in this operation. A zero result is possible but not detected. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write registers PRODH: PRODL |

Example: `MULWF REG, 1`

Before Instruction
W = 0xC4
REG = 0xB5
PRODH = ?
PRODL = ?

After Instruction
W = 0xC4
REG = 0xB5
PRODH = 0x8A
PRODL = 0x94

**FIGURE 28-13: AVERAGE FOSC vs. VDD FOR VARIOUS Rs (RC MODE, C = 100 pF, TEMP = 25°C)**

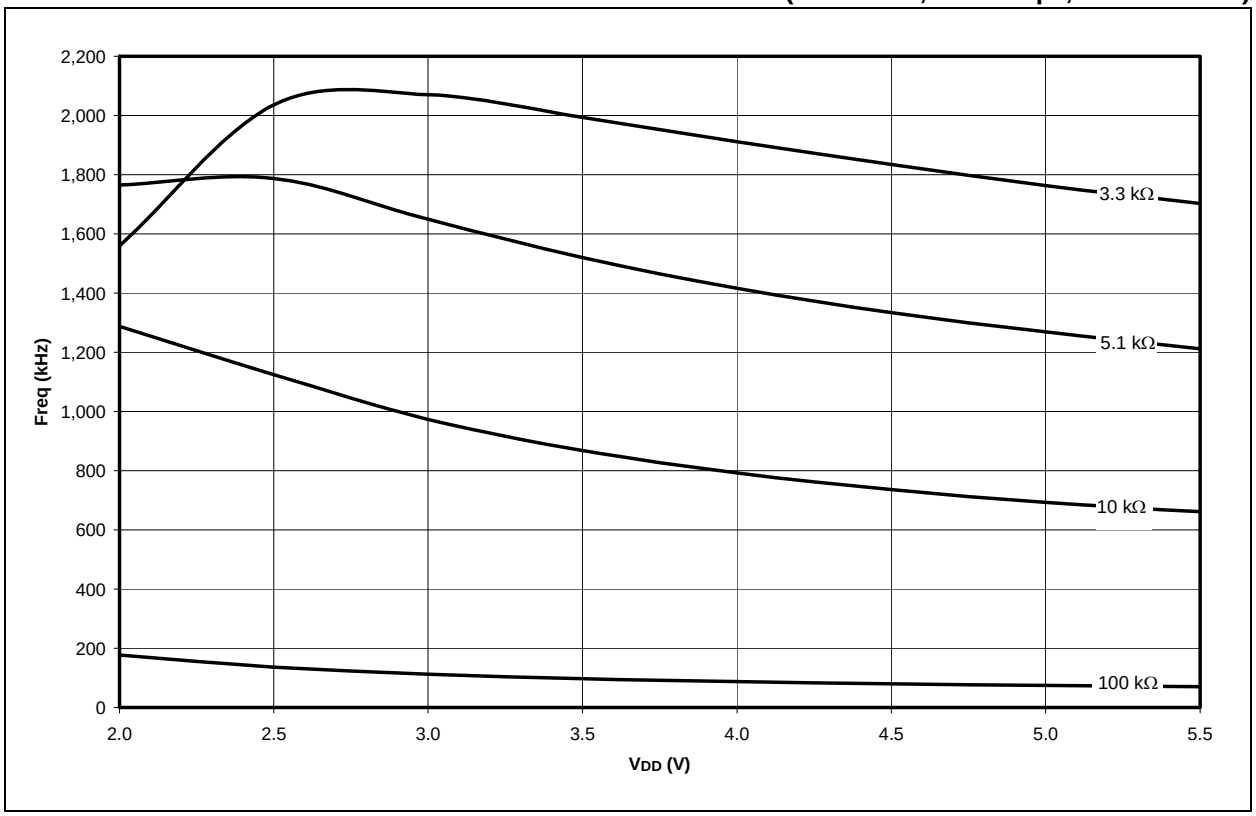
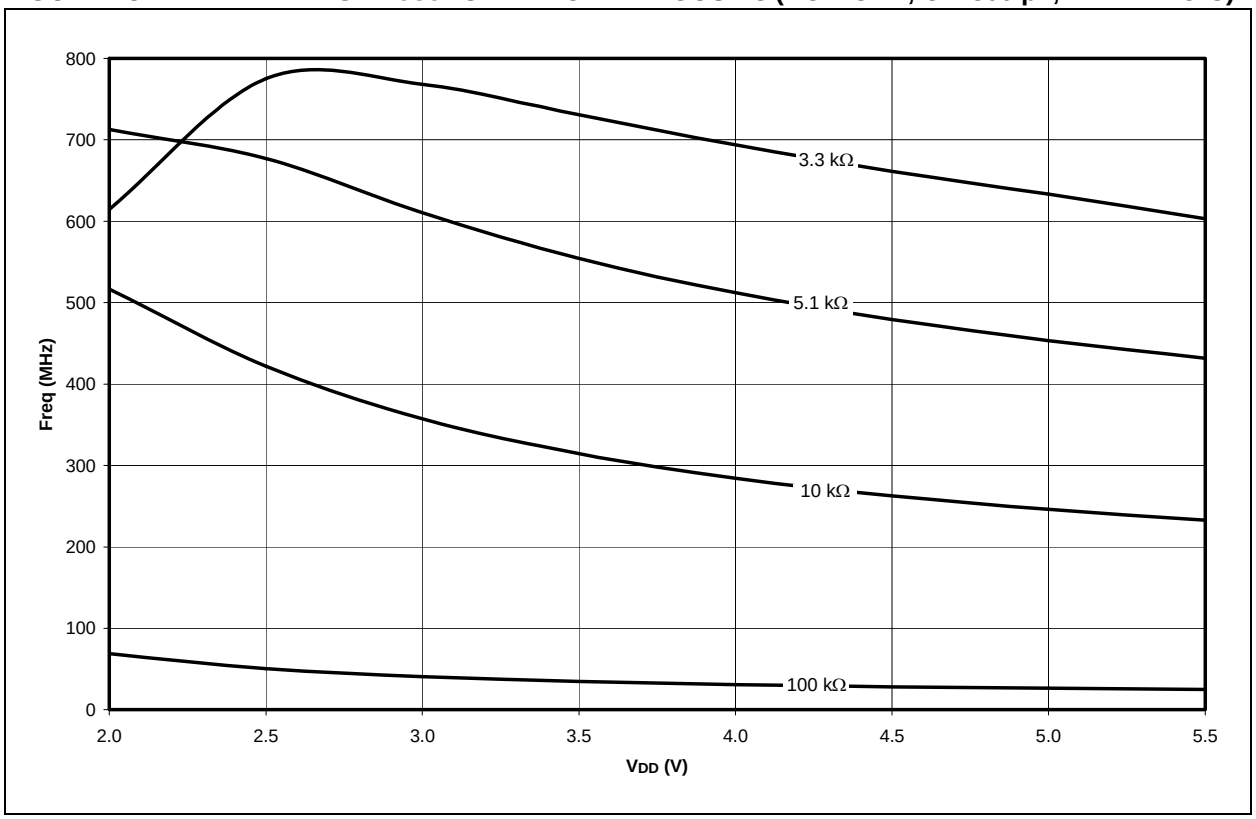



**FIGURE 28-14: AVERAGE FOSC vs. VDD FOR VARIOUS Rs (RC MODE, C = 300 pF, TEMP = 25°C)**



**NOTES:**

**NOTES:**