



Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	21
Program Memory Size	12KB (6K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	640 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f2439t-e-so">https://www.e-xfl.com/product-detail/microchip-technology/pic18f2439t-e-so</a>

# PIC18FXX39

**TABLE 1-1: PIC18FXX39 DEVICE FEATURES**

Features	PIC18F2439	PIC18F2539	PIC18F4439	PIC18F4539
Operating Frequency	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz
Program Memory (Bytes)	12K	24K	12K	24K
Program Memory (Instructions)	6144	12288	6144	12288
Data Memory (Bytes)	640	1408	640	1408
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	15	15	16	16
I/O Ports	Ports A, B, C	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	3	3	3	3
PWM Modules <sup>(1)</sup>	2	2	2	2
Single Phase Induction Motor Control	Yes	Yes	Yes	Yes
Serial Communications	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART
Parallel Communications	—	—	PSP	PSP
10-bit Analog-to-Digital Module	5 input channels	5 input channels	8 input channels	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions	75 Instructions	75 Instructions	75 Instructions
Packages	28-pin DIP 28-pin SOIC	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin TQFP 44-pin QFN	40-pin DIP 44-pin TQFP 44-pin QFN

**Note 1:** PWM modules are used exclusively in conjunction with the motor control kernel, and are not available for other applications.

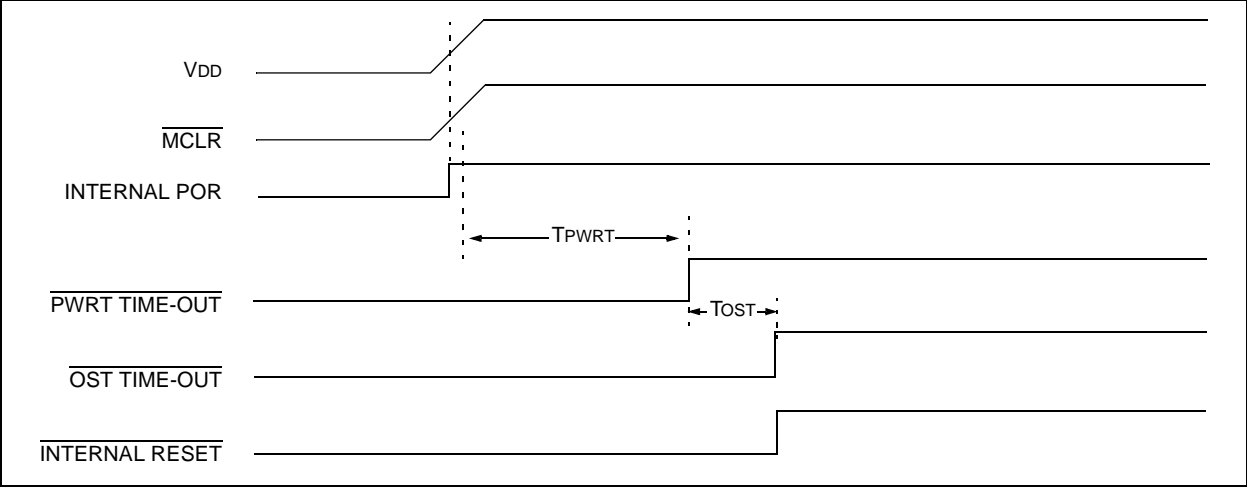
# PIC18FXX39

---

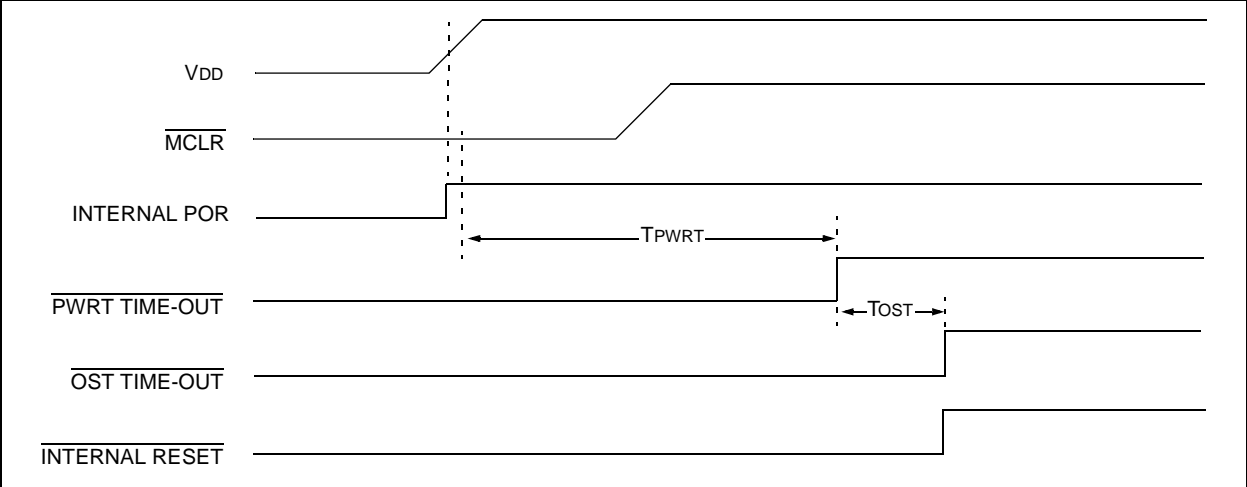
NOTES:

# PIC18FXX39

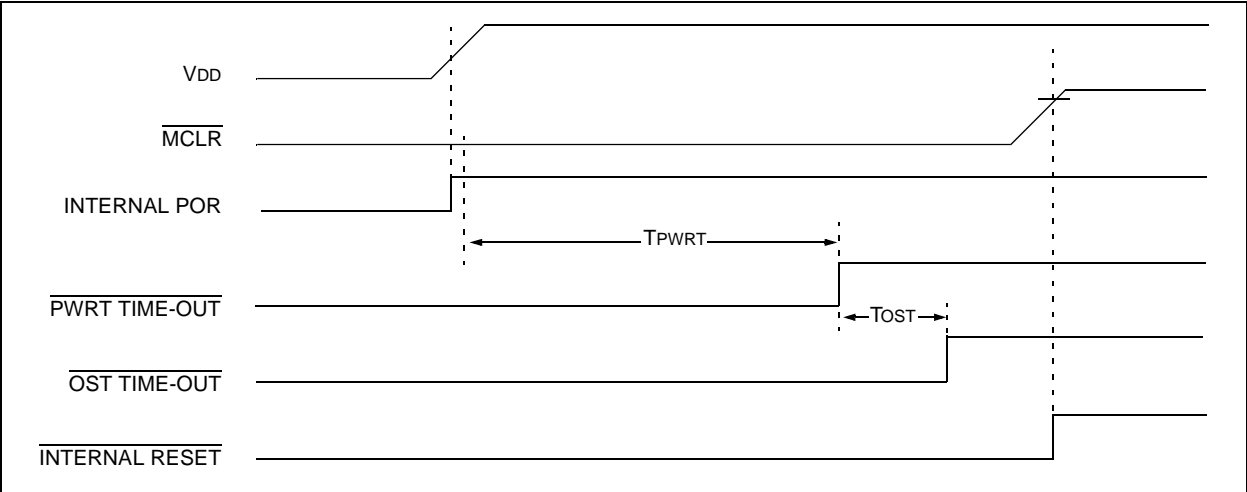
**FIGURE 3-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ )**



**FIGURE 3-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**



**FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**



## 5.5 Writing to FLASH Program Memory

The minimum programming block is 4 words or 8 bytes. Word or byte programming is not supported.

Table Writes are used internally to load the holding registers needed to program the FLASH memory. There are 8 holding registers used by the Table Writes for programming.

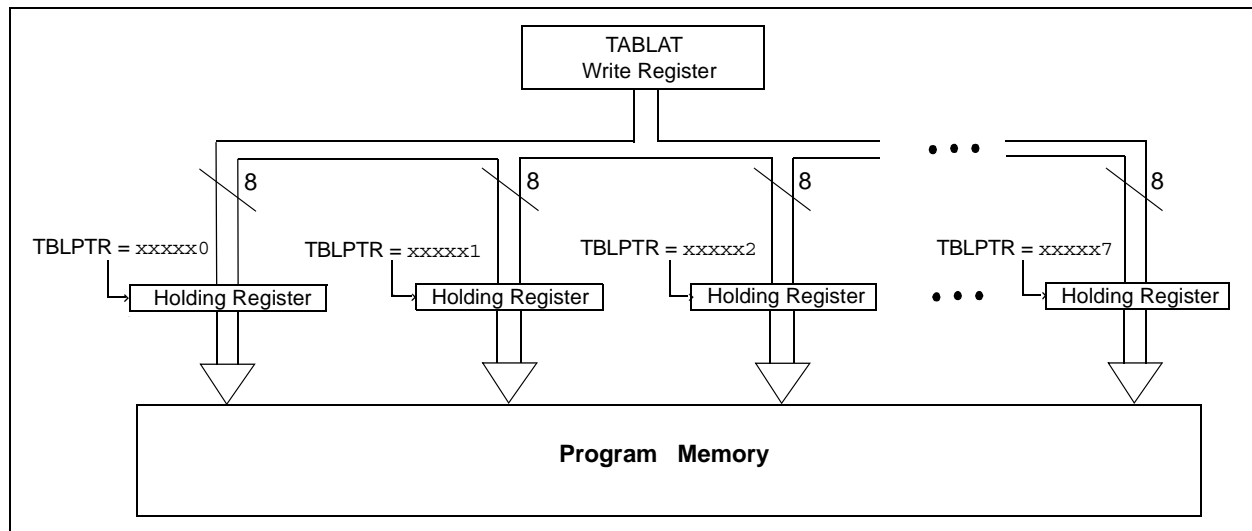
Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction has to be executed 8 times for each programming operation. All of the Table Write

operations will essentially be short writes, because only the holding registers are written. At the end of updating 8 registers, the EECON1 register must be written to, to start the programming operation with a long write.

The long write is necessary for programming the internal FLASH. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

**FIGURE 5-5: TABLE WRITES TO FLASH PROGRAM MEMORY**



### 5.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer with address being erased.
4. Do the row erase procedure.
5. Load Table Pointer with address of first byte being written.
6. Write the first 8 bytes into the holding registers with auto-increment (TBLWT\*+ or TBLWT+\*).
7. Set EEPGD bit to point to program memory, clear the CFGS bit to access program memory, and set WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.

10. Write AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Repeat steps 6-14 seven times, to write 64 bytes.
15. Verify the memory (Table Read).

This procedure will require about 18 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 5-3.

**Note:** Before setting the WR bit, the table pointer address needs to be within the intended address range of the 8 bytes in the holding registers.

# PIC18FXX39

## EXAMPLE 5-3: WRITING TO FLASH PROGRAM MEMORY

```
        MOVLW    D'64                ; number of bytes in erase block
        MOVWF    COUNTER
        MOVLW    BUFFER_ADDR_HIGH    ; point to buffer
        MOVWF    FSR0H
        MOVLW    BUFFER_ADDR_LOW
        MOVWF    FSR0L
        MOVLW    CODE_ADDR_UPPER      ; Load TBLPTR with the base
        MOVWF    TBLPTRU              ; address of the memory block
        MOVLW    CODE_ADDR_HIGH
        MOVWF    TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF    TBLPTRL

READ_BLOCK
        TBLRD*+                      ; read into TABLAT, and inc
        MOVF     TABLAT, W            ; get data
        MOVWF    POSTINC0            ; store data
        DECFSZ   COUNTER             ; done?
        BRA      READ_BLOCK          ; repeat

MODIFY_WORD
        MOVLW    DATA_ADDR_HIGH      ; point to buffer
        MOVWF    FSR0H
        MOVLW    DATA_ADDR_LOW
        MOVWF    FSR0L
        MOVLW    NEW_DATA_LOW         ; update buffer word
        MOVWF    POSTINC0
        MOVLW    NEW_DATA_HIGH
        MOVWF    INDF0

ERASE_BLOCK
        MOVLW    CODE_ADDR_UPPER      ; load TBLPTR with the base
        MOVWF    TBLPTRU              ; address of the memory block
        MOVLW    CODE_ADDR_HIGH
        MOVWF    TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF    TBLPTRL
        BSF      EECON1,EEPGD         ; point to FLASH program memory
        BCF      EECON1,CFG5          ; access FLASH program memory
        BSF      EECON1,WREN          ; enable write to memory
        BSF      EECON1,FREE          ; enable Row Erase operation
        BCF      INTCON,GIE           ; disable interrupts
        MOVLW    55h
        MOVWF    EECON2               ; write 55h
        MOVLW    AAh
        MOVWF    EECON2               ; write AAh
        BSF      EECON1,WR            ; start erase (CPU stall)
        BSF      INTCON,GIE           ; re-enable interrupts
        TBLRD*-                      ; dummy read decrement

WRITE_BUFFER_BACK
        MOVLW    8                    ; number of write buffer groups of 8 bytes
        MOVWF    COUNTER_HI
        MOVLW    BUFFER_ADDR_HIGH    ; point to buffer
        MOVWF    FSR0H
        MOVLW    BUFFER_ADDR_LOW
        MOVWF    FSR0L

PROGRAM_LOOP
        MOVLW    8                    ; number of bytes in holding register
        MOVWF    COUNTER

WRITE_WORD_TO_HREGS
        MOVF     POSTINC0, W          ; get low byte of buffer data
        MOVWF    TABLAT              ; present data to table latch
        TBLWT*+                      ; write data, perform a short write
                                      ; to internal TBLWT holding register.
        DECFSZ   COUNTER              ; loop until buffers are full
        BRA      WRITE_WORD_TO_HREGS
```

## 6.0 DATA EEPROM MEMORY

The Data EEPROM is readable and writable during normal operation over the entire VDD range. The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are four SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADR

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 256 bytes of data EEPROM with an address range from 0h to FFh.

The EEPROM data memory is rated for high erase/write cycles. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip to chip. Please refer to parameter D122 (Electrical Characteristics, Section 23.0) for exact limits.

## 6.1 EEADR

The address register can address up to a maximum of 256 bytes of data EEPROM.

## 6.2 EECON1 and EECON2 Registers

EECON1 is the control register for EEPROM memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the EEPROM write sequence.

Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset, or a WDT Time-out Reset during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR), due to the RESET condition forcing the contents of the registers to zero.

**Note:** Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared in software.

## 6.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD control bit (EECON1<7>), clear the CFGS control bit

(EECON1<6>), and then set control bit RD (EECON1<0>). The data is available for the very next instruction cycle; therefore, the EEDATA register can be read by the next instruction. EEDATA will hold this value until another read operation, or until it is written to by the user (during a write operation).

### EXAMPLE 6-1: DATA EEPROM READ

```
MOVLW DATA_EE_ADDR ;
MOVWF EEADR          ; Data Memory Address to read
BCF    EECON1, EEPGD ; Point to DATA memory
BCF    EECON1, CFGS  ; Access program FLASH or Data EEPROM memory
BSF    EECON1, RD     ; EEPROM Read
MOVF   EEDATA, W      ; W = EEDATA
```

## 6.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. Then, the sequence in Example 6-2 must be followed to initiate the write cycle.

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code exe-

cution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt, or poll this bit. EEIF must be cleared by software.

### EXAMPLE 6-2: DATA EEPROM WRITE

	MOVLW DATA_EE_ADDR ;
	MOVWF EEADR          ; Data Memory Address to read
	MOVLW DATA_EE_DATA  ;
	MOVWF EEDATA         ; Data Memory Value to write
	BCF    EECON1, EEPGD ; Point to DATA memory
	BCF    EECON1, CFGS  ; Access program FLASH or Data EEPROM memory
	BSF    EECON1, WREN  ; Enable writes
<b>Required Sequence</b>	BCF    INTCON, GIE   ; Disable interrupts
	MOVLW  55h          ;
	MOVWF  EECON2       ; Write 55h
	MOVLW  AAh          ;
	MOVWF  EECON2       ; Write AAh
	BSF    EECON1, WR    ; Set WR bit to begin write
	BSF    INTCON, GIE   ; Enable interrupts
	. ; user code execution
	. ;
	. ;
	BCF    EECON1, WREN  ; Disable writes on write complete (EEIF set)



# PIC18FXX39

## 8.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Enable registers (PIE1, PIE2). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

### REGISTER 8-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	—	TMR2IE <sup>(2)</sup>	TMR1IE
bit 7						bit 0	

bit 7 **PSPIE<sup>(1)</sup>**: Parallel Slave Port Read/Write Interrupt Enable bit

1 = Enables the PSP read/write interrupt  
0 = Disables the PSP read/write interrupt

bit 6 **ADIE**: A/D Converter Interrupt Enable bit

1 = Enables the A/D interrupt  
0 = Disables the A/D interrupt

bit 5 **RCIE**: USART Receive Interrupt Enable bit

1 = Enables the USART receive interrupt  
0 = Disables the USART receive interrupt

bit 4 **TXIE**: USART Transmit Interrupt Enable bit

1 = Enables the USART transmit interrupt  
0 = Disables the USART transmit interrupt

bit 3 **SSPIE**: Master Synchronous Serial Port Interrupt Enable bit

1 = Enables the MSSP interrupt  
0 = Disables the MSSP interrupt

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

bit 1 **TMR2IE<sup>(2)</sup>**: TMR2 to PR2 Match Interrupt Enable bit

1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE**: TMR1 Overflow Interrupt Enable bit

1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

**Note 1:** This bit is reserved on PIC18F2X39 devices; always maintain this bit clear.

**Note 2:** This bit is reserved for use by the ProMPT kernel; do not alter its value.

#### 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

## 10.1 Timer0 Operation

Timer0 can operate as a timer or as a counter.

Timer mode is selected by clearing the T0CS bit. In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0L register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0L register.

Counter mode is selected by setting the T0CS bit. In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit (T0SE). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed below.

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (Tosc). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

## 10.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not readable or writable.

The PSA and T0PS2:T0PS0 bits determine the prescaler assignment and prescale ratio.

Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values in power-of-2 increments, from 1:2 through 1:256, are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0L register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) will clear the prescaler count.

**Note:** Writing to TMR0L when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment.

### 10.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control; it can be changed “on-the-fly” during program execution.

## 10.3 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF bit. The interrupt can be masked by clearing the TMR0IE bit. The TMR0IE bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP, since the timer is shut-off during SLEEP.

## 10.4 16-bit Mode Timer Reads and Writes

TMR0H is not the high byte of the timer/counter in 16-bit mode, but is actually a buffered version of the high byte of Timer0 (see Figure 10-2). The high byte of the Timer0 counter/timer is not directly readable nor writable. TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte.

A write to the high byte of Timer0 must also take place through the TMR0H buffer register. Timer0 high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

**TABLE 10-1: REGISTERS ASSOCIATED WITH TIMER0**

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
TMR0L	Timer0 Module Low Byte Register								xxxx xxxx	uuuu uuuu
TMR0H	Timer0 Module High Byte Register								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111
TRISA	—	PORTA Data Direction Register							-111 1111	-111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

## REGISTER 16-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0

bit 7

bit 0

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).

0 = No overflow

**Note:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI, and  $\overline{SS}$  as serial port pins

0 = Disables serial port and configures these pins as I/O port pins

**Note:** When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

1 = IDLE state for clock is a high level

0 = IDLE state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

0101 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control disabled,  $\overline{SS}$  can be used as I/O pin

0100 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control enabled

0011 = Reserved

0010 = SPI Master mode, clock = Fosc/64

0001 = SPI Master mode, clock = Fosc/16

0000 = SPI Master mode, clock = Fosc/4

**Note:** Bit combinations not specifically listed here are either reserved, or implemented in I<sup>2</sup>C mode only.

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

# PIC18FXX39

## 16.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 16-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This then, would give waveforms for SPI communication as shown in

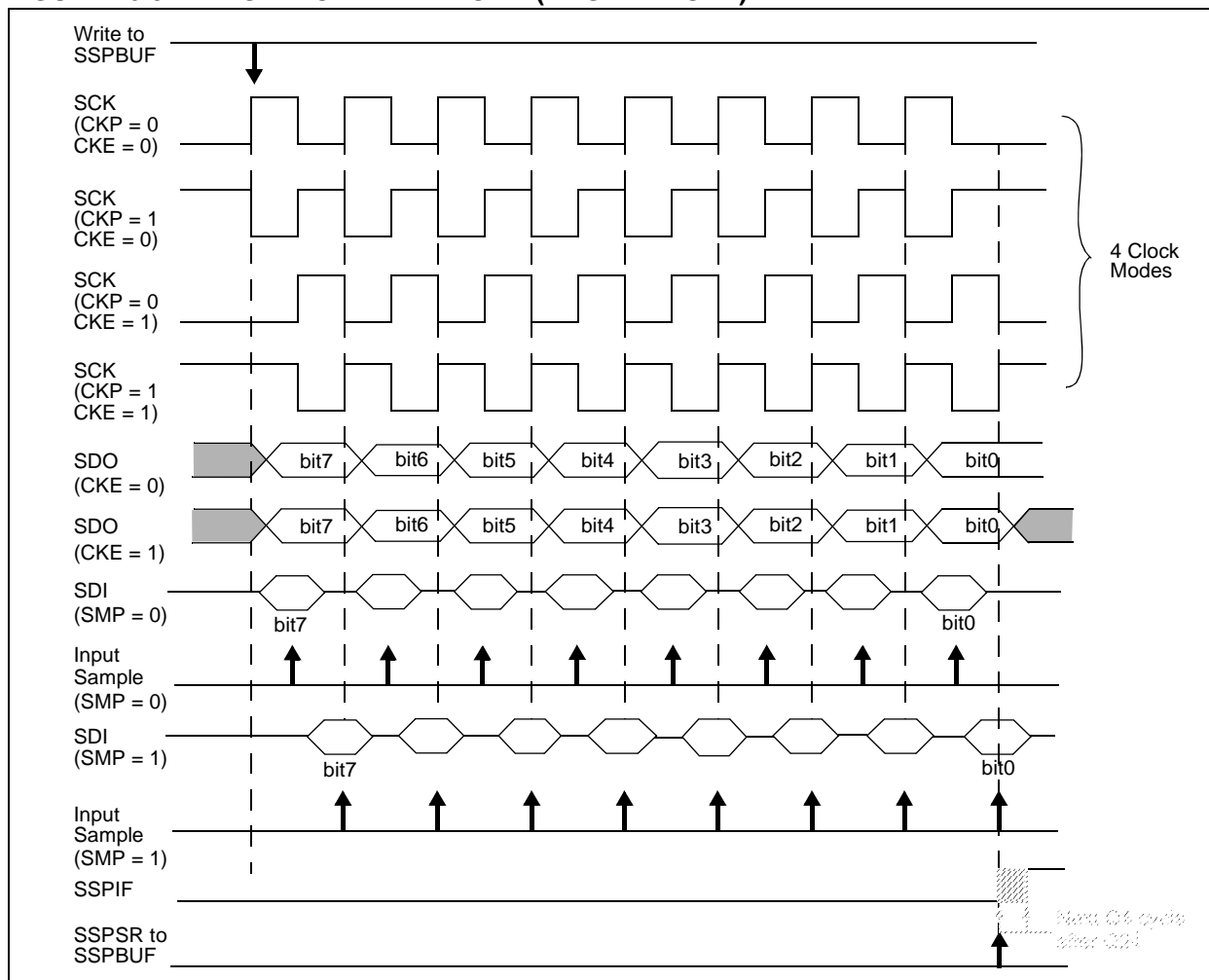
Figure 16-3, Figure 16-5, and Figure 16-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user-programmable to be one of the following:

- $F_{osc}/4$  (or  $T_{CY}$ )
- $F_{osc}/16$  (or  $4 \cdot T_{CY}$ )
- $F_{osc}/64$  (or  $16 \cdot T_{CY}$ )

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 16-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 16-3: SPI MODE WAVEFORM (MASTER MODE)**



## 19.2 Operation

Depending on the power source for the device voltage, the voltage normally decreases relatively slowly. This means that the LVD module does not need to be constantly operating. To decrease the current requirements, the LVD circuitry only needs to be enabled for short periods, where the voltage is checked. After doing the check, the LVD module may be disabled.

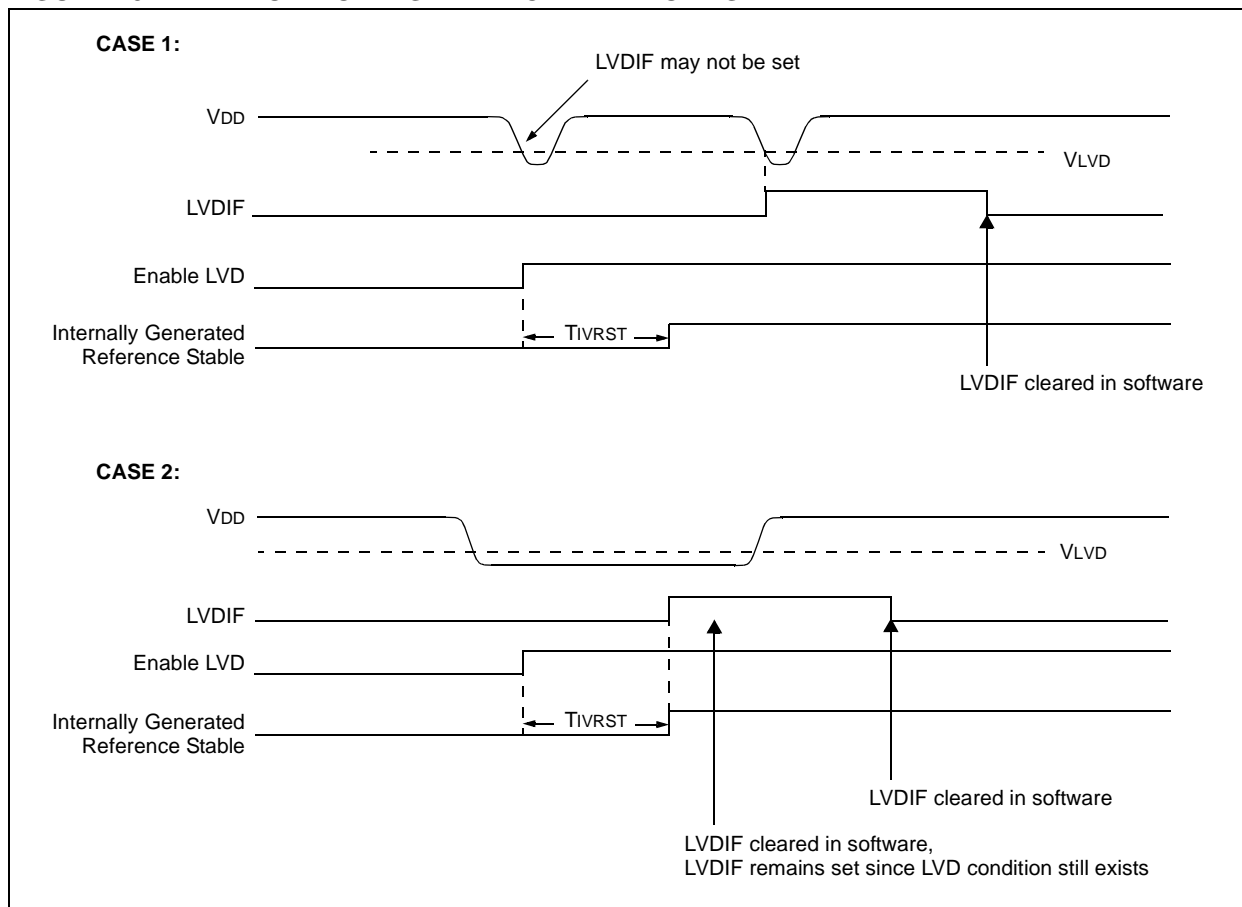
Each time that the LVD module is enabled, the circuitry requires some time to stabilize. After the circuitry has stabilized, all status flags may be cleared. The module will then indicate the proper state of the system.

The following steps are needed to set up the LVD module:

1. Write the value to the LVDL3:LVDL0 bits (LVDCON register), which selects the desired LVD Trip Point.
2. Ensure that LVD interrupts are disabled (the LVDIE bit is cleared or the GIE bit is cleared).
3. Enable the LVD module (set the LVDEN bit in the LVDCON register).
4. Wait for the LVD module to stabilize (the IRVST bit to become set).
5. Clear the LVD interrupt flag, which may have falsely become set until the LVD module has stabilized (clear the LVDIF bit).
6. Enable the LVD interrupt (set the LVDIE and the GIE bits).

Figure 19-4 shows typical waveforms that the LVD module may be used to detect.

**FIGURE 19-4: LOW VOLTAGE DETECT WAVEFORMS**



# PIC18FXX39

**TABLE 20-1: CONFIGURATION BITS AND DEVICE IDS**

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h CONFIG1H	—	—	— <sup>(1)</sup>	—	—	FOSC2	FOSC1	FOSC0	--1- -010
300002h CONFIG2L	—	—	—	—	BORV1	BORV0	BOREN	PWRTEN	---- 1111
300003h CONFIG2H	—	—	—	—	WDTPS2	WDTPS1	WDTPS0	WDTEN	---- 1111
300005h CONFIG3H	—	—	—	—	—	—	—	— <sup>(1)</sup>	---- ---1
300006h CONFIG4L	DEBUG	—	—	—	—	LVP	—	STVREN	1--- -1-1
300008h CONFIG5L	—	—	—	—	— <sup>(1)</sup>	CP2	CP1	CP0	---- 1111
300009h CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah CONFIG6L	—	—	—	—	— <sup>(1)</sup>	WRT2	WRT1	WRT0	---- 1111
30000Bh CONFIG6H	WRD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch CONFIG7L	—	—	—	—	— <sup>(1)</sup>	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	(2)
3FFFFFh DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 0100

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

**Note 1:** Unimplemented, but reserved; maintain this bit set.

**2:** See Register 20-11 for DEVID1 values.

**REGISTER 20-1: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)**

U-0	U-0	U-1	U-0	U-0	R/P-0	R/P-1	R/P-0
—	—	—	—	—	FOSC2	FOSC1	FOSC0
bit 7					bit 0		

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

bit 5 **Unimplemented and reserved:** Maintain as '1'

bit 4-3 **Unimplemented:** Read as '0'

bit 2-0 **FOSC2:FOSC0:** Oscillator Selection bits

111 = Reserved

110 = HS oscillator with PLL enabled; clock frequency = (4 x Fosc)

101 = EC oscillator w/ OSC2 configured as RA6

100 = EC oscillator w/ OSC2 configured as divide-by-4 clock output

011 = Reserved

010 = HS oscillator

001 = Reserved

000 = Reserved

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

## REGISTER 20-5: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	U-1	R/C-1	R/C-1	R/C-1
—	—	—	—	—	CP2 <sup>(1)</sup>	CP1	CP0
bit 7					bit 0		

- bit 7-4 **Unimplemented:** Read as '0'
- bit 3 **Unimplemented and reserved:** Maintain as '1'
- bit 2 **CP2:** Code Protection bit<sup>(1)</sup>  
 1 = Block 2 (004000-005FFFh) not code protected  
 0 = Block 2 (004000-005FFFh) code protected
- bit 1 **CP1:** Code Protection bit  
 1 = Block 1 (002000-003FFFh) not code protected  
 0 = Block 1 (002000-003FFFh) code protected
- bit 0 **CP0:** Code Protection bit  
 1 = Block 0 (000200-001FFFh) not code protected  
 0 = Block 0 (000200-001FFFh) code protected

**Note 1:** Unimplemented in PIC18FX439 devices; maintain this bit set.

### Legend:

R = Readable bit      C = Clearable bit      U = Unimplemented bit, read as '0'  
 - n = Value when device is unprogrammed      u = Unchanged from programmed state

## REGISTER 20-6: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD	CPB	—	—	—	—	—	—
bit 7				bit 0			

- bit 7 **CPD:** Data EEPROM Code Protection bit  
 1 = Data EEPROM not code protected  
 0 = Data EEPROM code protected
- bit 6 **CPB:** Boot Block Code Protection bit  
 1 = Boot block (000000-0001FFFh) not code protected  
 0 = Boot block (000000-0001FFFh) code protected
- bit 5-0 **Unimplemented:** Read as '0'

### Legend:

R = Readable bit      C = Clearable bit      U = Unimplemented bit, read as '0'  
 - n = Value when device is unprogrammed      u = Unchanged from programmed state

# PIC18FXX39

## REGISTER 20-11: DEVID1: DEVICE ID REGISTER 1 FOR PIC18FXX39 (BYTE ADDRESS 3FFFEh)

R	R	R	R	R	R	R	R	
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	
bit 7								bit 0

bit 7-5 **DEV2:DEV0:** Device ID bits

000 = PIC18F2539

001 = PIC18F4539

100 = PIC18F2439

101 = PIC18F4439

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

### Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

## REGISTER 20-12: DEVID2: DEVICE ID REGISTER 2 FOR PIC18FXX39 (BYTE ADDRESS 3FFFFh)

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7			bit 0				

bit 7-0 **DEV10:DEV3:** Device ID bits

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

### Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state



# PIC18FXX39

**TABLE 21-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
a	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
d	Destination select bit d = 0: store result in WREG, d = 1: store result in file register f.
dest	Destination, either the WREG register or the specified register file location.
f	8-bit Register file address (0x00 to 0xFF).
fs	12-bit Register file address (0x000 to 0xFFF). This is the source address.
fd	12-bit Register file address (0x000 to 0xFFF). This is the destination address.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the Table Read and Table Write instructions. Only used with Table Read and Table Write instructions:
*	No Change to register (such as TBLPTR with Table Reads and Writes).
*+	Post-Increment register (such as TBLPTR with Table Reads and Writes).
*-	Post-Decrement register (such as TBLPTR with Table Reads and Writes).
+*	Pre-Increment register (such as TBLPTR with Table Reads and Writes).
n	The relative address (2's complement number) for relative branch instructions, or the direct address for Call/Branch and Return instructions.
PRODH	Product of Multiply high byte.
PRODL	Product of Multiply low byte.
s	Fast Call/Return mode select bit s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
u	Unused or Unchanged.
WREG	Working register (accumulator).
x	Don't care (0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
TBLPTR	21-bit Table Pointer (points to a Program Memory location).
TABLAT	8-bit Table Latch.
TOS	Top-of-Stack.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
GIE	Global Interrupt Enable bit.
WDT	Watchdog Timer.
T $\overline{O}$	Time-out bit.
P $\overline{D}$	Power-down bit.
C, DC, Z, OV, N	ALU status bits Carry, Digit Carry, Zero, Overflow, Negative.
[ ]	Optional.
( )	Contents.
→	Assigned to.
< >	Register bit field.
∈	In the set of.
<i>italics</i>	User defined term (font is courier).

## CLRF Clear f

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

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

Operation:  $000h \rightarrow f$   
 $1 \rightarrow Z$

Status Affected: Z

Encoding: 

0110	101a	ffff	ffff
------	------	------	------

Description: Clears the contents of the specified register. 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 register 'f'

**Example:** CLRF FLAG\_REG, 1

Before Instruction

FLAG\_REG = 0x5A

After Instruction

FLAG\_REG = 0x00

## CLRWDT Clear Watchdog Timer

Syntax: [ *label* ] CLRWDT

Operands: None

Operation:  $000h \rightarrow WDT$ ,  
 $000h \rightarrow WDT$  postscaler,  
 $1 \rightarrow \overline{TO}$ ,  
 $1 \rightarrow \overline{PD}$

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

Encoding: 

0000	0000	0000	0100
------	------	------	------

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits  $\overline{TO}$  and  $\overline{PD}$  are set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

**Example:** CLRWDT

Before Instruction

WDT Counter = ?

After Instruction

WDT Counter = 0x00

WDT Postscaler = 0

$\overline{TO}$  = 1

$\overline{PD}$  = 1

# PIC18FXX39

FIGURE 23-1: PIC18FXX39 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

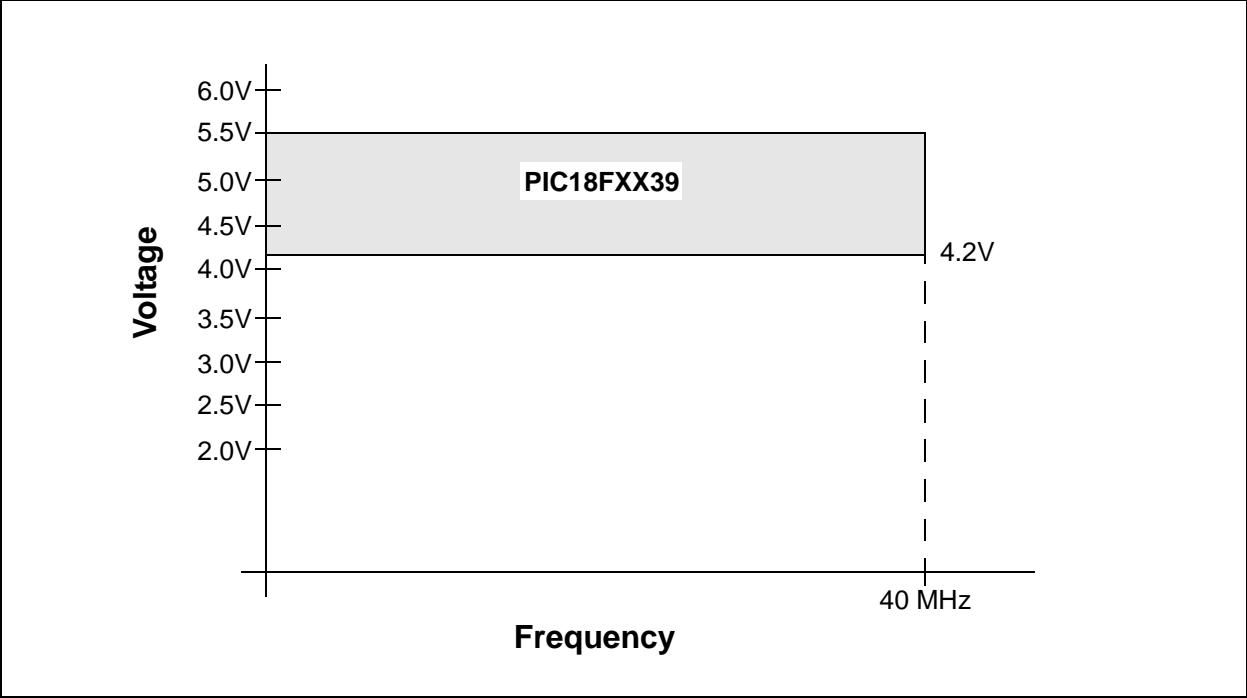
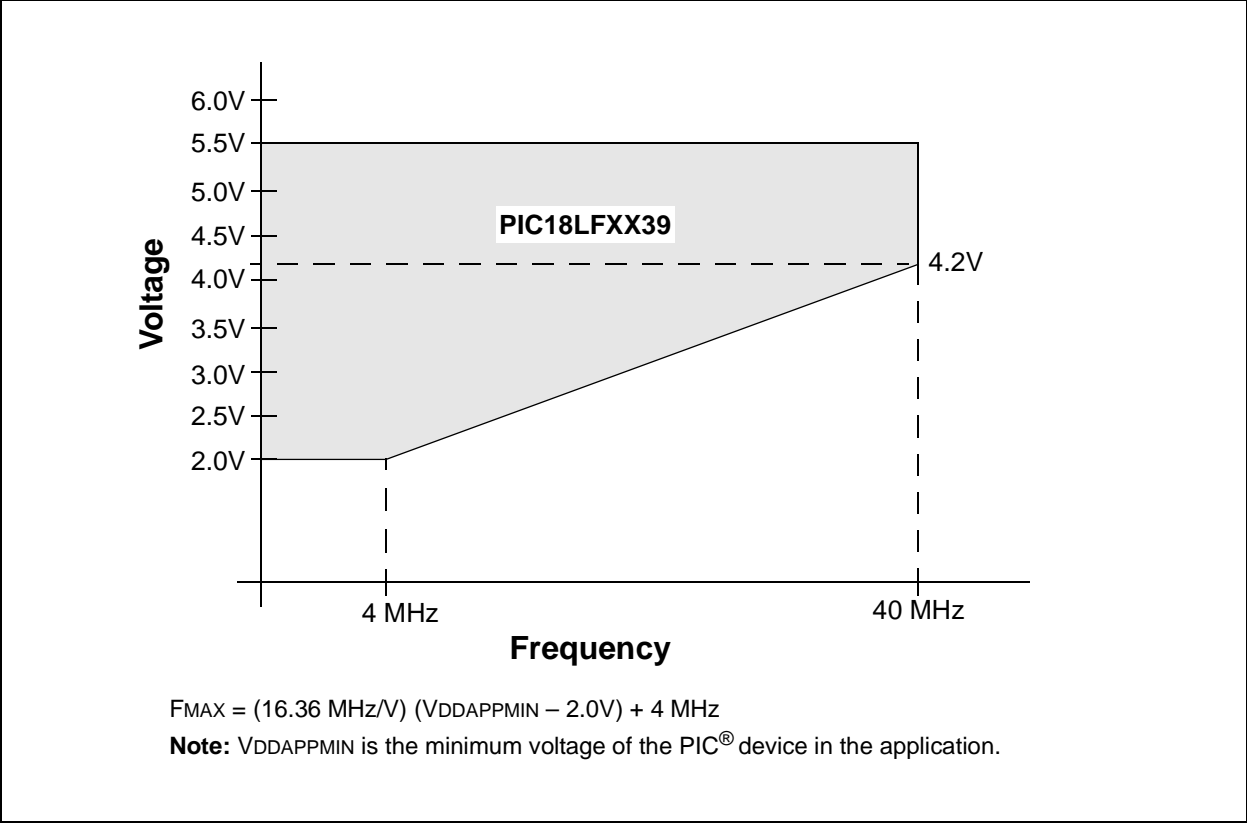
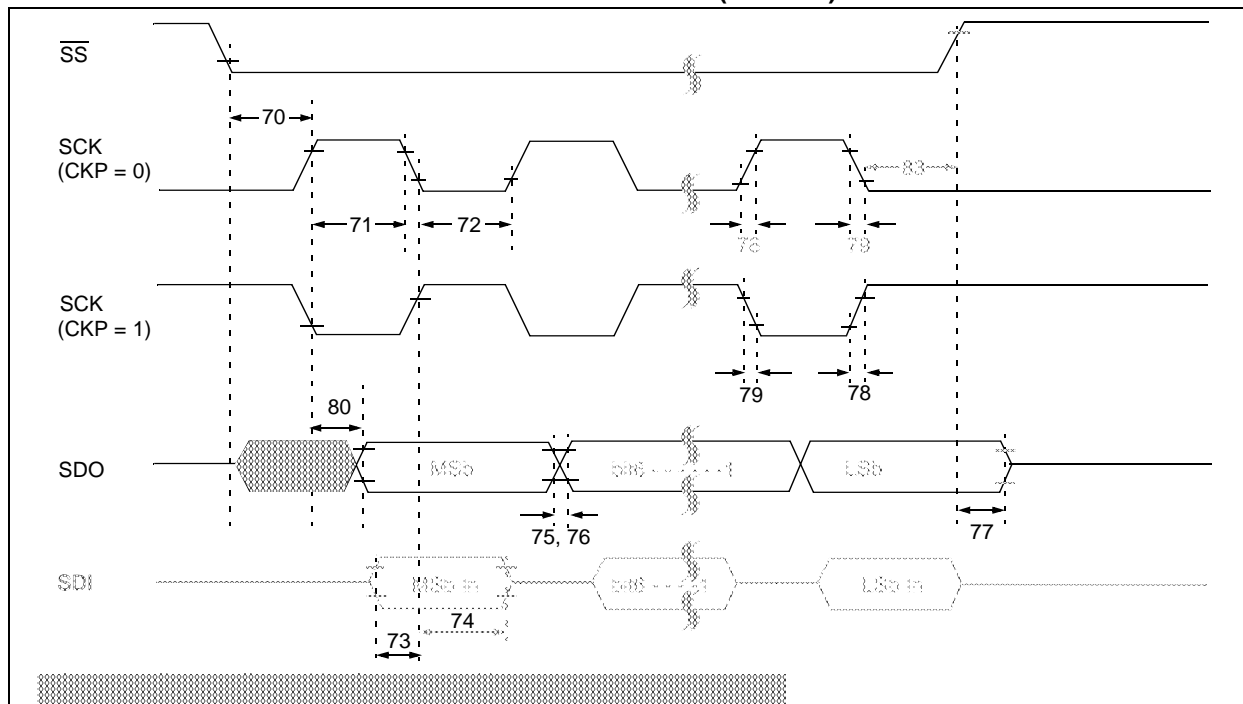


FIGURE 23-2: PIC18LFXX39 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



# PIC18FXX39

**FIGURE 23-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)**



**TABLE 23-13: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING (CKE = 0))**

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ input		T <sub>CY</sub>	—	ns	
71	TscH	SCK input high time (Slave mode)	Continuous	1.25 T <sub>CY</sub> + 30	—	ns	
71A			Single Byte	40	—	ns	(Note 1)
72	TscL	SCK input low time (Slave mode)	Continuous	1.25 T <sub>CY</sub> + 30	—	ns	
72A			Single Byte	40	—	ns	(Note 1)
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge		100	—	ns	
73A	Tb2B	Last clock edge of Byte 1 to the first clock edge of Byte 2		1.5 T <sub>CY</sub> + 40	—	ns	(Note 2)
74	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge		100	—	ns	
75	TdoR	SDO data output rise time	PIC18FXXXX	—	25	ns	
			PIC18LFXXXX	—	60	ns	V <sub>DD</sub> = 2V
76	TdoF	SDO data output fall time	PIC18FXXXX	—	25	ns	
			PIC18LFXXXX	—	60	ns	V <sub>DD</sub> = 2V
77	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance		10	50	ns	
78	TscR	SCK output rise time (Master mode)	PIC18FXXXX	—	25	ns	
			PIC18LFXXXX	—	60	ns	V <sub>DD</sub> = 2V
79	TscF	SCK output fall time (Master mode)	PIC18FXXXX	—	25	ns	
			PIC18LFXXXX	—	60	ns	V <sub>DD</sub> = 2V
80	Tsch2doV, TscL2doV	SDO data output valid after SCK edge		PIC18FXXXX	—	50	ns
				PIC18LFXXXX	—	150	ns
83	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge		1.5 T <sub>CY</sub> + 40	—	ns	

**Note 1:** Requires the use of Parameter # 73A.

**Note 2:** Only if Parameter # 71A and # 72A are used.

Multi-Master Mode .....	159	MOVFF .....	236
Operation .....	138	MOVLB .....	236
Read/Write Bit Information (R/W Bit) .....	138, 139	MOVLW .....	237
Registers .....	134	MOVWF .....	237
Serial Clock (RC3/SCK/SCL) .....	139	MULLW .....	238
Slave Mode .....	138	MULWF .....	238
Addressing .....	138	NEGF .....	239
Reception .....	139	NOP .....	239
Transmission .....	139	POP .....	240
SLEEP Operation .....	159	PUSH .....	240
STOP Condition Timing .....	158	RCALL .....	241
ICEPIC In-Circuit Emulator .....	254	RESET .....	241
ID Locations .....	195, 210	RETFIE .....	242
INCF .....	232	RETLW .....	242
INCFSZ .....	233	RETURN .....	243
In-Circuit Debugger .....	210	RLCF .....	243
In-Circuit Serial Programming (ICSP) .....	195, 210	RLNCF .....	244
Indirect Addressing .....	48	RRCF .....	244
INDF and FSR Registers .....	47	RRNCF .....	245
Operation .....	47	SETF .....	245
Indirect Addressing Operation .....	48	SLEEP .....	246
Indirect File Operand .....	39	SUBFWB .....	246
INFSNZ .....	233	SUBLW .....	247
Instruction Cycle .....	36	SUBWF .....	247
Instruction Flow/Pipelining .....	37	SUBWFB .....	248
Instruction Format .....	213	SWAPF .....	248
Instruction Set .....	211	TBLRD .....	249
ADDLW .....	217	TBLWT .....	250
ADDWF .....	217	TSTFSZ .....	251
ADDWFC .....	218	XORLW .....	251
ANDLW .....	218	XORWF .....	252
ANDWF .....	219	Summary Table .....	214
BC .....	219	Instructions in Program Memory .....	37
BCF .....	220	Two-Word Instructions .....	38
BN .....	220	INT Interrupt (RB0/INT). See Interrupt Sources	
BNC .....	221	INTCON Register	
BNN .....	221	RBIF Bit .....	86
BNOV .....	222	INTCON Registers .....	71–73
BNZ .....	222	Inter-Integrated Circuit. See I <sup>2</sup> C	
BOV .....	225	Interrupt Sources .....	195
BRA .....	223	A/D Conversion Complete .....	184
BSF .....	223	INT0 .....	81
BTFSC .....	224	Interrupt-on-Change (RB7:RB4) .....	86
BTFSS .....	224	PORTB, Interrupt-on-Change .....	81
BTG .....	225	RB0/INT Pin, External .....	81
BZ .....	226	TMR0 .....	81
CALL .....	226	TMR0 Overflow .....	101
CLRF .....	227	TMR1 Overflow .....	103, 105
CLRWDT .....	227	TMR2 to PR2 Match (PWM) .....	123
COMF .....	228	TMR3 Overflow .....	109, 111
CPFSEQ .....	228	USART Receive/Transmit Complete .....	165
CPFSGT .....	229	Interrupts .....	69
CPFSLT .....	229	Logic .....	70
DAW .....	230	Interrupts, Flag Bits	
DCFSNZ .....	231	A/D Converter Flag (ADIF Bit) .....	183
DECF .....	230	Interrupt-on-Change (RB7:RB4) Flag	
DECFSZ .....	231	(RBIF Bit) .....	86
GOTO .....	232	IORLW .....	234
INCF .....	232	IORWF .....	234
INCFSZ .....	233	IPR Registers .....	78–79
INFSNZ .....	233		
IORLW .....	234	<b>K</b>	
IORWF .....	234	KEELOQ Evaluation and Programming Tools .....	256
LFSR .....	235		
MOVF .....	235		