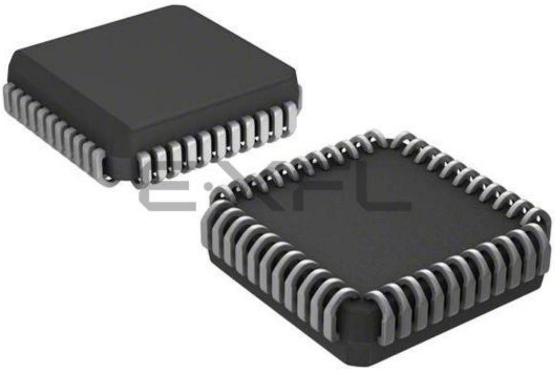


Welcome to E-XFL.COM

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	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	33
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f458t-i-l

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
							PORTD is a bidirectional I/O port. These pins have TTL input buffers when external memory is enabled.
RD0/PSP0/C1IN+	—	19	38	21			
RD0					I/O	ST	Digital I/O.
PSP0					I/O	TTL	Parallel Slave Port data.
C1IN+					I	Analog	Comparator 1 input.
RD1/PSP1/C1IN-	—	20	39	22			
RD1					I/O	ST	Digital I/O.
PSP1					I/O	TTL	Parallel Slave Port data.
C1IN-					I	Analog	Comparator 1 input.
RD2/PSP2/C2IN+	—	21	40	23			
RD2					I/O	ST	Digital I/O.
PSP2					I/O	TTL	Parallel Slave Port data.
C2IN+					I	Analog	Comparator 2 input.
RD3/PSP3/C2IN-	—	22	41	24			
RD3					I/O	ST	Digital I/O.
PSP3					I/O	TTL	Parallel Slave Port data.
C2IN-					I	Analog	Comparator 2 input.
RD4/PSP4/ECCP1/P1A	—	27	2	30			
RD4					I/O	ST	Digital I/O.
PSP4					I/O	TTL	Parallel Slave Port data.
ECCP1					I/O	ST	ECCP1 capture/compare.
P1A					O	—	ECCP1 PWM output A.
RD5/PSP5/P1B	—	28	3	31			
RD5					I/O	ST	Digital I/O.
PSP5					I/O	TTL	Parallel Slave Port data.
P1B					O	—	ECCP1 PWM output B.
RD6/PSP6/P1C	—	29	4	32			
RD6					I/O	ST	Digital I/O.
PSP6					I/O	TTL	Parallel Slave Port data.
P1C					O	—	ECCP1 PWM output C.
RD7/PSP7/P1D	—	30	5	33			
RD7					I/O	ST	Digital I/O.
PSP7					I/O	TTL	Parallel Slave Port data.
P1D					O	—	ECCP1 PWM output D.

Legend:

TTL	= TTL compatible input	CMOS	= CMOS compatible input or output
ST	= Schmitt Trigger input with CMOS levels	Analog	= Analog input
I	= Input	O	= Output
P	= Power	OD	= Open-Drain (no P diode to V_{DD})

3.7 Time-out Sequence

On power-up, the time-out sequence is as follows: First, PWRT time-out is invoked after the POR time delay has expired, then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 3-3, Figure 3-4, Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately (Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC18FXX8 device operating in parallel.

Table 3-2 shows the Reset conditions for some Special Function Registers, while Table 3-3 shows the Reset conditions for all registers.

TABLE 3-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up[(2)]		Brown-out[(2)]	Wake-up from Sleep or Oscillator Switch
	$\overline{\text{PWRTEN}}$ = 0	$\overline{\text{PWRTEN}}$ = 1		
HS with PLL enabled[(1)]	72 ms + 1024 TOSC + 2 ms	1024 TOSC + 2 ms	72 ms + 1024 TOSC + 2 ms	1024 TOSC + 2 ms
HS, XT, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
EC	72 ms	—	72 ms	—
External RC	72 ms	—	72 ms	—

Note 1: 2 ms = Nominal time required for the 4x PLL to lock.

2: 72 ms is the nominal Power-up Timer delay.

REGISTER 3-1: RCON REGISTER BITS AND POSITIONS

R/W-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1
IPEN	—	—	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

TABLE 3-2: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter	RCON Register	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	STKFUL	STKUNF
Power-on Reset	0000h	0--1 110q	1	1	1	0	0	u	u
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	0--0 011q	u	u	u	u	u	u	u
Software Reset during normal operation	0000h	0--0 011q	0	u	u	u	u	u	u
Stack Full Reset during normal operation	0000h	0--0 011q	u	u	u	1	1	u	1
Stack Underflow Reset during normal operation	0000h	0--0 011q	u	u	u	1	1	1	u
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	0--0 011q	u	1	0	u	u	u	u
WDT Reset	0000h	0--0 011q	u	0	1	u	u	u	u
WDT Wake-up	PC + 2	0--1 101q	u	0	0	u	u	u	u
Brown-out Reset	0000h	0--1 110q	1	1	1	u	0	u	u
Interrupt wake-up from Sleep	PC + 2[(1)]	0--1 101q	u	1	0	u	u	u	u

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bits are set, the PC is loaded with the interrupt vector (000008h or 000018h).

4.10 Access Bank

The Access Bank is an architectural enhancement that is very useful for C compiler code optimization. The techniques used by the C compiler are also useful for programs written in assembly.

This data memory region can be used for:

- Intermediate computational values
- Local variables of subroutines
- Faster context saving/switching of variables
- Common variables
- Faster evaluation/control of SFRs (no banking)

The Access Bank is comprised of the upper 160 bytes in Bank 15 (SFRs) and the lower 96 bytes in Bank 0. These two sections will be referred to as Access Bank High and Access Bank Low, respectively. Figure 4-6 indicates the Access Bank areas.

A bit in the instruction word specifies if the operation is to occur in the bank specified by the BSR register or in the Access Bank.

When forced in the Access Bank (a = `0`), the last address in Access Bank Low is followed by the first address in Access Bank High. Access Bank High maps most of the Special Function Registers so that these registers can be accessed without any software overhead.

4.11 Bank Select Register (BSR)

The need for a large general purpose memory space dictates a RAM banking scheme. The data memory is partitioned into sixteen banks. When using direct addressing, the BSR should be configured for the desired bank.

BSR<3:0> holds the upper 4 bits of the 12-bit RAM address. The BSR<7:4> bits will always read '`0`'s and writes will have no effect.

A `MOVLB` instruction has been provided in the instruction set to assist in selecting banks.

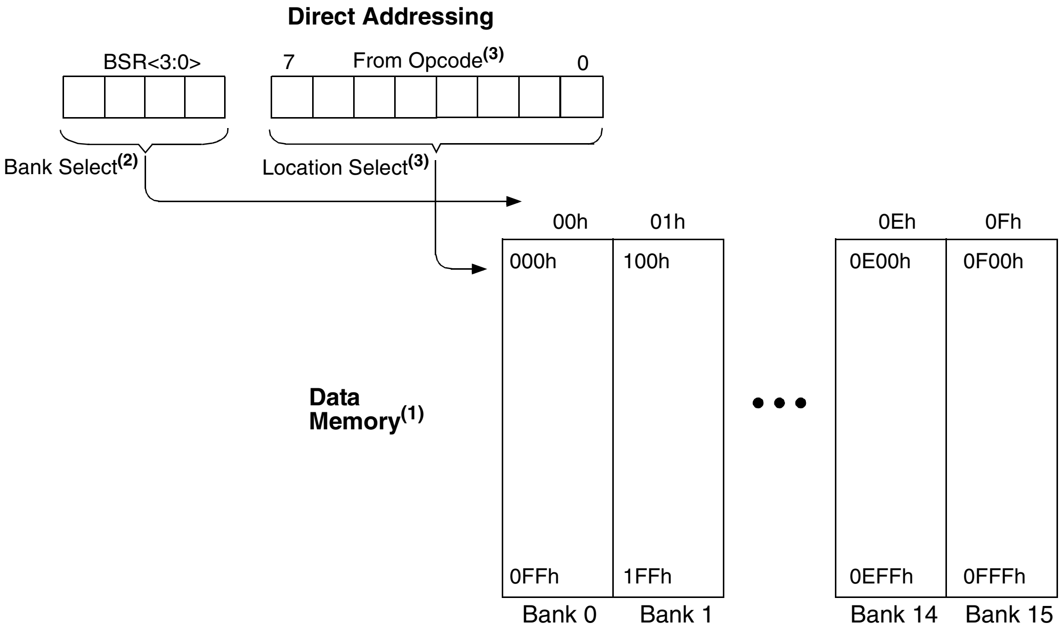
If the currently selected bank is not implemented, any read will return all '`0`'s and all writes are ignored. The Status register bits will be set/cleared as appropriate for the instruction performed.

Each Bank extends up to FFh (256 bytes). All data memory is implemented as static RAM.

A `MOVFF` instruction ignores the BSR since the 12-bit addresses are embedded into the instruction word.

Section 4.12 "Indirect Addressing, INDF and FSR Registers" provides a description of indirect addressing, which allows linear addressing of the entire RAM space.

FIGURE 4-7: DIRECT ADDRESSING



Note 1: For register file map detail, see Table 4-1.

2: The access bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

3: The `MOVFF` instruction embeds the entire 12-bit address in the instruction.

REGISTER 5-1: EECON1: EEPROM CONTROL REGISTER 1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	$\overline{\text{WR}}$	$\overline{\text{RD}}$
bit 7							bit 0

bit 7 **EEPGD:** Flash Program or Data EEPROM Memory Select bit

1 = Access program Flash memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EE or Configuration Select bit

1 = Access Configuration registers
0 = Access program Flash or data EEPROM memory

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **FREE:** Flash Row Erase Enable bit

1 = Erase the program memory row addressed by TBLPTR on the next $\overline{\text{WR}}$ command (reset by hardware)
0 = Perform write only

bit 3 **WRERR:** Write Error Flag bit

1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or any WDT Reset during self-timed programming in normal operation)
0 = The write operation completed

Note: When a WRERR occurs, the EEPGD or FREE bits are not cleared. This allows tracing of the error condition.

bit 2 **WREN:** Write Enable bit

1 = Allows write cycles
0 = Inhibits write to the EEPROM or Flash memory

bit 1 **$\overline{\text{WR}}$:** Write Control bit

1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once write is complete. The $\overline{\text{WR}}$ bit can only be set (not cleared) in software.)
0 = Write cycle is complete

bit 0 **$\overline{\text{RD}}$:** Read Control bit

1 = Initiates an EEPROM read (Read takes one cycle. $\overline{\text{RD}}$ is cleared in hardware. The $\overline{\text{RD}}$ bit can only be set (not cleared) in software. $\overline{\text{RD}}$ bit cannot be set when EEPGD = 1.)
0 = Does not initiate an EEPROM read

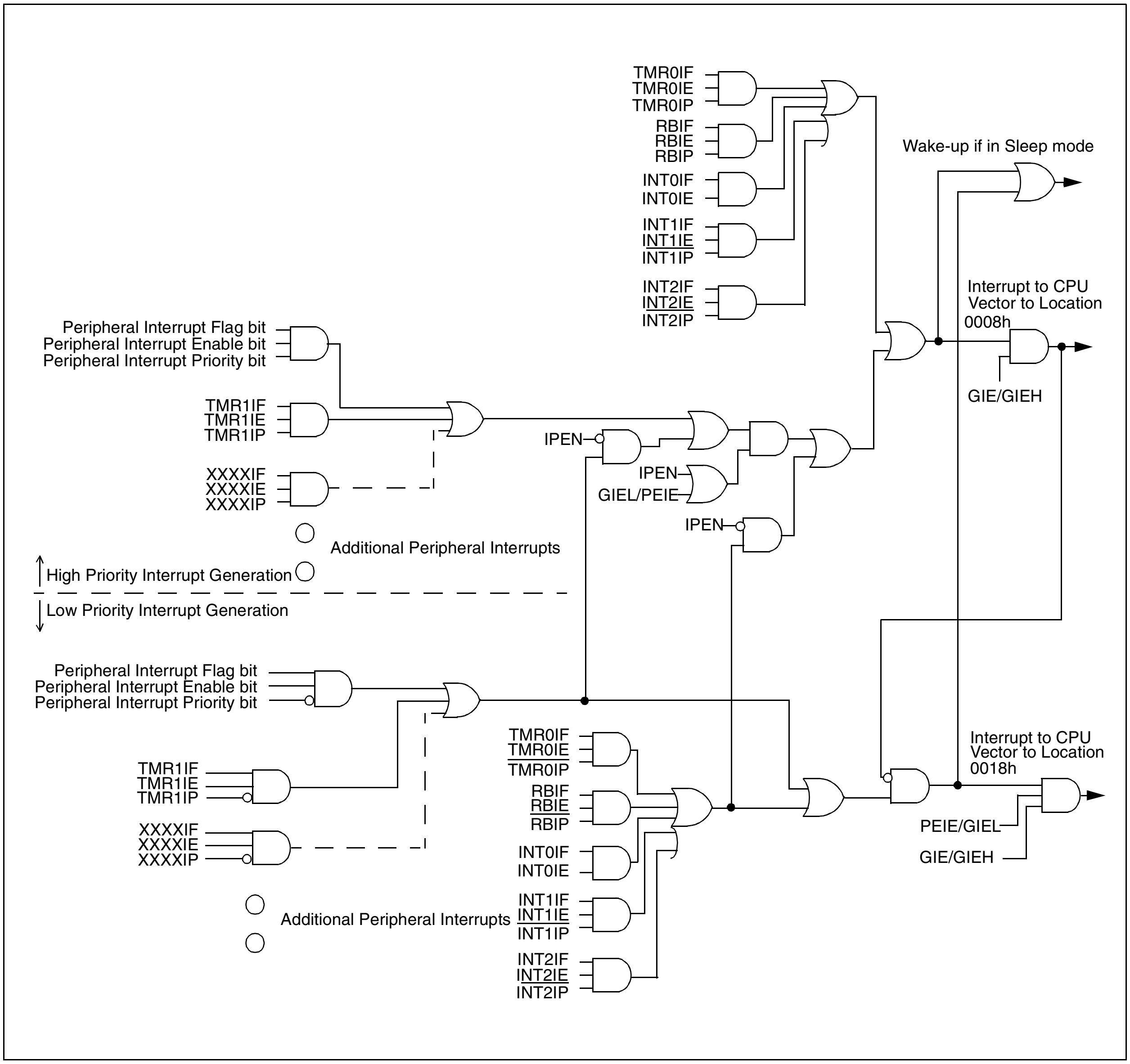
Legend:

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

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

```
               MOVLW  D'64                      ; number of bytes in erase block
               MOVWF  COUNTER
               MOVLW  high (BUFFER_ADDR)        ; point to buffer
               MOVWF  FSR0H
               MOVLW  low (BUFFER_ADDR)
               MOVWF  FSR0L
               MOVLW  upper (CODE_ADDR)         ; Load TBLPTR with the base
               MOVWF  TBLPTRU                   ; address of the memory block
               MOVLW  high (CODE_ADDR)
               MOVWF  TBLPTRH
               MOVLW  low (CODE_ADDR)
               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  upper (CODE_ADDR)         ; load TBLPTR with the base
               MOVWF  TBLPTRU                   ; address of the memory block
               MOVLW  high (CODE_ADDR)
               MOVWF  TBLPTRH
               MOVLW  low (CODE_ADDR)
               MOVWF  TBLPTRL
               BSF    EECON1, EEPGD             ; point to FLASH program memory
               BCF    EECON1, CFGS              ; 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
Required       MOVWF  EECON2                    ; write 55H
Sequence       MOVLW  0AAh
               MOVWF  EECON2                    ; write AAH
               BSF    EECON1, WR                ; start erase (CPU stall)
               NOP
               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  high (BUFFER_ADDR)        ; point to buffer
               MOVWF  FSR0H
               MOVLW  low (BUFFER_ADDR)
               MOVWF  FSR0L
PROGRAM_LOOP
               MOVLW  8                         ; number of bytes in holding register
               MOVWF  COUNTER
WRITE_WORD_TO_HREGS
               MOVFW  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
```

FIGURE 8-1: INTERRUPT LOGIC



8.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits. Because of the number of interrupts to be controlled, PIC18FXX8 devices have three INTCON registers. They are detailed in Register 8-1 through Register 8-3.

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 software polling.

REGISTER 8-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

bit 7 **GIE/GIEH:** Global Interrupt Enable bit

When IPEN (RCON<7>) = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts

When IPEN (RCON<7>) = 1:
1 = Enables all high priority interrupts
0 = Disables all priority interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit

When IPEN (RCON<7>) = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts

When IPEN (RCON<7>) = 1:
1 = Enables all low priority peripheral interrupts
0 = Disables all low priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software by reading PORTB)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Note: A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

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

9.5 PORTE, TRISE and LATE Registers

Note:	This port is only available on the PIC18F448 and PIC18F458.

PORTE is a 3-bit wide, bidirectional port. PORTE has three pins (RE0/AN5/$\overline{RD}$, RE1/AN6/$\overline{WR}$/C1OUT and RE2/AN7/$\overline{CS}$/C2OUT) which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

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

The corresponding Data Direction register for the port is TRISE. Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., put the contents of the output latch on the selected pin).

The TRISE register also controls the operation of the Parallel Slave Port through the control bits in the upper half of the register. These are shown in Register 9-1.

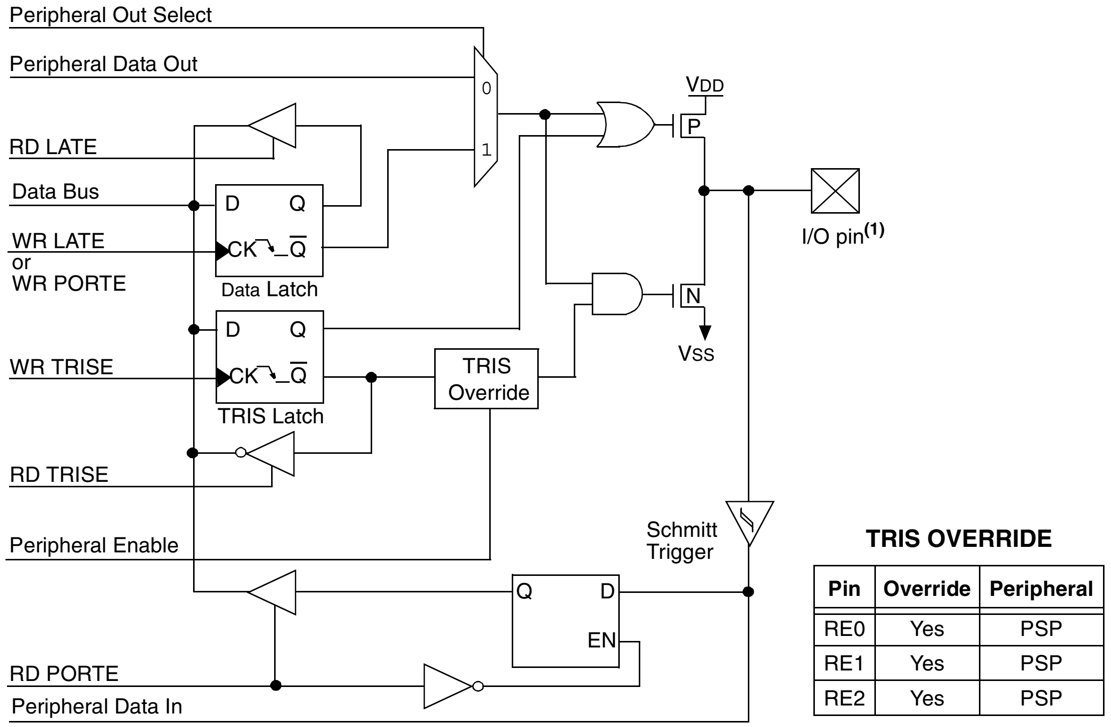
When the Parallel Slave Port is active, the PORTE pins function as its control inputs. For additional details, refer to **Section 10.0 "Parallel Slave Port"**.

PORTE pins are also multiplexed with inputs for the A/D converter and outputs for the analog comparators. When selected as an analog input, these pins will read as '0's. Direction bits TRISE<2:0> control the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

EXAMPLE 9-5: INITIALIZING PORTE

```
CLRF    PORTE          ; Initialize PORTE by
                       ; clearing output
                       ; data latches
CLRF    LATE           ; Alternate method
                       ; to clear output
                       ; data latches
MOVLW   03h            ; Value used to
                       ; initialize data
                       ; direction
MOVWF   TRISE          ; Set RE1:RE0 as inputs
                       ; RE2 as an output
                       ; (RE4=0 - PSPMODE Off)
```

FIGURE 9-10: PORTE BLOCK DIAGRAM



TRIS OVERRIDE

Pin	Override	Peripheral
RE0	Yes	PSP
RE1	Yes	PSP
RE2	Yes	PSP

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

13.0 TIMER2 MODULE

The Timer2 module timer has the following features:

- 8-bit timer (TMR2 register)
- 8-bit period register (PR2)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match of PR2
- SSP module optional use of TMR2 output to generate clock shift

Register 13-1 shows the Timer2 Control register. Timer2 can be shut-off by clearing control bit TMR2ON (T2CON register) to minimize power consumption. Figure 13-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

13.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 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 T2CKPS1:T2CKPS0 (T2CON register). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, PIR registers).

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

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

Note: Timer2 is disabled on POR.

REGISTER 13-1: T2CON: TIMER2 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
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 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

15.1 CCP1 Module

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

Table 15-1 shows the timer resources of the CCP module modes.

TABLE 15-1: CCP1 MODE – TIMER RESOURCE

CCP1 Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

15.2 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 or TMR3 register when an event occurs on pin RC2/CCP1. An event is defined as:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR registers), is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

15.2.1 CCP PIN CONFIGURATION

In Capture mode, the RC2/CCP1 pin should be configured as an input by setting the TRISC<2> bit.

Note: If the RC2/CCP1 is configured as an output, a write to the port can cause a capture condition.

15.2.2 TIMER1/TIMER3 MODE SELECTION

The timers used with the capture feature (either Timer1 and/or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer used with each CCP module is selected in the T3CON register.

TABLE 15-2: INTERACTION OF CCP1 AND ECCP1 MODULES

CCP1 Mode	ECCP1 Mode	Interaction
Capture	Capture	TMR1 or TMR3 time base. Time base can be different for each CCP.
Capture	Compare	The compare could be configured for the special event trigger which clears either TMR1 or TMR3, depending upon which time base is used.
Compare	Compare	The compare(s) could be configured for the special event trigger which clears TMR1 or TMR3, depending upon which time base is used.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt).
PWM	Capture	None.
PWM	Compare	None.

17.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep. Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit (SSPCON1<4>).

17.3.7 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

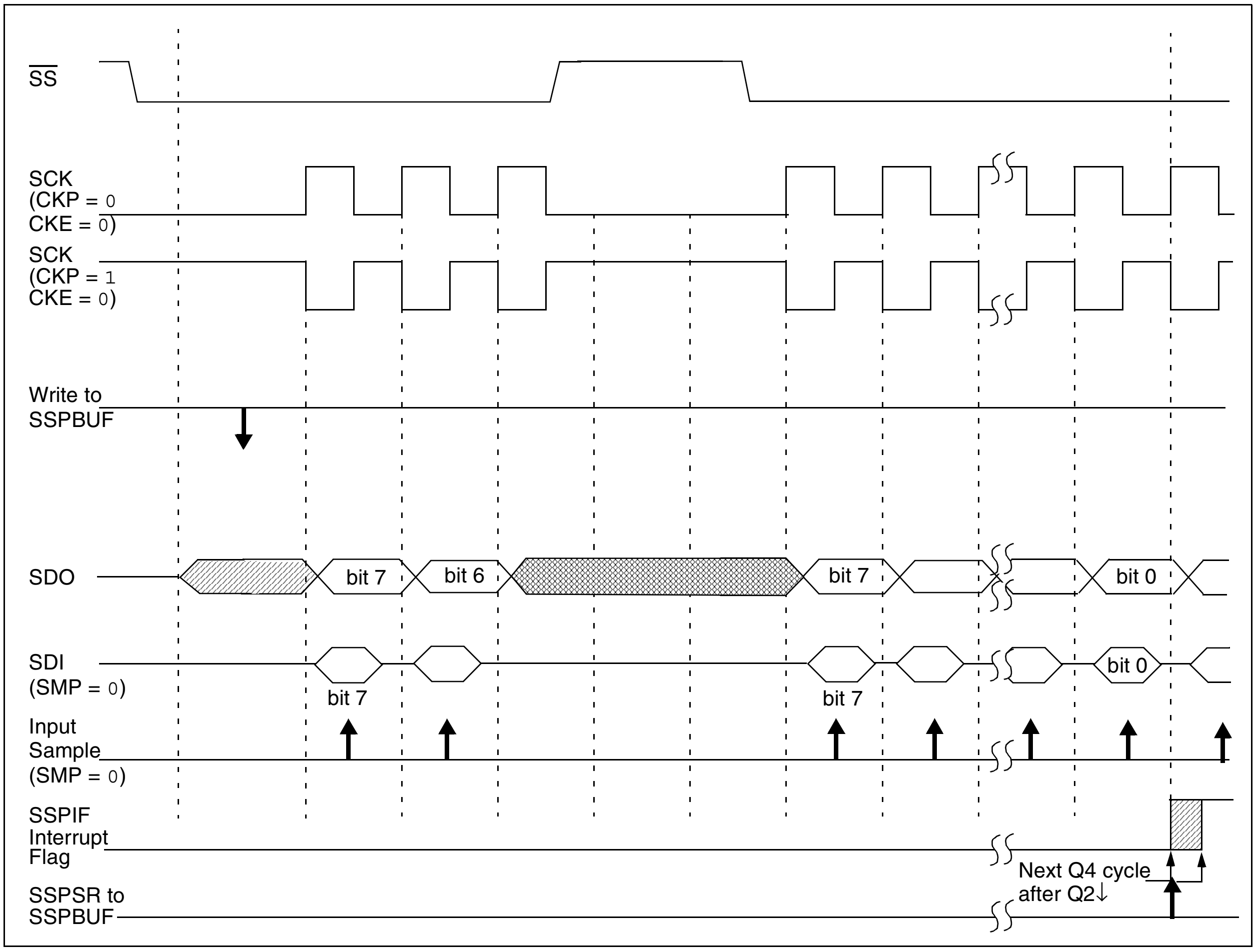
Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

2: If the SPI is used in Slave mode with CKE set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to ‘0’. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 17-4: SLAVE SYNCHRONIZATION WAVEFORM



17.4.10 I^{2}C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter #106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter #107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{\text{ACK}}$ bit during the ninth bit time, if an address match occurred, or if data was received properly. The status of $\overline{\text{ACK}}$ is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 17-21).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{\text{W}}$ bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF bit is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

17.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

17.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

17.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{\text{ACK}}$ = 0) and is set when the slave does not Acknowledge ($\overline{\text{ACK}}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

17.4.11 I^{2}C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

Note:	The RCEN bit should be set after the $\overline{\text{ACK}}$ sequence is complete or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

17.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

17.4.11.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

17.4.11.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 17-27: BUS COLLISION DURING START CONDITION (SCL = 0)

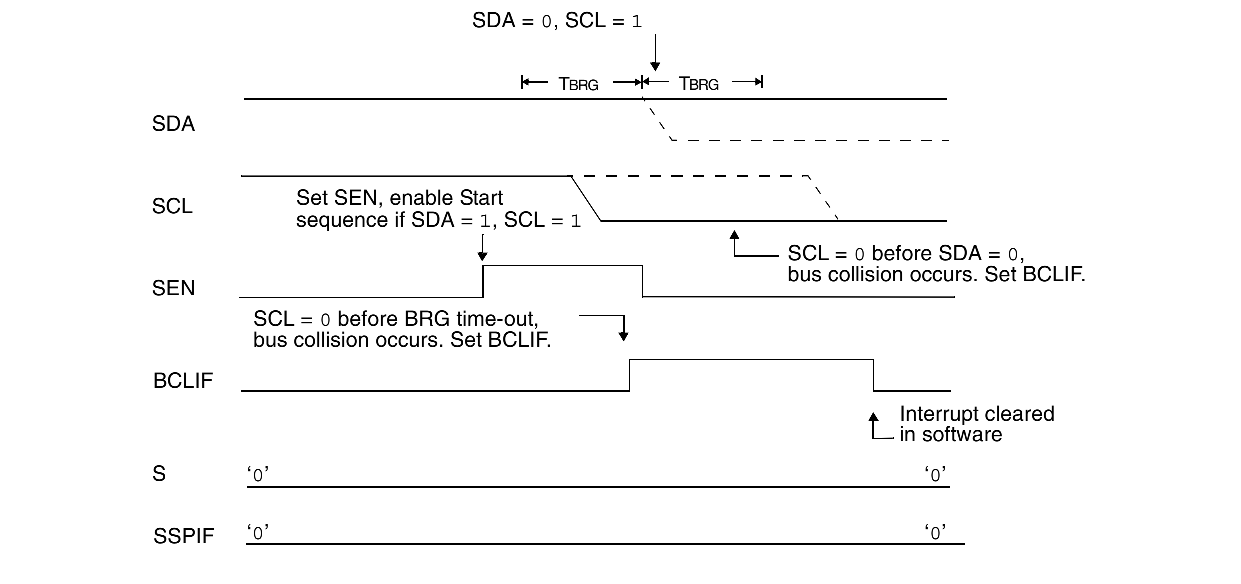
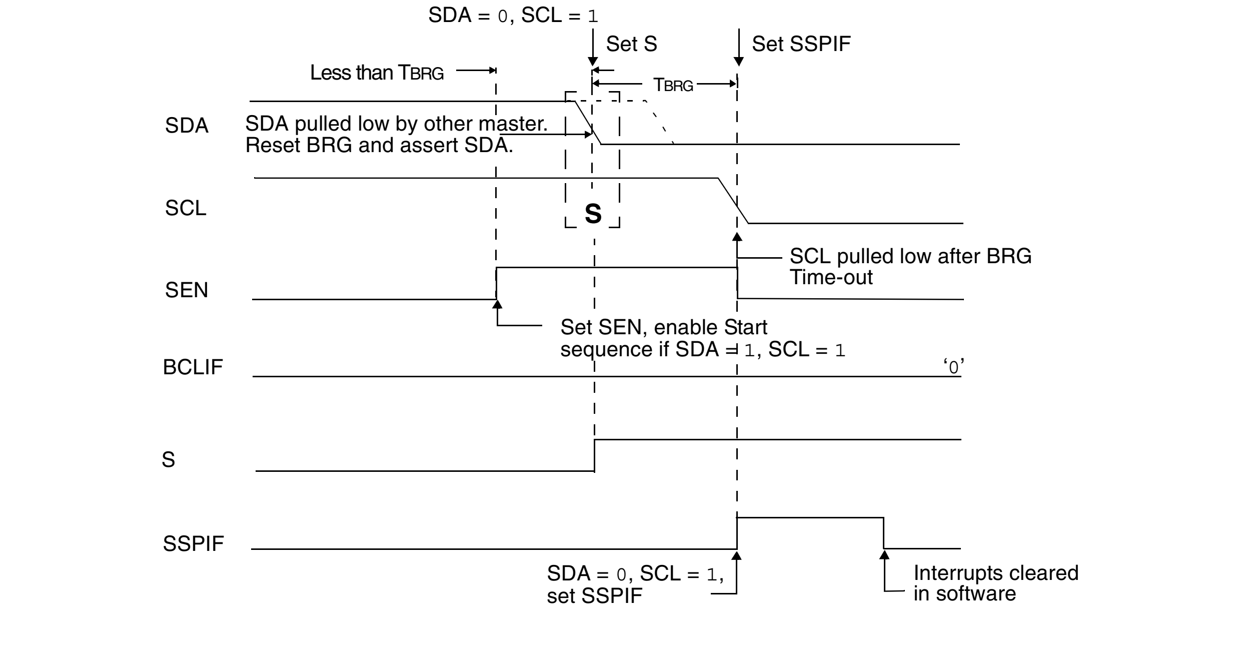

SDA = 0, SCL = 1
TBRG
TBRG
SDA
SCL
Set SEN, enable Start sequence if SDA = 1, SCL = 1
SCL = 0 before SDA = 0, bus collision occurs. Set BCLIF.
SEN
SCL = 0 before BRG time-out, bus collision occurs. Set BCLIF.
BCLIF
Interrupt cleared in software
S
'0'
'0'
SSPIF
'0'
'0'

FIGURE 17-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION


SDA = 0, SCL = 1
Set S
Set SSPIF
Less than TBRG
TBRG
SDA
SDA pulled low by other master. Reset BRG and assert SDA.
SCL
S
SCL pulled low after BRG Time-out
SEN
Set SEN, enable Start sequence if SDA = 1, SCL = 1
BCLIF
'0'
S
SSPIF
SDA = 0, SCL = 1, set SSPIF
Interrupts cleared in software

TBLRD Table Read

Syntax: [*label*] TBLRD (*; *+; *-; +*)

Operands: None

Operation:
if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change;
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR;
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR;
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT

Status Affected: None

Encoding:

`0000`	`0000`	`0000`	`10nn`
			`nn=0 *`
			`=1 *+`
			`=2 *-`
			`=3 +*`

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR[0] = `0`:	Least Significant Byte of Program Memory Word
TBLPTR[0] = `1`:	Most Significant Byte of Program Memory Word

The `TBLRD` instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read Program Memory)	No operation	No operation (Write TABLAT)

Example 1: `TBLRD *+ ;`

Before Instruction

TABLAT	=	0x55
TBLPTR	=	0x00A356
MEMORY(0x00A356)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x00A357

Example 2: `TBLRD +* ;`

Before Instruction

TABLAT	=	0xAA
TBLPTR	=	0x01A357
MEMORY(0x01A357)	=	0x12
MEMORY(0x01A358)	=	0x34

After Instruction

TABLAT	=	0x34
TBLPTR	=	0x01A358

TBLWT Table Write

Syntax: [*label*] TBLWT (*; *+; *-; +*)

Operands: None

Operation:

```
if TBLWT*,
(TABLAT) → Holding Register;
TBLPTR – No Change;
if TBLWT*+,
(TABLAT) → Holding Register;
(TBLPTR) + 1 → TBLPTR;
if TBLWT*-,
(TABLAT) → Holding Register;
(TBLPTR) – 1 → TBLPTR;
if TBLWT+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register;
```

Status Affected: None

Encoding:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*

Description: This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 “Flash Program Memory”** for additional details.)

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = `0`:	Least Significant Byte of Program Memory Word
TBLPTR[0] = `1`:	Most Significant Byte of Program Memory Word

The `TBLWT` instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

TBLWT Table Write (Continued)

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

Example 1: `TBLWT *+;`

Before Instruction

TABLAT	=	0x55
TBLPTR	=	0x00A356
HOLDING REGISTER (0x00A356)	=	0xFF

After Instructions (table write completion)

TABLAT	=	0x55
TBLPTR	=	0x00A357
HOLDING REGISTER (0x00A356)	=	0x55

Example 2: `TBLWT +*;`

Before Instruction

TABLAT	=	0x34
TBLPTR	=	0x01389A
HOLDING REGISTER (0x01389A)	=	0xFF
HOLDING REGISTER (0x01389B)	=	0xFF

After Instruction (table write completion)

TABLAT	=	0x34
TBLPTR	=	0x01389B
HOLDING REGISTER (0x01389A)	=	0xFF
HOLDING REGISTER (0x01389B)	=	0x34

27.1 DC Characteristics (Continued)

PIC18LFXX8 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial
PIC18FXX8 (Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended

Param No.	Symbol	Characteristic/ Device	Min	Typ	Max	Units	Conditions
	ΔI_{WDT}	**Module Differential Current**					
D022		**Watchdog Timer** PIC18LFXX8	— — —	0.75 0.8 7	1.5 8 25	µA µA µA	VDD = 2.5V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C
D022		**Watchdog Timer** PIC18FXX8	— — —	7 7 7	25 25 45	µA µA µA	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C
D022A	ΔI_{BOR}	**Brown-out Reset**(5) PIC18LFXX8	— — —	38 42 49	50 55 65	µA µA µA	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C
D022A		**Brown-out Reset**(5) PIC18FXX8	— — —	46 49 50	65 65 75	µA µA µA	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C
D022B	ΔI_{LVD}	**Low-Voltage Detect**(5) PIC18LFXX8	— — —	36 40 47	50 55 65	µA µA µA	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C
D022B		**Low-Voltage Detect**(5) PIC18FXX8	— — —	44 47 47	65 65 75	µA µA µA	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C
D025	ΔI_{TMR1}	**Timer1 Oscillator** PIC18LFXX8	— — —	6.2 6.2 7.5	40 45 55	µA µA µA	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C
D025		**Timer1 Oscillator** PIC18FXX8	— — —	7.5 7.5 7.5	55 55 65	µA µA µA	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C

Legend: Rows are shaded for improved readability.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD and VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, ...).

4: For RC oscillator configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2 REXT (mA) with REXT in kOhm.

5: The LVD and BOR modules share a large portion of circuitry. The ΔI_{BOR} and ΔI_{LVD} currents are not additive. Once one of these modules is enabled, the other may also be enabled without further penalty.

27.3.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 27-6: EXTERNAL CLOCK TIMING

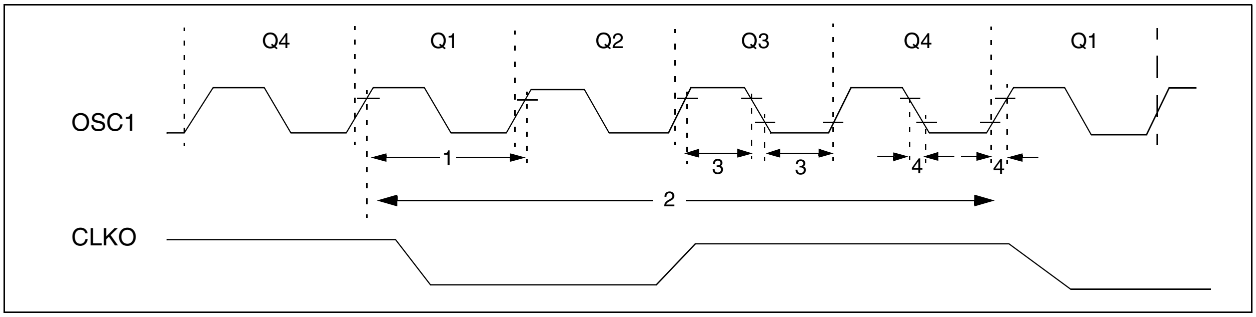
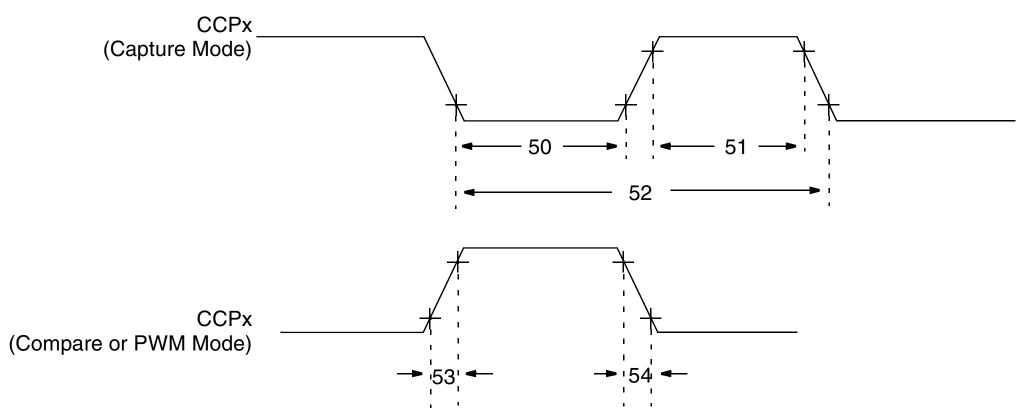


TABLE 27-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	FOSC	External CLKI Frequency(1)	DC	40	MHz	EC, ECIO oscillator, -40°C to +85°C
		Oscillator Frequency(1)	DC	25	MHz	EC, ECIO oscillator, +85°C to +125°C
			DC	4	MHz	RC oscillator
			0.1	4	MHz	XT oscillator
			4	25	MHz	HS oscillator, -40°C to +85°C
			4	25	MHz	HS oscillator, +85°C to +125°C
			4	10	MHz	HS + PLL oscillator, -40°C to +85°C
			4	6.25	MHz	HS + PLL oscillator, +85°C to +125°C
			DC	200	kHz	LP oscillator
1	TOSC	External CLKI Period(1)	25	—	ns	EC, ECIO oscillator, -40°C to +85°C
		Oscillator Period(1)	40	—	ns	EC, ECIO oscillator, +85°C to +125°C
			250	—	ns	RC oscillator
			250	10,000	ns	XT oscillator
			40	—	ns	HS oscillator, -40°C to +85°C
			40	—	ns	HS oscillator, +85°C to +125°C
			100	250	ns	HS + PLL oscillator, -40°C to +85°C
			160	250	ns	HS + PLL oscillator, +85°C to +125°C
			5	200	µs	LP oscillator
2	TCY	Instruction Cycle Time(1)	100	—	ns	TCY = 4/FOSC, -40°C to +85°C
			160	—	ns	TCY = 4/FOSC, +85°C to +125°C
3	TosL,	External Clock in (OSC1)	30	—	ns	XT oscillator
	TosH	High or Low Time	2.5	—	ns	LP oscillator
			10	—	µs	HS oscillator
4	TosR,	External Clock in (OSC1)	—	20	ns	XT oscillator
	TosF	Rise or Fall Time	—	50	ns	LP oscillator
			—	7.5	ns	HS oscillator

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

FIGURE 27-11: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND ECCP1)

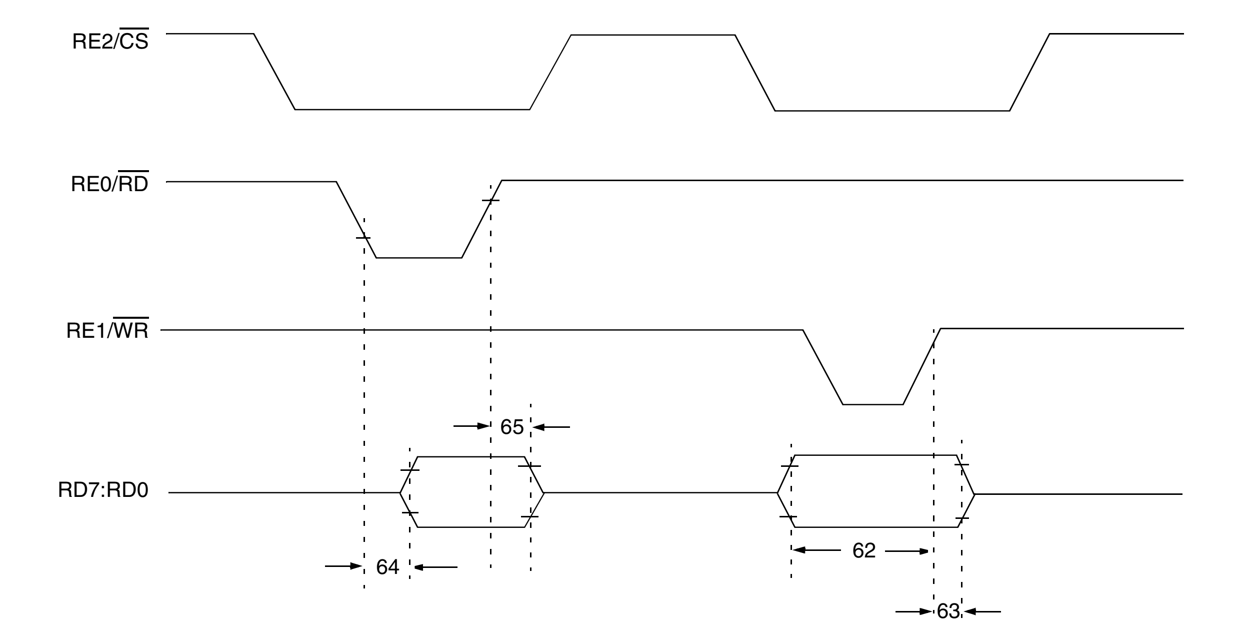


Note: Refer to Figure 27-5 for load conditions.

TABLE 27-11: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND ECCP1)

Param No.	Symbol	Characteristic			Min	Max	Units	Conditions
50	TccL	CCPx Input Low Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18**F**XX8	10	—	ns	
				PIC18**LF**XX8	20	—	ns	
51	TccH	CCPx Input High Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18**F**XX8	10	—	ns	
				PIC18**LF**XX8	20	—	ns	
52	TccP	CCPx Input Period			$\frac{3\,\mathrm{TCY} + 40}{N}$	—	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fall Time		PIC18**F**XX8	—	25	ns	
				PIC18**LF**XX8	—	45	ns	
54	TccF	CCPx Output Fall Time		PIC18**F**XX8	—	25	ns	
				PIC18**LF**XX8	—	45	ns	

FIGURE 27-12: PARALLEL SLAVE PORT TIMING (PIC18F248 AND PIC18F458)



Note: Refer to Figure 27-5 for load conditions.

TABLE 27-12: PARALLEL SLAVE PORT REQUIREMENTS (PIC18F248 AND PIC18F458)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
62	TdtV2wrH	Data-In Valid before $\overline{WR}$ ↑ or $\overline{CS}$ ↑ (setup time)		20 25	— —	ns ns	 Extended Temp. range
63	TwrH2dtI	$\overline{WR}$ ↑ or $\overline{CS}$ ↑ to Data-In Invalid (hold time)	PIC18**F**XX8	20	—	ns	
			PIC18**LF**XX8	35	—	ns	
64	TrdL2dtV	$\overline{RD}$ ↓ and $\overline{CS}$ ↓ to Data-Out Valid		— —	80 90	ns ns	 Extended Temp. range
65	TrdH2dtI	$\overline{RD}$ ↑ or $\overline{CS}$ ↓ to Data-Out Invalid		10	30	ns	
66	TibfINH	Inhibit the IBF flag bit being cleared from $\overline{WR}$ ↑ or $\overline{CS}$ ↑		—	3 TCY	ns	