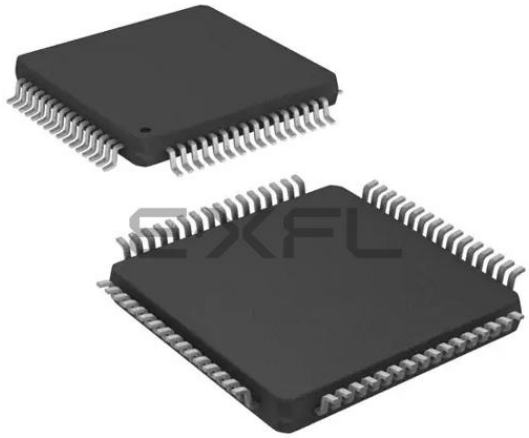


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	49
Program Memory Size	96KB (48K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66j55-i-pt

1.1.5 EXTERNAL MEMORY BUS

In the event that 128 Kbytes of memory are inadequate for an application, the 80-pin members of the PIC18F87J10 family also implement an External Memory Bus (EMB). This allows the controller's internal program counter to address a memory space of up to 2 Mbytes, permitting a level of data access that few 8-bit devices can claim. This allows additional memory options, including:

- Using combinations of on-chip and external memory up to the 2-Mbyte limit
- Using external Flash memory for reprogrammable application code or large data tables
- Using external RAM devices for storing large amounts of variable data

1.1.6 EXTENDED INSTRUCTION SET

The PIC18F87J10 family implements the optional extension to the PIC18 instruction set, adding 8 new instructions and an Indexed Addressing mode. Enabled as a device configuration option, the extension has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as 'C'.

1.1.7 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also aids in migrating to the next larger device. This is true when moving between the 64-pin members, between the 80-pin members, or even jumping from 64-pin to 80-pin devices.

The PIC18F87J10 family is also pin compatible with other PIC18 families, such as the PIC18F87J10, PIC18F87J11, PIC18F8720 and PIC18F8722. This allows a new dimension to the evolution of applications, allowing developers to select different price points within Microchip's PIC18 portfolio, while maintaining the same feature set.

1.2 Other Special Features

- **Communications:** The PIC18F87J10 family incorporates a range of serial and parallel communication peripherals, including a fully featured Universal Serial Bus communications module that is compliant with the USB Specification Revision 2.0. This device also includes 2 independent Enhanced USARTs and 2 Master SSP modules, capable of both SPI and I2C™ (Master and Slave) modes of operation. The device also has a parallel port and can be configured to serve as either a Parallel Master Port or as a Parallel Slave Port.
- **CCP Modules:** All devices in the family incorporate two Capture/Compare/PWM (CCP) modules and three Enhanced CCP modules to maximize flexibility in control applications. Up to four different time bases may be used to perform several different operations at once. Each of the three ECCPs offers up to four PWM outputs, allowing for a total of 12 PWMs. The ECCPs also offer many beneficial features, including polarity selection, programmable dead time, auto-shutdown and restart and Half-Bridge and Full-Bridge Output modes.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and thus, reducing code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit prescaler, allowing an extended time-out range that is stable across operating voltage and temperature. See **Section 28.0 "Electrical Characteristics"** for time-out periods.

1.3 Details on Individual Family Members

Devices in the PIC18F87J10 family are available in 64-pin and 80-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2. The devices are differentiated from each other in two ways:

1. Flash program memory (six sizes, ranging from 32 Kbytes for PIC18FX5J50 devices to 128 Kbytes for PIC18FX7J50).
2. I/O ports (7 bidirectional ports on 64-pin devices, 9 bidirectional ports on 80-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1 and Table 1-2.

The pinouts for all devices are listed in Table 1-3 and Table 1-4.

TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
PMADDRH	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1H	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
PMADDRL	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	Feature1	PIC18F8XJ5X	0000 0000	0000 0000	uuuu uuuu
UCON	Feature1	PIC18F8XJ5X	-0x0 000-	-0x0 000-	-uuu uuu-
USTAT	Feature1	PIC18F8XJ5X	-xxx xxx-	-xxx xxx-	-uuu uuu-
UEIR	Feature1	PIC18F8XJ5X	0--0 0000	0--0 0000	u--u uuuu
UIR	Feature1	PIC18F8XJ5X	-000 0000	-000 0000	-uuu uuuu
UFRMH	Feature1	PIC18F8XJ5X	---- -xxx	---- -xxx	---- -uuu
UFRML	Feature1	PIC18F8XJ5X	xxxx xxxx	xxxx xxxx	uuuu uuuu
UCFG	Feature1	PIC18F8XJ5X	00-0 0000	00-0 0000	uu-u uuuu
UADDR	Feature1	PIC18F8XJ5X	-000 0000	-uuu uuuu	-uuu uuuu
UEIE	Feature1	PIC18F8XJ5X	0--0 0000	0--0 0000	u--u uuuu
UIE	Feature1	PIC18F8XJ5X	-000 0000	-000 0000	-uuu uuuu
UEP15	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP14	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP13	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP12	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP11	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP10	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP9	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP8	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP7	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP6	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP5	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP4	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP3	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP2	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP1	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
UEP0	Feature1	PIC18F8XJ5X	---0 0000	---0 0000	---u uuuu
PMCONH	Feature1	PIC18F8XJ5X	0-00 0000	0-00 0000	u-uu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

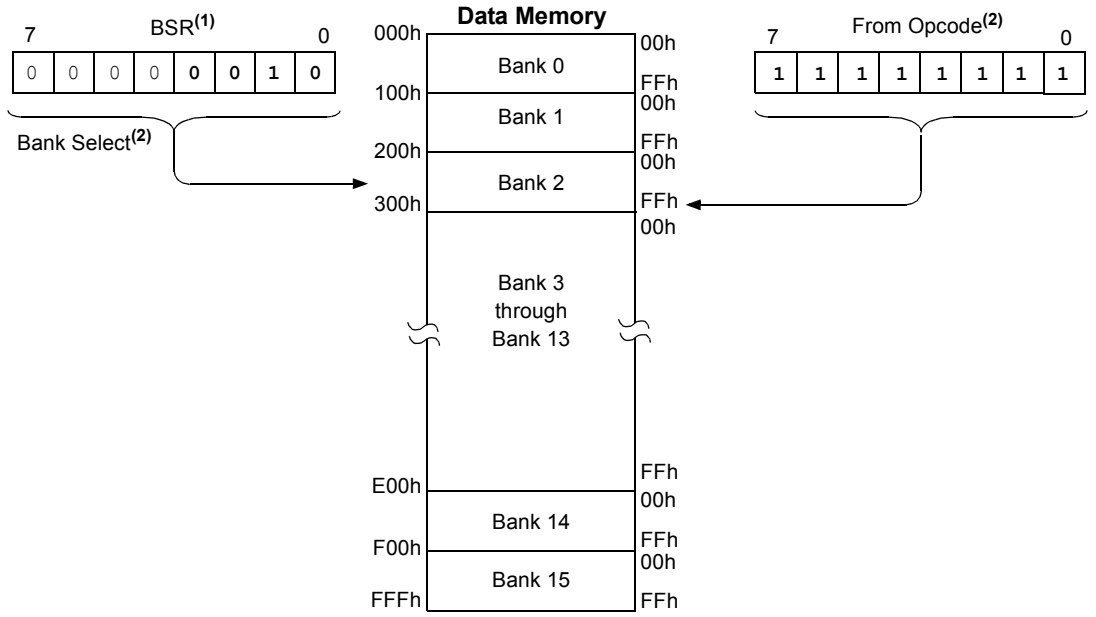
Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 4-1 for Reset value for specific condition.

FIGURE 5-8: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM 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.

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

5.3.3 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from or written to the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the “Access RAM” and is composed of GPRs. The upper half is where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 5-7).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘`1`’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘`0`’, however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = `1`). This is discussed in more detail in **Section 5.6.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

5.3.4 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2, PIE3). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

REGISTER 9-7: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 7 **PMPIE:** Parallel Master Port Read/Write Interrupt Enable bit
1 = Enables the PM read/write interrupt
0 = Disables the PM 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 **RC1IE:** EUSART1 Receive Interrupt Enable bit
1 = Enables the EUSART1 receive interrupt
0 = Disables the EUSART1 receive interrupt

bit 4 **TX1IE:** EUSART1 Transmit Interrupt Enable bit
1 = Enables the EUSART1 transmit interrupt
0 = Disables the EUSART1 transmit interrupt

bit 3 **SSP1IE:** Master Synchronous Serial Port Interrupt Enable bit (MSSP1 module)
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt

bit 2 **CCP1IE:** ECCP1 Interrupt Enable bit
1 = Enables the ECCP1 interrupt
0 = Disables the ECCP1 interrupt

bit 1 **TMR2IE:** 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

REGISTER 10-4: PADCFG1: PAD CONFIGURATION CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
bit 7							bit 0

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

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit
1 = PMP module uses TTL input buffers
0 = PMP module uses Schmitt Trigger input buffers

10.2 PORTA, TRISA and LATA Registers

PORTA is a 6-bit wide, bidirectional port. The corresponding Data Direction register is TRISA. The corresponding Output Latch register is LATA.

The RA4 pin is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. It is also multiplexed as the Parallel Master Port Data pin. The other PORTA pins are multiplexed with the analog VREF+ and VREF- inputs. The operation of pins RA5:RA0 as A/D Converter inputs is selected by clearing or setting the control bits in the ANCON0 register.

Note 1: The RA5 (RA5/PMD4/AN4/C2INA) pin is a multiplexed A/D convertor, Parallel Master Port data and also a Comparator 2 input A. (PMP pin placement depends on the PMPMX Configuration bit.)

2: RA5 and RA3:RA0 are configured as analog inputs on any Reset and are read as '0'. RA4 is configured as a digital input.

The RA4/T0CKI pin is a Schmitt Trigger input. All other PORTA pins have TTL input levels and full CMOS output drivers.

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

OSC2/CLKO/RA6 and OSC1/CLKI/RA7 normally serve as the external circuit connections for the external (primary) oscillator circuit (HS and HSPLL Oscillator modes), or the external clock input (EC and ECPLL Oscillator modes). In these cases, RA6 and RA7 are not available as digital I/O and their corresponding TRIS and LAT bits are read as '0'.

For INTOSCx and INTOSCPLLx Oscillator modes (FOSC2 Configuration bit is '0'), either RA7, or both RA6 and RA7, automatically become available as digital I/O, depending on the oscillator mode selected. When RA6 is not configured as a digital I/O, in these cases, it provides a clock output at FOSC/4. A list of the possible configurations for RA6 and RA7, based on oscillator mode, is provided in Register 10-3. For these pins, the corresponding PORTA, TRISA and LATA bits are only defined when the pins are configured as I/O.

TABLE 10-3: FUNCTION OF RA7:RA6 IN INTOSC AND INTOSCPLL MODES

Oscillator Mode (FOSC2:FOSC0 Configuration bits)	RA6	RA7
INTOSCPLLO (011)	CLKO	I/O
INTOSCPLL (010)	I/O	I/O
INTOSCO (001)	CLKO	I/O
INTOSC (000)	I/O	I/O

Legend: CLKO = FOSC/4 clock output; I/O = digital port.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
CLRF    LATA          ; Alternate method to
                      ; clear  data latches
BSF     WDTCON,ADSHR  ; Enable write/read to
                      ; the shared SFR
MOVLW   1Fh           ; Configure A/D
MOVWF   ANCON0        ; for digital inputs
BCF     WDTCON,ADSHR  ; Disable write/read
                      ; to the shared SFR
MOVLW   0CFh          ; Value used to
                      ; initialize
                      ; data direction
MOVWF   TRISA         ; Set RA<3:0> as inputs,
                      ; RA<5:4> as outputs
```

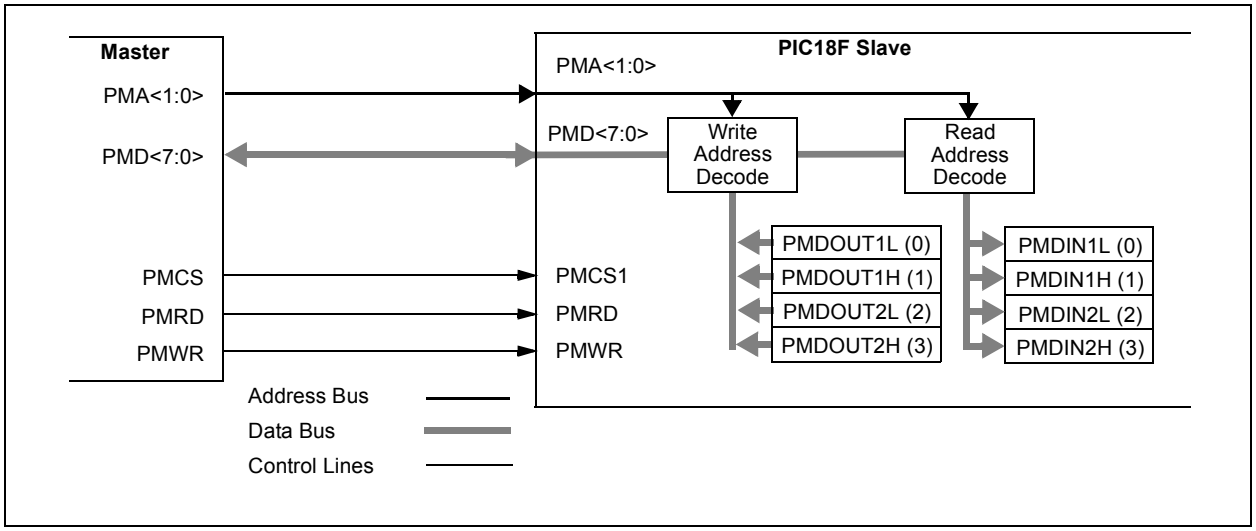
11.2.5 ADDRESSABLE PARALLEL SLAVE PORT MODE

In the Addressable Parallel Slave Port mode (PMMODEH<1:0> = 01), the module is configured with two extra inputs, PMA<1:0>, which are the address lines 1 and 0. This makes the 4-byte buffer space directly addressable as fixed pairs of read and write buffers. As with legacy Buffered mode, data is output from PMDOUT1L, PMDOUT1H, PMDOUT2L and PMDOUT2H, and is read in PMDIN1L, PMDIN1H, PMDIN2L and PMDIN2H. Table 11-2 shows the buffer addressing for the incoming address to the input and output registers.

TABLE 11-2: SLAVE MODE BUFFER ADDRESSING

PMADDR<1:0>	Output Register (Buffer)	Input Register (Buffer)
00	PMDOUT1L (0)	PMDIN1L (0)
01	PMDOUT1H (1)	PMDIN1H (1)
10	PMDOUT2L (2)	PMDIN2L (2)
11	PMDOUT2H((3)	PMDIN2H (3)

FIGURE 11-6: PARALLEL MASTER/SLAVE CONNECTION ADDRESSED BUFFER EXAMPLE



11.3 MASTER PORT MODES

In its Master modes, the PMP module provides an 8-bit data bus, up to 16 bits of address, and all the necessary control signals to operate a variety of external parallel devices, such as memory devices, peripherals and slave microcontrollers. To use the PMP as a master, the module must be enabled (PMPEN = 1) and the mode must be set to one of the two possible Master modes (PMMODEH<1:0> = 10 or 11).

Because there are a number of parallel devices with a variety of control methods, the PMP module is designed to be extremely flexible to accommodate a range of configurations. Some of these features include:

- 8 and 16-Bit Data modes on an 8-bit data bus
- Configurable address/data multiplexing
- Up to two chip select lines
- Up to 16 selectable address lines
- Address auto-increment and auto-decrement
- Selectable polarity on all control lines
- Configurable wait states at different stages of the read/write cycle

11.3.1 PMP AND I/O PIN CONTROL

Multiple control bits are used to configure the presence or absence of control and address signals in the module. These bits are PTBEEN, PTWREN, PTRDEN, and PTEN<15:0>. They give the user the ability to conserve pins for other functions and allow flexibility to control the external address. When any one of these bits is set, the associated function is present on its associated pin; when clear, the associated pin reverts to its defined I/O port function.

Setting a PTEN bit will enable the associated pin as an address pin and drive the corresponding data contained in the PMADDR register. Clearing the PTENx bit will force the pin to revert to its original I/O function.

For the pins configured as chip select (PMCS1 or PMCS2) with the corresponding PTENx bit set. The PTEN0 and PTEN1 bits also control the PMALL and PMALH signals. When multiplexing is used, the associated address latch signals should be enabled.

11.3.2 READ/WRITE CONTROL

The PMP module supports two distinct read/write signaling methods. In Master Mode 1, read and write strobe are combined into a single control line, PMRD/PMWR. A second control line, PMENB, determines when a read or write action is to be taken. In Master Mode 2, separate Read and Write strobes (PMRD and PMWR) are supplied on separate pins.

All control signals (PMRD, PMWR, PMBE, PMENB, PMAL and PMCSx) can be individually configured as either positive or negative polarity. Configuration is controlled by separate bits in the PMCONL register. Note that the polarity of control signals that share the same output pin (for example, PMWR and PMENB) are controlled by the same bit; the configuration depends on which Master Port mode is being used.

11.3.3 DATA WIDTH

The PMP supports data widths of both 8 and 16 bits. The data width is selected by the MODE16 bit (PMMODEH<2>). Because the data path into and out of the module is only 8 bits wide, 16-bit operations are always handled in a multiplexed fashion, with the Least Significant Byte of data being presented first. To differentiate data bytes, the Byte Enable control strobe, PMBE, is used to signal when the Most Significant Byte of data is being presented on the data lines.

11.3.4 ADDRESS MULTIPLEXING

In either of the Master modes (PMMODEH<1:0> = 1x), the user can configure the address bus to be multiplexed together with the data bus. This is accomplished using the ADRMUX1:ADRMUX0 bits (PMCONH<4:3>). There are three address multiplexing modes available; typical pinout configurations for these modes are shown in Figure 11-9, Figure 11-10 and Figure 11-11.

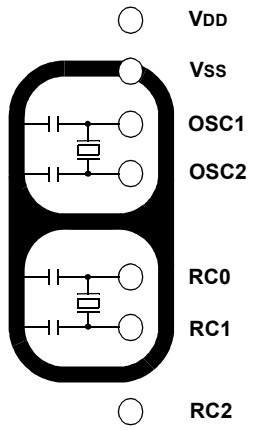
In Demultiplexed mode (PMCONH<4:3> = 00), data and address information are completely separated. Data bits are presented on PMD<7:0>, and address bits are presented on PMADDRH<7:0> and PMADDRL<7:0>

In Partially Multiplexed mode (PMCONH<4:3> = 01), the lower eight bits of the address are multiplexed with the data pins on PMD<7:0>. The upper eight bits of address are unaffected and are presented on PMADDRH<7:0>. The PMA0 pin is used as an Address Latch, and presents the Address Latch Low enable strobe (PMALL). The read and write sequences are extended by a complete CPU cycle during which the address is presented on the PMD<7:0> pins.

In Fully Multiplexed mode (PMCONH<4:3> = 10), the entire 16 bits of the address are multiplexed with the data pins on PMD<7:0>. The PMA0 and PMA1 pins are used to present Address Latch Low enable (PMALL) and Address Latch High enable (PMALH) strobes, respectively. The read and write sequences are extended by two complete CPU cycles. During the first cycle, the lower eight bits of the address are presented on the PMD<7:0> pins with the PMALL strobe active. During the second cycle, the upper eight bits of the address are presented on the PMD<7:0> pins with the PMALH strobe active. In the event the upper address bits are configured as chip select pins, the corresponding address bits are automatically forced to '0'.

If a high-speed circuit must be located near the oscillator (such as the ECCP1 pin in Output Compare or PWM mode, or the primary oscillator using the OSC2 pin), a grounded guard ring around the oscillator circuit, as shown in Figure 13-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

FIGURE 13-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING



Note: Not drawn to scale.

13.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled or disabled by setting or clearing the Timer1 Interrupt Enable bit, TMR1IE (PIE1<0>).

13.5 Resetting Timer1 Using the ECCP Special Event Trigger

If ECCP1 or ECCP2 is configured to use Timer1 and to generate a Special Event Trigger in Compare mode (CCPxM3:CCPxM0 = 1011), this signal will reset Timer3. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 18.2.1 "Special Event Trigger"** for more information).

The module must be configured as either a timer or a synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a period register for Timer1.

If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a Special Event Trigger, the write operation will take precedence.

Note: The Special Event Triggers from the ECCPx module will not set the TMR1IF interrupt flag bit (PIR1<0>).

13.6 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 13.3 "Timer1 Oscillator"**) gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, RTCisr, shown in Example 13-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow triggers the interrupt and calls the routine which increments the seconds counter by one. Additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it. The simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1) as shown in the routine, RTCinit. The Timer1 oscillator must also be enabled and running at all times.

13.7 Considerations in Asynchronous Counter Mode

Following a Timer1 interrupt and an update to the TMR1 registers, the Timer1 module uses a falling edge on its clock source to trigger the next register update on the rising edge. If the update is completed after the clock input has fallen, the next rising edge will not be counted.

If the application can reliably update TMR1 before the timer input goes low, no additional action is needed. Otherwise, an adjusted update can be performed following a later Timer1 increment. This can be done by monitoring TMR1L within the interrupt routine until it increments, and then updating the TMR1H:TMR1L register pair while the clock is low, or one-half of the period of the clock source. Assuming that Timer1 is being used as a Real-Time Clock, the clock source is a 32.768 kHz crystal oscillator. In this case, one-half period of the clock is 15.25 μs.

19.4.3.5 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and the SDAx line is held low ($\overline{ACK}$).

When the address byte overflow condition exists, then the no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit, BF (SSPxSTAT<0>), is set or bit, SSPOV (SSPxCON1<6>), is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPxIF, must be cleared in software. The SSPxSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPxCON2<0> = 1), SCLx will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPxCON1<4>). See **Section 19.4.4 "Clock Stretching"** for more details.

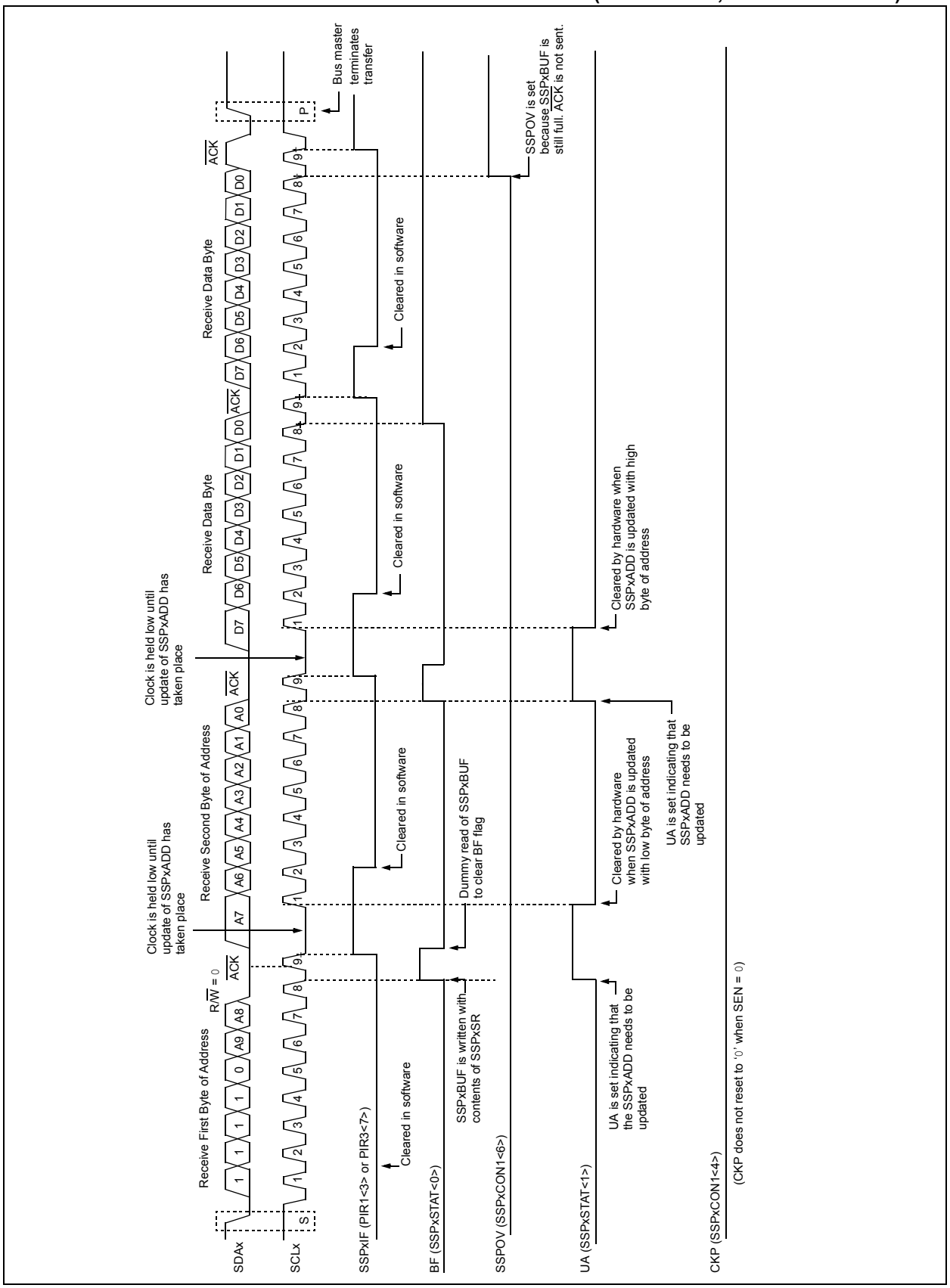
19.4.3.6 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin SCLx is held low regardless of SEN (see **Section 19.4.4 "Clock Stretching"** for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then, pin SCLx should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 19-10).

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not $\overline{ACK}$), then the data transfer is complete. In this case, when the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets the SSPxSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, pin SCLx must be enabled by setting bit, CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared in software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

FIGURE 19-12: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)



19.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPxCON2<4>). When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG; the SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into an inactive state (Figure 19-25).

19.4.12.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

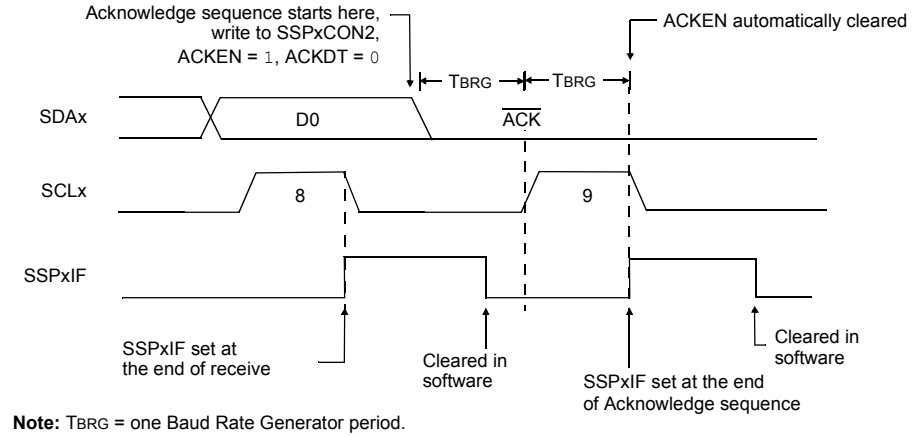
19.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPxCON2<2>). At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit (SSPxSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 19-26).

19.4.13.1 WCOL Status Flag

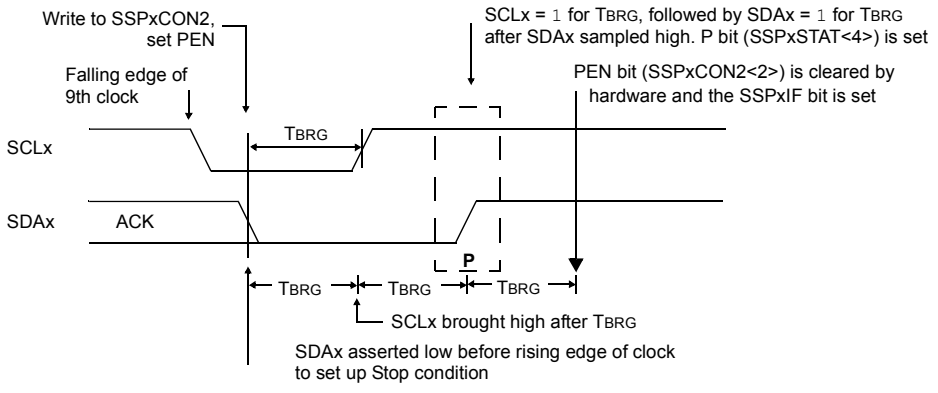
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 19-25: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 19-26: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

19.4.14 SLEEP OPERATION

While in Sleep mode, the I^2C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

19.4.15 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

19.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit (SSPxSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDAx line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLxIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

19.4.17 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDAx pin, arbitration takes place when the master outputs a '1' on SDAx, by letting SDAx float high and another master asserts a '0'. When the SCLx pin floats high, data should be stable. If the expected data on SDAx is a '1' and the data sampled on the SDAx pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF and reset the I^2C port to its Idle state (Figure 19-27).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDAx and SCLx lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDAx and SCLx lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine, and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDAx and SCLx pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 19-27: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE

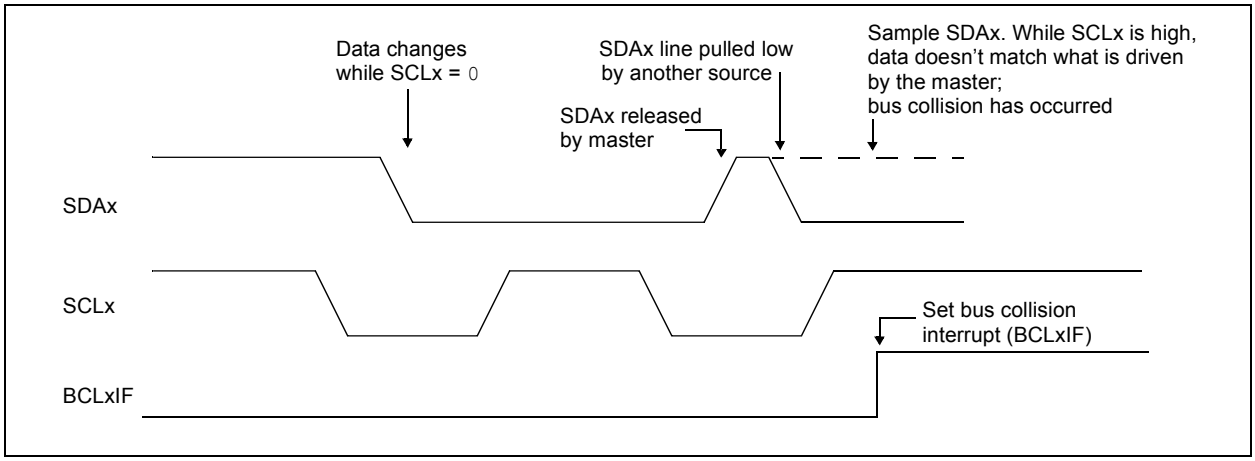


TABLE 20-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	61
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	64
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	64
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	64
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	64
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	64
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	64
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	63
RCREGx	EUSARTx Receive Register								63
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	63
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	65
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								65
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								65

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

REGISTER 22-2: UCFG: USB CONFIGURATION REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	—	—	UPUEN(1,2)	UTRDIS(1)	FSEN(1)	PPB1	PPB0
bit 7							bit 0

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

bit 7 **UTEYE:** USB Eye Pattern Test Enable bit
1 = Eye pattern test enabled
0 = Eye pattern test disabled

bit 6 **Unimplemented:** Always should be programmed to '0'(3)

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

bit 4 **UPUEN:** USB On-Chip Pull-up Enable bit(1,2)
1 = On-chip pull-up enabled (pull-up on D+ with FSEN = 1 or D- with FSEN = 0)
0 = On-chip pull-up disabled

bit 3 **UTRDIS:** On-Chip Transceiver Disable bit(1)
1 = On-chip transceiver disabled
0 = On-chip transceiver active

bit 2 **FSEN:** Full-Speed Enable bit(1)
1 = Full-speed device: controls transceiver edge rates; requires input clock at 48 MHz
0 = Low-speed device: controls transceiver edge rates; requires input clock at 6 MHz

bit 1-0 **PPB1:PPB0:** Ping-Pong Buffers Configuration bits
11 = Even/Odd ping-pong buffers enabled for Endpoints 1 to 15
10 = Even/Odd ping-pong buffers enabled for all endpoints
01 = Even/Odd ping-pong buffer enabled for OUT Endpoint 0
00 = Even/Odd ping-pong buffers disabled

Note 1: The UPUEN, UTRDIS and FSEN bits should never be changed while the USB module is enabled. These values must be preconfigured prior to enabling the module.
2: This bit is only valid when the on-chip transceiver is active (UTRDIS = 0); otherwise, it is ignored.
3: Firmware should never set this bit. Doing so may cause unexpected behavior.

22.5.2 USB INTERRUPT ENABLE REGISTER (UIE)

The USB Interrupt Enable register (Register 22-8) contains the enable bits for the USB status interrupt sources. Setting any of these bits will enable the respective interrupt source in the UIR register.

The values in this register only affect the propagation of an interrupt condition to the microcontroller's interrupt logic. The flag bits are still set by their interrupt conditions, allowing them to be polled and serviced without actually generating an interrupt.

REGISTER 22-8: UIE: USB INTERRUPT ENABLE REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
bit 7							bit 0

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

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

bit 6 **SOFIE:** Start-of-Frame Token Interrupt Enable bit
1 = Start-of-Frame token interrupt enabled
0 = Start-of-Frame token interrupt disabled

bit 5 **STALLIE:** STALL Handshake Interrupt Enable bit
1 = STALL interrupt enabled
0 = STALL interrupt disabled

bit 4 **IDLEIE:** Idle Detect Interrupt Enable bit
1 = Idle detect interrupt enabled
0 = Idle detect interrupt disabled

bit 3 **TRNIE:** Transaction Complete Interrupt Enable bit
1 = Transaction interrupt enabled
0 = Transaction interrupt disabled

bit 2 **ACTVIE:** Bus Activity Detect Interrupt Enable bit
1 = Bus activity detect interrupt enabled
0 = Bus activity detect interrupt disabled

bit 1 **UERRIE:** USB Error Interrupt Enable bit
1 = USB error interrupt enabled
0 = USB error interrupt disabled

bit 0 **URSTIE:** USB Reset Interrupt Enable bit
1 = USB Reset interrupt enabled
0 = USB Reset interrupt disabled

22.7 Oscillator

The USB module has specific clock requirements. For full-speed operation, the clock source must be 48 MHz. Even so, the microcontroller core and other peripherals are not required to run at that clock speed. Available clocking options are described in detail in **Section 2.3 "Oscillator Settings for USB"**.

22.8 USB Firmware and Drivers

Microchip provides a number of application-specific resources, such as USB firmware and driver support. Refer to www.microchip.com for the latest firmware and driver support.

TABLE 22-4: REGISTERS ASSOCIATED WITH USB MODULE OPERATION(1)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Details on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	81
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	LVDIP	TMR3IP	CCP2IP	85
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	LVDIF	TMR3IF	CCP2IF	85
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	LVDIE	TMR3IE	CCP2IE	85
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	87
UCFG	UTEYE	—	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	87
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	87
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	87
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	87
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	87
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	87
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	87
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	87
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	87
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	88
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	87

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the USB module.

Note 1: This table includes only those hardware mapped SFRs located in Bank 15 of the data memory space. The Buffer Descriptor registers, which are mapped into Bank 4 and are not true SFRs, are listed separately in Table 22-3.

REGISTER 25-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

Legend:			
R = Readable bit	WO = Write-Once bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IESO:** Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit
1 = Two-Speed Start-up enabled
0 = Two-Speed Start-up disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **FOSC2:FOSC0:** Oscillator Selection bits
111 = ECPLL oscillator with PLL enabled, CLKO on RA6, ECPLL oscillator used by USB
110 = EC oscillator with CLKO on RA6 , EC oscillator used by USB
101 = HSPLL oscillator with PLL enabled, HSPLL oscillator used by USB
100 = HS oscillator, HS oscillator used by USB
011 = INTOSCPLLO, internal oscillator with INTOSCPLL enabled, CLKO on RA6 and port function RA7
010 = INTOSCPLL, Internal oscillator with Port function on RA6 and RA7
001 = INTOSCO internal oscillator block (INTRC/INTOSC) with CLKO on RA6 Port function on RA7
000 = INTOSC internal oscillator block (INTRC/INTOSC) Port function on RA6 and RA7

SLEEP **Enter Sleep Mode**

Syntax: SLEEP

Operands: None

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

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

Encoding:

0000	0000	0000	0011

Description: The Power-Down status bit ($\overline{PD}$) is cleared. The Time-out status bit ($\overline{TO}$) is set. The Watchdog Timer and its postscaler are cleared.

The processor is put into Sleep mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to Sleep

Example: SLEEP

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?
After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared.

SUBFWB **Subtract f from W with Borrow**

Syntax: SUBFWB f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

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

Status Affected: N, OV, C, DC, Z

Encoding:

0101	01da	ffff	ffff

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

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: SUBFWB REG, 1, 0

Before Instruction
REG = 3
W = 2
C = 1
After Instruction
REG = FF
W = 2
C = 0
Z = 0
N = 1 ; result is negative

Example 2: SUBFWB REG, 0, 0

Before Instruction
REG = 2
W = 5
C = 1
After Instruction
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; result is positive

Example 3: SUBFWB REG, 1, 0

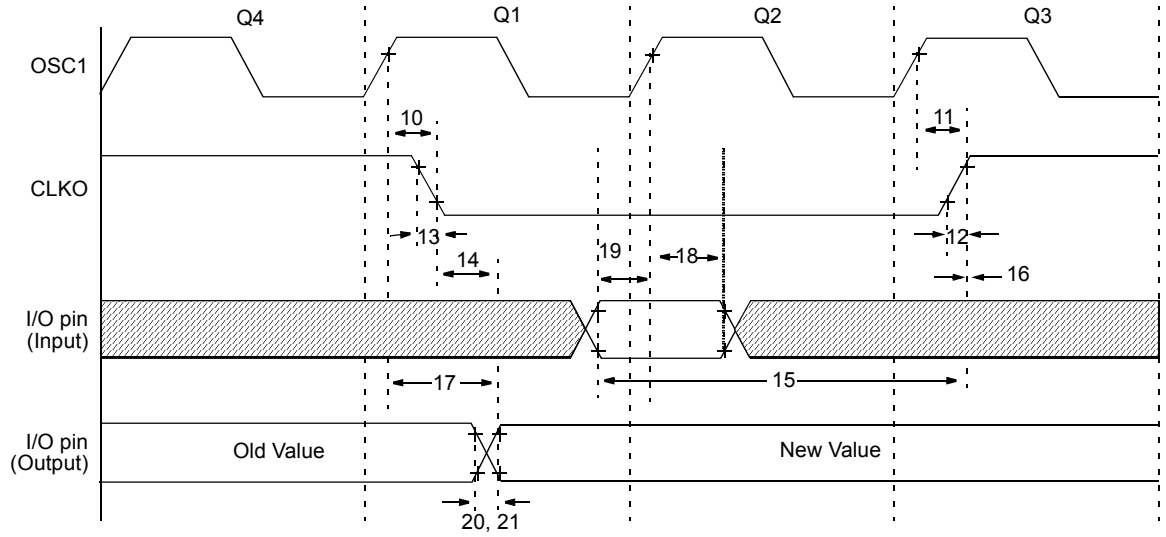
Before Instruction
REG = 1
W = 2
C = 0
After Instruction
REG = 0
W = 2
C = 1
Z = 1 ; result is zero
N = 0

TABLE 28-5: USB MODULE SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Sym**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D313	VUSB	USB Voltage	3.0	—	3.6	V	Voltage on VUSB pin must be in this range for proper USB operation
D314	IIL	Input Leakage on pin	—	—	±1	μA	VSS ≤ VPIN ≤ VDD pin at high impedance
D315	VILUSB	Input Low Voltage for USB Buffer	—	—	0.8	V	For VUSB range
D316	VIHUSB	Input High Voltage for USB Buffer	2.0	—	—	V	For VUSB range
D318	VDIFS	Differential Input Sensitivity	—	—	0.2	V	The difference between D+ and D- must exceed this value while VCM is met
D319	VCM	Differential Common Mode Range	0.8	—	2.5	V	
D320	ZOUT	Driver Output Impedance(1)	28	—	44	Ω	
D321	VOL	Voltage Output Low	0.0	—	0.3	V	1.5 kΩ load connected to 3.6V
D322	VOH	Voltage Output High	2.8	—	3.6	V	1.5 kΩ load connected to ground

Note 1: The D+ and D- signal lines have built-in impedance matching resistors. No external resistors, capacitors or magnetic components are necessary on the D+/D- signal paths between the PIC18F87J10 family device and USB cable.

FIGURE 28-5: CLKO AND I/O TIMING



Note: Refer to Figure 28-3 for load conditions.

TABLE 28-10: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
10	TosH2ckL	OSC1 ↑ to CLKO ↓	—	75	200	ns	**(Note 1)**
11	TosH2ckH	OSC1 ↑ to CLKO ↑	—	75	200	ns	**(Note 1)**
12	TckR	CLKO Rise Time	—	15	30	ns	**(Note 1)**
13	TckF	CLKO Fall Time	—	15	30	ns	**(Note 1)**
14	TckL2ioV	CLKO ↓ to Port Out Valid	—	—	0.5 TCY + 20	ns	
15	TioV2ckH	Port In Valid before CLKO ↑	0.25 TCY + 25	—	—	ns	
16	TckH2ioI	Port In Hold after CLKO ↑	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 cycle) to Port Out Valid	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	100	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns	
20	TioR	Port Output Rise Time	—	—	6	ns	
21	TioF	Port Output Fall Time	—	—	5	ns	
22†	TINP	INTx pin High or Low Time	TCY	—	—	ns	
23†	TRBP	RB7:RB4 Change INTx High or Low Time	TCY	—	—	ns	

† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in EC mode, where CLKO output is 4 x TOSC.