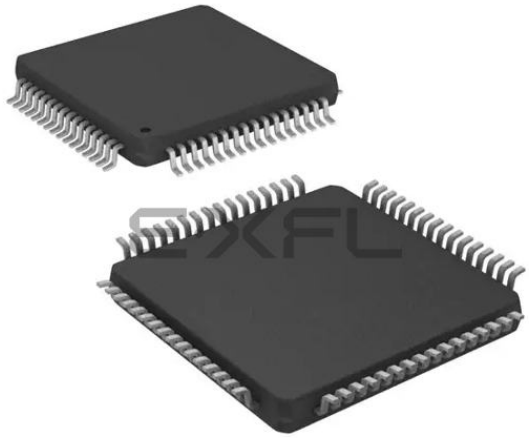


Welcome to E-XFL.COM

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	128KB (64K 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/pic18f67j50t-i-pt

FIGURE 4-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 2

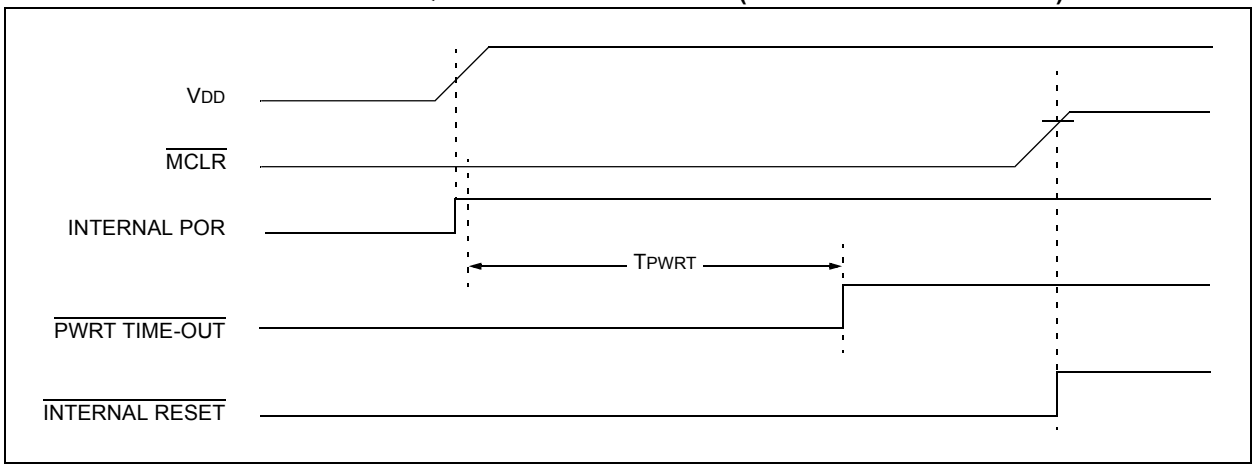
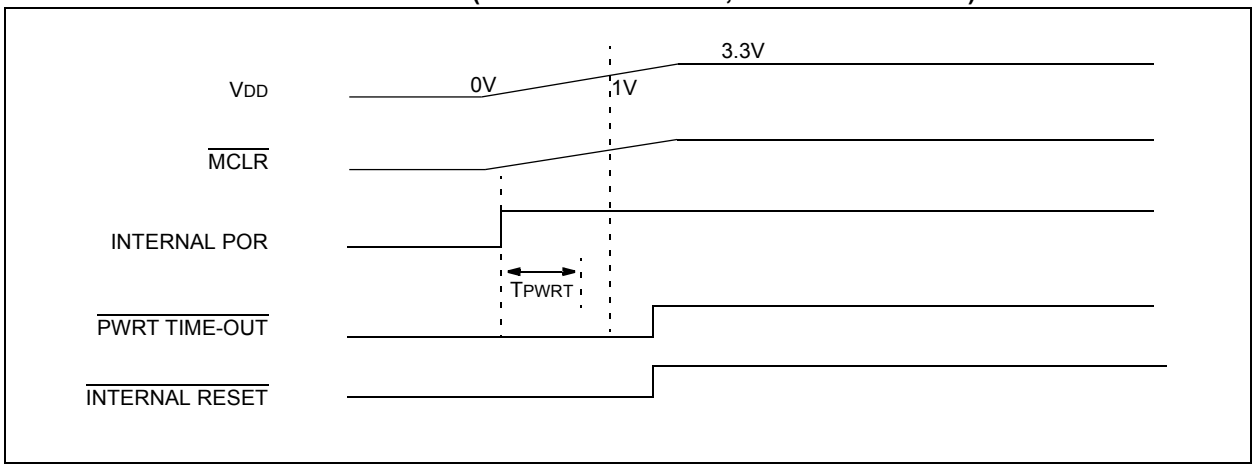


FIGURE 4-6: SLOW RISE TIME ($\overline{MCLR}$ TIED TO V_{DD}, V_{DD} RISE > T_{PWRT})



5.3.5.1 Shared Address SFRs

In several locations in the SFR bank, a single address is used to access two different hardware registers. In these cases, a “legacy” register of the standard PIC18 SFR set (such as OSCCON, T1CON, etc.) shares its address with an alternate register. These alternate registers are associated with enhanced configuration options for peripherals, or with new device features not included in the standard PIC18 SFR map. A complete list of shared register addresses and the registers associated with them is provided in Table 5-4.

Access to the alternate registers is enabled in software by setting the ADSHR bit in the WDTCON register (Register 5-3). ADSHR must be manually set or cleared to access the alternate or legacy registers, as required. Since the bit remains in a given state until changed, users should always verify the state of ADSHR before writing to any of the shared SFR addresses.

5.3.5.2 Context Defined SFRs

In addition to the shared address SFRs, there are several registers that share the same address in the SFR space, but are not accessed with the ADSHR bit. Instead, the register’s definition and use depends on the operating mode of its associated peripheral. These registers are:

- SSPxADD and SSPxMSK: These are two separate hardware registers, accessed through a single SFR address. The operating mode of the MSSP modules determines which register is being accessed. See **Section 19.4.3.4 “7-Bit Address Masking Mode”** for additional details.
- PMADDRH/L and PMDOUT2H/L: In this case, these named buffer pairs are actually the same physical registers. The PMP module’s operating mode determines what function the registers take on. See **Section 11.1.2 “Data Registers”** for additional details.

TABLE 5-4: SHARED SFR ADDRESSES FOR PIC18F87J50 FAMILY DEVICES

Address		Name	Address		Name	Address		Name
FD3h	(D)	OSCCON	FCDh	(D)	T1CON	FC2h	(D)	ADCON0
	(A)	REFOCON		(A)	ODCON3		(A)	ANCON1
FCFh	(D)	TMR1H	FCCh	(D)	TMR2	FC1h	(D)	ADCON1
	(A)	ODCON1		(A)	PADCFG1		(A)	ANCON0
FCEh	(D)	TMR1L	FCBh	(D)	PR2	F77h	(D)	PR4
	(A)	ODCON2		(A)	MEMCON(1)		(A)	CVRCON

Legend: (D) = Default SFR, accessible only when ADSHR = 0; (A) = Alternate SFR, accessible only when ADSHR = 1.
Note 1: Implemented in 80-pin devices only.

REGISTER 5-3: WDTCON: WATCHDOG TIMER CONTROL REGISTER

R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN
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 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit
For details of bit operation, see Register 25-9 on page 359.

bit 6 **LVDSTAT:** Low-Voltage Detect Status bit
1 = V_{DDCORE} > 2.45V nominal
0 = V_{DDCORE} < 2.45V nominal

bit 5 **Unimplemented**: Read as ‘0’

bit 4 **ADSHR:** Shared Address SFR Select bit
1 = Alternate SFR is selected
0 = Default (legacy) SFR is selected

bit 3-1 **Unimplemented**: Read as ‘0’

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit
For details of bit operation, see Register 25-9.

6.5 Writing to Flash Program Memory

The programming block is 32 words or 64 bytes. Programming one word or two bytes at a time is also supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the `TBLWT` instruction may need to be executed 64 times for each programming operation (if WPROG = `0`). All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

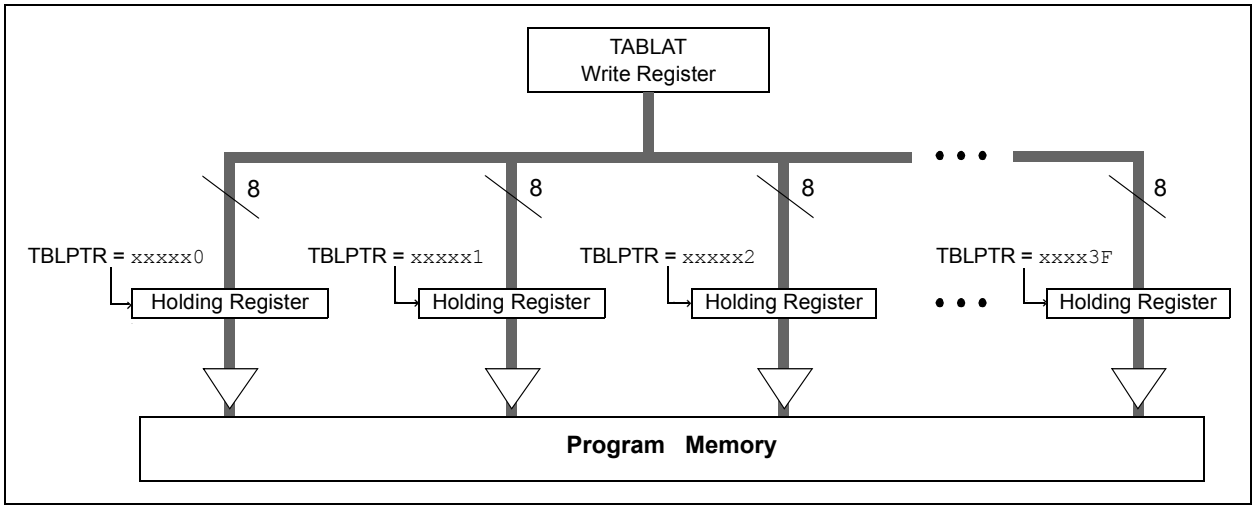
The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The 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.

Note 1: Unlike previous PIC® devices, members of the PIC18F87J10 family do not reset the holding registers after a write occurs. The holding registers must be cleared or overwritten before a programming sequence.

2: To maintain the endurance of the program memory cells, each Flash byte should not be programmed more than one time between erase operations. Before attempting to modify the contents of the target cell a second time, a row erase of the target row, or a bulk erase of the entire memory, must be performed.

FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY



6.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

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

1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the row erase procedure.
5. Load Table Pointer register with address of first byte being written, minus 1.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the WREN bit (EECON1<2>) to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write for T_{IW} (see parameter D133A).
13. Re-enable interrupts.
14. Repeat steps 6 through 13 until all 1024 bytes are written to program memory.
15. Verify the memory (table read).

An example of the required code is shown in Example 6-3 on the following page.

Note: Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

REGISTER 9-6: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
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 **SSP2IF:** Master Synchronous Serial Port 2 Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 6 **BCL2IF:** Bus Collision Interrupt Flag bit (MSSP2 module)
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 5 **RC2IF:** EUSART2 Receive Interrupt Flag bit
1 = The EUSART2 receive buffer, RCREG2, is full (cleared when RCREG2 is read)
0 = The EUSART2 receive buffer is empty

bit 4 **TX2IF:** EUSART2 Transmit Interrupt Flag bit
1 = The EUSART2 transmit buffer, TXREG2, is empty (cleared when TXREG2 is written)
0 = The EUSART2 transmit buffer is full

bit 3 **TMR4IF:** TMR4 to PR4 Match Interrupt Flag bit
1 = TMR4 to PR4 match occurred (must be cleared in software)
0 = No TMR4 to PR4 match occurred

bit 2 **CCP5IF:** CCP5 Interrupt Flag bit
Capture mode:
1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred
Compare mode:
1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **CCP4IF:** CCP4 Interrupt Flag bit
Capture mode:
1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred
Compare mode:
1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 register compare match occurred
PWM mode:
Unused in this mode.

bit 0 **CCP3IF:** ECCP3 Interrupt Flag bit
Capture mode:
1 = A TMR1/TMR3 register capture occurred (must be cleared in software)
0 = No TMR1/TMR3 register capture occurred
Compare mode:
1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)
0 = No TMR1/TMR3 register compare match occurred
PWM mode:
Unused in this mode.

9.6 INTx Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/INT3 pins are edge-triggered. If the corresponding INTEDGx bit in the INTCON2 register is set (= 1), the interrupt is triggered by a rising edge; if the bit is clear, the trigger is on the falling edge. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit, INTxIF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxIE. Flag bit, INTxIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from the power-managed modes if bit INTxIE was set prior to going into the power-managed modes. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up.

Interrupt priority for INT1, INT2 and INT3 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0. It is always a high-priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh → 00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 12.0 "Timer0 Module"** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit, RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack. If a fast return from interrupt is not used (see **Section 5.3 "Data Memory Organization"**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF     W_TEMP                    ; W_TEMP is in virtual bank
MOVFF     STATUS, STATUS_TEMP       ; STATUS_TEMP located anywhere
MOVFF     BSR, BSR_TEMP             ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF     BSR_TEMP, BSR             ; Restore BSR
MOVF      W_TEMP, W                 ; Restore WREG
MOVFF     STATUS_TEMP, STATUS       ; Restore STATUS
```

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to nine ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Each port has three memory-mapped registers for its operation:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Output Latch register)

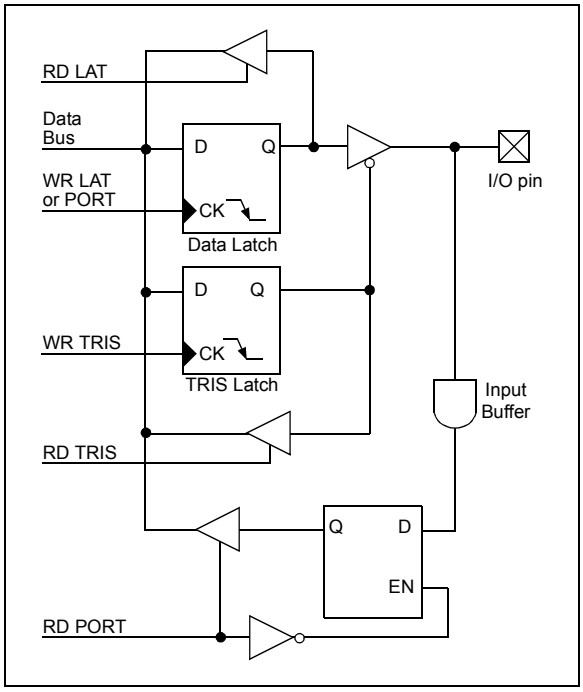
Reading the PORT register reads the current status of the pins, whereas writing to the PORT register writes to the output latch (LAT) register.

Setting a TRIS bit (= 1) makes the corresponding PORT pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRIS bit (= 0) makes the corresponding PORT pin an output (i.e., put the contents of the corresponding LAT bit on the selected pin).

The Data Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving. Read-modify-write operations on the LAT register read and write the latched output value for PORT register.

A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 10-1.

FIGURE 10-1: GENERIC I/O PORT OPERATION



10.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

10.1.1 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind (such as A/D and comparator inputs) can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided.

Table 10-1 summarizes the input capabilities. Refer to **Section 28.0 "Electrical Characteristics"** for more details.

TABLE 10-1: INPUT VOLTAGE LEVELS

Port or Pin	Tolerated Input	Description
PORTA<5:0>	VDD	Only VDD input levels tolerated.
PORTC<1:0>		
PORTF<6:1>		
PORTH<7:4>(1)		
PORTB<7:0>	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTC<7:2>		
PORTD<7:0>		
PORTE<7:0>		
PORTF<7>		
PORTG<4:0>		
PORTH<3:0>(1)		
PORTJ<7:0>(1)		

Note 1: These ports are not available on 64-pin devices.

10.1.2 PIN OUTPUT DRIVE

When used as digital I/O, the output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. In general, there are three classes of output pins in terms of drive capability.

PORTB and PORTC, as well as PORTA<7:6>, are designed to drive higher current loads, such as LEDs. PORTD, PORTE and PORTJ are capable of driving digital circuits associated with external memory devices. They can also drive LEDs, but only those with smaller current requirements. PORTF, PORTG and PORTH, along with PORTA<5:0>, have the lowest drive level, but are capable of driving normal digital circuit loads with a high input impedance.

REGISTER 10-1: ODCON1: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD
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 7-5 **Unimplemented:** Read as '0'

bit 4-3 **CCP5OD:CCP4OD:** CCPx Open-Drain Output Enable bits
1 = Open-drain output on CCPx pin (Capture/PWM modes) enabled
0 = Open-drain output disabled

bit 2-0 **ECCP3OD:ECCP1OD:** ECCPx Open-Drain Output Enable bits
1 = Open-drain output on ECCPx pin (Capture mode) enabled
0 = Open-drain output disabled

REGISTER 10-2: ODCON2: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U2OD	U1OD
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 7-2 **Unimplemented:** Read as '0'

bit 1-0 **U2OD:U1OD:** EUSARTx Open-Drain Output Enable bits
1 = Open-drain output on TXx/CKx pin enabled
0 = Open-drain output disabled

REGISTER 10-3: ODCON3: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPI2OD	SPI1OD
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 7-2 **Unimplemented:** Read as '0'

bit 1-0 **SPI2OD:SPI1OD:** SPI Open-Drain Output Enable bits
1 = Open-drain output on SDOx pin enabled
0 = Open-drain output disabled

REGISTER 11-2: PMCONL: PARALLEL PORT CONTROL REGISTER LOW BYTE

R/W-0	R/W-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP
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 7-6 **CSF1:CSF0:** Chip Select Function bits
11 = Reserved
10 = PMCS1 and PMCS2 function as chip select
01 = PMCS2 functions as chip select, PMCS1 used as address bit 14 (PMADDRH address bit 6)
00 = PMCS2 and PMCS1 used as address bits 15 and 14 (PMADDRH address bits 7 and 6)

bit 5 **ALP:** Address Latch Polarity bit(1)
1 = Active-high (PMALL and PMALH)
0 = Active-low ($\overline{\text{PMALL}}$ and $\overline{\text{PMALH}}$)

bit 4 **CS2P:** Chip Select 2 Polarity bit(1)
1 = Active-high (PMCS2)
0 = Active-low ($\overline{\text{PMCS2}}$)

bit 3 **CS1P:** Chip Select 1 Polarity bit(1)
1 = Active-high (PMCS1/PMCS)
0 = Active-low ($\overline{\text{PMCS1}}$/$\overline{\text{PMCS}}$)

bit 2 **BEP:** Byte Enable Polarity bit
1 = Byte enable active-high (PMBE)
0 = Byte enable active-low ($\overline{\text{PMBE}}$)

bit 1 **WRSP:** Write Strobe Polarity bit
For Slave modes and Master Mode 2 (PMMODEH<1:0> = 00,01,10):
1 = Write strobe active-high (PMWR)
0 = Write strobe active-low ($\overline{\text{PMWR}}$)
For Master Mode 1 (PMMODEH<1:0> = 11):
1 = Enable strobe active-high (PMENB)
0 = Enable strobe active-low ($\overline{\text{PMENB}}$)

bit 0 **RDSP:** Read Strobe Polarity bit
For Slave modes and Master Mode 2 (PMMODEH<1:0> = 00,01,10):
1 = Read strobe active-high (PMRD)
0 = Read strobe active-low ($\overline{\text{PMRD}}$)
For Master Mode 1 (PMMODEH<1:0> = 11):
1 = Read/write strobe active-high (PMRD/PMWR)
0 = Read/write strobe active-low ($\overline{\text{PMRD}}$/$\overline{\text{PMWR}}$)

Note 1: These bits have no effect when their corresponding pins are used as address lines.

REGISTER 11-3: PMMODEH: PARALLEL PORT MODE REGISTER HIGH BYTE

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
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 **BUSY:** Busy bit (Master mode only)
1 = Port is busy
0 = Port is not busy

bit 6-5 **IRQM1:IRQM0:** Interrupt Request Mode bits
11 = Interrupt generated when read buffer 3 is read or write buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = No interrupt generated, processor stall activated
01 = Interrupt generated at the end of the read/write cycle
00 = No interrupt generated

bit 4-3 **INCM1:INCM0:** Increment Mode bits
11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
10 = Decrement ADDR<15,13:0> by 1 every read/write cycle
01 = Increment ADDR<15,13:0> by 1 every read/write cycle
00 = No increment or decrement of address

bit 2 **MODE16:** 8/16-Bit Mode bit
1 = 16-Bit mode: data register is 16 bits, a read or write to the data register invokes two 8-bit transfers
0 = 8-Bit mode: data register is 8 bits, a read or write to the data register invokes one 8-bit transfer

bit 1-0 **MODE1:MODE0:** Parallel Port Mode Select bits
11 = Master Mode 1 (PMCSx, PMRD/PMWR, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
10 = Master Mode 2 (PMCSx, PMRD, PMWR, PMBE, PMA<x:0> and PMD<7:0>)
01 = Enhanced PSP, control signals (PMRD, PMWR, PMCS, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port, control signals (PMRD, PMWR, PMCS and PMD<7:0>)

18.1 ECCP Outputs and Configuration

Each of the Enhanced CCP modules may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated PxA through PxD, are multiplexed with various I/O pins. Some ECCP pin assignments are constant, while others change based on device configuration. For those pins that do change, the controlling bits are:

- CCP2MX Configuration bit
- ECCPMX Configuration bit (80-pin devices only)
- Program Memory Operating mode, set by the EMB Configuration bits (80-pin devices only)

The pin assignments for the Enhanced CCP modules are summarized in Table 18-1, Table 18-2 and Table 18-3. To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the PxMx and CCPxMx bits (CCPxCON<7:6> and <3:0>, respectively). The appropriate TRIS direction bits for the corresponding port pins must also be set as outputs.

18.1.1 ECCP1/ECCP3 OUTPUTS AND PROGRAM MEMORY MODE

In 80-pin devices, the use of Extended Microcontroller mode has an indirect effect on the use of ECCP1 and ECCP3 in Enhanced PWM modes. By default, PWM outputs, P1B/P1C and P3B/P3C, are multiplexed to PORTE pins along with the high-order byte of the External Memory Bus. When the bus is active in Extended Microcontroller mode, it overrides the Enhanced CCP outputs and makes them unavailable. Because of this, ECCP1 and ECCP3 can only be used in compatible (single output) PWM modes when the device is in Extended Microcontroller mode and default pin configuration.

An exception to this configuration is when a 12-bit address width is selected for the external bus (EMB1:EMB0 Configuration bits = 01). In this case, the upper pins of PORTE continue to operate as digital I/O, even when the external bus is active. P1B/P1C and P3B/P3C remain available for use as Enhanced PWM outputs.

If an application requires the use of additional PWM outputs during enhanced microcontroller operation, the P1B/P1C and P3B/P3C outputs can be reassigned to the upper bits of PORTH. This is done by clearing the ECCPMX Configuration bit.

18.1.2 ECCP2 OUTPUTS AND PROGRAM MEMORY MODES

For 80-pin devices, the program memory mode of the device (**Section 5.1.3 "PIC18F87J50 Family Program Memory Modes"**) also impacts pin multiplexing for the module. The ECCP2 input/output (ECCP2/P2A) can be multiplexed to one of three pins. The default assignment (CCP2MX Configuration bit is set) for all devices is RC1. Clearing CCP2MX reassigns ECCP2/P2A to RE7.

An additional option exists for 80-pin devices. When these devices are operating in Microcontroller mode, the multiplexing options described above still apply. In Extended Microcontroller mode, clearing CCP2MX reassigns ECCP2/P2A to RB3.

Changing the pin assignment of ECCP2 does not automatically change any requirements for configuring the port pin. Users must always verify that the appropriate TRIS register is configured correctly for ECCP2 operation regardless of where it is located.

18.1.3 USE OF CCP4 AND CCP5 WITH ECCP1 AND ECCP3

Only the ECCP2 module has four dedicated output pins that are available for use. Assuming that the I/O ports or other multiplexed functions on those pins are not needed, they may be used whenever needed without interfering with any other CCP module.

ECCP1 and ECCP3, on the other hand, only have three dedicated output pins: ECCPx/PxA, PxB and PxC. Whenever these modules are configured for Quad PWM mode, the pin normally used for CCP4 or CCP5 becomes the PxD output pins for ECCP3 and ECCP1, respectively. The CCP4 and CCP5 modules remain functional but their outputs are overridden.

18.1.4 ECCP MODULES AND TIMER RESOURCES

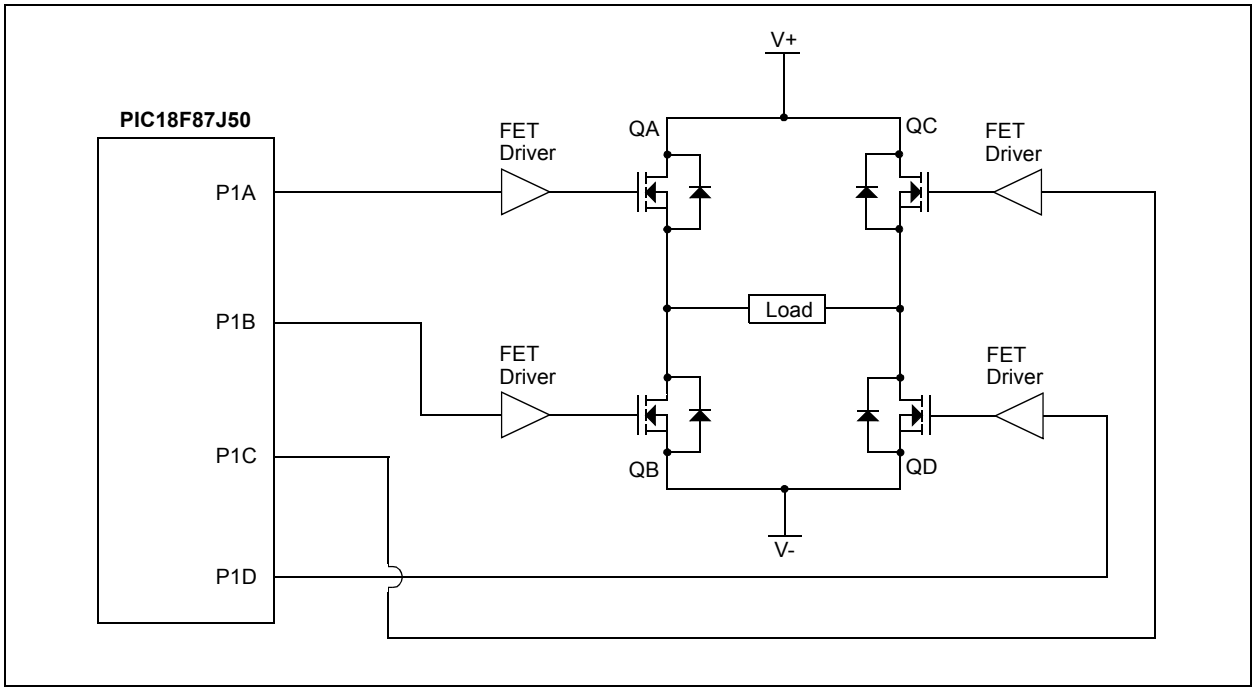
Like the standard CCP modules, the ECCP modules can utilize Timers 1, 2, 3 or 4, depending on the mode selected. Timer1 and Timer3 are available for modules in Capture or Compare modes, while Timer2 and Timer4 are available for modules in PWM mode. Additional details on timer resources are provided in **Section 17.1.1 "CCP Modules and Timer Resources"**.

18.1.5 OPEN-DRAIN OUTPUT OPTION

When operating in compare or standard PWM modes, the drivers for the ECCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 10.1.4 "Open-Drain Outputs"**.

The open-drain output option is controlled by the bits in the ODCON1 register. Setting the appropriate bit configures the pin for the corresponding module for open-drain operation. The ODCON1 memory shares the same address space as of TMR1H. The ODCON1 register can be accessed by setting the ADSHR bit in the WDTCON register (WDTCON<4>).

FIGURE 18-7: EXAMPLE OF FULL-BRIDGE OUTPUT APPLICATION



18.4.5.1 Direction Change in Full-Bridge Output Mode

In the Full-Bridge Output mode, the P1M1 bit in the CCP1CON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will assume the new direction on the next PWM cycle.

Just before the end of the current PWM period, the modulated outputs (P1B and P1D) are placed in their inactive state, while the unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction. This occurs in a time interval of (4 TOSC * (Timer2 Prescale Value) before the next PWM period begins. The Timer2 prescaler will be either 1, 4 or 16, depending on the value of the T2CKPS bits (T2CON<1:0>). During the interval from the switch of the unmodulated outputs to the beginning of the next period, the modulated outputs (P1B and P1D) remain inactive. This relationship is shown in Figure 18-8.

Note that in the Full-Bridge Output mode, the ECCP1 module does not provide any dead-band delay. In general, since only one output is modulated at all times, dead-band delay is not required. However, there is a situation where a dead-band delay might be required. This situation occurs when both of the following conditions are true:

1. The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
2. The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

Figure 18-9 shows an example where the PWM direction changes from forward to reverse at a near 100% duty cycle. At time t1, the outputs, P1A and P1D, become inactive, while output, P1C, becomes active. In this example, since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current may flow through power devices, QC and QD (see Figure 18-7), for the duration of ‘t’. The same phenomenon will occur to power devices, QA and QB, for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, one of the following requirements must be met:

1. Reduce PWM for a PWM period before changing directions.
2. Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

19.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDAx or SCLx is sampled low at the beginning of the Start condition (Figure 19-28).

b) SCLx is sampled low before SDAx is asserted low (Figure 19-29).

During a Start condition, both the SDAx and the SCLx pins are monitored.

If the SDAx pin is already low, or the SCLx pin is already low, then all of the following occur:

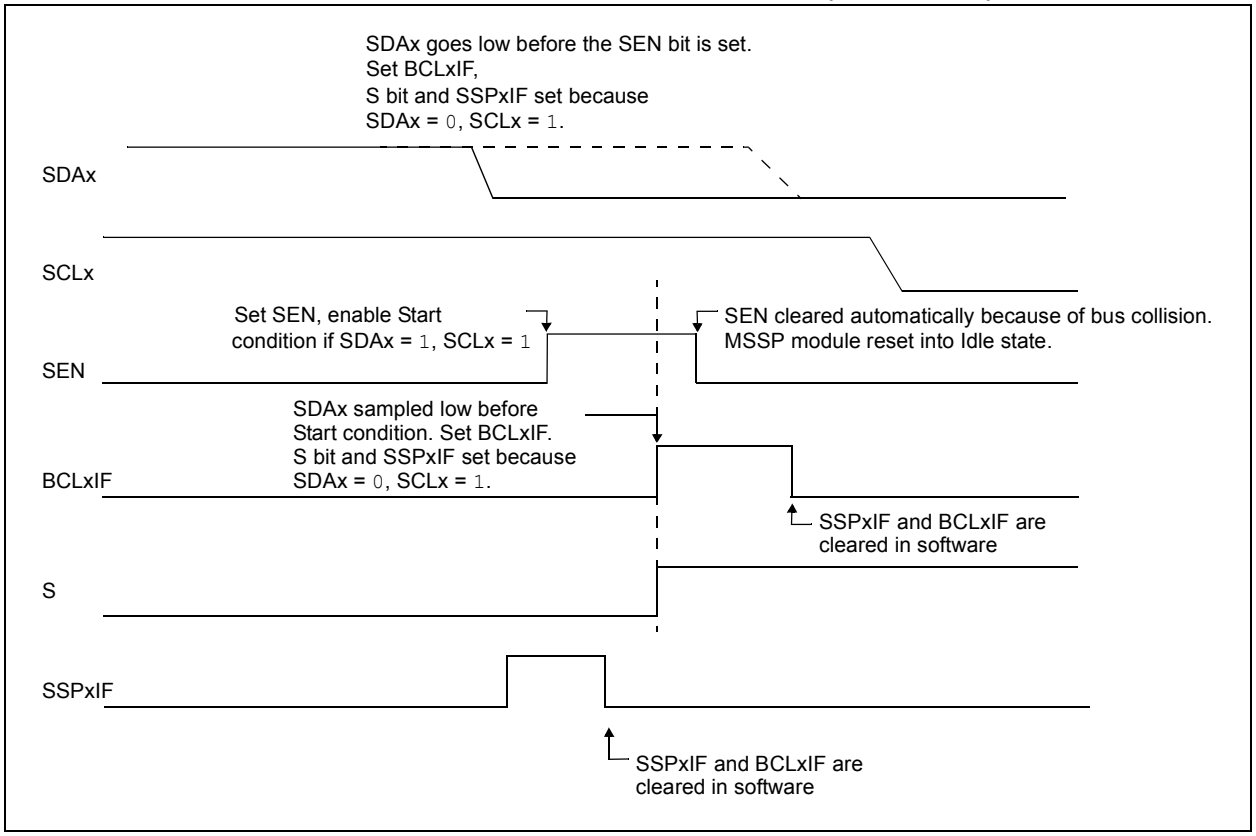
- the Start condition is aborted,
- the BCLxIF flag is set and
- the MSSP module is reset to its inactive state (Figure 19-28)

The Start condition begins with the SDAx and SCLx pins deasserted. When the SDAx pin is sampled high, the Baud Rate Generator is loaded from SSPxADD<6:0> and counts down to 0. If the SCLx pin is sampled low while SDAx is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDAx pin is sampled low during this count, the BRG is reset and the SDAx line is asserted early (Figure 19-30). If, however, a '1' is sampled on the SDAx pin, the SDAx pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0. If the SCLx pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCLx pin is asserted low.

Note:	The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDAx before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 19-28: BUS COLLISION DURING START CONDITION (SDAx ONLY)



21.1 A/D Acquisition Requirements

For the A/D Converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 21-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 21-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 21-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 21-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 21-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$
or
$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)$$

EQUATION 21-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$
$$T_{AMP} = 0.2\ \mu s$$
$$T_{COFF} = (\text{Temp} - 25°C)(0.02\ \mu s/°C)$$
$$(85°C - 25°C)(0.02\ \mu s/°C)$$
$$1.2\ \mu s$$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, T_{COFF} = 0 ms.

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)\ \mu s$$
$$-(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega)\ln(0.0004883)\ \mu s$$
$$1.05\ \mu s$$

$$T_{ACQ} = 0.2\ \mu s + 1.05\ \mu s + 1.2\ \mu s$$
$$2.45\ \mu s$$

26.0 INSTRUCTION SET SUMMARY

The PIC18F87J10 family of devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

26.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® instruction sets, while maintaining an easy migration from these PIC instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 26-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 26-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator, ‘f’, specifies which file register is to be used by the instruction. The destination designator, ‘d’, specifies where the result of the operation is to be placed. If ‘d’ is ‘0’, the result is placed in the WREG register. If ‘d’ is ‘1’, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator, ‘f’, represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 26-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The instruction set summary, shown in Table 26-2, lists the standard instructions recognized by the Microchip MPASM™ Assembler.

Section 26.1.1 “Standard Instruction Set” provides a description of each instruction.

TABLE 26-2: PIC18F87J50 FAMILY INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb LSb	Status Affected	Notes
LITERAL OPERATIONS						
ADDLW	k	Add Literal and WREG	1	0000 1111 kkkk kkkk	C, DC, Z, OV, N	
ANDLW	k	AND Literal with WREG	1	0000 1011 kkkk kkkk	Z, N	
IORLW	k	Inclusive OR Literal with WREG	1	0000 1001 kkkk kkkk	Z, N	
LFSR	f, k	Move Literal (12-bit) 2nd word to FSR(f) 1st word	2	1110 1110 00ff kkkk 1111 0000 kkkk kkkk	None	
MOVLB	k	Move Literal to BSR<3:0>	1	0000 0001 0000 kkkk	None	
MOVLW	k	Move Literal to WREG	1	0000 1110 kkkk kkkk	None	
MULLW	k	Multiply Literal with WREG	1	0000 1101 kkkk kkkk	None	
RETLW	k	Return with Literal in WREG	2	0000 1100 kkkk kkkk	None	
SUBLW	k	Subtract WREG from Literal	1	0000 1000 kkkk kkkk	C, DC, Z, OV, N	
XORLW	k	Exclusive OR Literal with WREG	1	0000 1010 kkkk kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS						
TBLRD*		Table Read	2	0000 0000 0000 1000	None	
TBLRD*+		Table Read with Post-Increment		0000 0000 0000 1001	None	
TBLRD*-		Table Read with Post-Decrement		0000 0000 0000 1010	None	
TBLRD+*		Table Read with Pre-Increment		0000 0000 0000 1011	None	
TBLWT*		Table Write	2	0000 0000 0000 1100	None	
TBLWT*+		Table Write with Post-Increment		0000 0000 0000 1101	None	
TBLWT*-		Table Write with Post-Decrement		0000 0000 0000 1110	None	
TBLWT+*		Table Write with Pre-Increment		0000 0000 0000 1111	None	

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a `NOP` unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

BNOV **Branch if Not Overflow**

Syntax: BNOV n

Operands: -128 ≤ n ≤ 127

Operation: if Overflow bit is '0',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0101	nnnn	nnnn

Description: If the Overflow bit is '0', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNOV Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If Overflow = 0;
PC = address `(Jump)`
If Overflow = 1;
PC = address `(HERE + 2)`

BNZ **Branch if Not Zero**

Syntax: BNZ n

Operands: -128 ≤ n ≤ 127

Operation: if Zero bit is '0',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0001	nnnn	nnnn

Description: If the Zero bit is '0', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

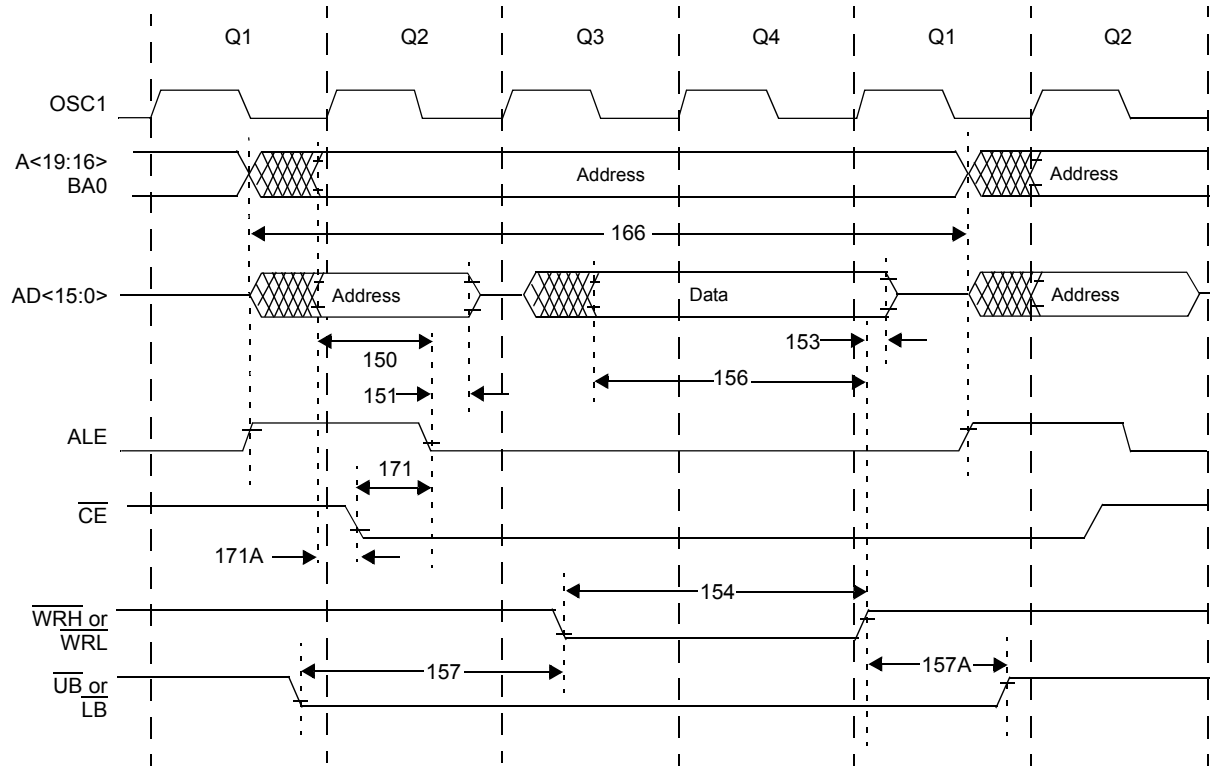
If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BNZ Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If Zero = 0;
PC = address `(Jump)`
If Zero = 1;
PC = address `(HERE + 2)`

FIGURE 28-7: PROGRAM MEMORY WRITE TIMING DIAGRAM

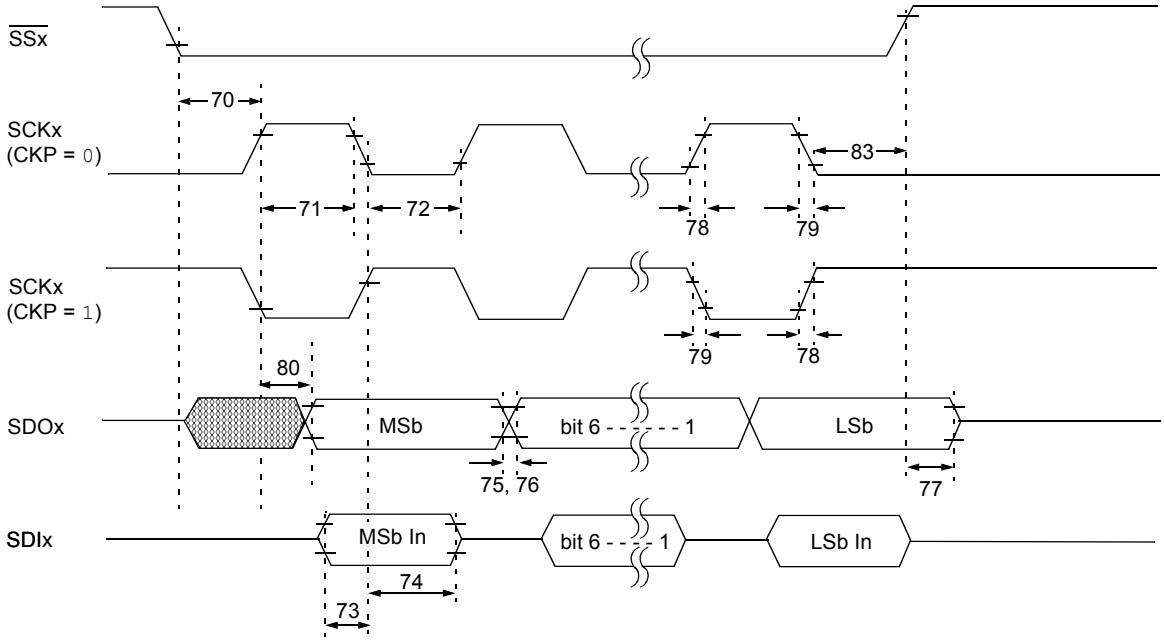


Operating Conditions: 2.0V < VCC < 3.6V, -40°C < TA < +125°C unless otherwise stated.

TABLE 28-12: PROGRAM MEMORY WRITE TIMING REQUIREMENTS

Param. No	Symbol	Characteristics	Min	Typ	Max	Units
150	TadV2alL	Address Out Valid to ALE ↓ (address setup time)	0.25 TCY – 10	—	—	ns
151	TalL2adl	ALE ↓ to Address Out Invalid (address hold time)	5	—	—	ns
153	TwrH2adl	$\overline{WRn}$ ↑ to Data Out Invalid (data hold time)	5	—	—	ns
154	TwrL	$\overline{WRn}$ Pulse Width	0.5 TCY – 5	0.5 TCY	—	ns
156	TadV2wrH	Data Valid before $\overline{WRn}$ ↑ (data setup time)	0.5 TCY – 10	—	—	ns
157	TbsV2wrL	Byte Select Valid before $\overline{WRn}$ ↓ (byte select setup time)	0.25 TCY	—	—	ns
157A	TwrH2bsI	$\overline{WRn}$ ↑ to Byte Select Invalid (byte select hold time)	0.125 TCY – 5	—	—	ns
166	TalH2alH	ALE ↑ to ALE ↑ (cycle time)	—	TCY	—	ns
171	TalH2csL	Chip Enable Active to ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD Valid to Chip Enable Active	—	—	10	ns

FIGURE 28-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)



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

TABLE 28-20: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ to Write to SSPxBUF		3 Tcy	—	ns	
71	TscH	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single byte	40	—	ns	**(Note 1)**
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single byte	40	—	ns	**(Note 1)**
73	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge		100	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge		100	—	ns	
75	TdoR	SDOx Data Output Rise Time		—	25	ns	
76	TdoF	SDOx Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance		10	50	ns	
78	TscR	SCKx Output Rise Time (Master mode)		—	25	ns	
79	TscF	SCKx Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ after SCKx Edge		1.5 Tcy + 40	—	ns	

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

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

APPENDIX A: REVISION HISTORY

Revision A (February 2007)

Original data sheet for the PIC18F87J10 family of devices.

Revision B (May 2007)

Updated electrical specification data.

Revision C (October 2009)

Removed “Preliminary” marking.

APPENDIX B: DEVICE DIFFERENCES

The differences between the devices listed in this data sheet are shown in Table B-1,

TABLE B-1: DEVICE DIFFERENCES BETWEEN PIC18F87J50 FAMILY MEMBERS

Features	PIC18F65J50	PIC18F66J50	PIC18F66J55	PIC18F67J50	PIC18F85J50	PIC18F86J50	PIC18F86J55	PIC18F87J50
Program Memory	32K	64K	96K	128K	32K	64K	96K	128K
Program Memory (Instructions)	16380	32764	49148	65532	16380	32764	49148	65532
I/O Ports (Pins)	Ports A, B, C, D, E, F, G				Ports A, B, C, D, E, F, G, H, J			
EMB	No				Yes			
10-Bit ADC Module	8 Input Channels				12 Input Channels			
Packages	64-Pin TQFP				80-Pin TQFP			

T