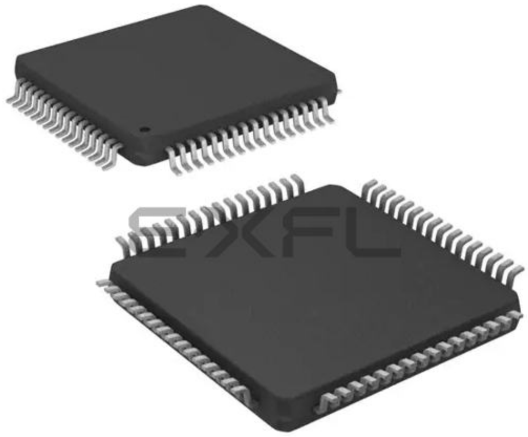
Microchip Technology - PIC18F66J10T-I/PT Datasheet



What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	50
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 11x10b
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/pic18f66j10t-i-pt

3.0 OSCILLATOR CONFIGURATIONS

3.1 Oscillator Types

The PIC18F87J10 family of devices can be operated in five different oscillator modes:

1. HS High-Speed Crystal/Resonator
2. HSPLL High-Speed Crystal/Resonator with Software PLL Control
3. EC External Clock with FOSC/4 Output
4. ECPLL External Clock with Software PLL Control
5. INTRC Internal 31 kHz Oscillator

Four of these are selected by the user by programming the FOSC<2:0> Configuration bits. The fifth mode (INTRC) may be invoked under software control; it can also be configured as the default mode on device Resets.

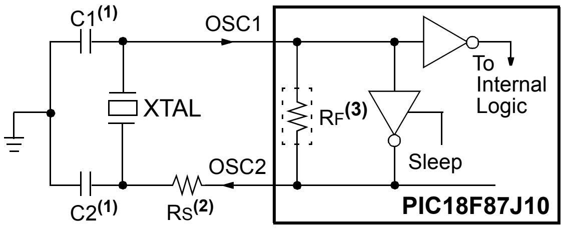
3.2 Crystal Oscillator/Ceramic Resonators (HS Modes)

In HS or HSPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 3-1 shows the pin connections.

The oscillator design requires the use of a parallel cut crystal.

Note: Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

FIGURE 3-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS OR HSPLL CONFIGURATION)



Note 1: See Table 3-1 and Table 3-2 for initial values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the oscillator mode chosen.

TABLE 3-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

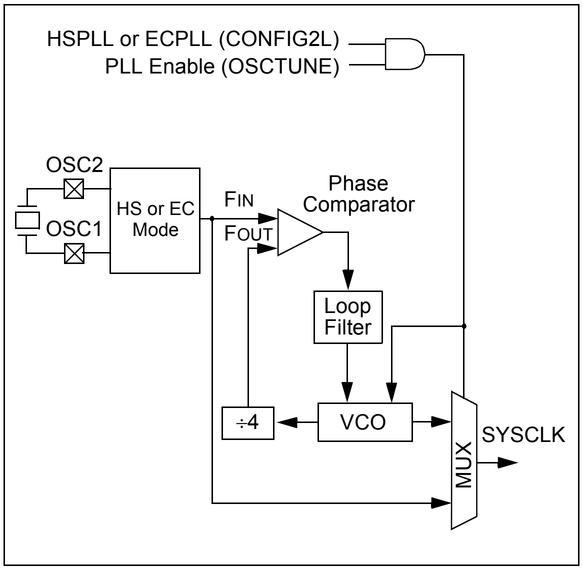
Typical Capacitor Values Used:			
Mode	**Freq.**	**OSC1**	**OSC2**
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF
Capacitor values are for design guidance only. These capacitors were tested with the resonators listed below for basic start-up and operation. **These values are not optimized**. Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. See the notes following Table 3-2 for additional information.			
Resonators Used:			
4.0 MHz			
8.0 MHz			
16.0 MHz			

3.4 PLL Frequency Multiplier

A Phase Locked Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals, or users who require higher clock speeds from an internal oscillator. For these reasons, the HSPLL and ECPLL modes are available.

The HSPLL and ECPLL modes provide the ability to selectively run the device at 4 times the external oscillating source to produce frequencies up to 40 MHz. The PLL is enabled by setting the PLLEN bit in the OSCTUNE register (Register 3-1).

FIGURE 3-4: PLL BLOCK DIAGRAM



REGISTER 3-1: OSCTUNE: PLL CONTROL REGISTER

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	PLLEN[1]	—	—	—	—	—	—
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

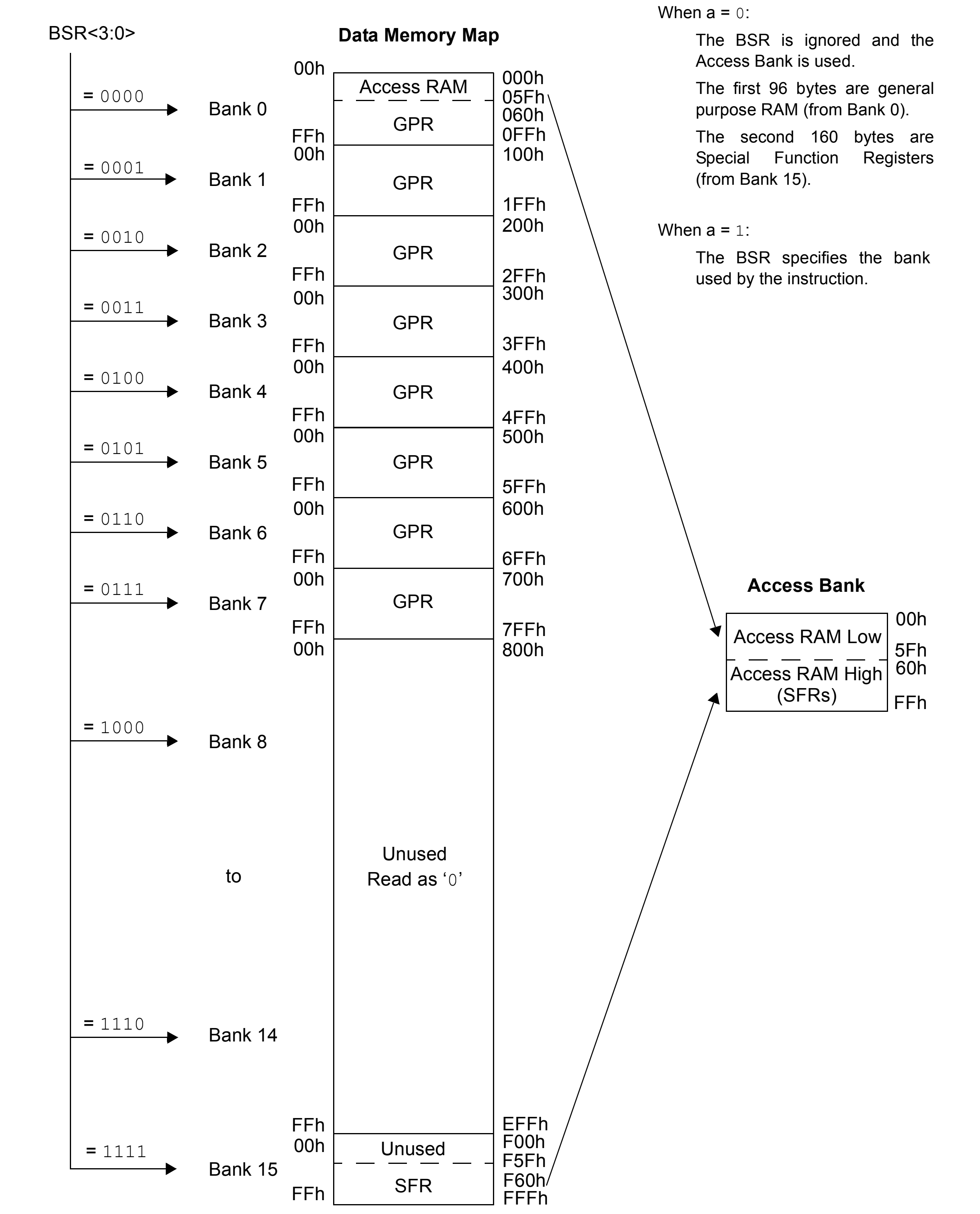
bit 7 **Unimplemented:** Read as ‘0’

bit 6 **PLLEN:** Frequency Multiplier PLL Enable bit[1]
1 = PLL enabled
0 = PLL disabled

bit 5-0 **Unimplemented:** Read as ‘0’

Note 1: Available only for ECPLL and HSPLL oscillator configurations; otherwise, this bit is unavailable and read as ‘0’.

FIGURE 6-7: DATA MEMORY MAP FOR PIC18FX5J10/X5J15/X6J10 DEVICES



When a = 0:

The BSR is ignored and the Access Bank is used.

The first 96 bytes are general purpose RAM (from Bank 0).

The second 160 bytes are Special Function Registers (from Bank 15).

When a = 1:

The BSR specifies the bank used by the instruction.

TABLE 6-4: REGISTER FILE SUMMARY (PIC18F87J10 FAMILY) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
TXREG1	EUSART1 Transmit Register								`xxxx xxxx`	55, 249, 250
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	`0000 0010`	55, 240
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 000x`	55, 241
EECON2	Program Memory Control Register 2 (not a physical register)								`---- ----`	55
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	`---0 x00-`	55
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	`1111 1111`	55, 123
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	`0000 0000`	55, 117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	`0000 0000`	55, 120
IPR2	OSCFIP	CMIP	—	—	BCL1IP	—	TMR3IP	CCP2IP	`11-- 1-11`	55, 121
PIR2	OSCFIF	CMIF	—	—	BCL1IF	—	TMR3IF	CCP2IF	`00-- 0-00`	55, 115
PIE2	OSCFIE	CMIE	—	—	BCL1IE	—	TMR3IE	CCP2IE	`00-- 0-00`	55, 120
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	`1111 1111`	55, 120
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	`0000 0000`	55, 114
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	`0000 0000`	55, 117
MEMCON[3]	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	`0-00 --00`	55, 96
OSCTUNE	—	PLLEN[4]	—	—	—	—	—	—	`-0-- ----`	33, 55
TRISJ[2]	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	`1111 1111`	56, 147
TRISH[2]	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	`1111 1111`	56, 145
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	`---1 1111`	56, 143
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	`1111 111-`	56, 141
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	`1111 1111`	56, 139
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	`1111 1111`	56, 136
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	`1111 1111`	56, 133
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	`1111 1111`	56, 130
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`--11 1111`	56, 127
LATJ[2]	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	`xxxx xxxx`	56, 147
LATH[2]	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	`xxxx xxxx`	56, 145
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	`---x xxxx`	56, 143
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	`xxxx xxx-`	56, 141
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	`xxxx xxxx`	56, 139
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	`xxxx xxxx`	56, 136
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	`xxxx xxxx`	56, 133
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	`xxxx xxxx`	56, 130
LATA	—	—	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	`--xx xxxx`	56, 127
PORTJ[2]	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	`xxxx xxxx`	56, 147
PORTH[2]	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	`0000 xxxx`	56, 145
PORTG	RDPU	REPU	RJPU[2]	RG4	RG3	RG2	RG1	RG0	`111x xxxx`	56, 143
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	`x000 000-`	56, 141
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	`xxxx xxxx`	56, 139
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	`xxxx xxxx`	56, 136
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	`xxxx xxxx`	56, 133
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	56, 130
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	`--0x 0000`	56, 127

Legend: `x` = unknown, `u` = unchanged, `-` = unimplemented, `q` = value depends on condition

Note 1: Bit 21 of the PC is only available in Serial Programming modes.

2: These bits and/or registers are only available in 80-pin devices; otherwise, they are unimplemented and read as '`0`'. Reset values are shown for 80-pin devices.

3: This register and its bits are not implemented in 64-pin devices. In 80-pin devices, the bits are unwritable and read as '`0`' in Microcontroller mode.

4: The PLLEN bit is available only when either ECPLL or HSPLL Oscillator modes are selected; otherwise, the bit is read as '`0`'.

5: Reset value is '`0`' when Two-Speed Start-up is enabled and '`1`' if disabled.

REGISTER 7-1: EECON1: EEPROM CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0
—	—	—	FREE	WRERR	WREN	WR	—
bit 7							bit 0

Legend:	S = Settable bit		
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 **FREE:** Flash Erase Enable bit

1 = Erase the program memory block addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
0 = Perform write-only

bit 3 **WRERR:** Flash Program Error Flag bit

1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation or an improper write attempt)
0 = The write operation completed

bit 2 **WREN:** Flash Program Write Enable bit

1 = Allows write cycles to Flash program memory
0 = Inhibits write cycles to Flash program memory

bit 1 **WR:** Write Control bit

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

bit 0 **Unimplemented:** Read as ‘0’

7.4 Erasing Flash Program Memory

The minimum erase block is 512 words or 1024 bytes. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 1024 bytes of program memory is erased. The Most Significant 12 bits of the TBLPTR<21:10> point to the block being erased. TBLPTR<9:0> are ignored.

The EECON1 register commands the erase operation. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for EECON2 must be used.

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

7.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer register with the address of the block being erased.
2. Set the WREN and FREE bits (EECON1<2,4>) to enable the erase operation.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the erase cycle.
7. The CPU will stall for duration of the erase for T_{IE} (see parameter D133B).
8. Re-enable interrupts.

EXAMPLE 7-2: ERASING FLASH PROGRAM MEMORY

```
                   MOVLW   CODE_ADDR_UPPER        ; load TBLPTR with the base
                   MOVWF   TBLPTRU                ; address of the memory block
                   MOVLW   CODE_ADDR_HIGH
                   MOVWF   TBLPTRH
                   MOVLW   CODE_ADDR_LOW
                   MOVWF   TBLPTRL
       ERASE_BLOCK
                   BSF     EECON1, WREN           ; enable write to memory
                   BSF     EECON1, FREE           ; enable Erase operation
                   BCF     INTCON, GIE            ; disable interrupts
    Required       MOVLW   55h
    Sequence       MOVWF   EECON2                 ; write 55h
                   MOVLW   0AAh
                   MOVWF   EECON2                 ; write 0AAh
                   BSF     EECON1, WR             ; start erase (CPU stall)
                   BSF     INTCON, GIE            ; re-enable interrupts
```

8.1 External Memory Bus Control

The operation of the interface is controlled by the MEMCON register (Register 8-1). This register is available in all program memory operating modes except Microcontroller mode. In this mode, the register is disabled and cannot be written to.

The EBDIS bit (MEMCON<7>) controls the operation of the bus and related port functions. Clearing EBDIS enables the interface and disables the I/O functions of the ports, as well as any other functions multiplexed to those pins. Setting the bit enables the I/O ports and other functions, but allows the interface to override everything else on the pins when an external memory operation is required. By default, the external bus is always enabled and disables all other I/O.

The operation of the EBDIS bit is also influenced by the program memory mode being used. This is discussed in more detail in **Section 8.5 "Program Memory Modes and the External Memory Bus"**.

The WAIT bits allow for the addition of wait states to external memory operations. The use of these bits is discussed in **Section 8.3 "Wait States"**.

The WM bits select the particular operating mode used when the bus is operating in 16-Bit Data Width mode. These are discussed in more detail in **Section 8.6 "16-Bit Data Width Modes"**. These bits have no effect when an 8-Bit Data Width mode is selected.

REGISTER 8-1: MEMCON: EXTERNAL MEMORY BUS CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 7							bit 0

Legend:	S = Settable bit		
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 **EBDIS**: External Bus Disable bit

1 = External bus enabled when microcontroller accesses external memory; otherwise, all external bus drivers are mapped as I/O ports
0 = External bus always enabled, I/O ports are disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5-4 **WAIT<1:0>:** Table Reads and Writes Bus Cycle Wait Count bits

11 = Table reads and writes will wait 0 TCY
10 = Table reads and writes will wait 1 TCY
01 = Table reads and writes will wait 2 TCY
00 = Table reads and writes will wait 3 TCY

bit 3-2 **Unimplemented:** Read as ‘0

bit 1-0 **WM<1:0>:** TBLWT Operation with 16-Bit Data Bus Width Select bits

1x = Word Write mode: TABLAT0 and TABLAT1 word output; $\overline{\text{WRH}}$ active when TABLAT1 written
01 = Byte Select mode: TABLAT data copied on both MSB and LSB; $\overline{\text{WRH}}$ and ($\overline{\text{UB}}$ or $\overline{\text{LB}}$) will activate
00 = Byte Write mode: TABLAT data copied on both MSB and LSB; $\overline{\text{WRH}}$ or $\overline{\text{WRL}}$ will activate

8.7 8-Bit Mode

In 8-Bit Data Width mode, the external memory bus operates only in Multiplexed mode; that is, data shares the 8 Least Significant bits of the address bus.

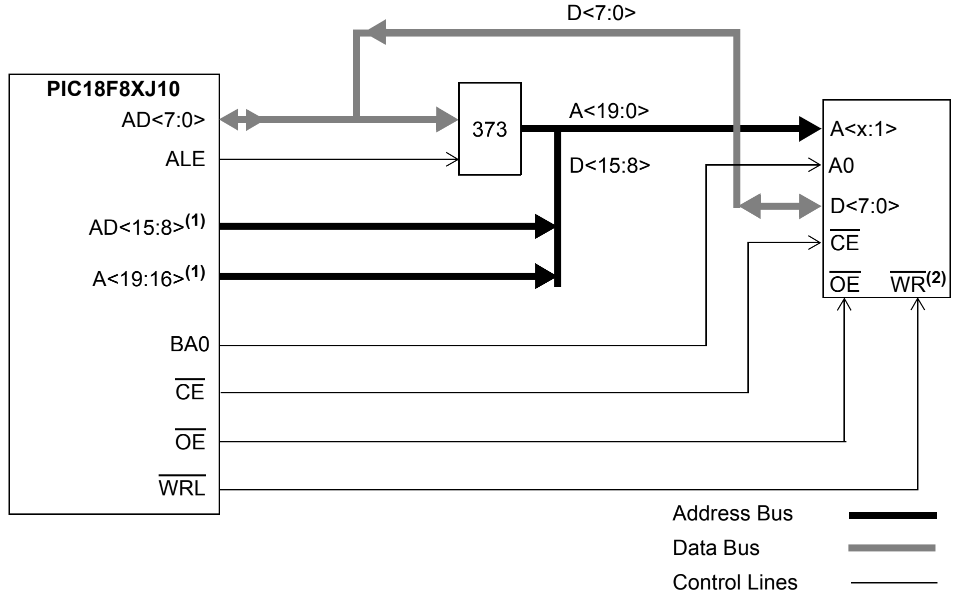
Figure 8-6 shows an example of 8-Bit Multiplexed mode for 80-pin devices. This mode is used for a single 8-bit memory connected for 16-bit operation. The instructions will be fetched as two 8-bit bytes on a shared data/address bus. The two bytes are sequentially fetched within one instruction cycle (TCY). Therefore, the designer must choose external memory devices according to timing calculations based on 1/2 TCY (2 times the instruction rate). For proper memory speed selection, glue logic propagation delay times must be considered, along with setup and hold times.

The Address Latch Enable (ALE) pin indicates that the address bits, AD<15:0>, are available on the external memory interface bus. The Output Enable signal ($\overline{\text{OE}}$) will enable one byte of program memory for a portion of the instruction cycle, then BA0 will change and the second byte will be enabled to form the 16-bit instruction word. The Least Significant bit of the address, BA0, must be connected to the memory devices in this mode. The Chip Enable signal ($\overline{\text{CE}}$) is active at any time that the microcontroller accesses external memory, whether reading or writing. It is inactive (asserted high) whenever the device is in Sleep mode.

This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a `TBLWT` instruction cycle, the TABLAT data is presented on the upper and lower bytes of the AD<15:0> bus. The appropriate level of the BA0 control line is strobed on the LSb of the TBLPTR.

FIGURE 8-6: 8-BIT MULTIPLEXED MODE EXAMPLE



Note 1: Upper order address bits are only used for 20-bit address width. The upper AD byte is used for all address widths except 8-bit.

2: This signal only applies to table writes. See **Section 7.1 "Table Reads and Table Writes"**.

TABLE 19-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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	53
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	55
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	55
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	55
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	55
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	55
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	55
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	56
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	56
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	56
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								54
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	54, 57
SSPxSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	54, 57
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								57

Legend: Shaded cells are not used by the MSSP module in SPI mode.

20.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

All members of the PIC18F87J10 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full duplex) with:
  - Auto-Wake-up on character reception
  - Auto-Baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1) and PORTG (RG1/TX2/CK2 and RG2/RX2/DT2), respectively. In order to configure these pins as an EUSART:

- For EUSART1:
  - bit, SPEN (RCSTA1<7>), must be set (= `1`)
  - bit, TRISC<7>, must be set (= `1`)
  - bit, TRISC<6>, must be cleared (= `0`) for Asynchronous and Synchronous Master modes
  - bit, TRISC<6>, must be set (= `1`) for Synchronous Slave mode
- For EUSART2:
  - bit, SPEN (RCSTA2<7>), must be set (= `1`)
  - bit, TRISG<2>, must be set (= `1`)
  - bit, TRISG<1>, must be cleared (= `0`) for Asynchronous and Synchronous Master modes
  - bit, TRISC<6> must be set (= `1`) for Synchronous Slave mode

Note:	The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of each Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTAx)
- Receive Status and Control (RCSTAx)
- Baud Rate Control (BAUDCONx)

These are detailed on the following pages in Register 20-1, Register 20-2 and Register 20-3, respectively.

Note:	Throughout this section, references to register and bit names that may be associated with a specific EUSART module are referred to generically by the use of ‘x’ in place of the specific module number. Thus, “RCSTAx” might refer to the Receive Status register for either EUSART1 or EUSART2.

21.2 Selecting and Configuring Automatic Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/$\overline{\text{DONE}}$ bit is set.

When the GO/$\overline{\text{DONE}}$ bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/$\overline{\text{DONE}}$ bit. This occurs when the ACQT<2:0> bits (ADCON2<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/$\overline{\text{DONE}}$ bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/$\overline{\text{DONE}}$ bit.

In either case, when the conversion is completed, the GO/$\overline{\text{DONE}}$ bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

21.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable.

There are seven possible options for TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible but greater than the minimum TAD (see parameter 130 in Table 27-27 for more information).

Table 21-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 21-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS<2:0>	
2 TOSC	000	2.86 MHz
4 TOSC	100	5.71 MHz
8 TOSC	001	11.43 MHz
16 TOSC	101	22.86 MHz
32 TOSC	010	40.0 MHz
64 TOSC	110	40.0 MHz
RC(2)	x11	1.00 MHz(1)

Note 1: The RC source has a typical TAD time of 4 μs.

2: For device frequencies above 1 MHz, the device must be in Sleep mode for the entire conversion or the A/D accuracy may be out of specification.

21.4 Configuring Analog Port Pins

The ADCON1, TRISA, TRISF and TRISH registers control the operation of the A/D port pins. The port pins needed as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<3:0> bits and the TRIS bits.

Note 1: When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will be accurately converted.

2: Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

CPFSGT — Compare f with W, Skip if f > W

Syntax: CPFSGT f {,a}

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

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	010a	ffff	ffff

Description: Compares the contents of data memory location ‘f’ to the contents of the W by performing an unsigned subtraction.

If the contents of ‘f’ are greater than the contents of WREG, then the fetched instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank.

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 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE      CPFSGT REG, 0
NGREATER  :
GREATER   :
```

```
Before Instruction
    PC          =    Address (HERE)
    W           =    ?
After Instruction
    If REG      >    W;
        PC      =    Address (GREATER)
    If REG      ≤    W;
        PC      =    Address (NGREATER)
```

CPFSLT — Compare f with W, Skip if f < W

Syntax: CPFSLT f {,a}

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

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location ‘f’ to the contents of W by performing an unsigned subtraction.

If the contents of ‘f’ are less than the contents of W, then the fetched instruction is discarded and a `NOP` is executed instead, making this a two-cycle instruction.

If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE     CPFSLT REG, 1
NLESS    :
LESS     :
```

```
Before Instruction
    PC        =    Address (HERE)
    W         =    ?
After Instruction
    If REG    <    W;
    PC        =    Address (LESS)
    If REG    ≥    W;
    PC        =    Address (NLESS)
```

25.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, the PIC18F87J10 family of devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment Indirect and Indexed Addressing operations and the implementation of Indexed Literal Offset Addressing for many of the standard PIC18 instructions.

The additional features of the extended instruction set are disabled by default on unprogrammed devices. Users must properly set or clear the XINST Configuration bit during programming to enable or disable these features.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers, or use them for Indexed Addressing. Two of the instructions, `ADDFSR` and `SUBFSR`, each have an additional special instantiation for using FSR2. These versions (`ADDULNK` and `SUBULNK`) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- function pointer invocation
- software Stack Pointer manipulation
- manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 25-3. Detailed descriptions are provided in **Section 25.2.2 “Extended Instruction Set”**. The opcode field descriptions in Table 25-1 (page 294) apply to both the standard and extended PIC18 instruction sets.

Note:	The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

25.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of Indexed Addressing, it is enclosed in square brackets (“[ ]”). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 25.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**.

Note:	In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces (“{ }”).

TABLE 25-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word	Status Affected
				MSb LSb	
ADDFSR	f, k	Add Literal to FSR	1	`1110 1000 ffkk kkkk`	None
ADDULNK	k	Add Literal to FSR2 and Return	2	`1110 1000 11kk kkkk`	None
CALLW		Call Subroutine using WREG	2	`0000 0000 0001 0100`	None

MOVSF	z_s, f_d	Move z_s (source) to 1st word
		f_d (destination) 2nd word
			2	`1110 1011 0zzz zzzz`
				`1111 ffff ffff ffff`
					None

MOVSS	z_s, z_d	Move z_s (source) to 1st word
		z_d (destination) 2nd word
			2	`1110 1011 1zzz zzzz`
				`1111 xxxx xzzz zzzz`
					None

PUSHL	k	Store Literal at FSR2, Decrement FSR2	1	`1110 1010 kkkk kkkk`	None
SUBFSR	f, k	Subtract Literal from FSR	1	`1110 1001 ffkk kkkk`	None
SUBULNK	k	Subtract Literal from FSR2 and Return	2	`1110 1001 11kk kkkk`	None

MOVSS — Move Indexed to Indexed

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \leq z_s \leq 127$
$0 \leq z_d \leq 127$

Operation: $((\text{FSR2}) + z_s) \rightarrow ((\text{FSR2}) + z_d)$

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	zzzz_s
2nd word (dest.)	1111	xxxx	xzzz	zzzz_d

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The `MOVSS` instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a `NOP`.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: `MOVSS [05h], [06h]`

Before Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	11h

After Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	33h

PUSHL — Store Literal at FSR2, Decrement FSR2

Syntax: PUSHL k

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow (\text{FSR2})$,
$\text{FSR2} - 1 \rightarrow \text{FSR2}$

Status Affected: None

Encoding:

1111	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: `PUSHL 08h`

Before Instruction

FSR2H:FSR2L	=	01ECh
Memory (01ECh)	=	00h

After Instruction

FSR2H:FSR2L	=	01EBh
Memory (01ECh)	=	08h

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J10 Family (Industrial)

PIC18F87J10 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	Device	Typ	Max	Units	Conditions	
	Power-Down Current (IPD)[1]					
	All devices	27	69	μA	-40°C	VDD = 2.0V[5] (Sleep mode)
		43	69	μA	+25°C	
		121	149	μA	+85°C	
	All devices	49	104	μA	-40°C	VDD = 2.5V[5] (Sleep mode)
		69	104	μA	+25°C	
		166	184	μA	+85°C	
	All devices	75	203	μA	-40°C	VDD = 3.3V[6] (Sleep mode)
		100	203	μA	+25°C	
		140	289	μA	+85°C	

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS, and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

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

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

4: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

5: ENVREG tied to VSS, voltage regulator disabled.

6: ENVREG tied to VDD, voltage regulator enabled.

27.2 DC Characteristics: Power-Down and Supply Current PIC18F87J10 Family (Industrial) (Continued)

PIC18F87J10 Family (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (I_{DD})(2,3)						
	All devices	1.8	3.27	mA	-40°C	V_{DD} = 2.0V(5)	F_{OSC} = 1 MHz (**PRI_IDLE** mode, EC oscillator)
		1.8	3.27	mA	+25°C		
		1.9	3.27	mA	+85°C		
	All devices	4.0	5.57	mA	-40°C	V_{DD} = 2.5V(5)	
		3.7	5.57	mA	+25°C		
		3.5	5.57	mA	+85°C		
	All devices	4.2	5.97	mA	-40°C	V_{DD} = 3.3V(6)	
		4.0	5.97	mA	+25°C		
		3.8	5.97	mA	+85°C		
	All devices	2.4	4.47	mA	-40°C	V_{DD} = 2.0V(5)	F_{OSC} = 4 MHz (**PRI_IDLE** mode, EC oscillator)
		2.4	4.47	mA	+25°C		
		2.5	4.47	mA	+85°C		
	All devices	4.7	6.97	mA	-40°C	V_{DD} = 2.5V(5)	
		4.4	6.97	mA	+25°C		
		4.8	6.97	mA	+85°C		
	All devices	5.1	7.47	mA	-40°C	V_{DD} = 3.3V(6)	
		4.9	7.47	mA	+25°C		
		4.8	7.47	mA	+85°C		
	All devices	13.4	18.7	mA	-40°C	V_{DD} = 2.5V(5)	F_{OSC} = 40 MHz (**PRI_IDLE** mode, EC oscillator)
		13.0	18.7	mA	+25°C		
		13.0	18.7	mA	+85°C		
	All devices	14.8	19.7	mA	-40°C	V_{DD} = 3.3V(6)	
		14.4	19.7	mA	+25°C		
		14.5	19.7	mA	+85°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS}, and all features that add delta current disabled (such as WDT, Timer1 oscillator, BOR, etc.).

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

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be estimated by the formula $I_r = V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

4: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

5: ENVREG tied to V_{SS}, voltage regulator disabled.

6: ENVREG tied to V_{DD}, voltage regulator enabled.

27.3 DC Characteristics: PIC18F87J10 Family (Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	V_{IL}	**Input Low Voltage**				
		All I/O Ports:				
D030		with TTL buffer	V_{SS}	0.15 V_{DD}	V	V_{DD} < 3.3V
D030A			—	0.8	V	3.3V ≤ V_{DD} ≤ 3.6V
D031		with Schmitt Trigger Buffer	V_{SS}	0.2 V_{DD}	V	
D031A		RC3 and RC4	V_{SS}	0.3 V_{DD}	V	I^2C™ enabled
D031B			V_{SS}	0.8	V	SMBus enabled
D032		$\overline{\text{MCLR}}$	V_{SS}	0.2 V_{DD}	V	
D033		OSC1	V_{SS}	0.3 V_{DD}	V	HS, HSPLL modes
D033A		OSC1	V_{SS}	0.2 V_{DD}	V	EC, ECPLL modes[1]
D034		T1CKI	V_{SS}	0.3	V	
	V_{IH}	**Input High Voltage**				
		I/O Ports with non 5.5V Tolerance:[4]				
D040		with TTL Buffer	0.25 V_{DD} + 0.8V	V_{DD}	V	V_{DD} < 3.3V
D040A			2.0	V_{DD}	V	3.3V ≤ V_{DD} ≤ 3.6V
D041		with Schmitt Trigger Buffer	0.8 V_{DD}	V_{DD}	V	
		I/O Ports with 5.5V Tolerance:[4]				
D041A		RC3 and RC4	0.7 V_{DD}	V_{DD}	V	I^2C enabled
D041B			2.1	V_{DD}	V	SMBus enabled
Dxxx		with TTL Buffer	0.25 V_{DD} + 0.8V	5.5	V	V_{DD} < 3.3V
DxxxA			2.0	5.5	V	3.3V ≤ V_{DD} ≤ 3.6V
Dxxx		with Schmitt Trigger Buffer	0.8 V_{DD}	5.5	V	
D042		$\overline{\text{MCLR}}$	0.8 V_{DD}	V_{DD}	V	
D043		OSC1	0.7 V_{DD}	V_{DD}	V	HS, HSPLL modes
D043A		OSC1	0.8 V_{DD}	V_{DD}	V	EC, ECPLL modes
D044		T1CKI	1.6	V_{DD}	V	
	I_{IL}	**Input Leakage Current**[2,3]				
D060		I/O Ports with non 5.5V Tolerance:[4]	—	±1	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, Pin at high-impedance
D060A		I/O Ports with 5.5V Tolerance:[4]	—	±1	μA	V_{ss} ≤ V_{PIN} ≤ 5.5V. Pin at high-impedance
D061		$\overline{\text{MCLR}}$	—	±1	μA	V_{ss} ≤ V_{PIN} ≤ V_{DD}
D063		OSC1	—	±5	μA	V_{ss} ≤ V_{PIN} ≤ V_{DD}

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC® device be driven with an external clock while in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Refer to Table 11-2 for the pins that have corresponding tolerance limits.

TABLE 27-7: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.7V TO 3.6V)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	10	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	40	MHz	
F12	t_{rc}	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13	ΔCLK	CLKO Stability (Jitter)	-2	—	+2	%	

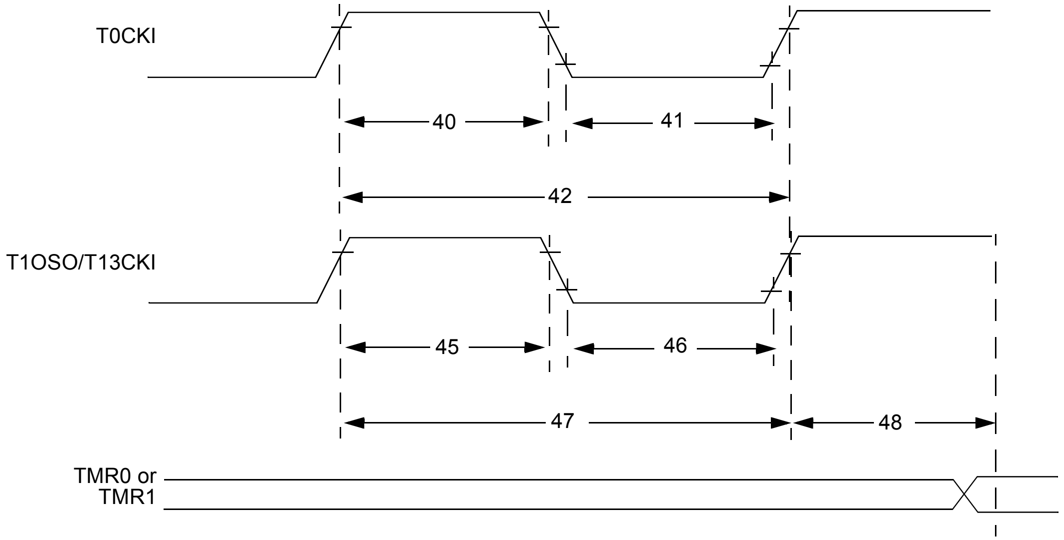
† Data in “Typ” column is at 5V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 27-8: AC CHARACTERISTICS: INTERNAL RC ACCURACY PIC18F87J10 FAMILY (INDUSTRIAL)

Param No.	Characteristic	Min	Typ	Max	Units	Conditions
	INTRC Accuracy @ Freq = 31 kHz(1)	21.88	—	40.63	kHz	-40°C to +85°C, VDD = 2.0-3.3V

Note 1: INTRC frequency after calibration. Change of INTRC frequency as VDD changes.

FIGURE 27-11: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 27-3 for load conditions.

TABLE 27-13: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
40	TT0H	T0CKI High Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
41	TT0L	T0CKI Low Pulse Width	No prescaler	0.5 TCY + 20	—	ns	
			With prescaler	10	—	ns	
42	TT0P	T0CKI Period	No prescaler	TCY + 10	—	ns	
			With prescaler	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4,..., 256)
45	TT1H	T13CKI High Time	Synchronous, no prescaler	0.5 TCY + 20	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
46	TT1L	T13CKI Low Time	Synchronous, no prescaler	0.5 TCY + 5	—	ns	
			Synchronous, with prescaler	10	—	ns	
			Asynchronous	30	—	ns	
47	TT1P	T13CKI Input Period	Synchronous	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	ns	
	FT1	T13CKI Oscillator Input Frequency Range		DC	50	kHz	
48	TCKE2TMRI	Delay from External T13CKI Clock Edge to Timer Increment		2 TOSC	7 TOSC	—	

A.1 Power Requirement Differences

The most significant difference between the PIC18F87J10 and PIC18F8722 device families is the power requirements. PIC18F87J10 devices are designed on a smaller process; this results in lower maximum voltage and higher leakage current.

The operating voltage range for PIC18F87J10 devices is 2.0V to 3.6V. In addition, these devices have split power requirements: one for the core logic and one for the I/O. One of the VDD pins is separated for the core logic supply (VDDCORE). This pin has specific voltage and capacitor requirements as described in **Section 27.0 "Electrical Characteristics"**.

A.2 Pin Differences

There are several differences in the pinouts between the PIC18F87J10 and the PIC18F8722 families:

- Input voltage tolerance
- Output current capabilities
- Available I/O

Pins on the PIC18F87J10 that have digital only input capability will tolerate voltages up to 5.5V and are thus tolerant to voltages above VDD. Table 11-1 in **Section 11.0 "I/O Ports"** contains the complete list.

In addition to input differences, there are output differences as well. PIC18F87J10 devices have three classes of pin output current capability: high, medium and low. Not all I/O pins can source or sink equal levels of current. Only PORTB and PORTC support the 25 mA source/sink capability that is supported by all output pins on the PIC18F8722. Table 11-2 in **Section 11.0 "I/O Ports"** contains the complete list of output capabilities.

There are additional differences in how some pin functions are implemented on PIC18F87J10 devices. First, the OSC1/OSC2 oscillator pins are strictly dedicated to the external oscillator function; there is no option to re-allocate these pins to I/O (RA6 or RA7) as on PIC18F8722 devices. Second, the $\overline{\text{MCLR}}$ pin is dedicated only to $\overline{\text{MCLR}}$ and cannot be configured as an input (RG5). Finally, RF0 does not exist on PIC18F87J10 devices.

All of these pin differences (including power pin differences) should be accounted for when making a conversion between PIC18F8722 and PIC18F87J10 devices.

A.3 Oscillator Differences

PIC18F8722 devices have a greater range of oscillator options than PIC18F87J10 devices. The latter family is limited primarily to operating modes that support HS and EC oscillators.

In addition, the PIC18F87J10 has an internal RC oscillator with only a fixed 32 kHz output. The higher frequency RC modes of the PIC18F8722 family are not available.

Both device families have an internal PLL. For the PIC18F87J10 family, however, the PLL must be enabled in software.

The clocking differences should be considered when making a conversion between the PIC18F8722 and PIC18F87J10 device families.

A.4 Peripherals

Peripherals must also be considered when making a conversion between the PIC18F87J10 and the PIC18F8722 families:

- **External Memory Bus:** The external memory bus on the PIC18F87J10 does not support Microcontroller mode; however, it does support external address offset.
- **A/D Converter:** There are only 15 channels on PIC18F87J10 devices. The converters for these devices also require a calibration step prior to normal operation.
- **Data EEPROM:** PIC18F87J10 devices do not have this module.
- **BOR:** PIC18F87J10 devices do not have a programmable BOR. Simple brown-out capability is provided through the use of the internal voltage regulator.
- **LVD:** PIC18F87J10 devices do not have this module.