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	EBI/EMI, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3930 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 15x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f87j11t-i-pt

REGISTER 3-2: OSCTUNE: OSCILLATOR TUNING REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled)
0 = 31 kHz device clock derived from INTRC 31 kHz oscillator

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

bit 5-0 **TUN<5:0>:** Fast RC Oscillator (INTOSC) Frequency Tuning bits
011111 = Maximum frequency
•
•
000001
000000 = Center frequency. Fast RC Oscillator is running at the calibrated frequency.
111111
•
•
100000 = Minimum frequency

3.3 Clock Sources and Oscillator Switching

Essentially, PIC18F87J11 family devices have three independent clock sources:

- Primary oscillators
- Secondary oscillators
- Internal oscillator

The **primary oscillators** can be thought of as the main device oscillators. These are any external oscillators connected to the OSC1 and OSC2 pins, and include the External Crystal and Resonator modes, and the External Clock modes. If selected by the FOSC<2:0> Configuration bits, the internal oscillator block (either the 31 kHz INTRC or the 8 MHz INTOSC source) may be considered a primary oscillator. The particular mode is defined by the FOSCx Configuration bits. The details of these modes are covered in **Section 3.4 "External Oscillator Modes"**.

The **secondary oscillators** are external clock sources that are not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode. PIC18F87J11 family devices offer the Timer1 oscillator as a secondary oscillator source. This oscillator, in all power-managed modes, is often the time base for functions, such as a Real-Time Clock (RTC). The Timer1 oscillator is discussed in greater detail in **Section 14.0 "Timer1 Module"**.

In addition to being a primary clock source in some circumstances, the **internal oscillator** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor. The internal oscillator block is discussed in more detail in **Section 3.5 "Internal Oscillator Block"**.

The PIC18F87J11 family includes features that allow the device clock source to be switched from the main oscillator, chosen by device configuration, to one of the alternate clock sources. When an alternate clock source is enabled, various power-managed operating modes are available.

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS(4) (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets, CM Resets	Wake-up via WDT or Interrupt
INDF2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PREINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
FSR2H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- 0000	---- uuuu
FSR2L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F6XJ1X	PIC18F8XJ1X	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6XJ1X	PIC18F8XJ1X	0110 q100	0110 q100	0110 q10u
REFOCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 0000	u-uu uuuu	u-uu uuuu
CM1CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	0001 1111	uuuu uuuu
CM2CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	0001 1111	uuuu uuuu
RCON(4)	PIC18F6XJ1X	PIC18F8XJ1X	0-11 1100	0-qq qquu	u-qq qquu
TMR1H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON1	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---u uuuu	---u uuuu
TMR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON2	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
T1CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	u0uu uuuu	uuuu uuuu
ODCON3	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
TMR2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PADCFG1	PIC18F6XJ1X	PIC18F8XJ1X	---- ---0	---- ---u	---- ---u
PR2	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	1111 1111
MEMCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 --00	0-00 --00	u-uu --uu
T2CON	PIC18F6XJ1X	PIC18F8XJ1X	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1MSK	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	uuuu uuuu	uuuu uuuu
SSP1STAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu

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

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

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

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

4: See Table 5-2 for Reset value for specific conditions.

6.3.4 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy more than the top half of Bank 15 (F5Ah to FFFh). A list of these registers is given inTable 6-3, Table 6-4 and Table 6-5.

The SFRs can be classified into two sets: those associated with the “core” device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and Interrupt registers are described in their respective chapters, while the ALU’s STATUS register is described later in this section. Registers related to the operation of the peripheral features are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as ‘0’s

Note:	Addresses, F5Ah through F5Fh, are not part of the Access Bank. These registers must always be accessed using the Bank Select Register. Addresses, F40h to F59h, are not implemented and are not accessible to the user.

TABLE 6-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18F87J11 FAMILY DEVICES

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(1)	FBFh	ECCP1AS	F9Fh	IPR1	F7Fh	SPBRGH1	F5Fh	PMDIN2H
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	ECCP1DEL	F9Eh	PIR1	F7Eh	BAUDCON1	F5Eh	PMDIN2L
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	CCPR1H	F9Dh	PIE1	F7Dh	SPBRGH2	F5Dh	PMEH
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	CCPR1L	F9Ch	RCSTA2	F7Ch	BAUDCON2	F5Ch	PMEL
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	CCP1CON	F9Bh	OSCTUNE	F7Bh	TMR3H	F5Bh	PMSTATH
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP2AS	F9Ah	TRISJ(2)	F7Ah	TMR3L	F5Ah	PMSTATL
FF9h	PCL	FD9h	FSR2L	FB9h	ECCP2DEL	F99h	TRISH(2)	F79h	T3CON	F59h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR2H	F98h	TRISG	F78h	TMR4	F58h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCPR2L	F97h	TRISF	F77h	PR4(3)	F57h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCP2CON	F96h	TRISE	F76h	T4CON	F56h	—
FF5h	TABLAT	FD5h	T0CON	FB5h	ECCP3AS	F95h	TRISD	F75h	CCPR4H	F55h	—
FF4h	PRODH	FD4h	—	FB4h	ECCP3DEL	F94h	TRISC	F74h	CCPR4L	F54h	—
FF3h	PRODL	FD3h	OSCCON(3)	FB3h	CCPR3H	F93h	TRISB	F73h	CCP4CON	F53h	—
FF2h	INTCON	FD2h	CM1CON	FB2h	CCPR3L	F92h	TRISA	F72h	CCPR5H	F52h	—
FF1h	INTCON2	FD1h	CM2CON	FB1h	CCP3CON	F91h	LATJ(2)	F71h	CCPR5L	F51h	—
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	LATH(2)	F70h	CCP5CON	F50h	—
FEFh	INDF0(1)	FCFh	TMR1H(3)	FAFh	RCREG1	F8Fh	LATG	F6Fh	SSP2BUF	F4Fh	—
FEEh	POSTINC0(1)	FCEh	TMR1L(3)	FAEh	TXREG1	F8Eh	LATF	F6Eh	SSP2ADD	F4Eh	—
FEDh	POSTDEC0(1)	FCDh	T1CON(3)	FADh	TXSTA1	F8Dh	LATE	F6Dh	SSP2STAT	F4Dh	—
FECh	PREINC0(1)	FCCh	TMR2(3)	FACh	RCSTA1	F8Ch	LATD	F6Ch	SSP2CON1	F4Ch	—
FEBh	PLUSW0(1)	FCBh	PR2(3)	FABh	SPBRG2	F8Bh	LATC	F6Bh	SSP2CON2	F4Bh	—
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	CMSTAT	F4Ah	—
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	PMADDRH(4)	F49h	—
FE8h	WREG	FC8h	SSP1ADD	FA8h	TXSTA2	F88h	PORTJ(2)	F68h	PMADDRL(4)	F48h	—
FE7h	INDF1(1)	FC7h	SSP1STAT	FA7h	EECON2	F87h	PORTH(2)	F67h	PMDIN1H	F47h	—
FE6h	POSTINC1(1)	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	PMDIN1L	F46h	—
FE5h	POSTDEC1(1)	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	PMCONH	F45h	—
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	PMCONL	F44h	—
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	PMMODEH	F43h	—
FE2h	FSR1H	FC2h	ADCON0(3)	FA2h	IPR2	F82h	PORTC	F62h	PMMODEL	F42h	—
FE1h	FSR1L	FC1h	ADCON1(3)	FA1h	PIR2	F81h	PORTB	F61h	PMDOUT2H	F41h	—
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	PMDOUT2L	F40h	—

Note 1: This is not a physical register.
2: This register is not available on 64-pin devices.
3: This register shares the same address with another register (see Table 6-4 for alternate register).
4: The PMADDRH/L and PMDOUT1H/L register pairs share the same address. PMADDR is used in Master modes and PMDOUT1 is used in Slave modes.
5: Addresses, F40 to F59, are not implemented and are not accessible to the user.

8.0 EXTERNAL MEMORY BUS

Note: The External Memory Bus (EMB) is not implemented on 64-pin devices.

The External Memory Bus allows the device to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program or data memory. It supports both 8 and 16-Bit Data Width modes and three address widths of up to 20 bits.

The bus is implemented with 28 pins, multiplexed across four I/O ports. Three ports (PORTD, PORTE and PORTH) are multiplexed with the address/data bus for a total of 20 available lines, while PORTJ is multiplexed with the bus control signals.

A list of the pins and their functions is provided in Table 8-1.

TABLE 8-1: PIC18F87J11 FAMILY EXTERNAL BUS – I/O PORT FUNCTIONS

Name	Port	Bit	External Memory Bus Function
RD0/AD0	PORTD	0	Address Bit 0 or Data Bit 0
RD1/AD1	PORTD	1	Address Bit 1 or Data Bit 1
RD2/AD2	PORTD	2	Address Bit 2 or Data Bit 2
RD3/AD3	PORTD	3	Address Bit 3 or Data Bit 3
RD4/AD4	PORTD	4	Address Bit 4 or Data Bit 4
RD5/AD5	PORTD	5	Address Bit 5 or Data Bit 5
RD6/AD6	PORTD	6	Address Bit 6 or Data Bit 6
RD7/AD7	PORTD	7	Address Bit 7 or Data Bit 7
RE0/AD8	PORTE	0	Address Bit 8 or Data Bit 8
RE1/AD9	PORTE	1	Address Bit 9 or Data Bit 9
RE2/AD10	PORTE	2	Address Bit 10 or Data Bit 10
RE3/AD11	PORTE	3	Address Bit 11 or Data Bit 11
RE4/AD12	PORTE	4	Address Bit 12 or Data Bit 12
RE5/AD13	PORTE	5	Address Bit 13 or Data Bit 13
RE6/AD14	PORTE	6	Address Bit 14 or Data Bit 14
RE7/AD15	PORTE	7	Address Bit 15 or Data Bit 15
RH0/A16	PORTH	0	Address Bit 16
RH1/A17	PORTH	1	Address Bit 17
RH2/A18	PORTH	2	Address Bit 18
RH3/A19	PORTH	3	Address Bit 19
RJ0/ALE	PORTJ	0	Address Latch Enable (ALE) Control Pin
RJ1/$\overline{OE}$	PORTJ	1	Output Enable ($\overline{OE}$) Control Pin
RJ2/$\overline{WRL}$	PORTJ	2	Write Low ($\overline{WRL}$) Control Pin
RJ3/$\overline{WRH}$	PORTJ	3	Write High ($\overline{WRH}$) Control Pin
RJ4/BA0	PORTJ	4	Byte Address Bit 0 (BA0)
RJ5/$\overline{CE}$	PORTJ	5	Chip Enable ($\overline{CE}$) Control Pin
RJ6/$\overline{LB}$	PORTJ	6	Lower Byte Enable ($\overline{LB}$) Control Pin
RJ7/$\overline{UB}$	PORTJ	7	Upper Byte Enable ($\overline{UB}$) Control Pin

Note: For the sake of clarity, only I/O port and external bus assignments are shown here. One or more additional multiplexed features may be available on some pins.

10.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits.

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 10-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(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 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts (if GIEH = 1)
0 = Disables all low-priority peripheral interrupts

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

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

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

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

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

bit 0 **RBIF:** RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

Note 1: A mismatch condition will continue to set this bit. Reading PORTB, and then waiting one additional instruction cycle, will end the mismatch condition and allow the bit to be cleared.

TABLE 11-10: PORTD FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD0/AD0/ PMD0	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	AD0(2)	x	O	DIG	External Memory Interface, Address/Data Bit 0 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 0 input.(1)
	PMD0(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RD1/AD1/ PMD1	RD1	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	AD1(2)	x	O	DIG	External Memory Interface, Address/Data bit 1 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 1 input.(1)
	PMD1(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RD2/AD2/ PMD2	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	AD2(2)	x	O	DIG	External Memory Interface, Address/Data Bit 2 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 2 input.(1)
	PMD2(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RD3/AD3/ PMD3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	AD3(2)	x	O	DIG	External Memory Interface, Address/Data Bit 3 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 3 input.(1)
	PMD3(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
RD4/AD4/ PMD4/SDO2	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	AD4(2)	x	O	DIG	External Memory Interface, Address/Data Bit 4 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 4 input.(1)
	PMD4(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SDO2	0	O	DIG	SPI data output (MSSP2 module); takes priority over port data.
RD5/AD5/ PMD5/SDI2/ SDA2	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	AD5(2)	x	O	DIG	External Memory Interface, Address/Data Bit 5 output.(1)
		x	I	TTL	External Memory Interface, Data Bit 5 input.(1)
	PMD5(3)	x	O	DIG	Parallel Master Port data out.
		x	I	TTL	Parallel Master Port data input.
	SDI2	1	I	ST	SPI data input (MSSP2 module).
	SDA2	1	O	DIG	I^2C™ data output (MSSP2 module); takes priority over port data.
		1	I	ST	I^2C data input (MSSP2 module); input type depends on module setting.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: External Memory Interface I/O takes priority over all other digital and PMP I/O.

2: These bits are available on 80-pin devices only.

3: Default configuration for PMP (PMPMX Configuration bit = 1).

FIGURE 12-25: READ TIMING, 16-BIT MULTIPLEXED DATA, FULLY MULTIPLEXED 16-BIT ADDRESS

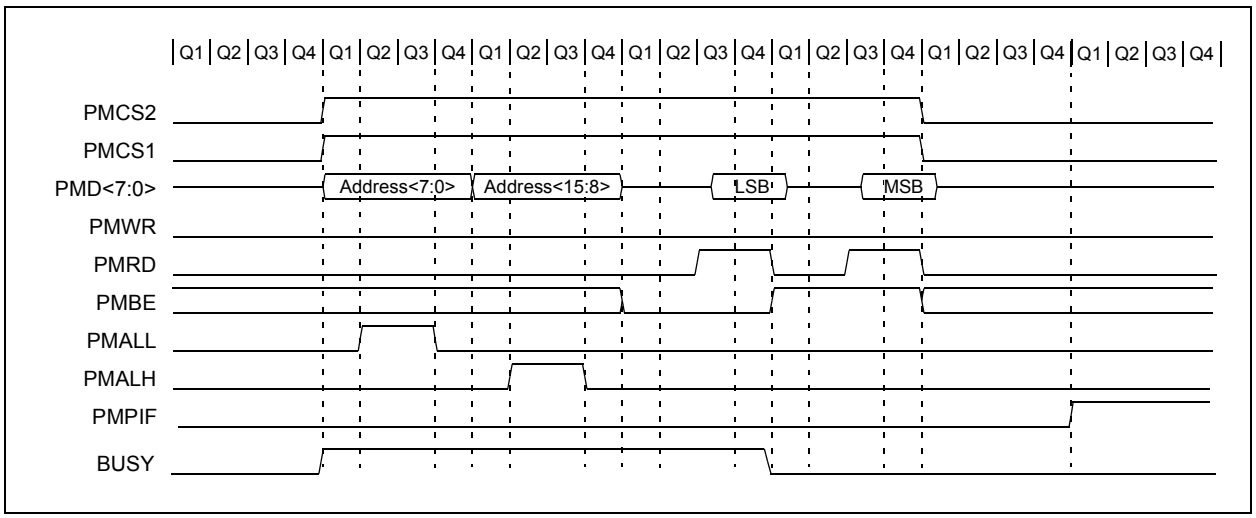
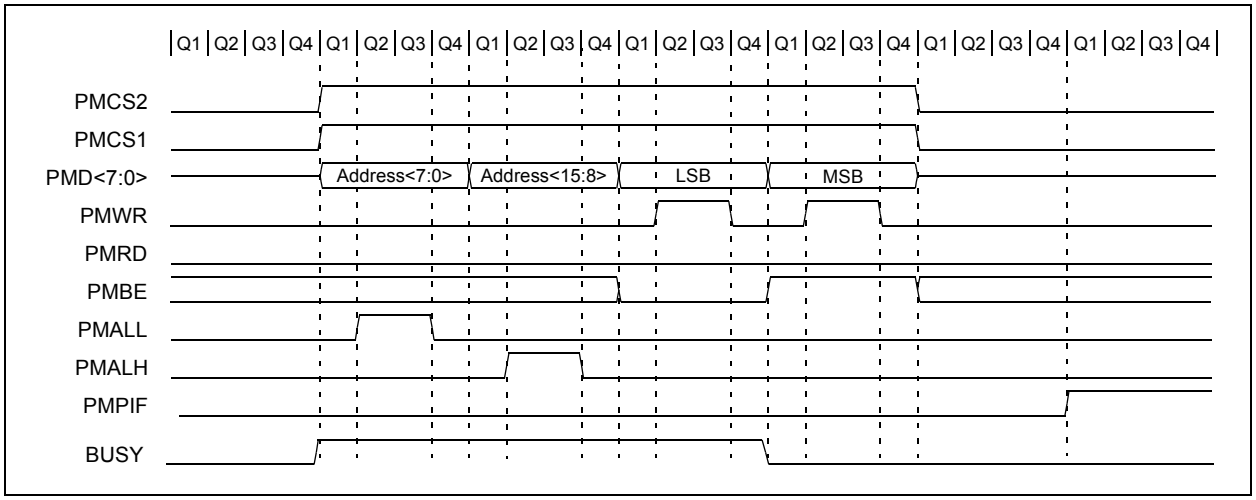


FIGURE 12-26: WRITE TIMING, 16-BIT MULTIPLEXED DATA, FULLY MULTIPLEXED 16-BIT ADDRESS



16.0 TIMER3 MODULE

The Timer3 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on ECCPx Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 16-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 16-2.

The Timer3 module is controlled through the T3CON register (Register 16-1). It also selects the clock source options for the CCP and ECCP modules; see **Section 18.1.1 "CCP Modules and Timer Resources"** for more information.

REGISTER 16-1: T3CON: TIMER3 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
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 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3CCP<2:1>:** Timer3 and Timer1 to ECCPx/CCPx Enable bits
11 = Timer3 and Timer4 are the clock sources for all ECCPx/CCPx modules
10 = Timer3 and Timer4 are the clock sources for ECCP3, CCP4 and CCP5; Timer1 and Timer2 are the clock sources for ECCP1 and ECCP2
01 = Timer3 and Timer4 are the clock sources for ECCP2, ECCP3, CCP4 and CCP5; Timer1 and Timer2 are the clock sources for ECCP1
00 = Timer1 and Timer2 are the clock sources for all ECCPx/CCPx modules

bit 5-4 **T3CKPS<1:0>**: Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the device clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Does not synchronize external clock input
0 = Synchronizes external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

16.1 Timer3 Operation

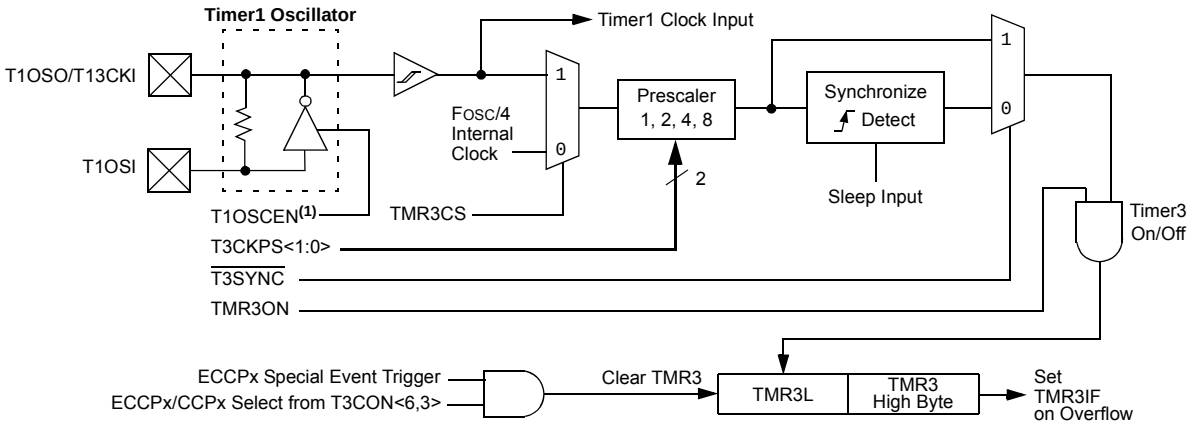
Timer3 can operate in one of three modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle (FOSC/4). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

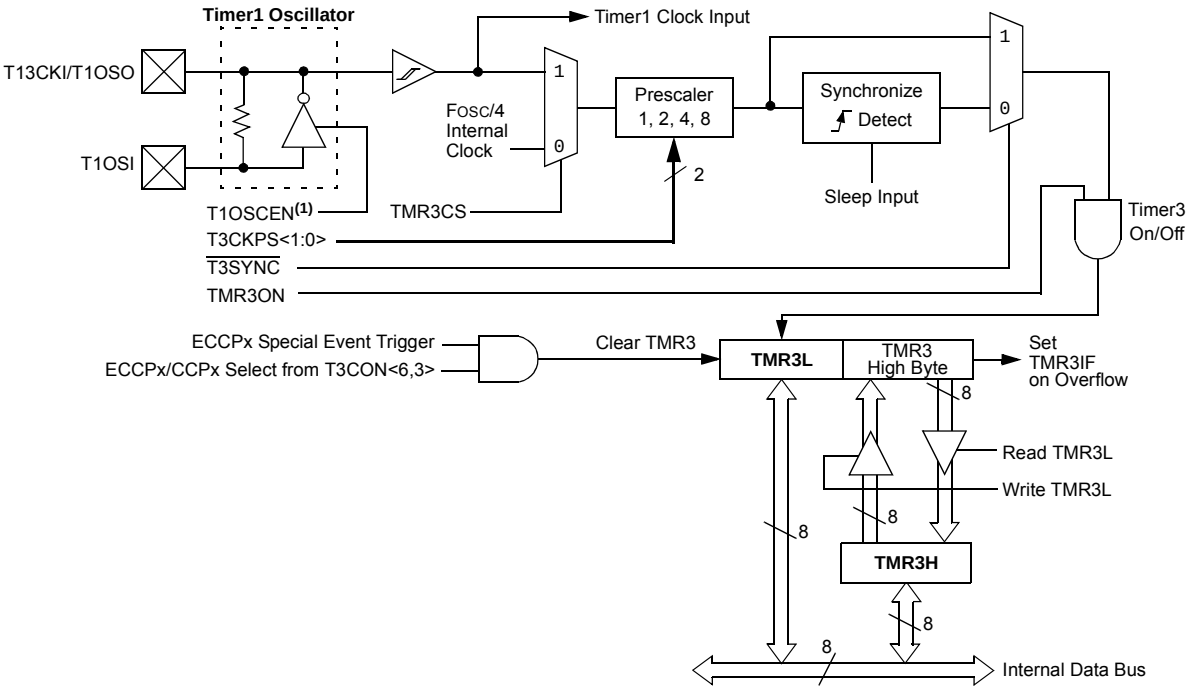
As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

FIGURE 16-1: TIMER3 BLOCK DIAGRAM



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

FIGURE 16-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

19.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The PWM duty cycle is calculated by the following equation:

EQUATION 19-2:

$$\text{PWM Duty Cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \bullet T_{OSC} \bullet (\text{TMR2 Prescale Value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time but the duty cycle value is not copied into CCPR1H until a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation. When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or two bits of the TMR2 prescaler, the ECCP1 pin is cleared. The maximum PWM resolution (bits) for a given PWM frequency is given by Equation 19-3.

EQUATION 19-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

Note: If the PWM duty cycle value is longer than the PWM period, the ECCP1 pin will not be cleared.

19.4.3 PWM OUTPUT CONFIGURATIONS

The P1M1:P1M0 bits in the CCP1CON register allow one of four configurations:

- Single Output
- Half-Bridge Output
- Full-Bridge Output, Forward mode
- Full-Bridge Output, Reverse mode

The Single Output mode is the standard PWM mode discussed in **Section 19.4 "Enhanced PWM Mode"**. The Half-Bridge and Full-Bridge Output modes are covered in detail in the sections that follow.

The general relationship of the outputs in all configurations is summarized in Figure 19-2.

TABLE 19-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	FFh	FFh	FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	6.58

FIGURE 20-16: I²C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)

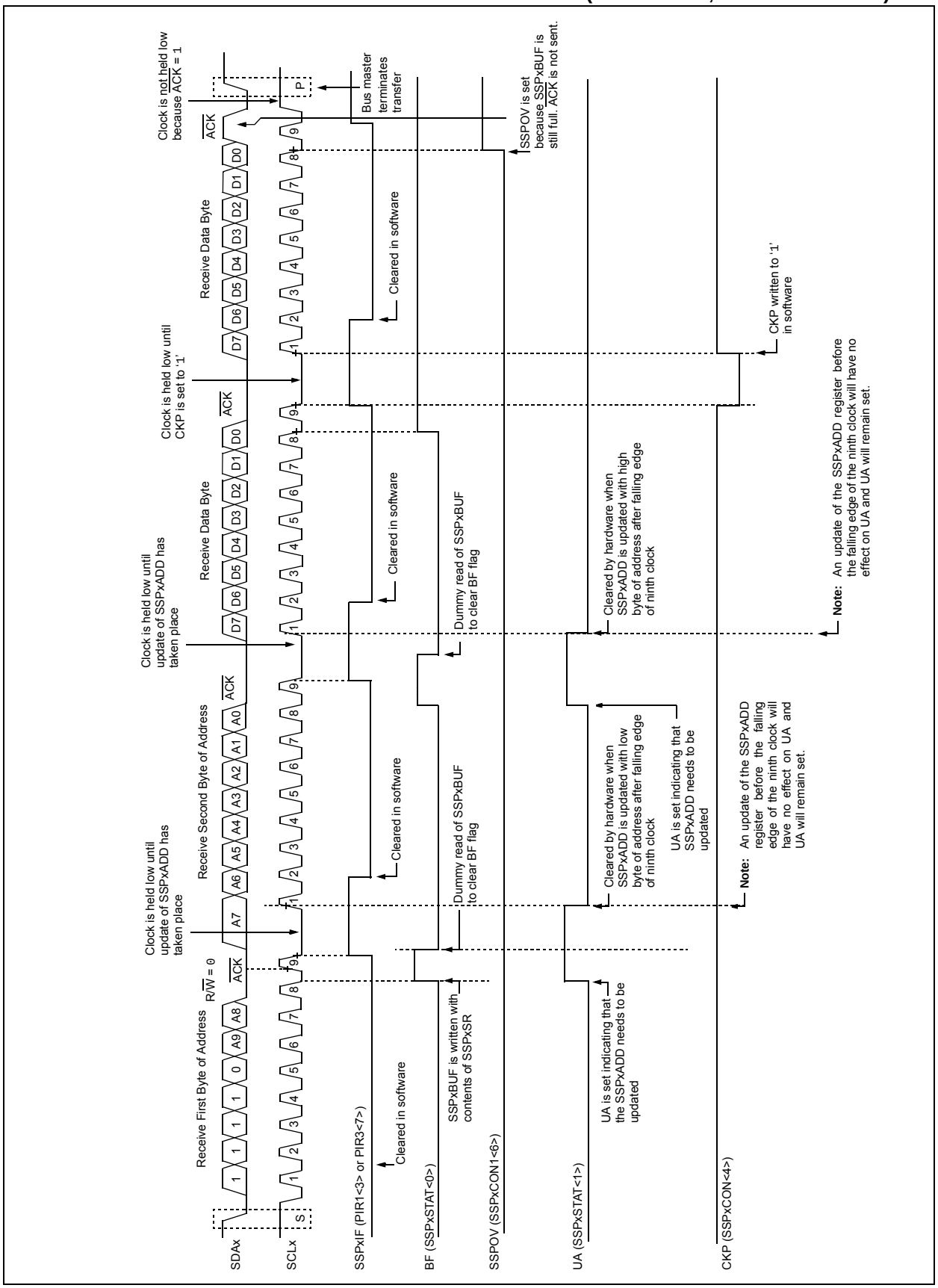
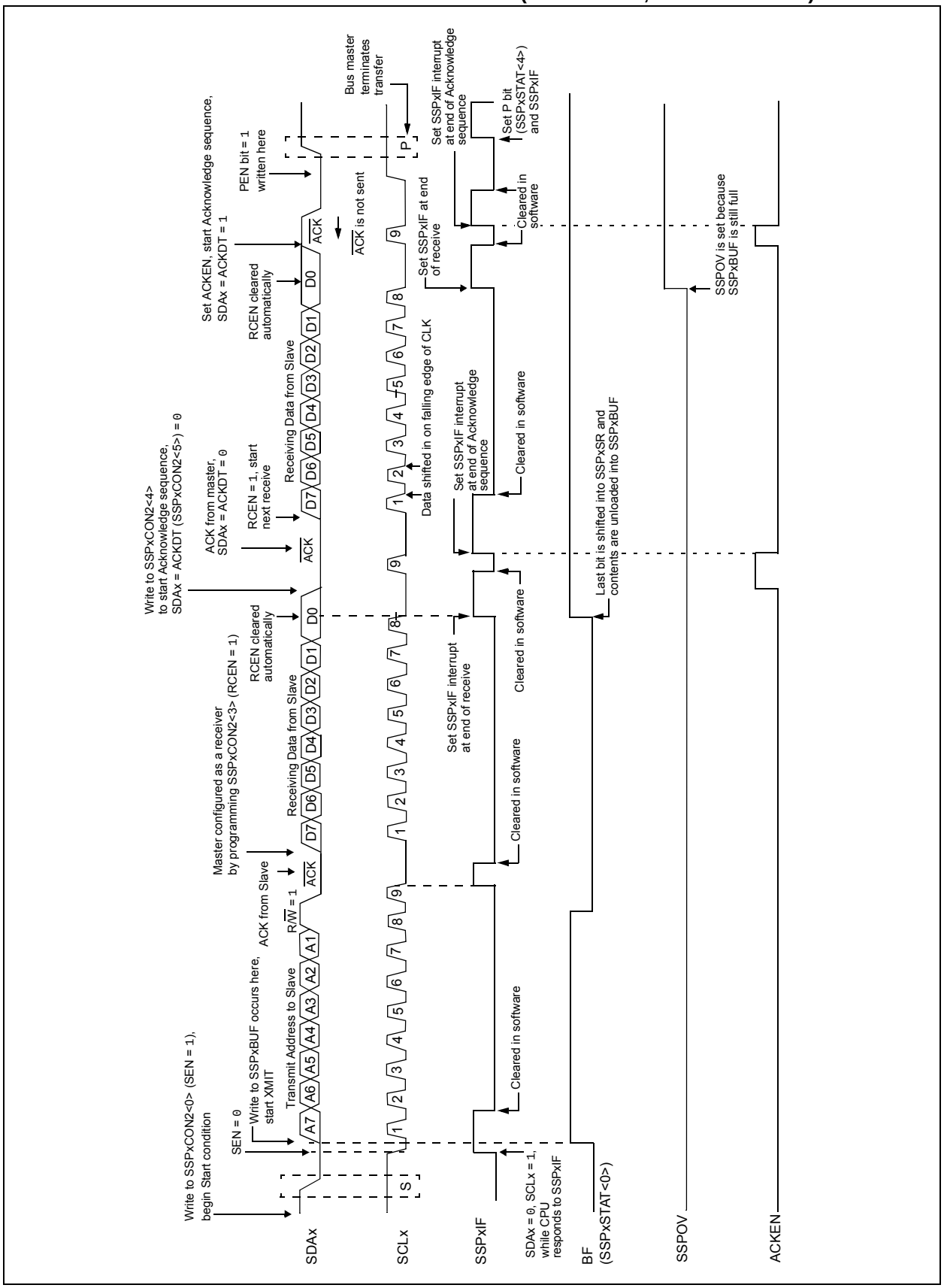


FIGURE 20-24: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



REGISTER 21-1: TXSTAx: EUSARTx TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care.
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-Bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit is enabled
0 = Transmit is disabled

bit 4 **SYNC:** EUSARTx Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Sends Sync Break on the next transmission (cleared by hardware upon completion)
0 = Sync Break transmission has completed
Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR is empty
0 = TSR is full

bit 0 **TX9D:** 9th bit of Transmit Data
This can be an address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

EXAMPLE 21-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, and 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGHx:SPBRGx] + 1))

Solving for SPBRGHx:SPBRGx:

X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

TABLE 21-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	63
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	63
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	65
SPBRGHx	EUSARTx Baud Rate Generator Register High Byte								65
SPBRGx	EUSARTx Baud Rate Generator Register Low Byte								65

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

TABLE 26-2: PIC18F87J11 FAMILY INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word MSb LSb	Status Affected	Notes
BYTE-ORIENTED OPERATIONS					
ADDWF f, d, a	Add WREG and f	1	0010 01da ffff ffff	C, DC, Z, OV, N	1, 2
ADDWFC f, d, a	Add WREG and Carry bit to f	1	0010 00da ffff ffff	C, DC, Z, OV, N	1, 2
ANDWF f, d, a	AND WREG with f	1	0001 01da ffff ffff	Z, N	1,2
CLRF f, a	Clear f	1	0110 101a ffff ffff	Z	2
COMF f, d, a	Complement f	1	0001 11da ffff ffff	Z, N	1, 2
CPFSEQ f, a	Compare f with WREG, Skip =	1 (2 or 3)	0110 001a ffff ffff	None	4
CPFSGT f, a	Compare f with WREG, Skip >	1 (2 or 3)	0110 010a ffff ffff	None	4
CPFSLT f, a	Compare f with WREG, Skip <	1 (2 or 3)	0110 000a ffff ffff	None	1, 2
DECF f, d, a	Decrement f	1	0000 01da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010 11da ffff ffff	None	1, 2, 3, 4
DCFSNZ f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100 11da ffff ffff	None	1, 2
INCF f, d, a	Increment f	1	0010 10da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011 11da ffff ffff	None	4
INFSNZ f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100 10da ffff ffff	None	1, 2
IORWF f, d, a	Inclusive OR WREG with f	1	0001 00da ffff ffff	Z, N	1, 2
MOVF f, d, a	Move f	1	0101 00da ffff ffff	Z, N	1
MOVFF f_s, f_d	Move f_s (source) to 1st word f_d (destination) 2nd word	2	1100 ffff ffff ffff 1111 ffff ffff ffff	None	
MOVWF f, a	Move WREG to f	1	0110 111a ffff ffff	None	
MULWF f, a	Multiply WREG with f	1	0000 001a ffff ffff	None	1, 2
NEGF f, a	Negate f	1	0110 110a ffff ffff	C, DC, Z, OV, N	
RLCF f, d, a	Rotate Left f through Carry	1	0011 01da ffff ffff	C, Z, N	1, 2
RLNCF f, d, a	Rotate Left f (No Carry)	1	0100 01da ffff ffff	Z, N	
RRCF f, d, a	Rotate Right f through Carry	1	0011 00da ffff ffff	C, Z, N	
RRNCF f, d, a	Rotate Right f (No Carry)	1	0100 00da ffff ffff	Z, N	
SETF f, a	Set f	1	0110 100a ffff ffff	None	1, 2
SUBFWB f, d, a	Subtract f from WREG with Borrow	1	0101 01da ffff ffff	C, DC, Z, OV, N	
SUBWF f, d, a	Subtract WREG from f	1	0101 11da ffff ffff	C, DC, Z, OV, N	1, 2
SUBWFB f, d, a	Subtract WREG from f with Borrow	1	0101 10da ffff ffff	C, DC, Z, OV, N	
SWAPF f, d, a	Swap Nibbles in f	1	0011 10da ffff ffff	None	4
TSTFSZ f, a	Test f, Skip if 0	1 (2 or 3)	0110 011a ffff ffff	None	1, 2
XORWF f, d, a	Exclusive OR WREG with f	1	0001 10da ffff ffff	Z, N	

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.

GOTO **Unconditional Branch**

Syntax: GOTO k

Operands: $0 \le k \le 1048575$

Operation: $k \rightarrow PC<20:1>$

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	1111	k_7kkk	$kkkk_0$
2nd word(k<19:8>)	1111	$k_{19}kkk$	kkkk	$kkkk_8$

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	No operation	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: GOTO THERE

After Instruction
PC = Address (THERE)

INCF **Increment f**

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

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

Operation: (f) + 1 → dest

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

Encoding:

0010	10da	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

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 $f \le 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: INCF CNT, 1, 0

Before Instruction
CNT = FFh
Z = 0
C = ?
DC = ?
After Instruction
CNT = 00h
Z = 1
C = 1
DC = 1

LFSR **Load FSR**

Syntax: LFSR f, k

Operands: $0 \le f \le 2$
$0 \le k \le 4095$

Operation: $k \rightarrow FSRf$

Status Affected: None

Encoding:

1110 1111	1110 0000	00ff $k_7$kkk	k_{11}kkk kkkk

Description: The 12-bit literal 'k' is loaded into the file select register pointed to by 'f'.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSRfH
Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSRfL

Example: `LFSR 2, 3ABh`

After Instruction
FSR2H = 03h
FSR2L = ABh

MOVF **Move f**

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

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

Operation: $f \rightarrow dest$

Status Affected: N, Z

Encoding:

0101	00da	ffff	ffff

Description: The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'. Location 'f' can be anywhere in the 256-byte bank.

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 $f \le 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVF REG, 0, 0`

Before Instruction
REG = 22h
W = FFh
After Instruction
REG = 22h
W = 22h

26.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note: Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing (**Section 6.6.1 "Indexed Addressing with Literal Offset"**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank (a = 0) or in a GPR bank designated by the BSR (a = 1). When the extended instruction set is enabled and a = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward-compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 26.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands"**).

Although the Indexed Literal Offset mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

26.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument 'f' in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value 'k'. As already noted, this occurs only when 'f' is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[ ]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within the brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled), when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM Assembler.

The destination argument 'd' functions as before.

In the latest versions of the MPASM Assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, /y, or the PE directive in the source listing.

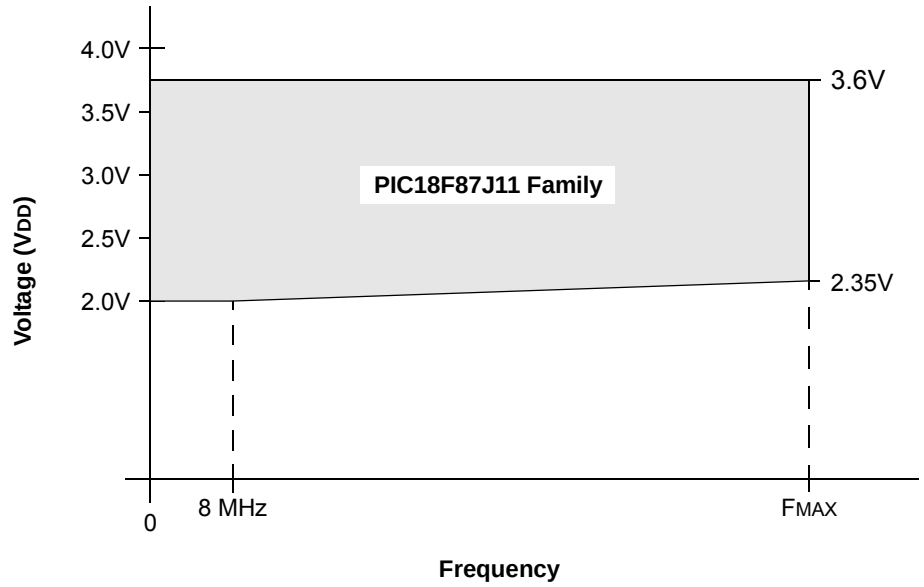
26.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

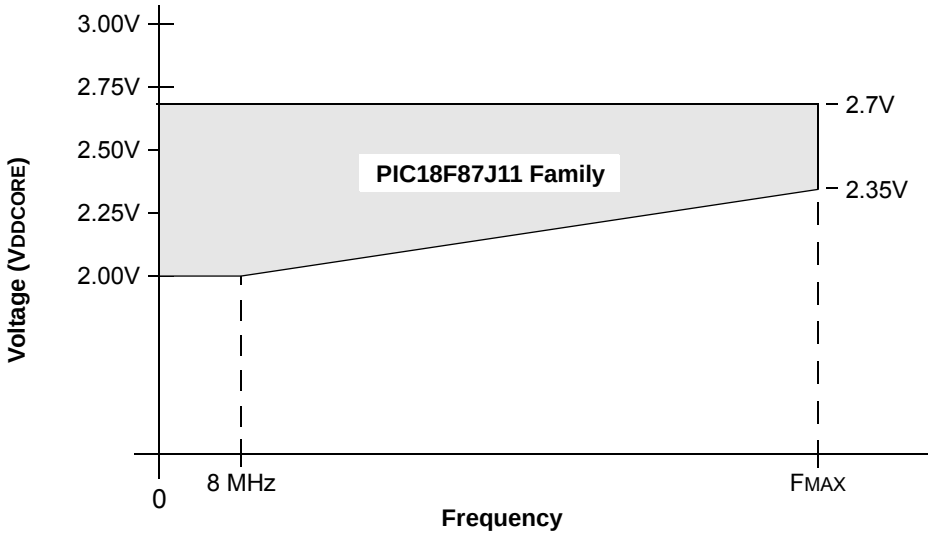
When porting an application to the PIC18F87J11 family, it is very important to consider the type of code. A large, re-entrant application that is written in C and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

FIGURE 28-1: PIC18F87J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL)



FMAX = 25 MHz in 8-Bit External Memory mode
FMAX = 48 MHz in all other modes

FIGURE 28-2: PIC18F87J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL)(0)



FMAX = 25 MHz in 8-Bit External Memory mode
FMAX = 48 MHz in all other modes

Note 1: When the on-chip voltage regulator is disabled, VDD and VDDCORE must be maintained so that VDDCORE ≤ VDD ≤ 3.6V.

28.2 DC Characteristics: Power-Down and Supply Current PIC18F87J11 Family (Industrial) (Continued)

PIC18F87J11 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD) Cont.(2,3)						
	All devices	0.10	0.26	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (**PRI_IDLE** mode, EC oscillator)
		0.07	0.18	mA	+25°C		
		0.09	0.22	mA	+85°C		
	All devices	0.25	0.48	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.13	0.30	mA	+25°C		
		0.10	0.26	mA	+85°C		
	All devices	0.45	0.68	mA	-40°C	VDD = 3.3V(5)	
		0.26	0.45	mA	+25°C		
		0.30	0.54	mA	+85°C		
	All devices	0.36	0.60	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (**PRI_IDLE** mode, EC oscillator)
		0.33	0.56	mA	+25°C		
		0.35	0.56	mA	+85°C		
	All devices	0.52	0.81	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.45	0.70	mA	+25°C		
		0.46	0.70	mA	+85°C		
	All devices	0.80	1.15	mA	-40°C	VDD = 3.3V(5)	
		0.66	0.98	mA	+25°C		
		0.65	0.98	mA	+85°C		
	All devices	5.2	6.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 48 MHz (**PRI_IDLE** mode, EC oscillator)
		4.9	5.9	mA	+25°C		
		3.4	4.5	mA	+85°C		
	All devices	6.2	12.4	mA	-40°C	VDD = 3.3V(5)	
		5.9	11.5	mA	+25°C		
		5.8	11.5	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 a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of the 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;
MCLR = VDD; WDT is enabled/disabled as specified.

3: 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.

4: Voltage regulator is disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator is enabled (ENVREG = 1, tied to VDD, REGSLP = 1).