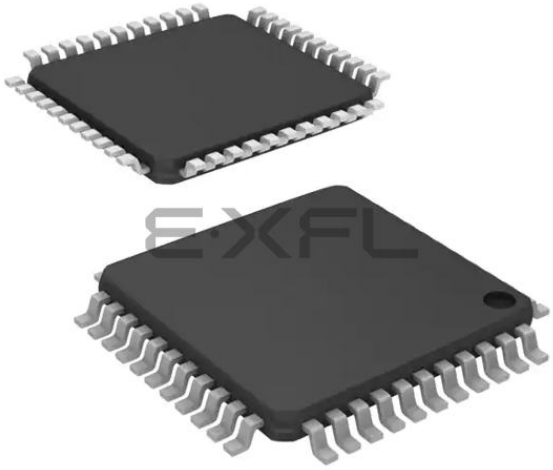


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k22-i-pt

TABLE 2: PIC18(L)F2XK22 PIN SUMMARY

28-SSOP, SOIC 28-SPDIP	28-QFN, UQFN	I/O	Analog	Comparator	CTMU	SR Latch	Reference	(E)CCP	EUSART	MSSP	Timers	Interrupts	Pull-up	Basic
2	27	RA0	AN0	C12IN0-										
3	28	RA1	AN1	C12IN1-										
4	1	RA2	AN2	C2IN+			VREF- DACOUT							
5	2	RA3	AN3	C1IN+			VREF+							
6	3	RA4		C1OUT		SRQ		CCP5			T0CKI			
7	4	RA5	AN4	C2OUT		SRNQ	HLVDIN			$\overline{SS}$1				
10	7	RA6												OSC2 CLKO
9	6	RA7												OSC1 CLKI
21	18	RB0	AN12			SRI		CCP4 FLT0		$\overline{SS}$2		INT0	Y	
22	19	RB1	AN10	C12IN3-				P1C		SCK2 SCL2		INT1	Y	
23	20	RB2	AN8		CTED1			P1B		SDI2 SDA2		INT2	Y	
24	21	RB3	AN9	C12IN2-	CTED2			CCP2 P2A(1)		SDO2			Y	
25	22	RB4	AN11					P1D			T5G	IOC	Y	
26	23	RB5	AN13					CCP3 P3A(4) P2B(3)			T1G T3CKI(2)	IOC	Y	
27	24	RB6							TX2/CK2			IOC	Y	PGC
28	25	RB7							RX2/DT2			IOC	Y	PGD
11	8	RC0						P2B(3)			SOSCO T1CKI T3CKI(2) T3G			
12	9	RC1						CCP2 P2A(1)			SOSCI			
13	10	RC2	AN14		CTPLS			CCP1 P1A			T5CKI			
14	11	RC3	AN15							SCK1 SCL1				
15	12	RC4	AN16							SDI1 SDA1				
16	13	RC5	AN17							SDO1				
17	14	RC6	AN18					CCP3 P3A(4)	TX1/CK1					
18	15	RC7	AN19					P3B	RX1/DT1					
1	26	RE3												$\overline{MCLR}$ VPP
8, 19 19	5, 16 16	VSS												VSS
20	17	VDD												VDD

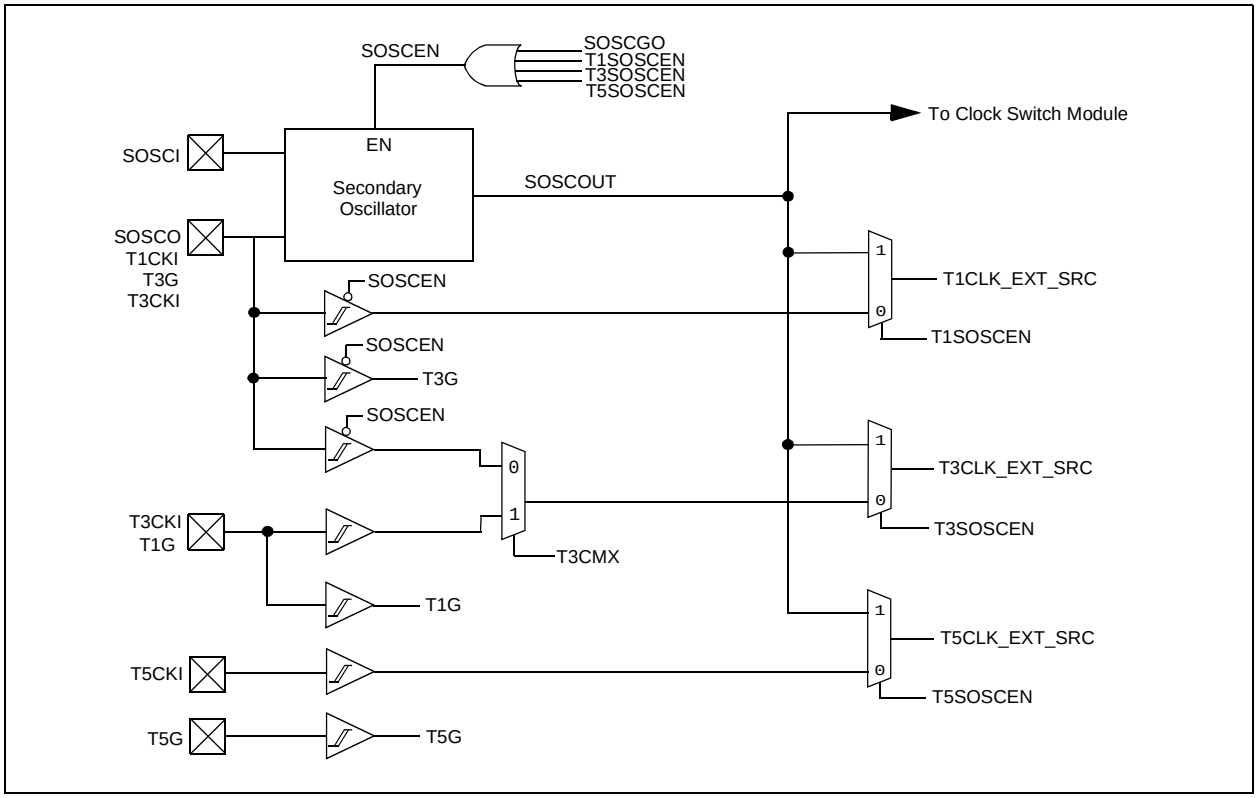
Note 1: CCP2/P2A multiplexed in fuses.
2: T3CKI multiplexed in fuses.
3: P2B multiplexed in fuses.
4: CCP3/P3A multiplexed in fuses.

TABLE 3: PIC18(L)F4XK22 PIN SUMMARY (CONTINUED)

40-PDIP	40-UQFN	44-TQFP	44-QFN	I/O	Analog	Comparator	CTMU	SR Latch	Reference	(E)CCP	EUSART	MSSP	Timers	Interrupts	Pull-up	Basic
9	24	26	26	RE1	AN6					P3B						
10	25	27	27	RE2	AN7					CCP5						
1	16	18	18	RE3											Y	$\overline{MCLR}$ V_{PP}
11, 32	7, 26	7, 28	7,8 28, 29	V_{DD}												V_{DD}
12, 31	6, 27	6, 29	6, 30, 31	V_{SS}												V_{SS}
—	—	12, 13 33, 34	13	NC												

Note 1: CCP2 multiplexed in fuses.
2: T3CKI multiplexed in fuses.
3: CCP3/P3A multiplexed in fuses.
4: P2B multiplexed in fuses.

FIGURE 2-4: SECONDARY OSCILLATOR AND EXTERNAL CLOCK INPUTS



3.6 Selective Peripheral Module Control

Idle mode allows users to substantially reduce power consumption by stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume power. There may be cases where the application needs what IDLE mode does not provide: the allocation of power resources to the CPU processing with minimal power consumption from the peripherals. PIC18(L)F2X/4XK22 family devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with control bits in the Peripheral Module Disable (PMD) registers. These bits generically named XXXMD are located in control registers PMD0, PMD1 or PMD2.

Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, power to the control and status registers associated with the peripheral is removed. Writes to these registers have no effect and read values are invalid. Clearing a set PMD bit restores power to the associated control and status registers, thereby setting those registers to their default values.

3.7 Register Definitions: Peripheral Module Disable

REGISTER 3-1: PMD0: PERIPHERAL MODULE DISABLE REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD
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 **UART2MD:** UART2 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 6 **UART1MD:** UART1 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 5 **TMR6MD:** Timer6 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 4 **TMR5MD:** Timer5 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 3 **TMR4MD:** Timer4 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 2 **TMR3MD:** Timer3 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 1 **TMR2MD:** Timer2 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

bit 0 **TMR1MD:** Timer1 Peripheral Module Disable Control bit
1 = Module is disabled, Clock Source is disconnected, module does not draw digital power
0 = Module is enabled, Clock Source is connected, module draws digital power

FIGURE 5-5: DATA MEMORY MAP FOR PIC18(L)F23K22 AND PIC18(L)F43K22 DEVICES

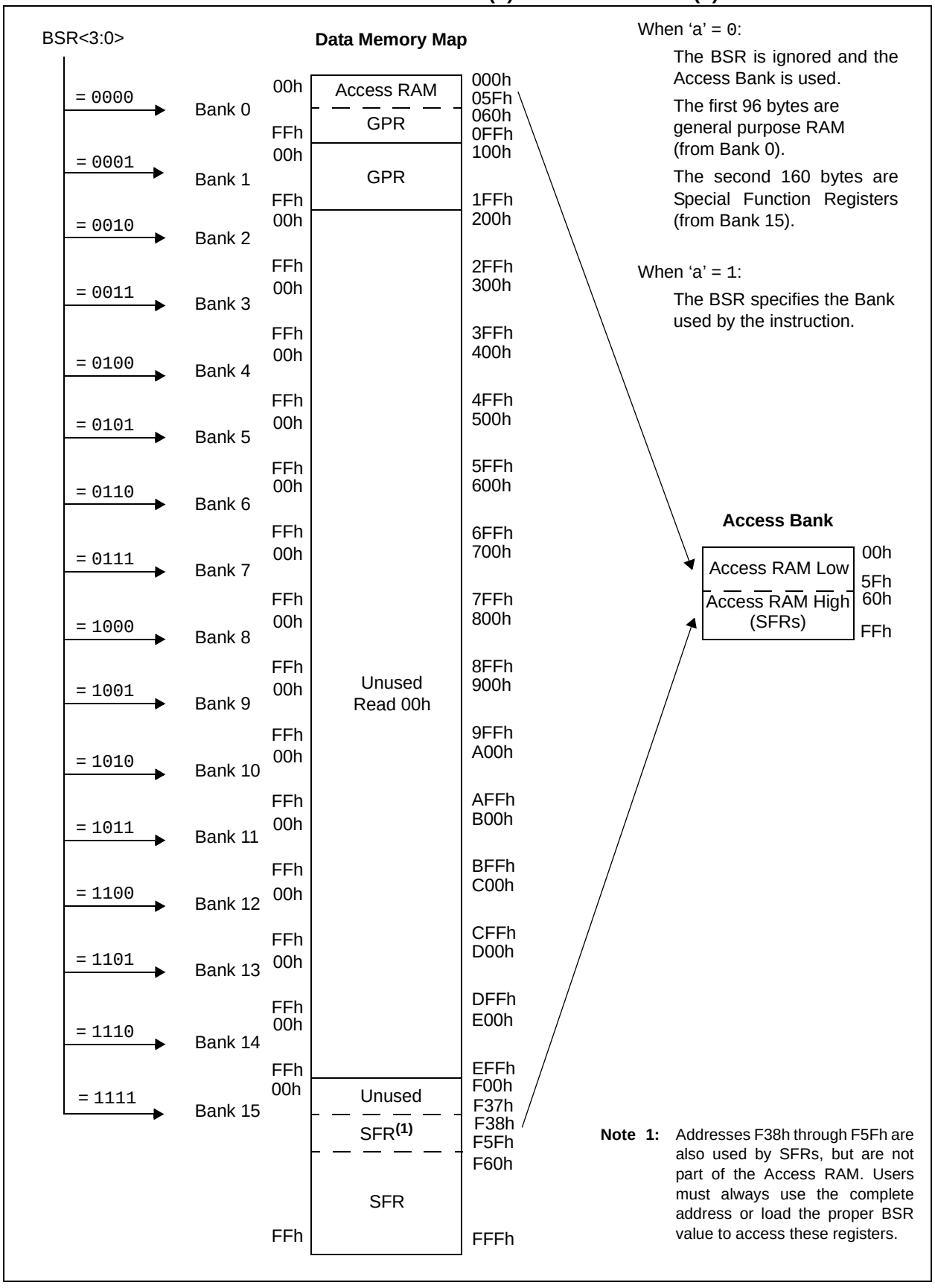
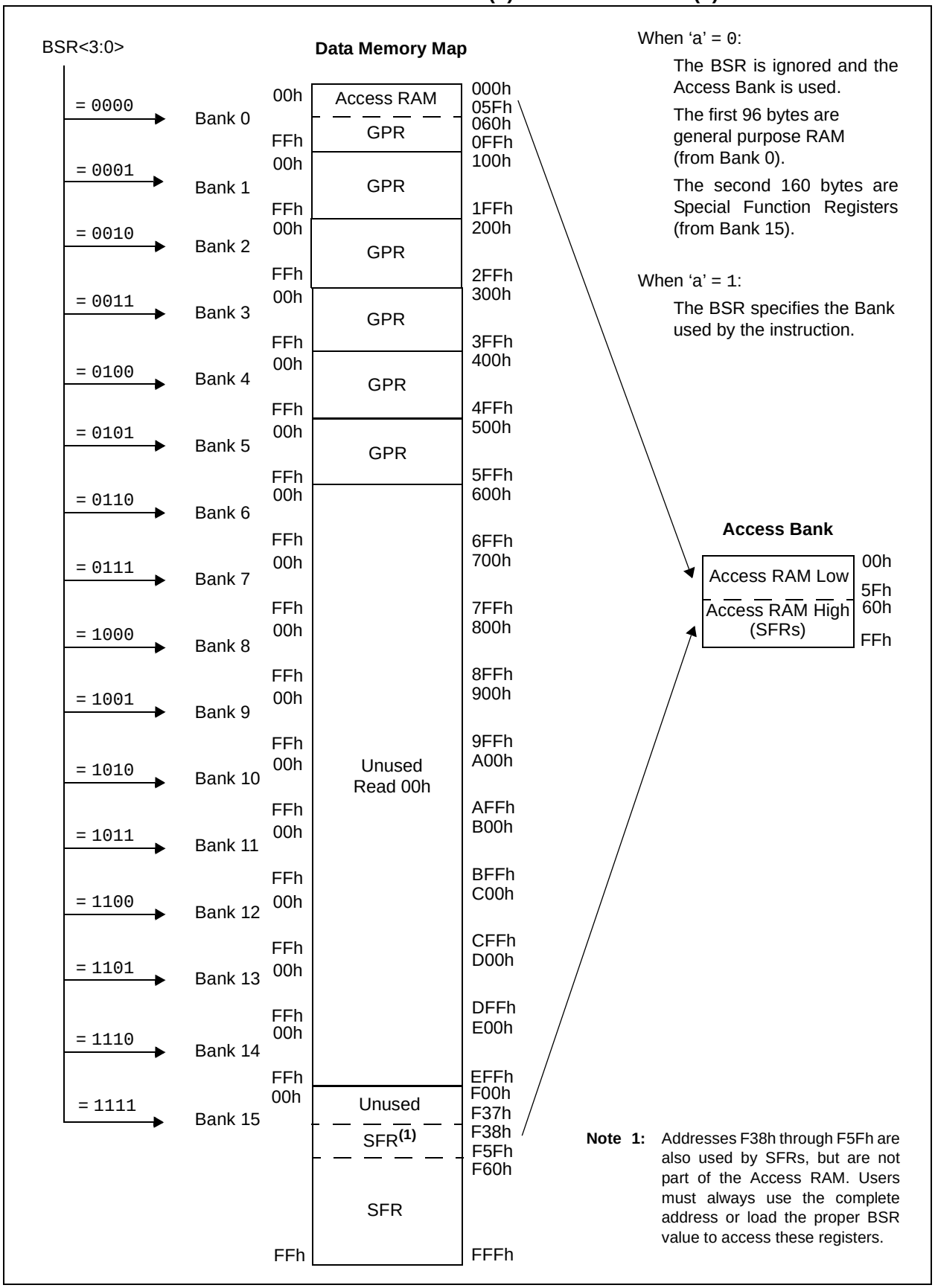


FIGURE 5-7: DATA MEMORY MAP FOR PIC18(L)F25K22 AND PIC18(L)F45K22 DEVICES



5.6 Data Addressing Modes

Note:	The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 5.7 "Data Memory and the Extended Instruction Set"** for more information.

While the program memory can be addressed in only one way – through the program counter – information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 5.7.1 "Indexed Addressing with Literal Offset"**.

5.6.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way but require an additional explicit argument in the opcode. This is known as Literal Addressing mode because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

5.6.2 DIRECT ADDRESSING

Direct addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of direct addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 5.4.3 "General Purpose Register File"**) or a location in the Access Bank (**Section 5.4.2 "Access Bank"**) as the data source for the instruction.

The Access RAM bit 'a' determines how the address is interpreted. When 'a' is '1', the contents of the BSR (**Section 5.4.1 "Bank Select Register (BSR)"**) are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

5.6.3 INDIRECT ADDRESSING

Indirect addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations which are to be read or written. Since the FSRs are themselves located in RAM as Special File Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures, such as tables and arrays in data memory.

The registers for indirect addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code, using loops, such as the example of clearing an entire RAM bank in Example 5-5.

EXAMPLE 5-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSR    FSR0, 100h ;
NEXT    CLRF    POSTINC0   ; Clear INDF
                           ; register then
                           ; inc pointer
        BTFSS   FSR0H, 1   ; All done with
                           ; Bank1?
        BRA     NEXT       ; NO, clear next
CONTINUE                   ; YES, continue
```

14.4.2 FULL-BRIDGE MODE

In Full-Bridge mode, all four pins are used as outputs. An example of full-bridge application is shown in Figure 14-10.

In the Forward mode, pin CCPx/PxA is driven to its active state, pin PxD is modulated, while PxB and PxC will be driven to their inactive state as shown in Figure 14-11.

In the Reverse mode, PxC is driven to its active state, pin PxB is modulated, while PxA and PxD will be driven to their inactive state as shown Figure 14-11.

PxA, PxB, PxC and PxD outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the PxA, PxB, PxC and PxD pins as outputs.

FIGURE 14-10: EXAMPLE OF FULL-BRIDGE APPLICATION

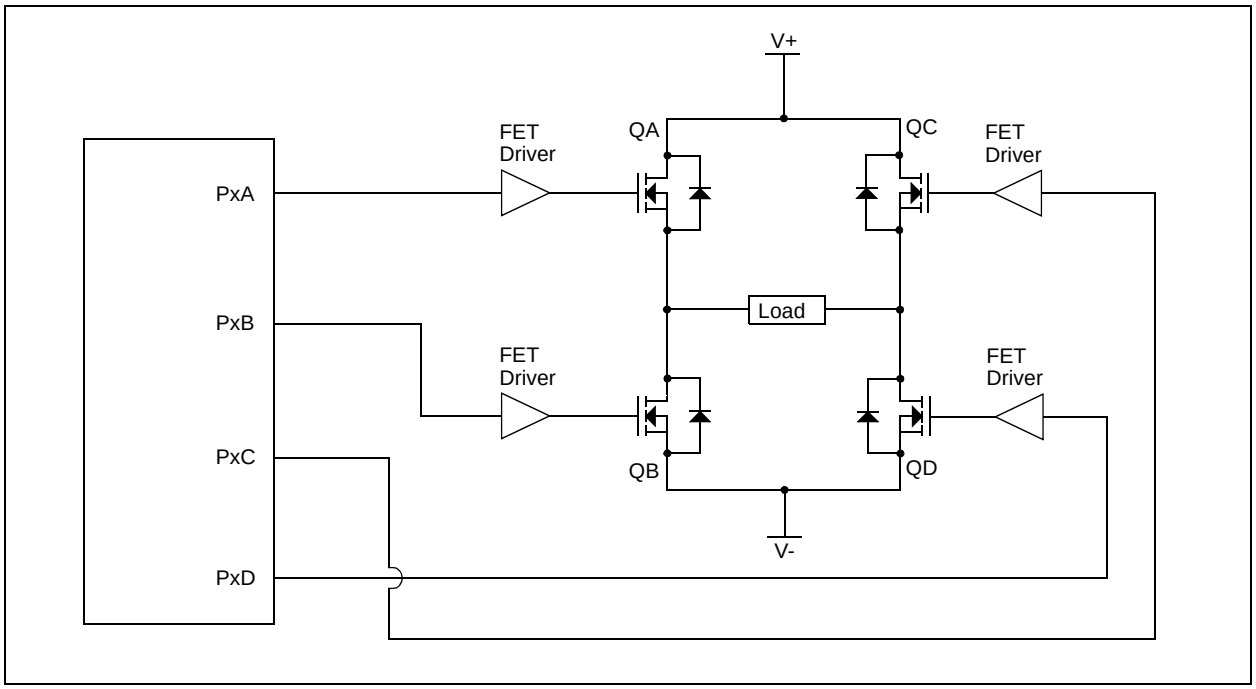
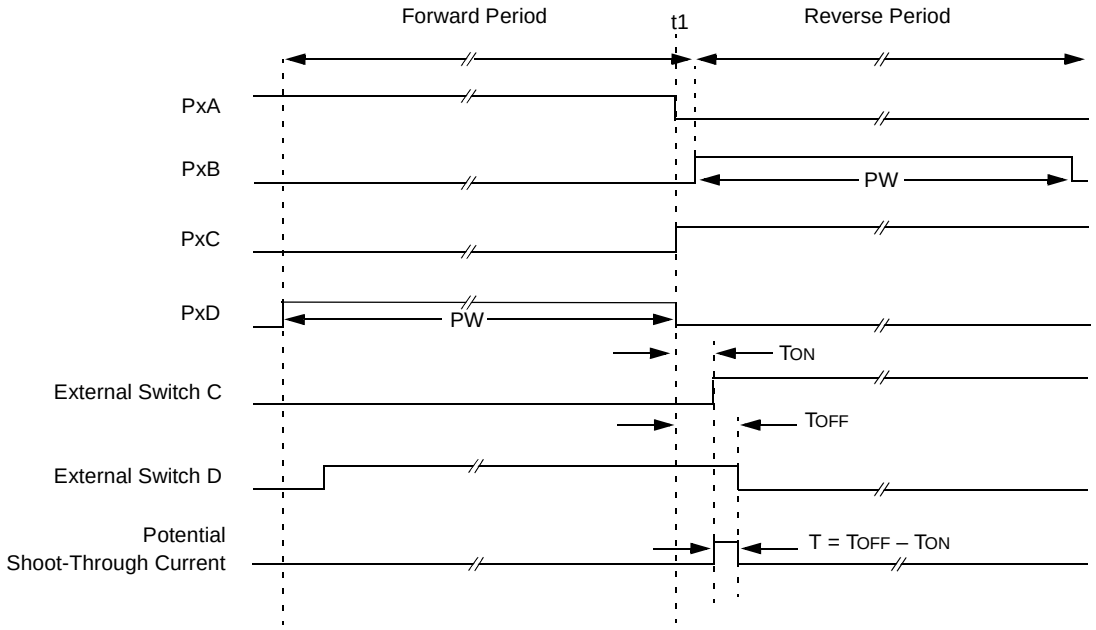


FIGURE 14-13: EXAMPLE OF PWM DIRECTION CHANGE AT NEAR 100% DUTY CYCLE



Note 1: All signals are shown as active-high.

2: TON is the turn-on delay of power switch QC and its driver.

3: TOFF is the turn-off delay of power switch QD and its driver.

14.4.3 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the CCPxAS<2:0> bits of the ECCPxAS register. A shutdown event may be generated by:

- A logic '0' on the INT pin
- Comparator Cx (async_CxOUT)
- Setting the CCPxASE bit in firmware

A shutdown condition is indicated by the CCPxASE (Auto-Shutdown Event Status) bit of the ECCPxAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:

The CCPxASE bit is set to '1'. The CCPxASE will remain set until cleared in firmware or an auto-restart occurs (see **Section 14.4.4 "Auto-Restart Mode"**).

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [PxA/PxC] and [PxB/PxD].

The state of each pin pair is determined by the PSSxAC<1:0> and PSSxBD<1:0> bits of the ECCPxAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.

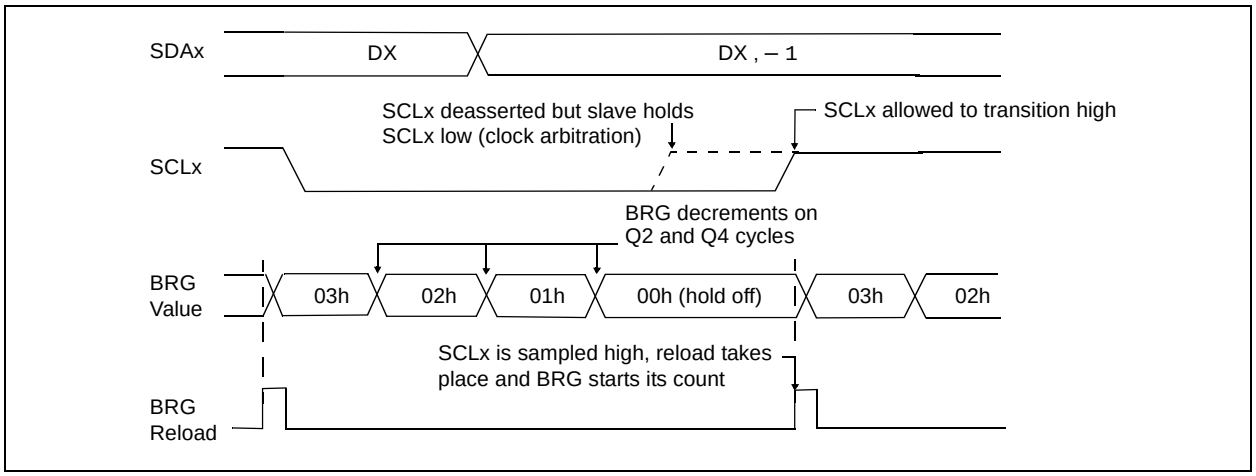
2: Writing to the CCPxASE bit is disabled while an auto-shutdown condition persists.

3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

15.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCLx pin is actually sampled high. When the SCLx pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCLx high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 15-25).

FIGURE 15-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



15.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not Idle.

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPxCON2 is disabled until the Start condition is complete.

15.8 Register Definitions: MSSP Control

REGISTER 15-2: SSPxSTAT: SSPx STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **SMP:** SPI Data Input Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)

In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

In I^2C mode only:
1 = Enable input logic so that thresholds are compliant with SMbus specification
0 = Disable SMbus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
(I^2C mode only. This bit is cleared when the MSSPx module is disabled, SSPxEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit
(I^2C mode only. This bit is cleared when the MSSPx module is disabled, SSPxEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ bit information (I^2C mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not $\overline{ACK}$ bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

Receive (SPI and I^2C modes):
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty

Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is empty

18.5 Operation During Sleep

The comparator, if enabled before entering Sleep mode, remains active during Sleep. The additional current consumed by the comparator is shown separately in **Section 27.0 "Electrical Specifications"**. If the comparator is not used to wake the device, power consumption can be minimized while in Sleep mode by turning off the comparator. Each comparator is turned off by clearing the CxON bit of the CMxCON0 register.

A change to the comparator output can wake-up the device from Sleep. To enable the comparator to wake the device from Sleep, the CxIE bit of the PIE2 register and the PEIE/GIEL bit of the INTCON register must be set. The instruction following the SLEEP instruction always executes following a wake from Sleep. If the GIE/GIEH bit of the INTCON register is also set, the device will then execute the Interrupt Service Routine.

18.6 Effects of a Reset

A device Reset forces the CMxCON0 and CM2CON1 registers to their Reset states. This forces both comparators and the voltage references to their Off states.Comparator Control Registers.

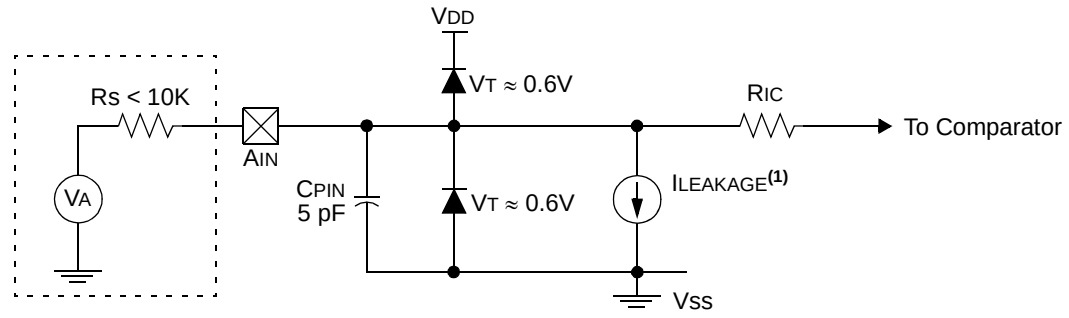
18.7 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 18-5. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1:	When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.
2:	Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 18-5: ANALOG INPUT MODEL



Legend: C_{PIN} = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_S = Source Impedance
V_A = Analog Voltage
V_T = Threshold Voltage

Note 1: See **Section 27.0 "Electrical Specifications"**.

24.5.1 PROGRAM MEMORY CODE PROTECTION

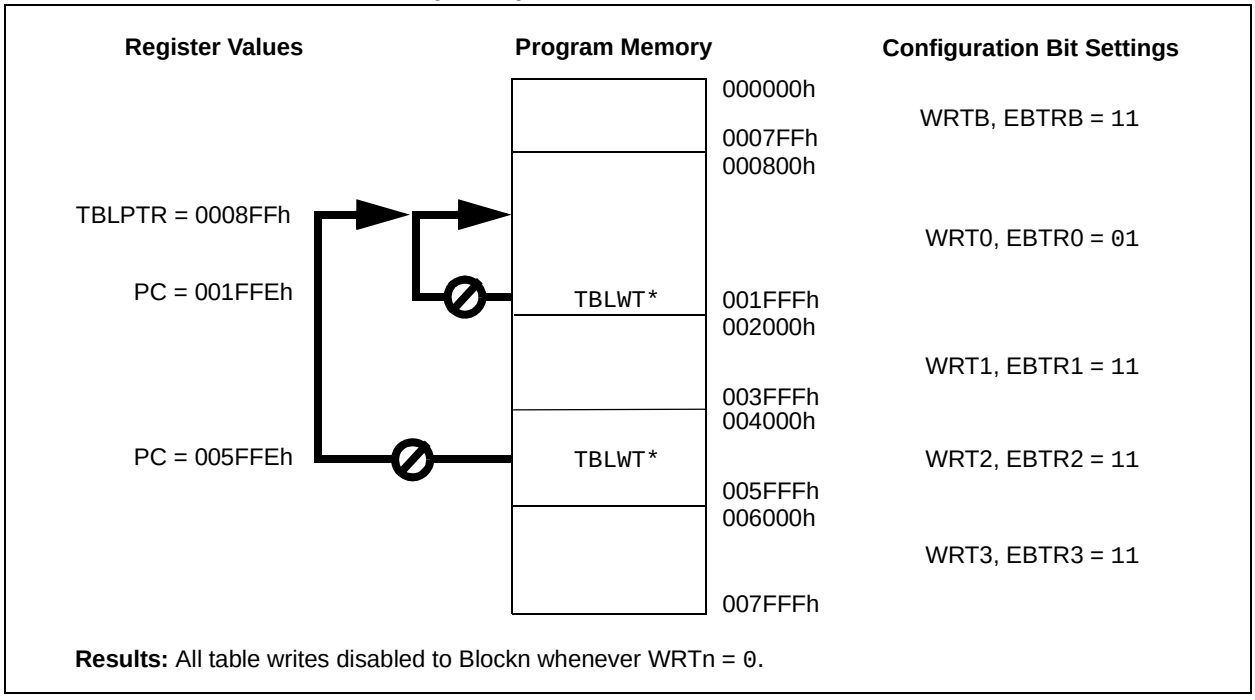
The program memory may be read to or written from any location using the table read and table write instructions. The device ID may be read with table reads. The Configuration registers may be read and written with the table read and table write instructions.

In Normal execution mode, the CPn bits have no direct effect. CPn bits inhibit external reads and writes. A block of user memory may be protected from table writes if the WRTn Configuration bit is '0'. The EBTRn bits control table reads. For a block of user memory with the EBTRn bit cleared to '0', a table READ instruction that executes from within that block is allowed to read.

A table read instruction that executes from a location outside of that block is not allowed to read and will result in reading '0's. Figures 24-3 through 24-5 illustrate table write and table read protection.

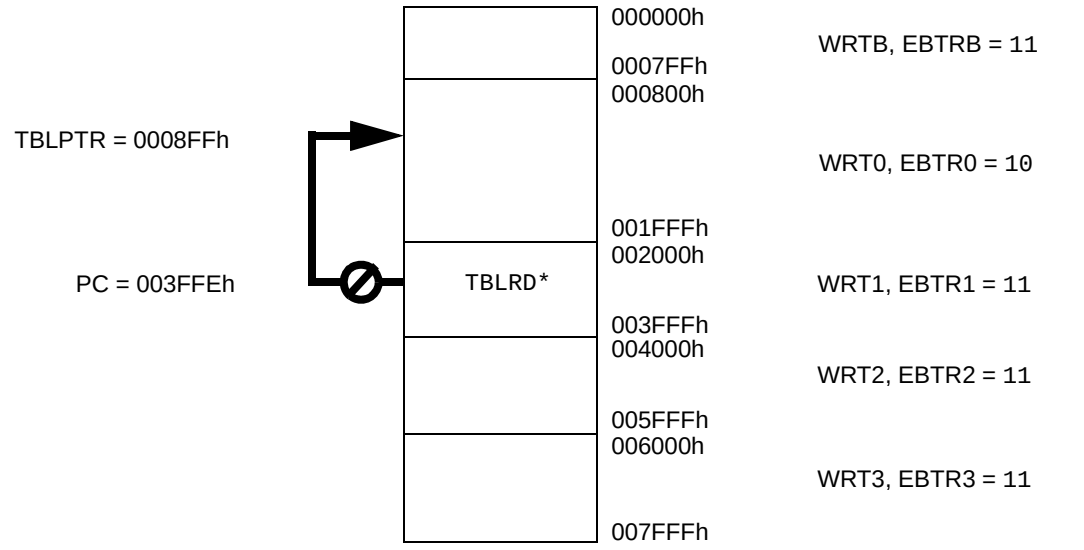
Note:	Code protection bits may only be written to a '0' from a '1' state. It is not possible to write a '1' to a bit in the '0' state. Code protection bits are only set to '1' by a full chip erase or block erase function. The full chip erase and block erase functions can only be initiated via ICSP™ or an external programmer.

FIGURE 24-3: TABLE WRITE (WRTn) DISALLOWED



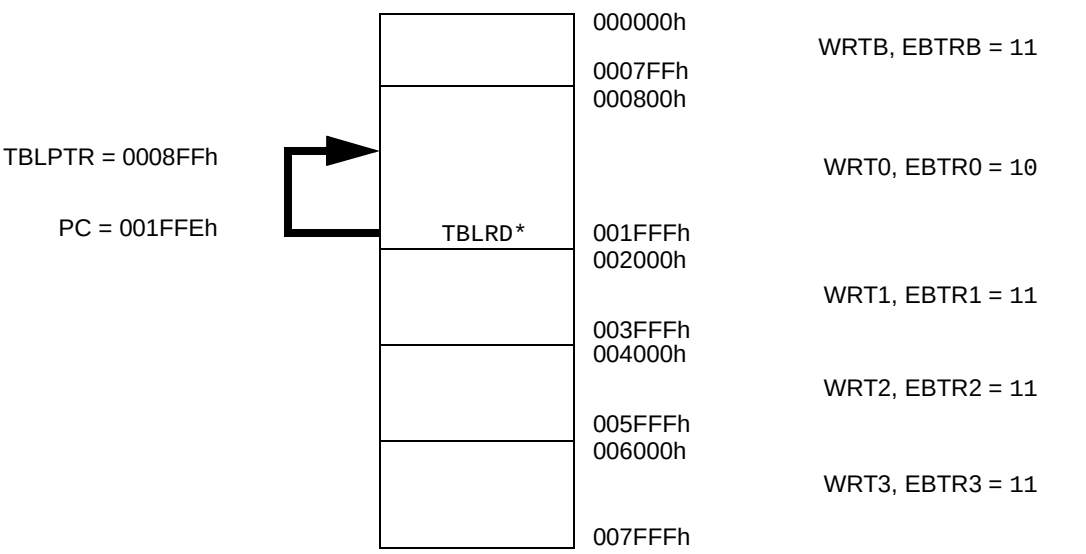
Results: All table writes disabled to Blockn whenever WRTn = 0.

FIGURE 24-4: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED



Results: All table reads from external blocks to Blockn are disabled whenever EBTRn = 0.
TABLAT register returns a value of '0'.

FIGURE 24-5: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



Results: Table reads permitted within Blockn, even when EBTRBn = 0.
TABLAT register returns the value of the data at the location TBLPTR.

MOVFF **Move f to f**

Syntax: MOVFF f_s,f_d

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)	1100	ffff	ffff$_s$
2nd word (destin.)	1111	ffff	ffff$_d$

Description: The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh.
Either source or destination can be W (a useful special situation).
MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).
The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVFF REG1, REG2

Before Instruction
REG1 = 33h
REG2 = 11h
After Instruction
REG1 = 33h
REG2 = 33h

MOVLB **Move literal to low nibble in BSR**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow$ BSR

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk

Description: The 8-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of k_7:k_4.

Words: 1

Cycles: 1

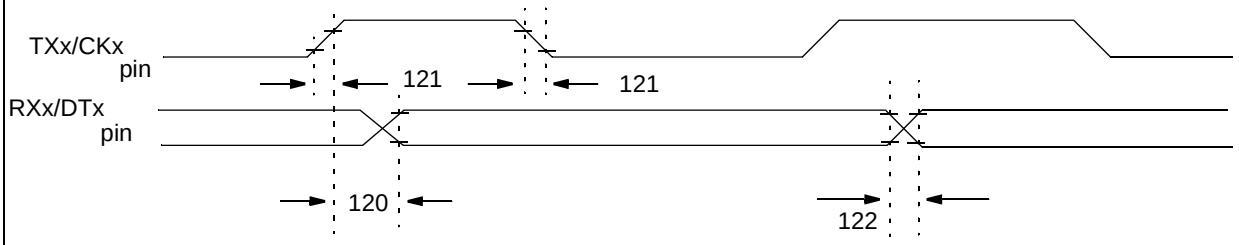
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

FIGURE 27-21: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

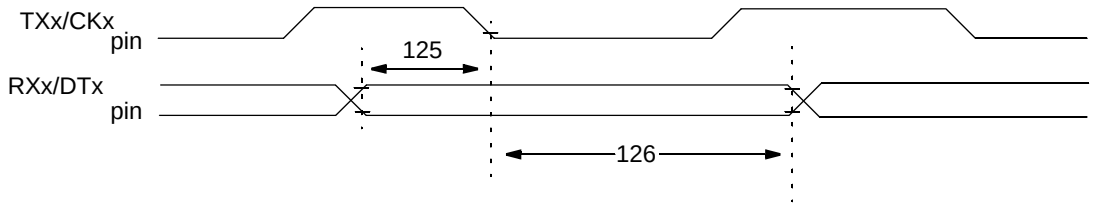


Note: Refer to Figure 27-6 for load conditions.

TABLE 27-19: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	—	40	ns	
121	Tckrf	Clock Out Rise Time and Fall Time (Master mode)	—	20	ns	
122	Tdtrf	Data Out Rise Time and Fall Time	—	20	ns	

FIGURE 27-22: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 27-6 for load conditions.

TABLE 27-20: EUSART SYNCHRONOUS RECEIVE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
125	TdtV2ckl	SYNC RCV (MASTER & SLAVE) Data Setup before CK ↓ (DT setup time)	10	—	ns	
126	TckL2dtl	Data Hold after CK ↓ (DT hold time)	15	—	ns	

FIGURE 28-60: PIC18LF2X/4XK22 TYPICAL IDD: PRI_IDLE EC MEDIUM POWER

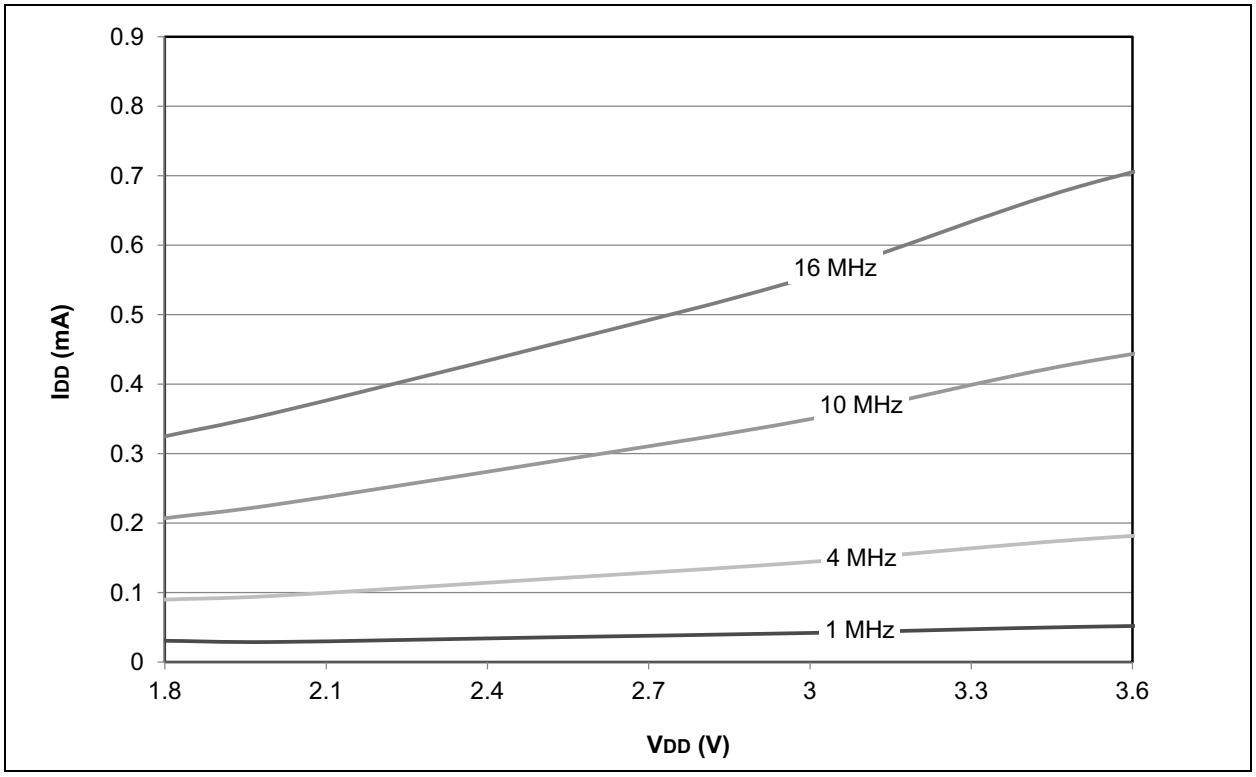


FIGURE 28-61: PIC18LF2X/4XK22 MAXIMUM IDD: PRI_IDLE EC MEDIUM POWER

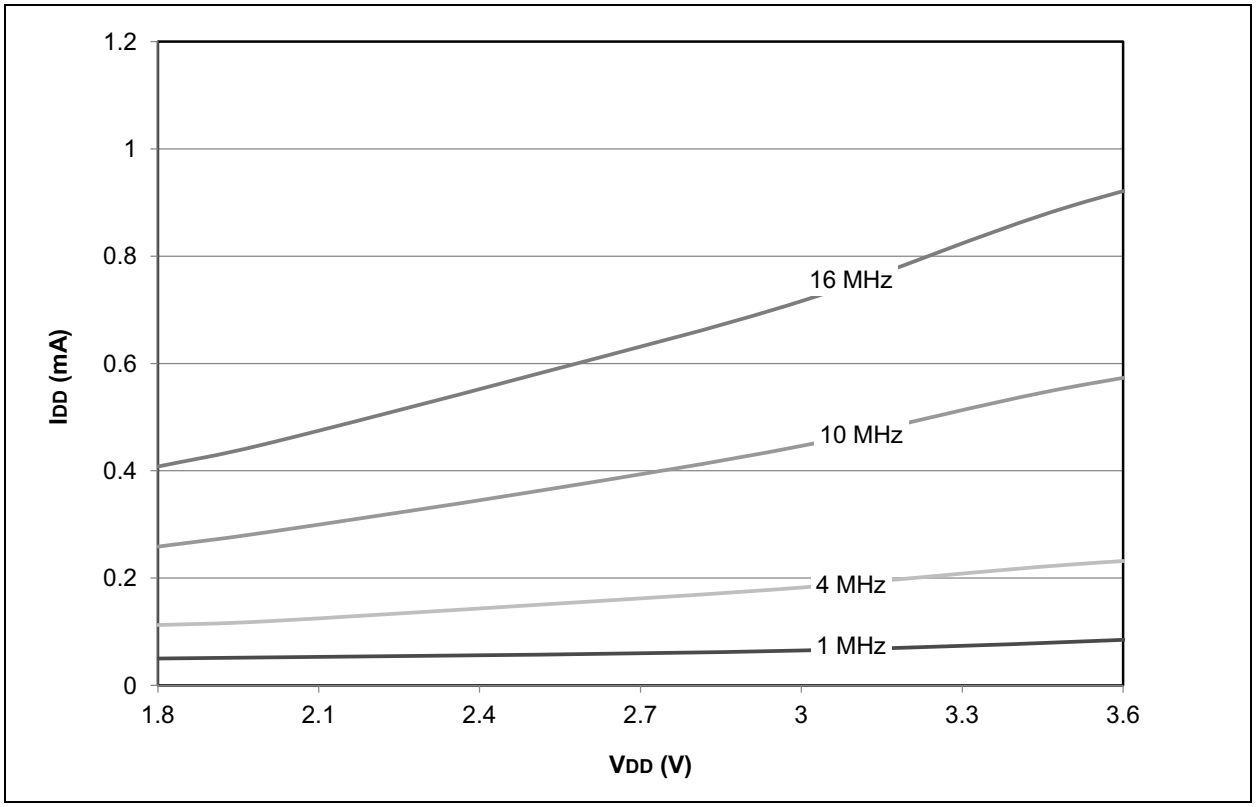


FIGURE 28-70: PIC18F2X/4XK22 TYPICAL IDD: PRI_IDLE EC with PLL

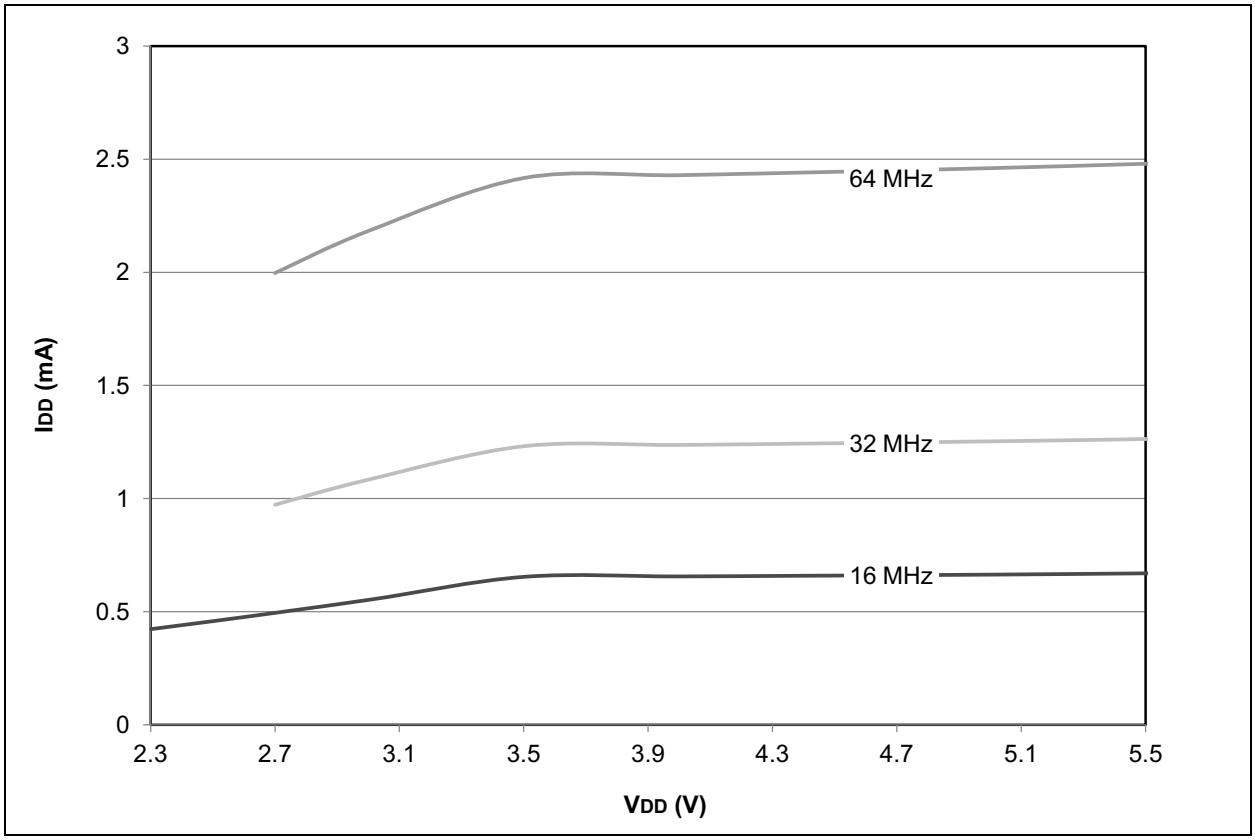


FIGURE 28-71: PIC18F2X/4XK22 MAXIMUM IDD: PRI_IDLE EC with PLL

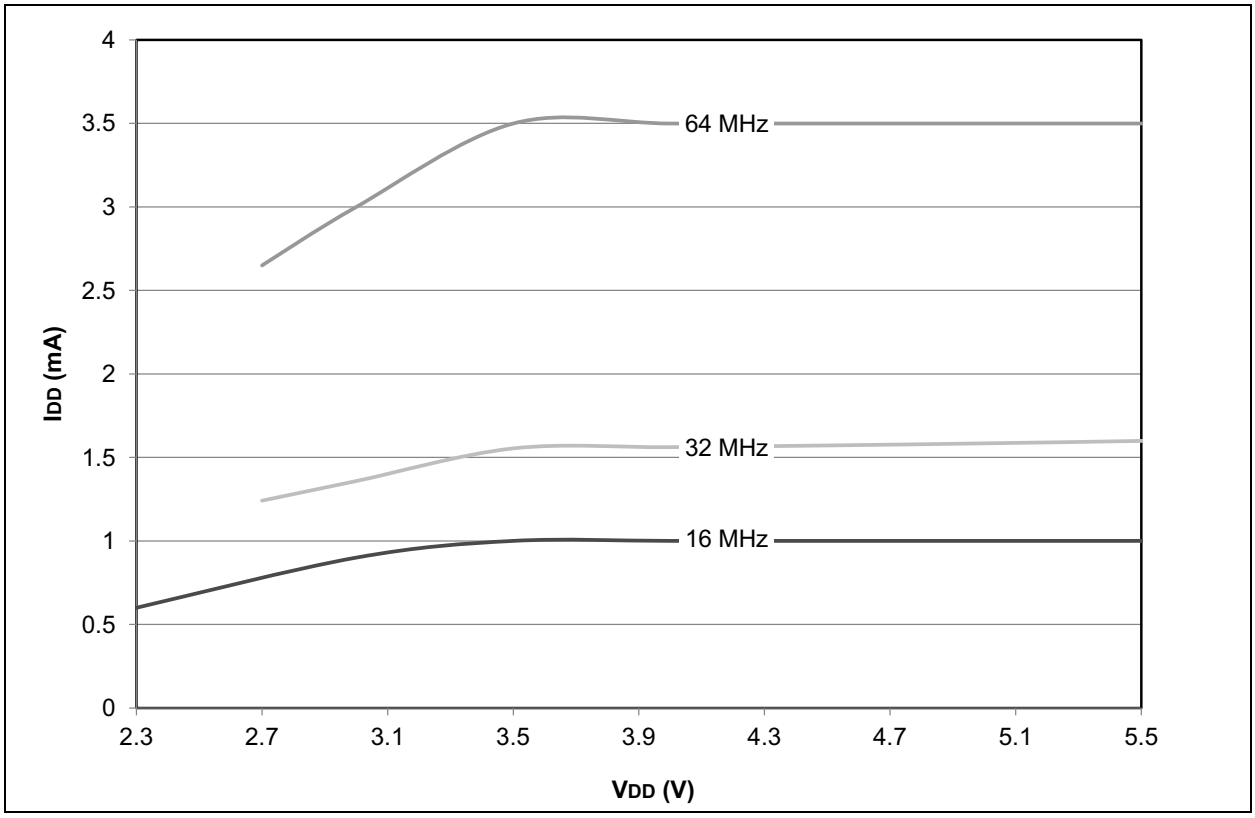


FIGURE 28-90: PIC18(L)F2X/4XK22 COMPARATOR OFFSET VOLTAGE, LOW-POWER MODE; VDD=5.5V

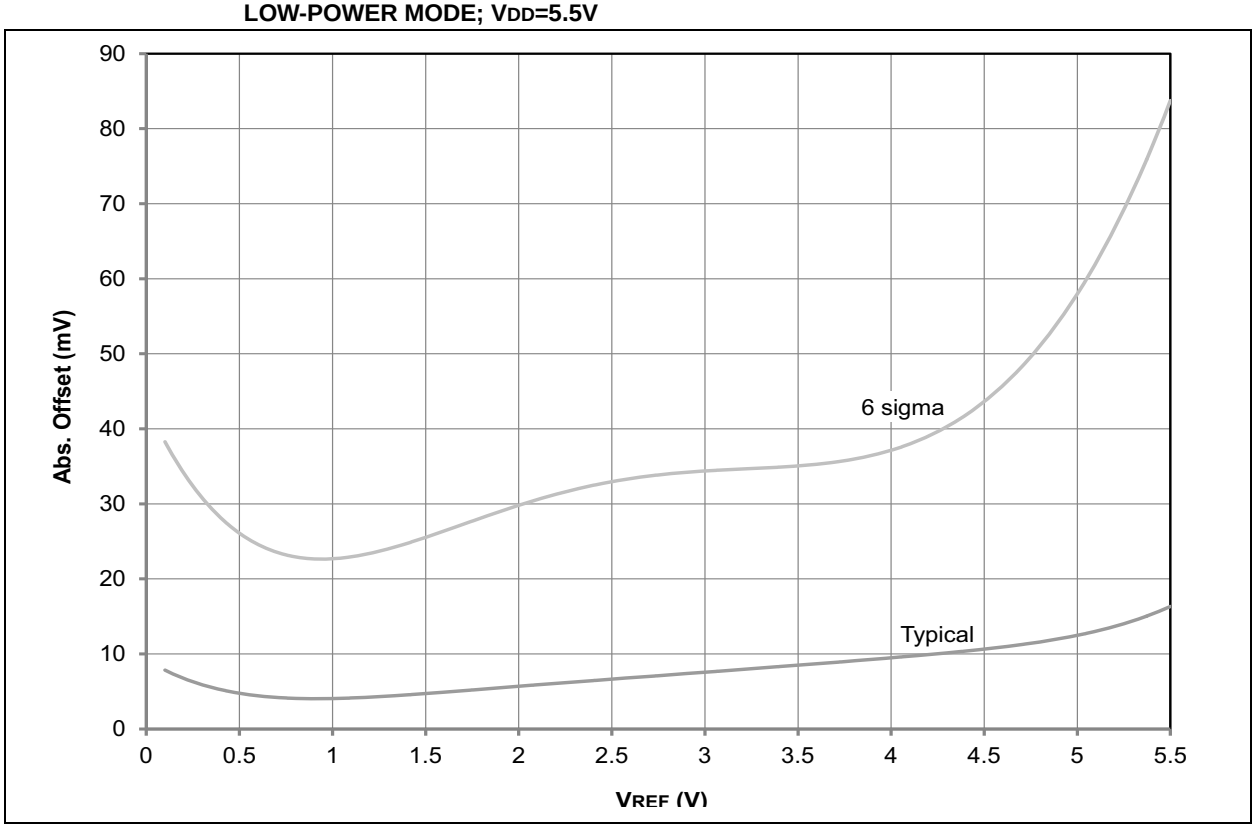
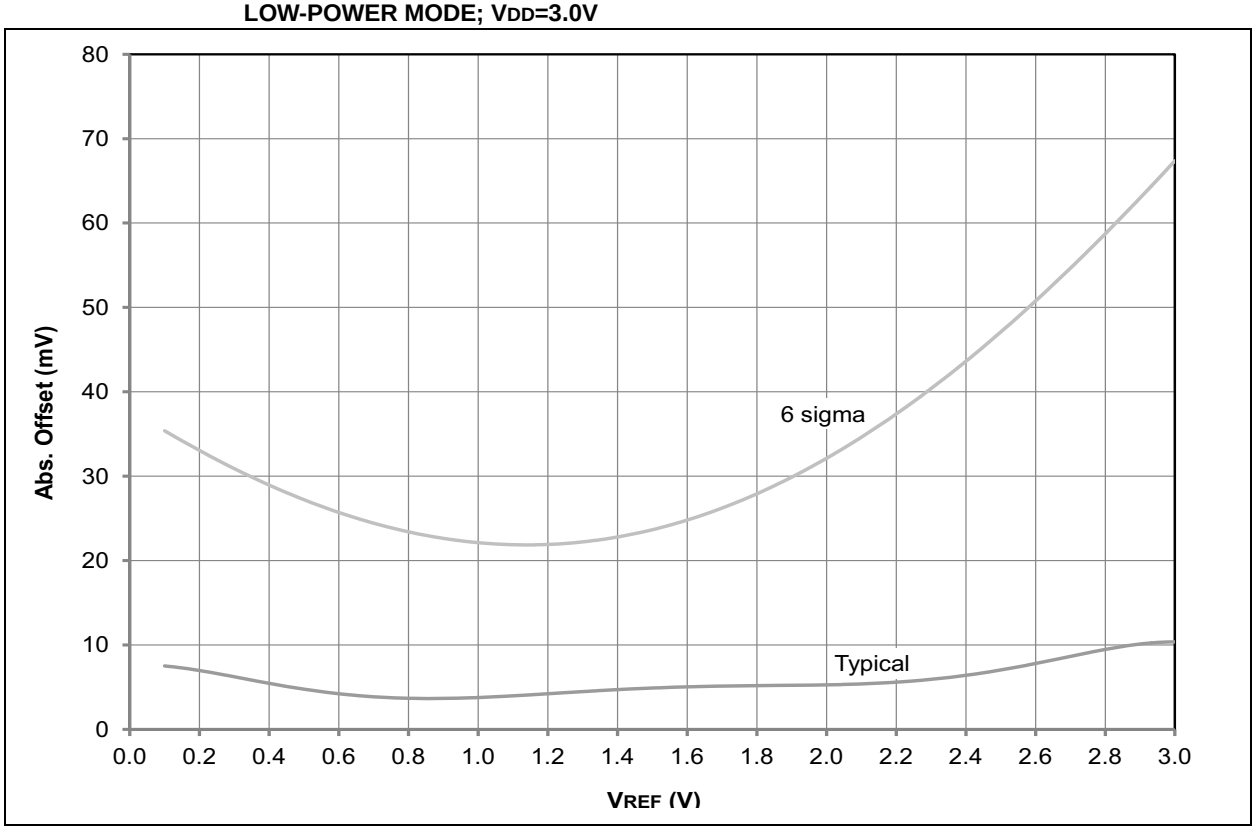


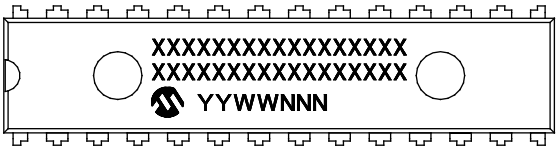
FIGURE 28-91: PIC18(L)F2X/4XK22 COMPARATOR OFFSET VOLTAGE, LOW-POWER MODE; VDD=3.0V



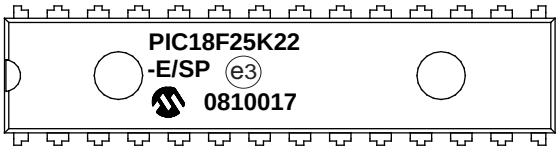
29.0 PACKAGING INFORMATION

29.1 Package Marking Information

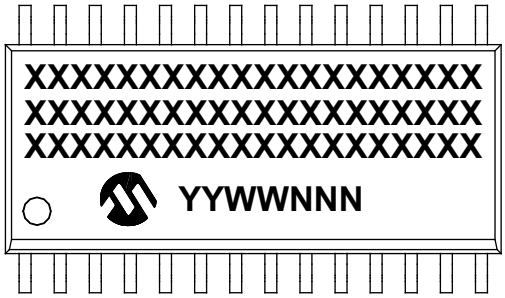
28-Lead SPDIP (.300”)



Example

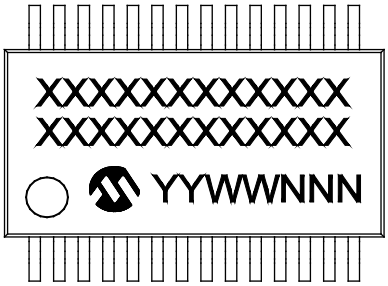


28-Lead SOIC (7.50 mm)

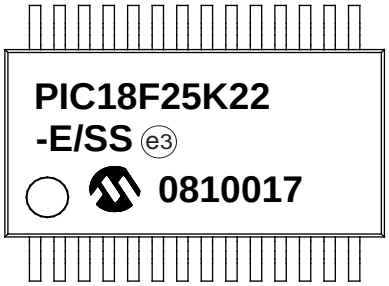


Example

28-Lead SSOP (5.30 mm)



Example



Legend:	XX...X	Customer-specific information or Microchip part number
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.