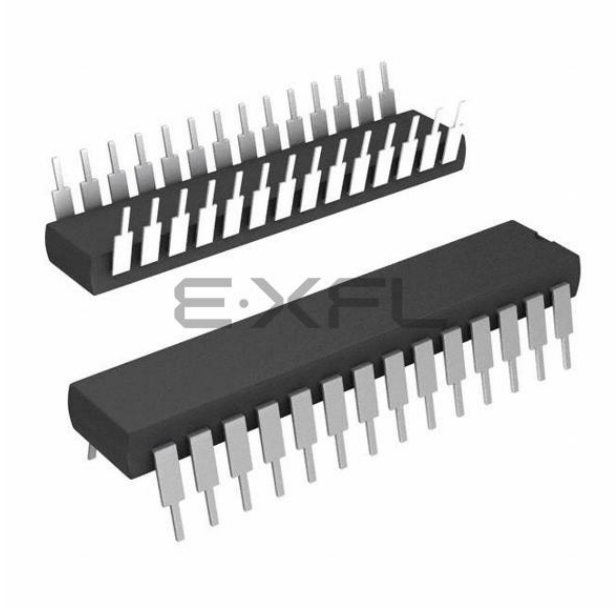


Welcome to E-XFL.COM

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	24
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 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25k22-i-sp

2.2 Oscillator Control

The OSCCON, OSCCON2 and OSCTUNE registers (Register 2-1 to Register 2-3) control several aspects of the device clock's operation, both in full-power operation and in power-managed modes.

- Main System Clock Selection (SCS)
- Primary Oscillator Circuit Shutdown (PRISD)
- Secondary Oscillator Enable (SOSCGO)
- Primary Clock Frequency 4x multiplier (PLLEN)
- Internal Frequency selection bits (IRCF, INTSRC)
- Clock Status bits (OSTS, HFIOFS, MFIOFS, LFIOFS. SOSCRUN, PLLRDY)
- Power management selection (IDLEN)

2.2.1 MAIN SYSTEM CLOCK SELECTION

The System Clock Select bits, SCS<1:0>, select the main clock source. The available clock sources are

- Primary clock defined by the FOSC<3:0> bits of CONFIG1H. The primary clock can be the primary oscillator, an external clock, or the internal oscillator block.
- Secondary clock (secondary oscillator)
- Internal oscillator block (HFINTOSC, MFINTOSC and LFINTOSC).

The clock source changes immediately after one or more of the bits is written to, following a brief clock transition interval. The SCS bits are cleared to select the primary clock on all forms of Reset.

2.2.2 INTERNAL FREQUENCY SELECTION

The Internal Oscillator Frequency Select bits (IRCF<2:0>) select the frequency output of the internal oscillator block. The choices are the LFINTOSC source (31.25 kHz), the MFINTOSC source (31.25 kHz, 250 kHz or 500 kHz) and the HFINTOSC source (16 MHz) or one of the frequencies derived from the HFINTOSC postscaler (31.25 kHz to 8 MHz). If the internal oscillator block is supplying the main clock, changing the states of these bits will have an immediate change on the internal oscillator's output. On device Resets, the output frequency of the internal oscillator is set to the default frequency of 1 MHz.

2.2.3 LOW FREQUENCY SELECTION

When a nominal output frequency of 31.25 kHz is selected (IRCF<2:0> = 000), users may choose which internal oscillator acts as the source. This is done with the INTSRC bit of the OSCTUNE register and MFIOSEL bit of the OSCCON2 register. See Figure 2-2 and Register 2-1 for specific 31.25 kHz selection. This option allows users to select a 31.25 kHz clock (MFINTOSC or HFINTOSC) that can be tuned using the TUN<5:0> bits in OSCTUNE register, while maintaining power savings with a very low clock speed. LFINTOSC always remains the clock source for features such as the Watchdog Timer and the Fail-Safe Clock Monitor, regardless of the setting of INTSRC and MFIOSEL bits

This option allows users to select the tunable and more precise HFINTOSC as a clock source, while maintaining power savings with a very low clock speed.

2.2.4 POWER MANAGEMENT

The IDLEN bit of the OSCCON register determines whether the device goes into Sleep mode or one of the Idle modes when the SLEEP instruction is executed.

TABLE 2-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

OSC Mode	OSC1 Pin	OSC2 Pin
RC, INTOSC with CLKOUT	Floating, external resistor should pull high	At logic low (clock/4 output)
RC with IO	Floating, external resistor should pull high	Configured as PORTA, bit 6
INTOSC with IO	Configured as PORTA, bit 7	Configured as PORTA, bit 6
EC with IO	Floating, pulled by external clock	Configured as PORTA, bit 6
EC with CLKOUT	Floating, pulled by external clock	At logic low (clock/4 output)
LP, XT, HS	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level

Note: See Table 4-2 in **Section 4.0 "Reset"** for time-outs due to Sleep and $\overline{\text{MCLR}}$ Reset.

2.11 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS<1:0>) bits of the OSCCON register.

PIC18(L)F2X/4XK22 devices contain circuitry to prevent clock "glitches" when switching between clock sources. A short pause in the device clock occurs during the clock switch. The length of this pause is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Clock transitions are discussed in greater detail in **Section 3.1.2 "Entering Power-Managed Modes"**.

2.11.1 SYSTEM CLOCK SELECT (SCS<1:0>) BITS

The System Clock Select (SCS<1:0>) bits of the OSCCON register select the system clock source that is used for the CPU and peripherals.

- When SCS<1:0> = 00, the system clock source is determined by configuration of the FOSC<3:0> bits in the CONFIG1H Configuration register.
- When SCS<1:0> = 10, the system clock source is chosen by the internal oscillator frequency selected by the INTSRC bit of the OSCTUNE register, the MFIOSEL bit of the OSCCON2 register and the IRCF<2:0> bits of the OSCCON register.
- When SCS<1:0> = 01, the system clock source is the 32.768 kHz secondary oscillator shared with Timer1, Timer3 and Timer5.

After a Reset, the SCS<1:0> bits of the OSCCON register are always cleared.

Note: Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS<1:0> bits of the OSCCON register. The user can monitor the SOSCRUN, MFIOFS and LFIOFS bits of the OSCCON2 register, and the HFIOFS and OSTS bits of the OSCCON register to determine the current system clock source.

2.11.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<3:0> bits in the CONFIG1H Configuration register, or from the internal clock source. In particular, when the primary oscillator is the source of the primary clock, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

3.0 POWER-MANAGED MODES

PIC18(L)F2X/4XK22 devices offer a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery-powered devices).

There are three categories of power-managed modes:

- Run modes
- Idle modes
- Sleep mode

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block). The Sleep mode does not use a clock source.

The power-managed modes include several power-saving features offered on previous PIC® microcontroller devices. One of the clock switching features allows the controller to use the secondary oscillator (SOSC) in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC microcontroller devices, where all device clocks are stopped.

3.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions:

- Whether or not the CPU is to be clocked
- The selection of a clock source

The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS<1:0> bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

3.1.1 CLOCK SOURCES

The SCS<1:0> bits allow the selection of one of three clock sources for power-managed modes. They are:

- the primary clock, as defined by the FOSC<3:0> Configuration bits
- the secondary clock (the SOSC oscillator)
- the internal oscillator block

3.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS<1:0> bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. Refer to **Section 2.11 “Clock Switching”** for more information.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

TABLE 3-1: POWER-MANAGED MODES

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN(1)	SCS<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – LP, XT, HS, RC, EC and Internal Oscillator Block(2). This is the normal full-power execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – SOSC Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal Oscillator Block(2)
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – SOSC Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block(2)

Note 1: IDLEN reflects its value when the SLEEP instruction is executed.

2: Includes HFINTOSC and HFINTOSC postscaler, as well as the LFINTOSC source.

© 2010-2016 Microchip Technology Inc.

4.7 Reset State of Registers

Some registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. All other registers are forced to a “Reset state” depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register, $\overline{RI}$, $\overline{TO}$, $\overline{PD}$, $\overline{POR}$ and $\overline{BOR}$, are set or cleared differently in different Reset situations, as indicated in Table 4-3. These bits are used by software to determine the nature of the Reset.

Table 5-2 describes the Reset states for all of the Special Function Registers. The table identifies differences between Power-On Reset (POR)/Brown-Out Reset (BOR) and all other Resets, (i.e., Master Clear, WDT Resets, STKFUL, STKUNF, etc.). Additionally, the table identifies register bits that are changed when the device receives a wake-up from WDT or other interrupts.

TABLE 4-3: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER

Condition	Program Counter	RCON Register						STKPTR Register	
		SBOREN	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	STKFUL	STKUNF
Power-on Reset	0000h	1	1	1	1	0	0	0	0
RESET Instruction	0000h	u(2)	0	u	u	u	u	u	u
Brown-out Reset	0000h	u(2)	1	1	1	u	0	u	u
$\overline{MCLR}$ during Power-Managed Run Modes	0000h	u(2)	u	1	u	u	u	u	u
$\overline{MCLR}$ during Power-Managed Idle Modes and Sleep Mode	0000h	u(2)	u	1	0	u	u	u	u
WDT Time-out during Full Power or Power-Managed Run Mode	0000h	u(2)	u	0	u	u	u	u	u
$\overline{MCLR}$ during Full Power Execution	0000h	u(2)	u	u	u	u	u	u	u
Stack Full Reset (STVREN = 1)	0000h	u(2)	u	u	u	u	u	1	u
Stack Underflow Reset (STVREN = 1)	0000h	u(2)	u	u	u	u	u	u	1
Stack Underflow Error (not an actual Reset, STVREN = 0)	0000h	u(2)	u	u	u	u	u	u	1
WDT Time-out during Power-Managed Idle or Sleep Modes	PC + 2	u(2)	u	0	0	u	u	u	u
Interrupt Exit from Power-Managed Modes	PC + 2(1)	u(2)	u	u	0	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an interrupt and the GIEH or GIEL bits are set, the PC is loaded with the interrupt vector (008h or 0018h).

2: Reset state is ‘1’ for SBOREN and unchanged for all other Resets when software BOR is enabled (BOREN<1:0> Configuration bits = 01). Otherwise, the Reset state is ‘0’.

TABLE 4-4: REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
RCON	IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	56
STKPTR	STKFUL	STKUNF	—	STKPTR<4:0>					67

Legend: — = unimplemented locations, read as ‘0’. Shaded bits are not used for Resets.

© 2010-2016 Microchip Technology Inc.

The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack (TOS) Special File Registers. Data can also be pushed to, or popped from the stack, using these registers.

A CALL type instruction causes a push onto the stack; the Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). A RETURN type instruction causes a pop from the stack; the contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

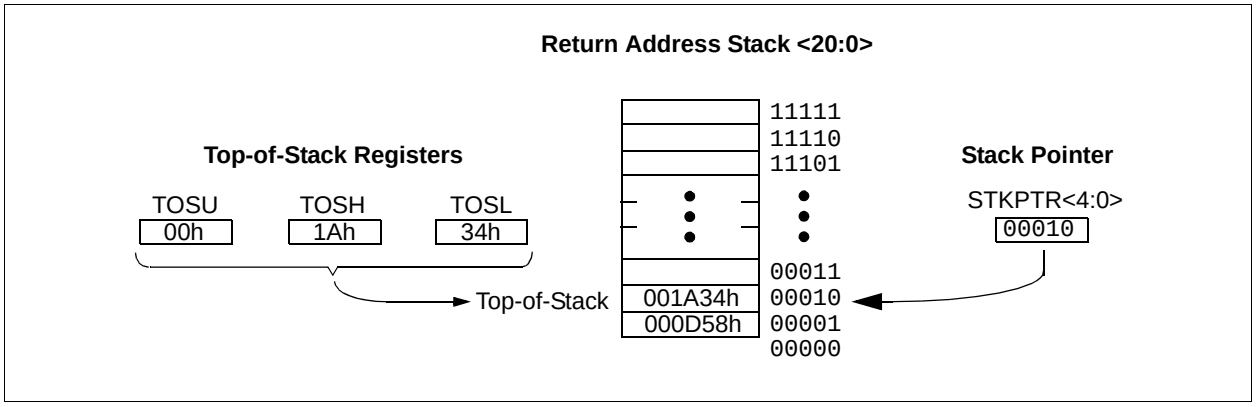
The Stack Pointer is initialized to '00000' after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of '00000'; this is only a Reset value. Status bits indicate if the stack is full or has overflowed or has underflowed.

5.1.2.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 5-2). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

The user must disable the Global Interrupt Enable (GIE) bits while accessing the stack to prevent inadvertent stack corruption.

FIGURE 5-2: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



5.1.2.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 5-1) contains the Stack Pointer value, the STKFUL (stack full) Status bit and the STKUNF (Stack Underflow) Status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. On Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 24.1 "Configuration Bits"** for a description of the device Configuration bits.) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and sets the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or until a POR occurs.

Note:	Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

© 2010-2016 Microchip Technology Inc.

TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FFFh	TOSU	—	—	—	Top-of-Stack, Upper Byte (TOS<20:16>)					---0 0000
FFEh	TOSH	Top-of-Stack, High Byte (TOS<15:8>)								0000 0000
FFDh	TOSL	Top-of-Stack, Low Byte (TOS<7:0>)								0000 0000
FFCh	STKPTR	STKFUL	STKUNF	—	STKPTR<4:0>					00-0 0000
FFBh	PCLATU	—	—	—	Holding Register for PC<20:16>					---0 0000
FFAh	PCLATH	Holding Register for PC<15:8>								0000 0000
FF9h	PCL	Holding Register for PC<7:0>								0000 0000
FF8h	TBLPTRU	—	—	Program Memory Table Pointer Upper Byte(TBLPTR<21:16>)						--00 0000
FF7h	TBLPTRH	Program Memory Table Pointer High Byte(TBLPTR<15:8>)								0000 0000
FF6h	TBLPTRL	Program Memory Table Pointer Low Byte(TBLPTR<7:0>)								0000 0000
FF5h	TABLAT	Program Memory Table Latch								0000 0000
FF4h	PRODH	Product Register, High Byte								xxxx xxxx
FF3h	PRODL	Product Register, Low Byte								xxxx xxxx
FF2h	INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x
FF1h	INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1
FF0h	INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00
FEFh	INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								---- ----
FEEh	POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								---- ----
FEDh	POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								---- ----
FECh	PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								---- ----
FEBh	PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W								---- ----
FEAh	FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0, High Byte				---- 0000
FE9h	FSR0L	Indirect Data Memory Address Pointer 0, Low Byte								xxxx xxxx
FE8h	WREG	Working Register								xxxx xxxx
FE7h	INDF1	Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)								---- ----
FE6h	POSTINC1	Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)								---- ----
FE5h	POSTDEC1	Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)								---- ----
FE4h	PREINC1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)								---- ----
FE3h	PLUSW1	Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W								---- ----
FE2h	FSR1H	—	—	—	—	Indirect Data Memory Address Pointer 1, High Byte				---- 0000
FE1h	FSR1L	Indirect Data Memory Address Pointer 1, Low Byte								xxxx xxxx
FE0h	BSR	—	—	—	—	Bank Select Register				---- 0000
FDFh	INDF2	Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)								---- ----
FDEh	POSTINC2	Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)								---- ----
FDDh	POSTDEC2	Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)								---- ----
FDCh	PREINC2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)								---- ----
FDBh	PLUSW2	Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W								---- ----
FDAh	FSR2H	—	—	—	—	Indirect Data Memory Address Pointer 2, High Byte				---- 0000
FD9h	FSR2L	Indirect Data Memory Address Pointer 2, Low Byte								xxxx xxxx
FD8h	STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx
FD7h	TMR0H	Timer0 Register, High Byte								0000 0000
FD6h	TMR0L	Timer0 Register, Low Byte								xxxx xxxx
FD5h	T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS<2:0>			1111 1111
FD3h	OSCCON	IDLEN	IRCF<2:0>			OSTS	HFIOFS	SCS<1:0>		0011 q000
FD2h	OSCCON2	PLLRDY	SOSCRUN	—	MFIOSEL	SOSCGO	PRISD	MFIOFS	LFIOFS	00-0 01x0

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

Note 1: PIC18(L)F4XK22 devices only.

2: PIC18(L)F2XK22 devices only.

3: PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.

4: PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

14.2 Compare Mode

The Compare mode function described in this section is identical for all CCP and ECCP modules available on this device family.

Compare mode makes use of the 16-bit TimerX resources, Timer1, Timer3 and Timer5. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMRxH:TMRxL register pair. When a match occurs, one of the following events can occur:

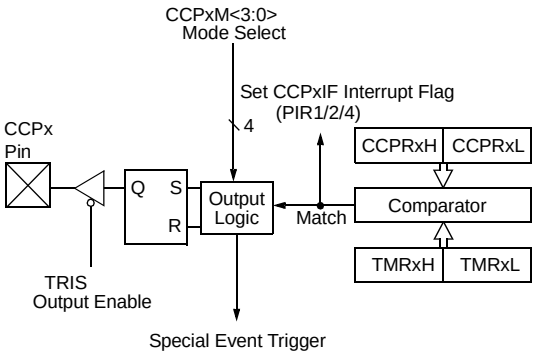
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

All Compare modes can generate an interrupt.

Figure 14-2 shows a simplified diagram of the Compare operation.

FIGURE 14-2: COMPARE MODE OPERATION BLOCK DIAGRAM



Special Event Trigger function on

- ECCP1, ECCP2, ECCP3, CCP4 and CCP5 will:
  - Reset TimerX – TMRxH:TMRxL = 0x0000
  - TimerX Interrupt Flag, (TMRxIF) is not set

Additional Function on

- CCP5 will
  - Set ADCON0<1>, GO/$\overline{\text{DONE}}$ bit to start an ADC Conversion if ADCON<0>, ADON = 1.

14.2.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Some CCPx outputs are multiplexed on a couple of pins. Table 14-2 shows the CCP output pin Multiplexing. Selection of the output pin is determined by the CCPxMX bits in Configuration register 3H (CONFIG3H). Refer to Register 24-4 for more details.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

14.2.2 TimerX MODE RESOURCE

In Compare mode, 16-bit TimerX resource must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 12.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring the 16-bit TimerX resources.

Note:	Clocking TimerX from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TimerX must be clocked from the instruction clock (FOSC/4) or from an external clock source.

14.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the CCPx module does not assert control of the CCPx pin (see the CCPxCON register).

TABLE 14-13: REGISTERS ASSOCIATED WITH ENHANCED PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ECCP1AS	CCP1ASE	CCP1AS<2:0>			PSS1AC<1:0>		PSS1BD<1:0>		202
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				198
ECCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		202
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
ECCP3AS	CCP3ASE	CCP3AS<2:0>			PSS3AC<1:0>		PSS3BD<1:0>		202
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
CCPTMRS0	C3TSEL<1:0>		—	C2TSEL<1:0>		—	C1TSEL<1:0>		201
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	122
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	124
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	118
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	120
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	113
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	115
PMD0	UART2MD	UART1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	52
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
PR2	Timer2 Period Register								—
PR4	Timer4 Period Register								—
PR6	Timer6 Period Register								—
PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	203
PSTR2CON	—	—	—	STR2SYNC	STR2D	STR2C	STR2B	STR2A	203
PSTR3CON	—	—	—	STR3SYNC	STR3D	STR3C	STR3B	STR3A	203
PWM1CON	P1RSEN	P1DC<6:0>							203
PWM2CON	P2RSEN	P2DC<6:0>							203
PWM3CON	P3RSEN	P3DC<6:0>							203
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		166
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		166
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		166
TMR2	Timer2 Register								—
TMR4	Timer4 Register								—
TMR6	Timer6 Register								—
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
TRISE	WPUE3	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)	151

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Enhanced PWM mode.
Note 1: These registers/bits are available on PIC18(L)F4XK22 devices.

TABLE 14-14: CONFIGURATION REGISTERS ASSOCIATED WITH ENHANCED PWM

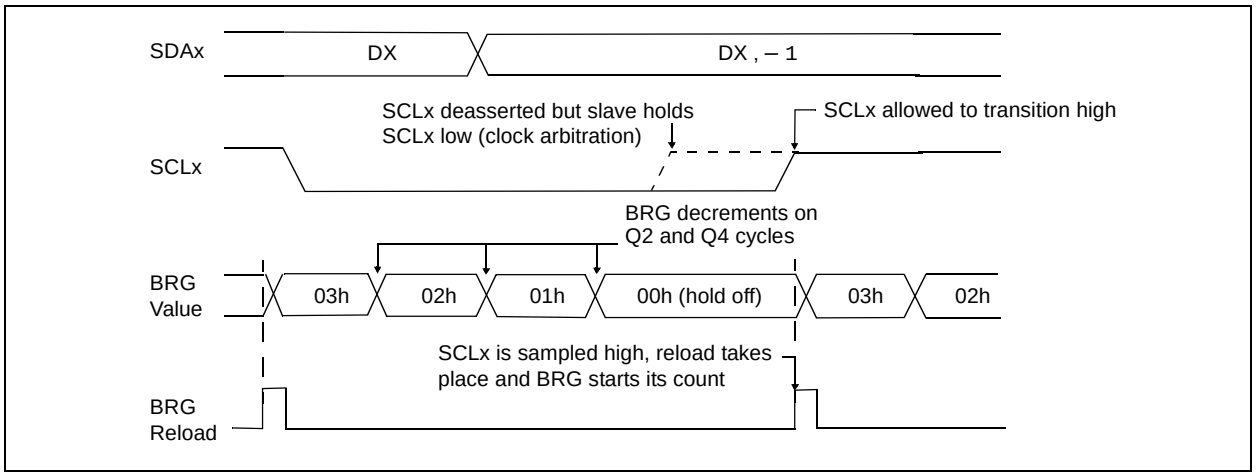
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348

Legend: — = Unimplemented location, read as '0'. Shaded bits are not used by Enhanced PWM mode.

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.

© 2010-2016 Microchip Technology Inc.

15.6.4 I²C MASTER MODE START CONDITION TIMING

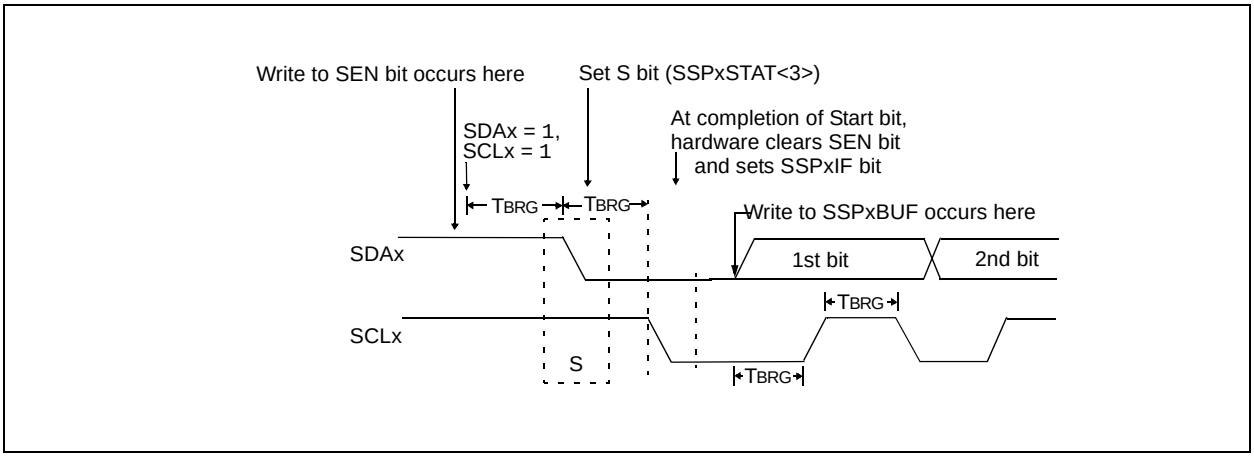
To initiate a Start condition (Figure 15-26), the user sets the Start Enable bit, SEN, of the SSPxCON2 register. If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count.

When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C Specification states that a bus collision cannot occur on a Start.

FIGURE 15-26: FIRST START BIT TIMING



18.0 COMPARATOR MODULE

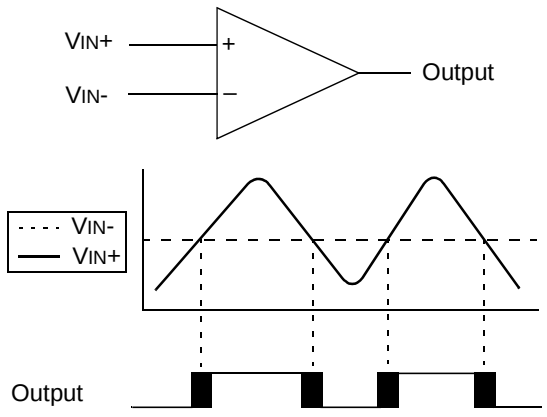
Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. The comparators are very useful mixed signal building blocks because they provide analog functionality independent of the program execution. The analog comparator module includes the following features:

- Independent comparator control
- Programmable input selection
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Programmable Speed/Power optimization
- PWM shutdown
- Programmable and fixed voltage reference
- Selectable Hysteresis

18.1 Comparator Overview

A single comparator is shown in Figure 18-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

FIGURE 18-1: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

© 2010-2016 Microchip Technology Inc.

The module uses the edge Status bits to control the current source output to external analog modules (such as the A/D Converter). Current is only supplied to external modules when only one (but not both) of the Status bits is set, and shuts current off when both bits are either set or cleared. This allows the CTMU to measure current only during the interval between edges. After both Status bits are set, it is necessary to clear them before another measurement is taken. Both bits should be cleared simultaneously, if possible, to avoid re-enabling the CTMU current source.

In addition to being set by the CTMU hardware, the edge Status bits can also be set by software. This is also the user's application to manually enable or disable the current source. Setting either one (but not both) of the bits enables the current source. Setting or clearing both bits at once disables the source.

19.1.5 INTERRUPTS

The CTMU sets its interrupt flag (PIR3<2>) whenever the current source is enabled, then disabled. An interrupt is generated only if the corresponding interrupt enable bit (PIE3<2>) is also set. If edge sequencing is not enabled (i.e., Edge 1 must occur before Edge 2), it is necessary to monitor the edge Status bits and determine which edge occurred last and caused the interrupt.

19.2 CTMU Module Initialization

The following sequence is a general guideline used to initialize the CTMU module:

1. Select the current source range using the IRNG bits (CTMUICON<1:0>).
2. Adjust the current source trim using the ITRIM bits (CTMUICON<7:2>).
3. Configure the edge input sources for Edge 1 and Edge 2 by setting the EDG1SEL and EDG2SEL bits (CTMUCONL<3:2 and 6:5>).
4. Configure the input polarities for the edge inputs using the EDG1POL and EDG2POL bits (CTMUCONL<4,7>). The default configuration is for negative edge polarity (high-to-low transitions).
5. Enable edge sequencing using the EDGSEQEN bit (CTMUCONH<2>). By default, edge sequencing is disabled.
6. Select the operating mode (Measurement or Time Delay) with the TGEN bit. The default mode is Time/Capacitance Measurement.
7. Discharge the connected circuit by setting the IDISSEN bit (CTMUCONH<1>); after waiting a sufficient time for the circuit to discharge, clear IDISSEN.
8. Disable the module by clearing the CTMUEN bit (CTMUCONH<7>).
9. Enable the module by setting the CTMUEN bit.
10. Clear the Edge Status bits: EDG2STAT and EDG1STAT (CTMUCONL<1:0>).
11. Enable both edge inputs by setting the EDGEN bit (CTMUCONH<3>).

Depending on the type of measurement or pulse generation being performed, one or more additional modules may also need to be initialized and configured with the CTMU module:

- Edge Source Generation: In addition to the external edge input pins, both Timer1 and the Output Compare/PWM1 module can be used as edge sources for the CTMU.
- Capacitance or Time Measurement: The CTMU module uses the A/D Converter to measure the voltage across a capacitor that is connected to one of the analog input channels.
- Pulse Generation: When generating system clock independent output pulses, the CTMU module uses Comparator 2 and the associated comparator voltage reference.

DAW **Decimal Adjust W Register**

Syntax: DAW

Operands: None

Operation: If [W<3:0> > 9] or [DC = 1] then
(W<3:0>) + 6 → W<3:0>;
else
(W<3:0>) → W<3:0>;

If [W<7:4> + DC > 9] or [C = 1] then
(W<7:4>) + 6 + DC → W<7:4> ;
else
(W<7:4>) + DC → W<7:4>

Status Affected: C

Encoding:

0000	0000	0000	0111

Description: DAW adjusts the 8-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example1:

```
DAW
```

Before Instruction
W = A5h
C = 0
DC = 0
After Instruction
W = 05h
C = 1
DC = 0

Example 2:

Before Instruction
W = CEh
C = 0
DC = 0
After Instruction
W = 34h
C = 1
DC = 0

DECF **Decrement f**

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

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

Operation: (f) – 1 → dest

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

Encoding:

0000	01da	ffff	ffff

Description: Decrement register ‘f’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default).
If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank.
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

Q Cycle Activity:

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

Example: DECF CNT, 1, 0

Before Instruction
CNT = 01h
Z = 0
After Instruction
CNT = 00h
Z = 1

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

27.11.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 27-7: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)

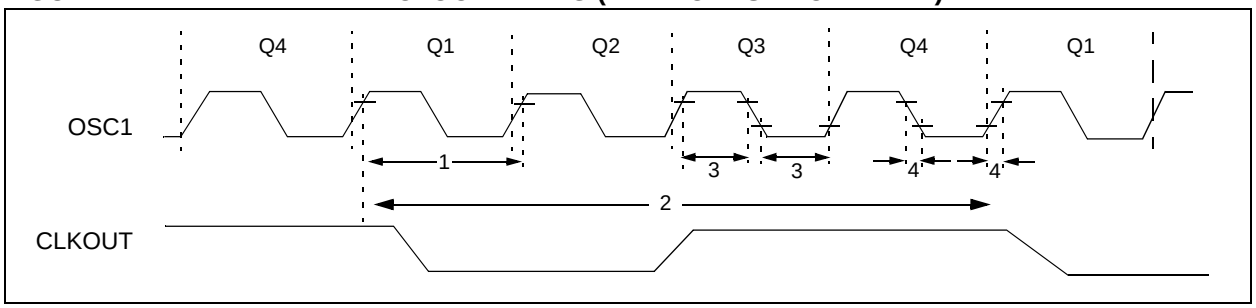


TABLE 27-7: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	FOSC	External CLKIN Frequency(1)	DC	0.5	MHz	EC, ECIO Oscillator mode (low power)
			DC	16	MHz	EC, ECIO Oscillator mode (medium power)
			DC	64	MHz	EC, ECIO Oscillator mode (high power)
		Oscillator Frequency(1)	DC	4	MHz	RC Oscillator mode
			5	200	kHz	LP Oscillator mode
			0.1	4	MHz	XT Oscillator mode
			4	4	MHz	HS Oscillator mode, VDD < 2.7V
			4	16	MHz	HS Oscillator mode, VDD ≥ 2.7V, Medium-Power mode (HSMP)
			4	20	MHz	HS Oscillator mode, VDD ≥ 2.7V, High-Power mode (HSHP)
1	TOSC	External CLKIN Period(1)	2.0	—	µs	EC, ECIO Oscillator mode (low power)
			62.5	—	ns	EC, ECIO Oscillator mode (medium power)
			15.6	—	ns	EC, ECIO Oscillator mode (high power)
		Oscillator Period(1)	250	—	ns	RC Oscillator mode
			5	200	µs	LP Oscillator mode
			0.25	10	µs	XT Oscillator mode
			250	250	ns	HS Oscillator mode, VDD < 2.7V
			62.5	250	ns	HS Oscillator mode, VDD ≥ 2.7V, Medium-Power mode (HSMP)
			50	250	ns	HS Oscillator mode, VDD ≥ 2.7V, High-Power mode (HSHP)
2	TCY	Instruction Cycle Time(1)	62.5	—	ns	TCY = 4/FOSC
3	TOSL, TOSH	External Clock in (OSC1) High or Low Time	2.5	—	µs	LP Oscillator mode
			30	—	ns	XT Oscillator mode
			10	—	ns	HS Oscillator mode
4	TOSR, TOSF	External Clock in (OSC1) Rise or Fall Time	—	50	ns	LP Oscillator mode
			—	20	ns	XT Oscillator mode
			—	7.5	ns	HS Oscillator mode

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

© 2010-2016 Microchip Technology Inc.

FIGURE 27-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)

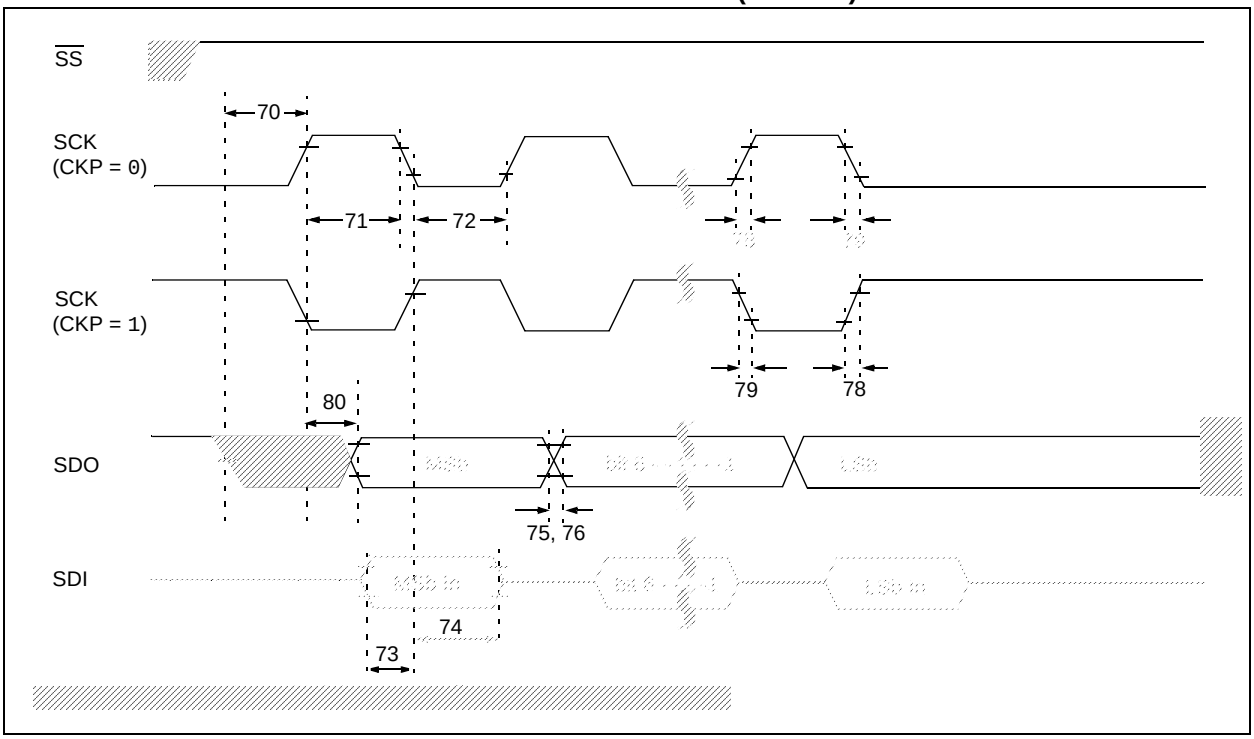
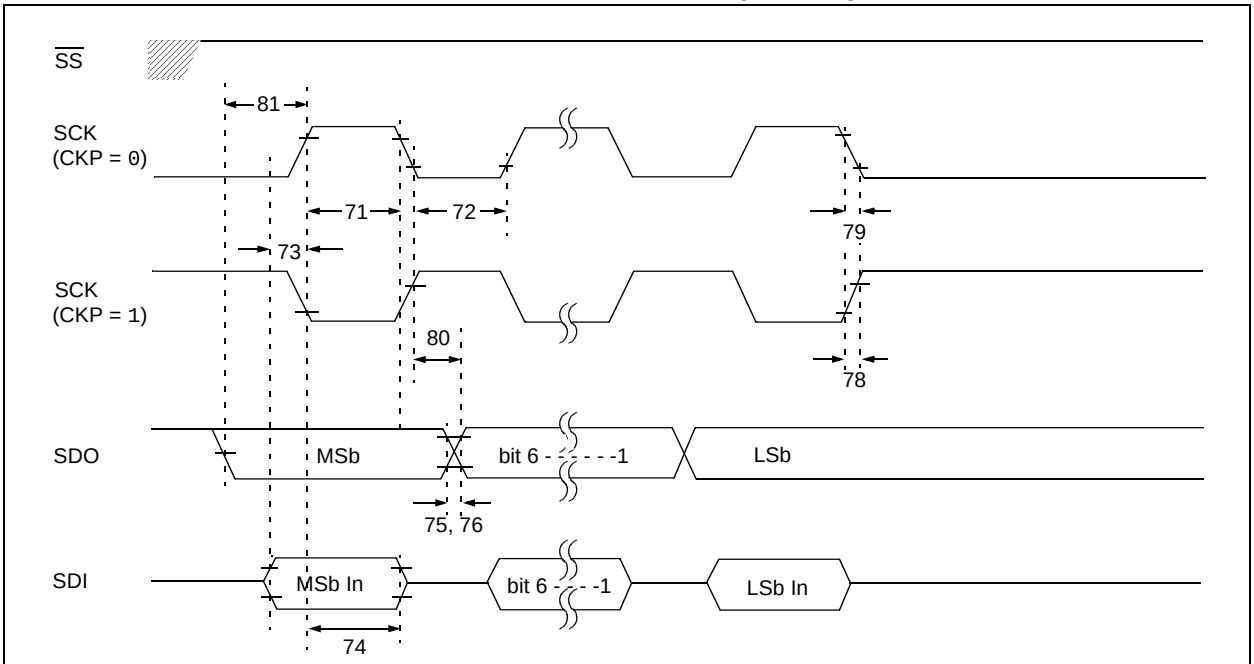


FIGURE 27-14: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)



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

FIGURE 28-7: PIC18LF2X/4XK22 DELTA IPD HIGH/LOW-VOLTAGE DETECT (HLVD)

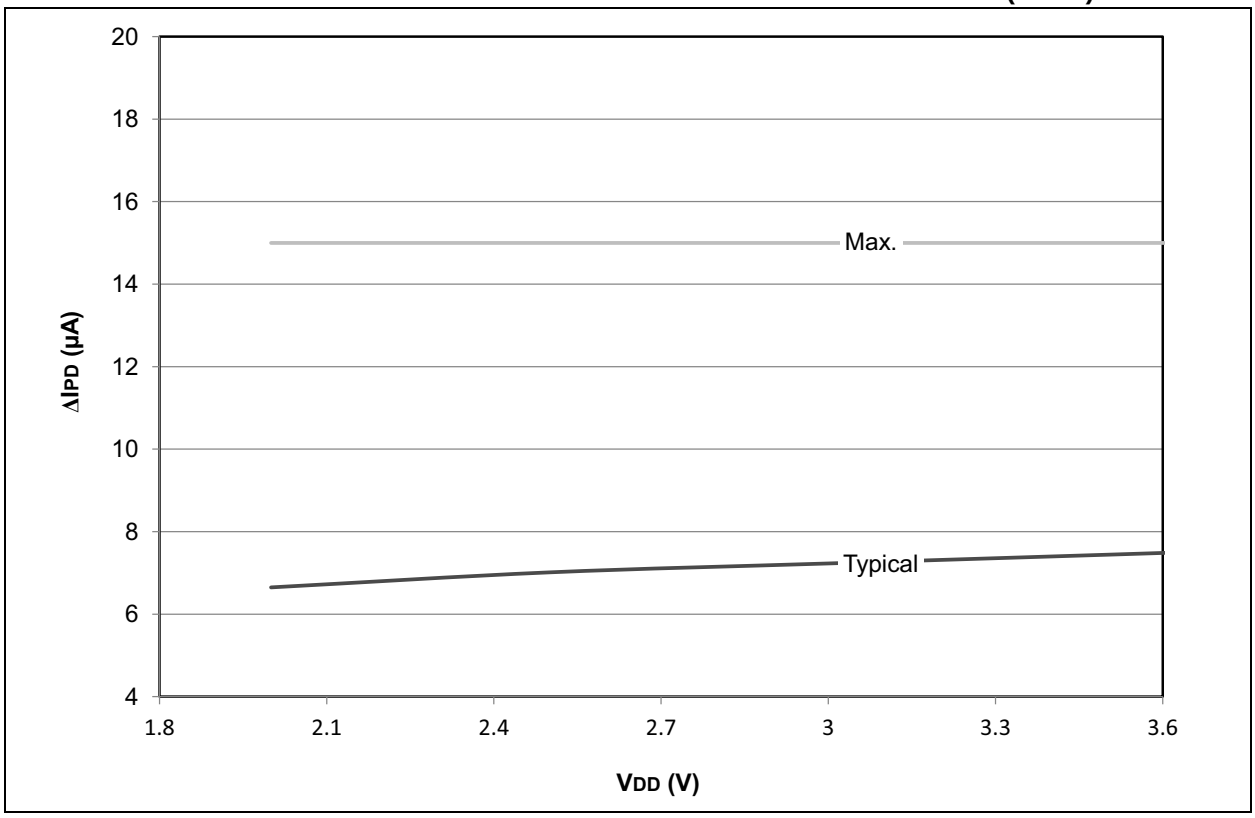


FIGURE 28-8: PIC18F2X/4XK22 DELTA IPD HIGH/LOW-VOLTAGE DETECT (HLVD)

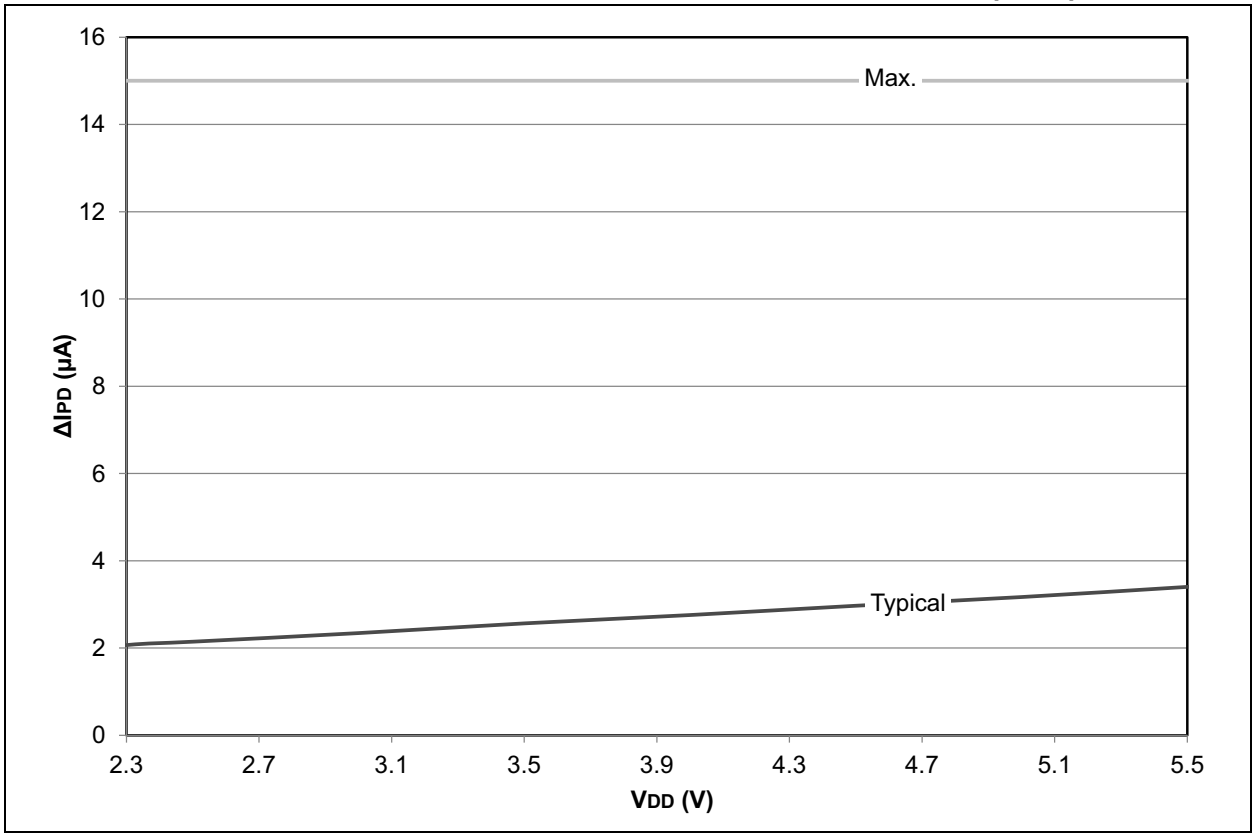


FIGURE 28-34: PIC18LF2X/4XK22 TYPICAL IDD: RC_IDLE LF-INTOSC 31 kHz

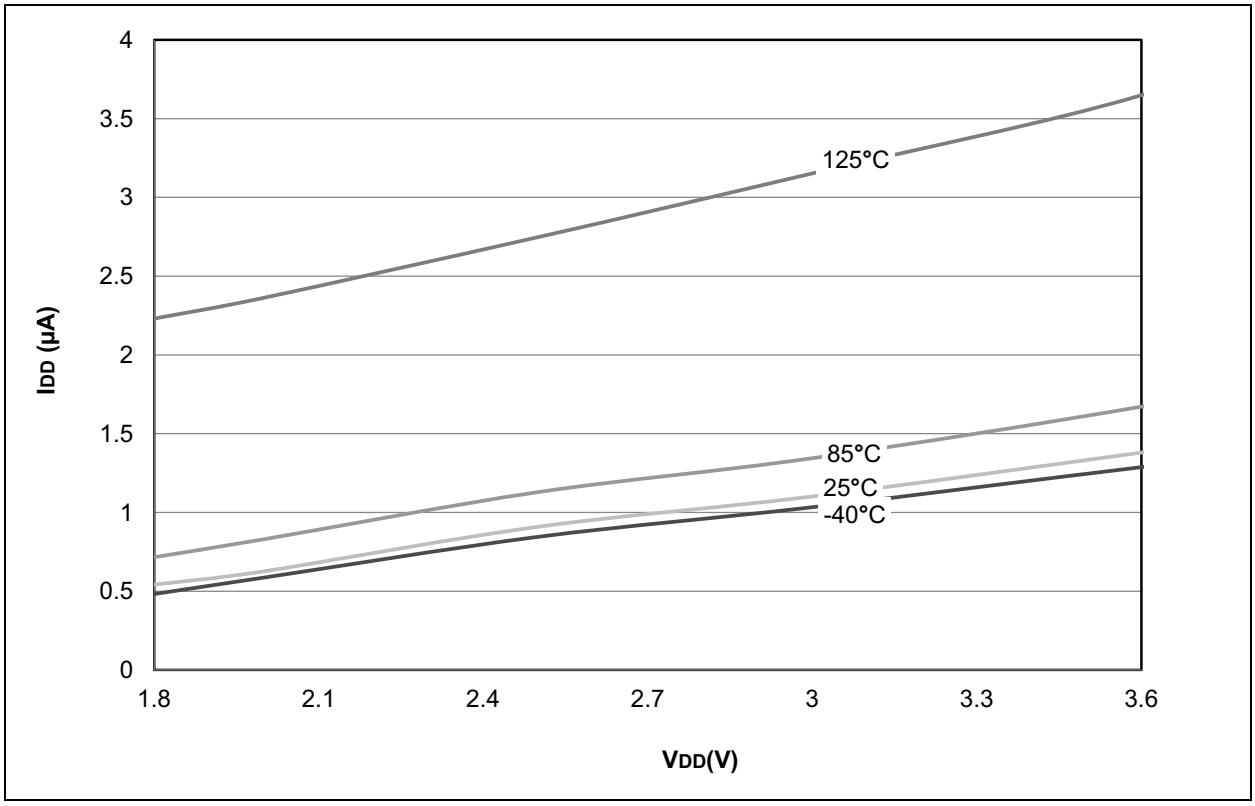


FIGURE 28-35: PIC18LF2X/4XK22 MAXIMUM IDD: RC_IDLE LF-INTOSC 31 kHz

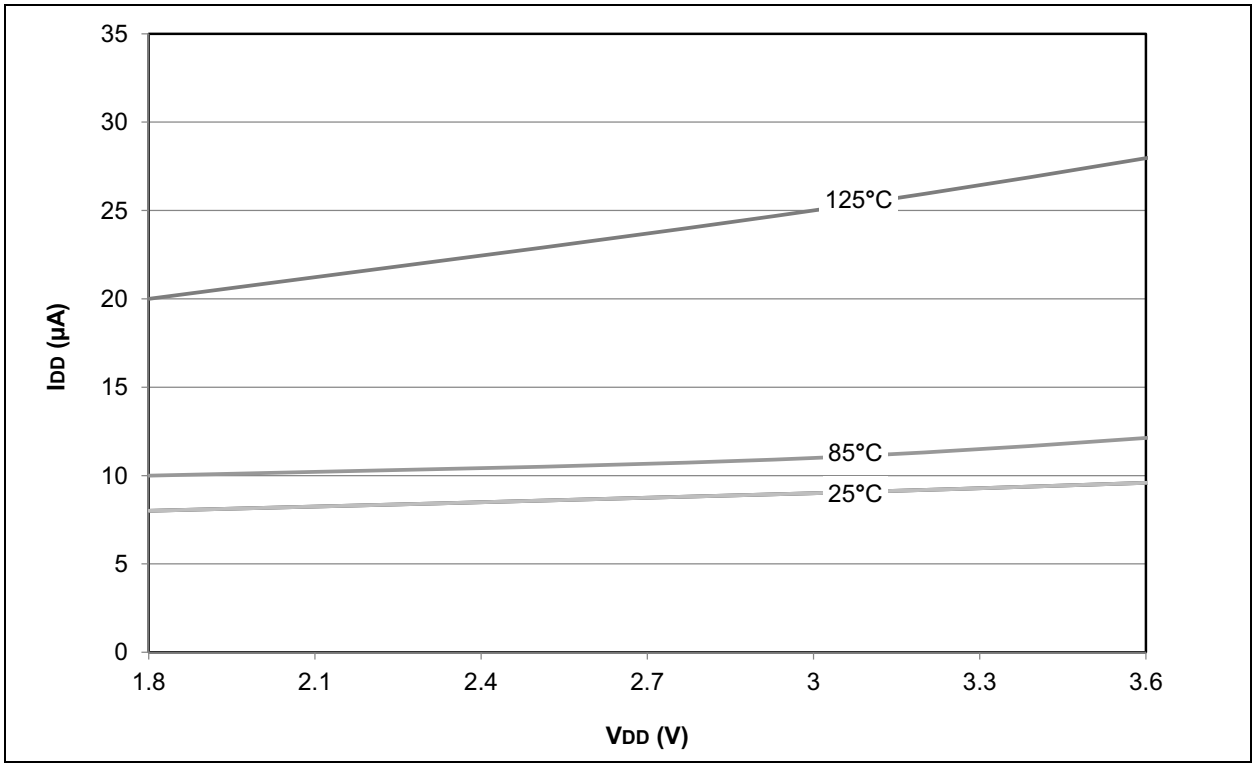


FIGURE 28-68: PIC18LF2X/4XK22 TYPICAL IDD: PRI_IDLE EC with PLL

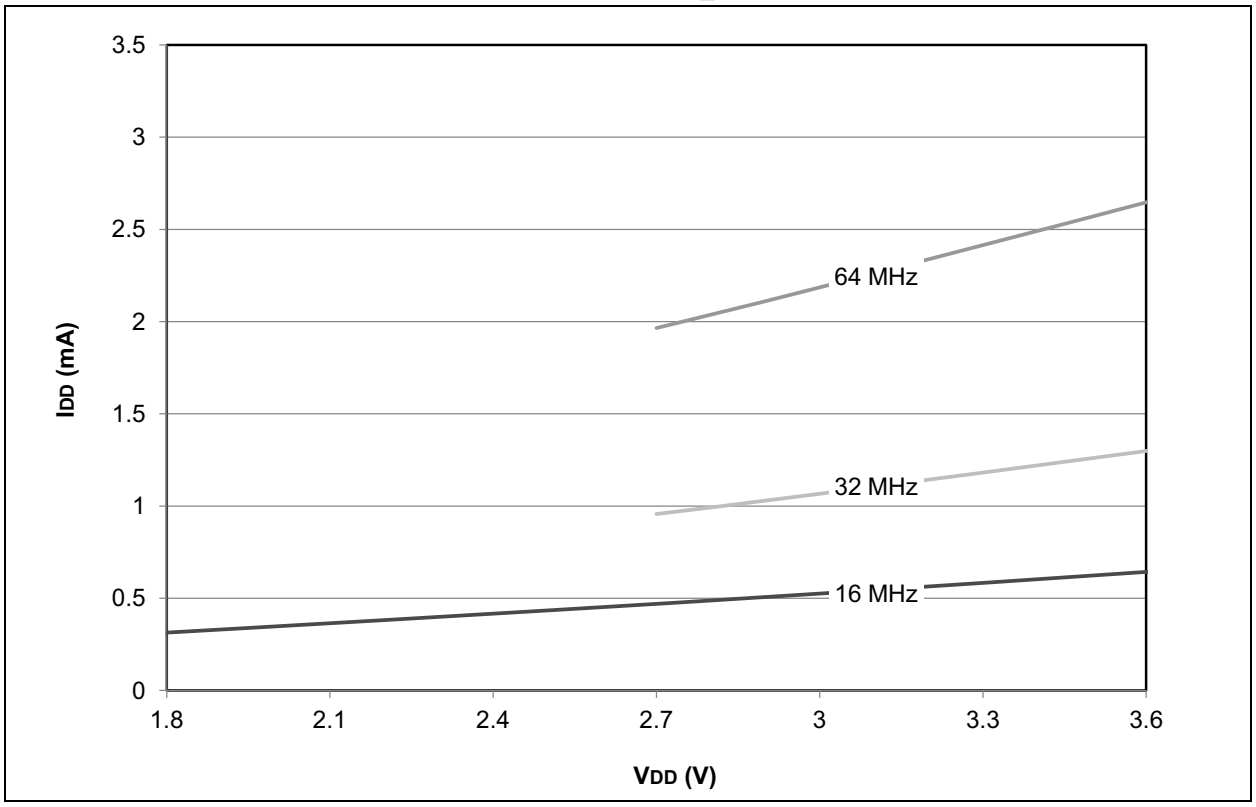
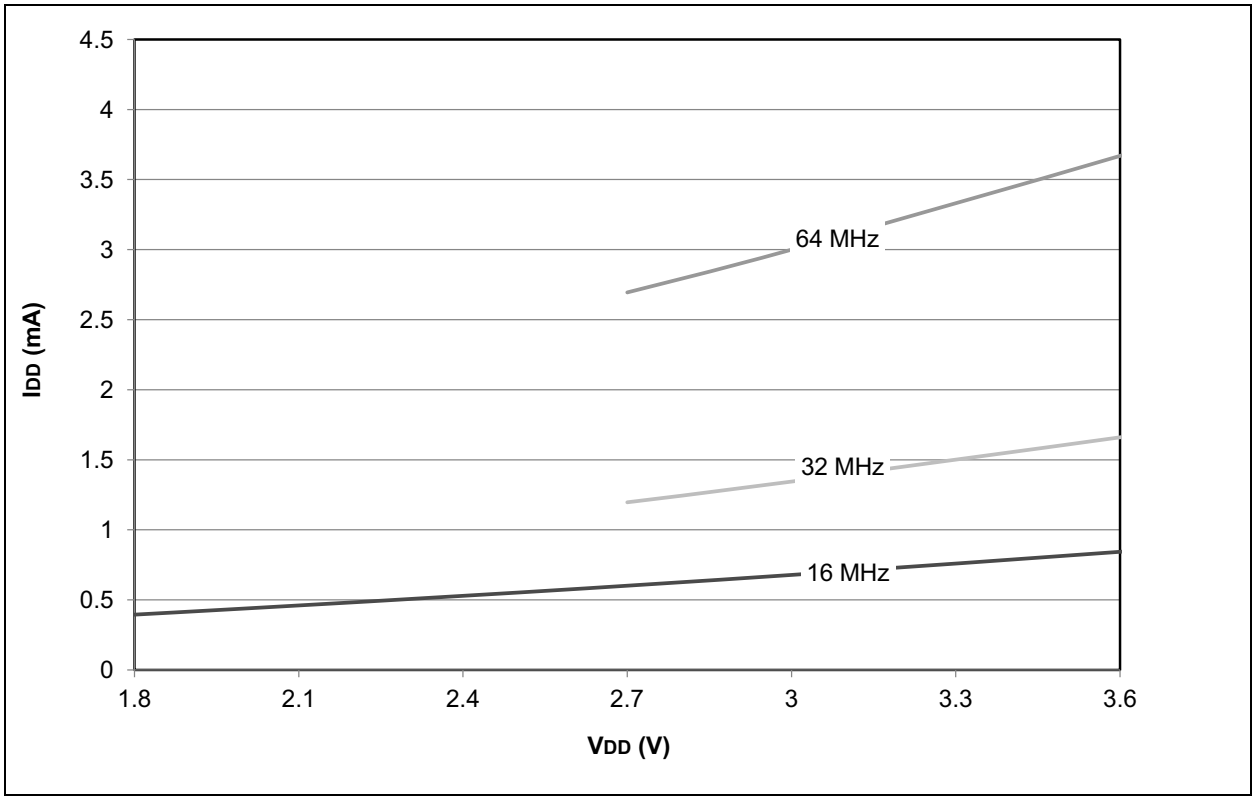


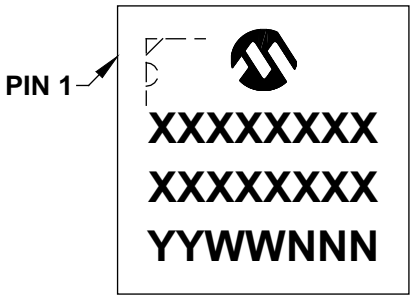
FIGURE 28-69: PIC18LF2X/4XK22 MAXIMUM IDD: PRI_IDLE EC with PLL



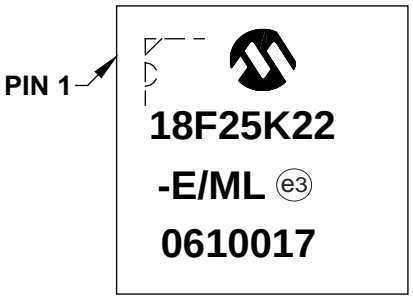
© 2010-2016 Microchip Technology Inc.

Package Marking Information (Continued)

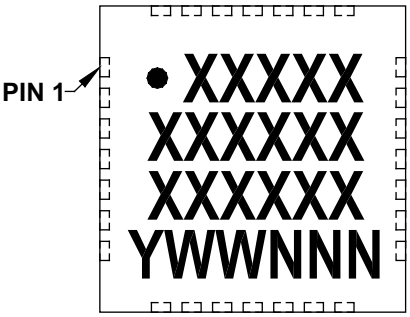
28-Lead QFN (6x6 mm)



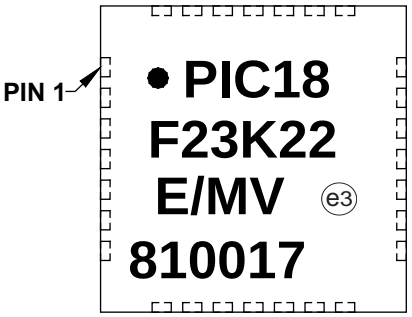
Example



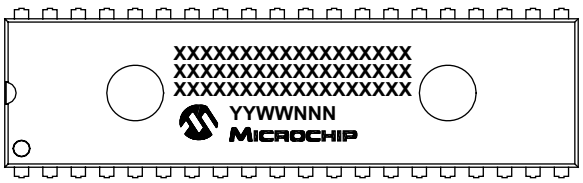
28-Lead UQFN (4x4x0.5 mm)



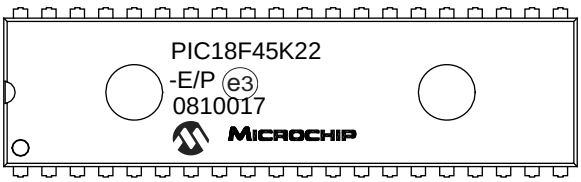
Example



40-Lead PDIP (600 mil)



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.

© 2010-2016 Microchip Technology Inc.