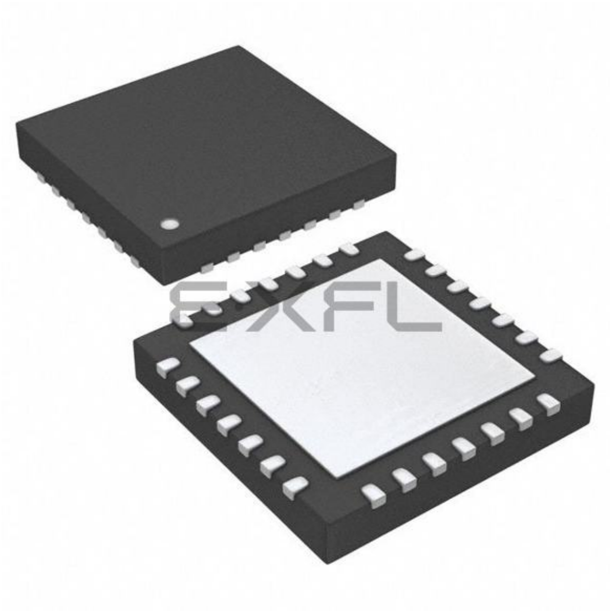


Welcome to E-XFL.COM

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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 35x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25k40-e-mv

TABLE 1-1: DEVICE FEATURES

Features	PIC18(L)F24K40	PIC18(L)F25K40
Program Memory (Bytes)	16384	32768
Program Memory (Instructions)	8192	16384
Data Memory (Bytes)	1024	2048
Data EEPROM Memory (Bytes)	256	256
I/O Ports	A,B,C,E[1]	A,B,C,E[1]
Capture/Compare/PWM Modules (CCP)	2	2
10-Bit Pulse-Width Modulator (PWM)	2	2
10-Bit Analog-to-Digital Module (ADC^2) with Computation Accelerator	4 internal 24 external	4 internal 24 external
Packages	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN	28-pin SPDIP 28-pin SOIC 28-pin SSOP 28-pin QFN 28-pin UQFN
Interrupt Sources	36	
Timers (16-/8-bit)	4/3	
Serial Communications	1 MSSP, 1 EUSART	
Enhanced Complementary Waveform Generator (ECWG)	1	
Zero-Cross Detect (ZCD)	1	
Data Signal Modulator (DSM)	1	
Peripheral Pin Select (PPS)	Yes	
Peripheral Module Disable (PMD)	Yes	
16-bit CRC with NVMSCAN	Yes	
Programmable High/Low-Voltage Detect (HLVD)	Yes	
Programmable Brown-out Reset (BOR)	Yes	
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Overflow, Stack Underflow ($\overline{\text{PWRT}}$, OST), $\overline{\text{MCLR}}$, WDT	
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	
Operating Frequency	DC – 64 MHz	

Note 1: PORTE contains the single RE3 input-only pin.

4.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

4.1 Overview

The oscillator module has multiple clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 4-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz-crystal resonators and ceramic resonators. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The RSTOSC bits of Configuration Word 1 (Register 3-1) determine the type of oscillator that will be used when the device runs after Reset, including when it is first powered up.

If an external clock source is selected, the FEXTOSC bits of Configuration Word 1 must be used in conjunction with the RSTOSC bits to select the External Clock mode.

The external oscillator module can be configured in one of the following clock modes, by setting the FEXTOSC<2:0> bits of Configuration Word 1:

1. ECL – External Clock Low-Power mode (below 100 kHz)
2. ECM – External Clock Medium Power mode (100 kHz to 8 MHz)
3. ECH – External Clock High-Power mode (above 8 MHz)
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (between 100 kHz and 8 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (above 4 MHz)

The ECH, ECM, and ECL Clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS Clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The internal oscillator block produces low and high-frequency clock sources, designated LFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 4-1). Multiple device clock frequencies may be derived from these clock sources.

TABLE 10-5: REGISTER FILE SUMMARY FOR PIC18(L)F24/25K40 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
F4Dh	SCANHADRH	HADR<15:8>								11111111
F4Ch	SCANHADRL	HADR<7:0>								11111111
F4Bh	SCANLADRU	—	—	LADR<21:16>						--000000
F4Ah	SCANLADRH	LADR<15:8>								00000000
F49h	SCANLADRL	LADR<7:0>								00000000
F48h	CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	00000000
F47h	CWG1AS1	—	—	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	--000000
F46h	CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	000101--
F45h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	--x-0000
F44h	CWG1CON0	EN	LD	—	—	—	MODE<2:0>			00---000
F43h	CWG1DBF	—	—	DBF<5:0>						--000000
F42h	CWG1DBR	—	—	DBR<5:0>						--000000
F41h	CWG1ISM	—	—	—	—	—	ISM<2:0>			-----000
F40h	CWG1CLKCON	—	—	—	—	—	—	—	CS	-------0
F3Fh	CLKRCLK	—	—	—	—	—	CLKRxCLK<2:0>			-----000
F3Eh	CLKRCON	CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>			0--10000
F3Dh	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	------00
F3Ch	CM1PCH	—	—	—	—	—	PCH<2:0>			-----000
F3Bh	CM1NCH	—	—	—	—	—	NCH<2:0>			-----000
F3Ah	CM1CON1	—	—	—	—	—	—	INTP	INTN	-----100
F39h	CM1CON0	EN	OUT	—	POL	—	—	HYS	SYNC	00-0--00
F38h	CM2PCH	—	—	—	—	—	C2PCH<2:0>			-----000
F37h	CM2NCH	—	—	—	—	—	C2NCH<2:0>			-----000
F36h	CM2CON1	—	—	—	—	—	—	INTP	INTN	-----100
F35h	CM2CON0	EN	OUT	—	POL	—	—	HYS	SYNC	00-0--00
F34h	DAC1CON1	—	—	—	DAC1R<4:0>					---xxxxx
F33h	DAC1CON0	EN	—	OE1	OE2	PSS<1:0>		—	NSS	0-0000-0
F32h	ZCDCON	SEN	—	OUT	POL	—	—	INTP	INTN	0-x0--00
F31h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0x000000
F30h	HLVDCON1	—	—	—	—	HLVDSEL<3:0>				----0000
F2Fh	HLVDCON0	EN	—	OUT	RDY	—	—	INTH	INTL	0-xx--00
F2Eh	—	Unimplemented								—
F2Dh	WPUE	—	—	—	—	WPUE3	—	—	—	----1---
F2Ch	—	Unimplemented								—
F2Bh	—	Unimplemented								—
F2Ah	INLVLE	—	—	—	—	INLVLE3	—	—	—	----1---
F29h	IOCEP	—	—	—	—	IOCEP3	—	—	—	----0---
F28h	IOCEN	—	—	—	—	IOCEN3	—	—	—	----0---
F27h	IOCEF	—	—	—	—	IOCEF3	—	—	—	----0---

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not available on LF devices.

FIGURE 11-5: PROGRAM FLASH MEMORY READ FLOWCHART

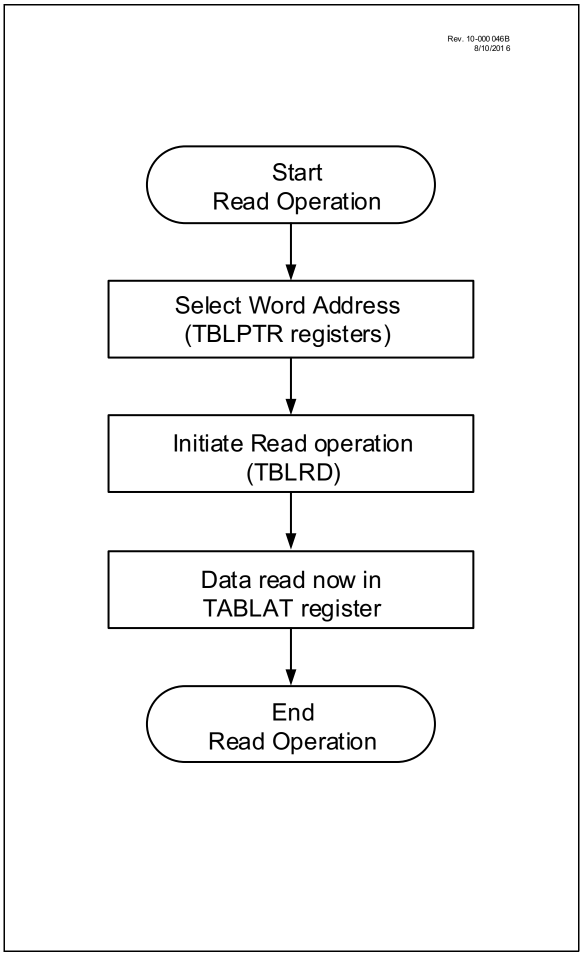
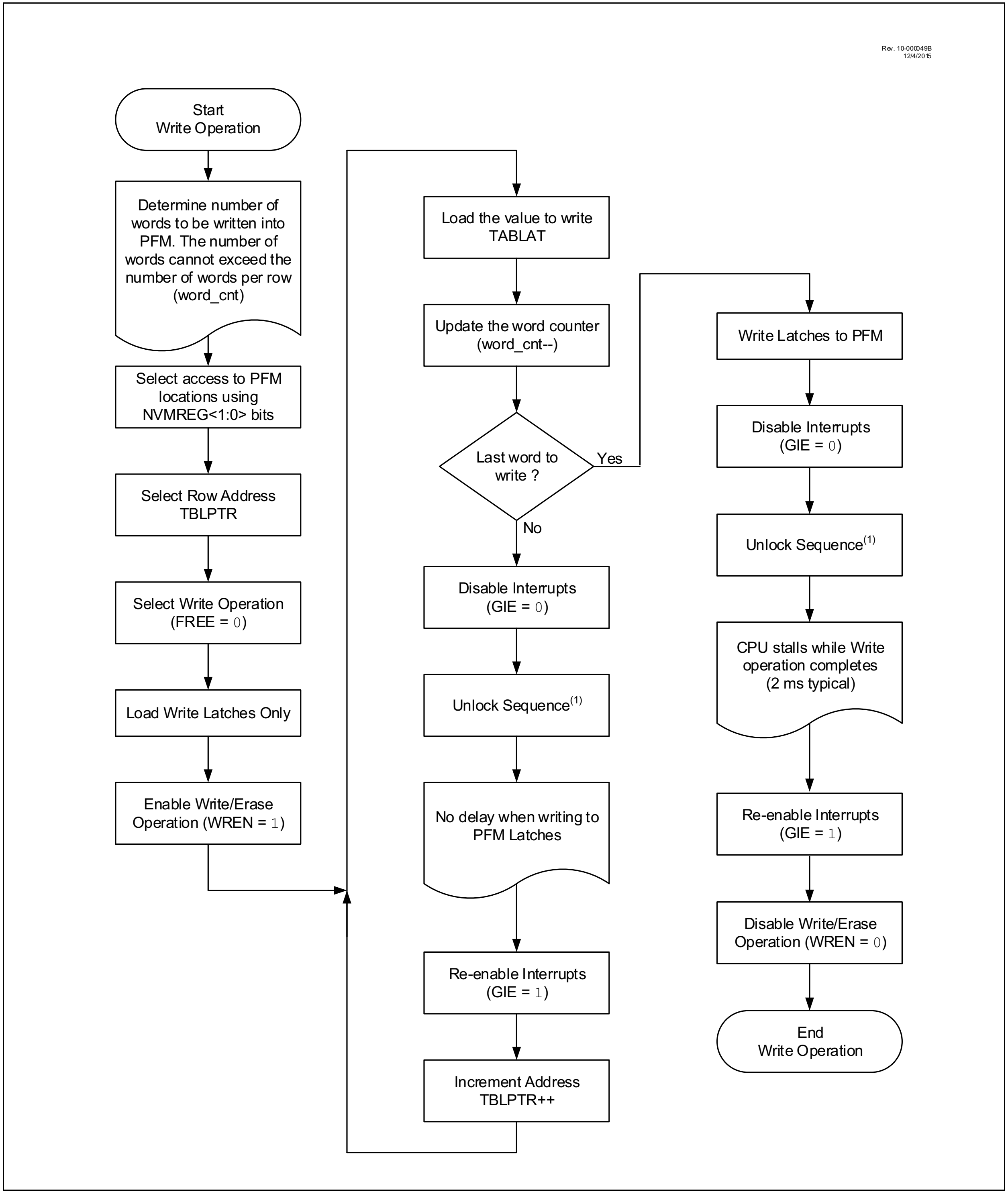


FIGURE 11-9: PROGRAM FLASH MEMORY (PFM) WRITE FLOWCHART


Rev. 10-000049B
12/4/2015
Start
Write Operation
Determine number of words to be written into PFM. The number of words cannot exceed the number of words per row (word_cnt)
Select access to PFM locations using NVMREG<1:0> bits
Select Row Address TBLPTR
Select Write Operation (FREE = 0)
Load Write Latches Only
Enable Write/Erase Operation (WREN = 1)
Load the value to write TABLAT
Update the word counter (word_cnt--)
Last word to write ?
Yes
No
Disable Interrupts (GIE = 0)
Unlock Sequence(1)
No delay when writing to PFM Latches
Re-enable Interrupts (GIE = 1)
Increment Address TBLPTR++
Write Latches to PFM
Disable Interrupts (GIE = 0)
Unlock Sequence(1)
CPU stalls while Write operation completes (2 ms typical)
Re-enable Interrupts (GIE = 1)
Disable Write/Erase Operation (WREN = 0)
End
Write Operation

14.4 INTCON Registers

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

14.5 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Request Flag registers (PIR0, PIR1, PIR2, PIR3, PIR4, PIR5, PIR6 and PIR7).

14.6 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Enable registers (PIE0, PIE1, PIE2, PIE3, PIE4, PIE5, PIE6 and PIE7). When IPEN = 0, the PEIE/GIEL bit must be set to enable any of these peripheral interrupts.

14.7 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are eight Peripheral Interrupt Priority registers (IPR0, IPR1, IPR2, IPR3, IPR4 and IPR5, IPR6 and IPR7). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

20.7 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 20-2. Refer to **Section 1.4.2.2 “Long Bit Names”** for more information.

TABLE 20-2:

Peripheral	Bit Name Prefix
Timer2	T2
Timer4	T4
Timer6	T6

TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		280
PWM3CON	EN	—	OUT	POL	—	—	—	—	279
PWM3DCH	DC<7:0>								281
PWM3DCL	DC<9:8>>		—	—	—	—	—	—	281
PWM4CON	EN	—	OUT	POL	—	—	—	—	279
PWM4DCH	DC<7:0>								281
PWM4DCL	DC<9:8>		—	—	—	—	—	—	281
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	166
PIE4	—	—	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	179
PIR4	—	—	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	171
IPR4	—	—	TMR6IP	TMR5IP	TMR4IP	TMR3IP	TMR2IP	TMR1IP	187
RxyPPS	—	—	—	RxyPPS<4:0>					213
TMR2	TMR2<7:0>								238*
PR2	PR2<7:0>								238*
T2CON	T2ON	T2CKPS<2:0>			T2OUTPS<3:0>				256
T2HLT	T2PSYNC	T2CPOL	T2CSYNC	T2MODE<4:0>					257
T2CLKCON	—	—	—	—	T2CS<3:0>				258
T2RST	—	—	—	—	T2RSEL<3:0>				259
PMD3	—	—	—	—	PWM4MD	PWM3MD	CCP2MD	CCP1MD	67

Legend: - = Unimplemented locations, read as ‘0’, u = unchanged, x = unknown. Shaded cells are not used by the PWM.
* Not a physical location.

REGISTER 25-5: MDSRC: MODULATION SOURCE CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	SRCS<3:0>			
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	

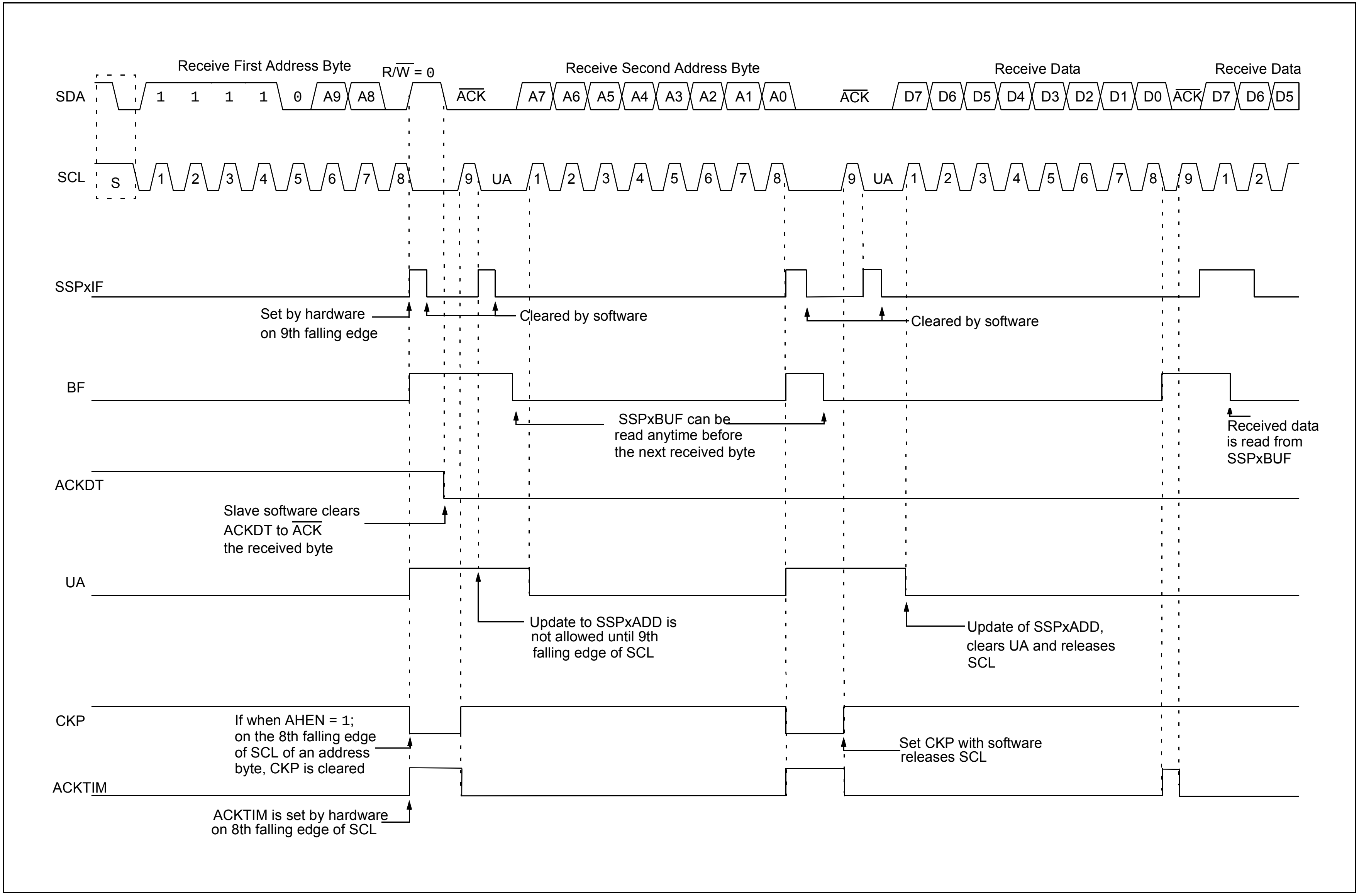
bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **SRCS<3:0>:** Modulator Source Selection bits
See Table 25-3 for signal list

TABLE 25-3: MDSRC SELECTION MUX CONNECTIONS

MDSRCS<3:0>		Connection
1011-1111		Reserved
1010	10	MSSP1 - SDO
1001	9	EUSART TX (TX/CK output)
1000	8	EUSART RX (DT output)
0111	7	CMP2 OUT
0110	6	CMP1 OUT
0101	5	PWM4 OUT
0100	4	PWM3 OUT
0011	3	CCP2 OUT
0010	2	CCP1 OUT
0001	1	MDBIT
0000	0	Pin selected by MDSRCPPS

FIGURE 26-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)


Receive First Address Byte
R/W̅ = 0
Receive Second Address Byte
Receive Data
Receive Data
SDA
1 1 1 1 0 A9 A8
ACK̅
A7 A6 A5 A4 A3 A2 A1 A0
ACK̅
D7 D6 D5 D4 D3 D2 D1 D0
ACK̅
D7 D6 D5
SCL
S
1 2 3 4 5 6 7 8 9 UA
1 2 3 4 5 6 7 8 9 UA
1 2 3 4 5 6 7 8 9 1 2
SSPxIF
Set by hardware on 9th falling edge
Cleared by software
Cleared by software
BF
SSPxBUF can be read anytime before the next received byte
Received data is read from SSPxBUF
ACKDT
Slave software clears ACKDT to ACK̅ the received byte
UA
Update to SSPxADD is not allowed until 9th falling edge of SCL
Update of SSPxADD, clears UA and releases SCL
CKP
If when AHEN = 1; on the 8th falling edge of SCL of an address byte, CKP is cleared
Set CKP with software releases SCL
ACKTIM
ACKTIM is set by hardware on 8th falling edge of SCL

26.10 I^{2}C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPxCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^{2}C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I^{2}C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPxIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSP module, when configured in I^{2}C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur

2: Master mode suspends Start/Stop detection when sending the Start/Stop condition by means of the SEN/PEN control bits. The SSPxIF bit is set at the end of the Start/Stop generation when hardware clears the control bit.

26.10.1 I^{2}C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I^{2}C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/$\overline{\text{Write}}$ (R/$\overline{\text{W}}$) bit. In this case, the R/$\overline{\text{W}}$ bit will be logic ‘0’. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{\text{W}}$ bit. In this case, the R/$\overline{\text{W}}$ bit will be logic ‘1’. Thus, the first byte transmitted is a 7-bit slave address followed by a ‘1’ to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 26.11 “Baud Rate Generator”** for more detail.

26.10.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL 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 26-25).

27.2 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a VOH Mark state which represents a '1' data bit, and a VOL Space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the Mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 27-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

27.2.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 27-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXxREG register.

27.2.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXxSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXxSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCxSTA register enables the EUSART and automatically configures the TXx/CKx I/O pin as an output. If the TXx/CKx pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note:	The TXxIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

27.2.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXxREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXxREG until the Stop bit of the previous character has been transmitted. The pending character in the TXxREG is then transferred to the TSR in one TCY immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXxREG.

27.2.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDxCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 27.5.1.2 "Clock Polarity"**.

27.2.1.4 Transmit Interrupt Flag

The TXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXxREG. In other words, the TXxIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXxREG. The TXxIF flag bit is not cleared immediately upon writing TXxREG. TXxIF becomes valid in the second instruction cycle following the write execution. Polling TXxIF immediately following the TXxREG write will return invalid results. The TXxIF bit is read-only, it cannot be set or cleared by software.

The TXxIF interrupt can be enabled by setting the TXxIE interrupt enable bit of the PIE3 register. However, the TXxIF flag bit will be set whenever the TXxREG is empty, regardless of the state of TXxIE enable bit.

To use interrupts when transmitting data, set the TXxIE bit only when there is more data to send. Clear the TXxIE interrupt enable bit upon writing the last character of the transmission to the TXxREG.

31.0 ANALOG-TO-DIGITAL CONVERTER WITH COMPUTATION (ADC^2) MODULE

The Analog-to-Digital Converter with Computation (ADC^2) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair).

Additionally, the following features are provided within the ADC module:

- 8-bit Acquisition Timer
- Hardware Capacitive Voltage Divider (CVD) support:
  - 8-bit precharge timer
  - Adjustable sample and hold capacitor array
  - Guard ring digital output drive
- Automatic repeat and sequencing:
  - Automated double sample conversion for CVD
  - Two sets of result registers (Result and Previous result)
  - Auto-conversion trigger
  - Internal retrigger
- Computation features:
  - Averaging and low-pass filter functions
  - Reference comparison
  - 2-level threshold comparison
  - Selectable interrupts

Figure 31-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion and upon threshold comparison. These interrupts can be used to wake-up the device from Sleep.

NEGF	Negate f			
Syntax:	NEGF f {,a}			
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]			
Operation:	$(\overline{f}) + 1 \rightarrow f$			
Status Affected:	N, OV, C, DC, Z			
Encoding:	`0110`	`110a`	`ffff`	`ffff`
Description:	Location ‘f’ is negated using two’s complement. The result is placed in the data memory location ‘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 ≤ 95 (5Fh). See **Section 35.2.3 “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 register ‘f’

Example: `NEGF    REG, 1`

Before Instruction
REG = `0011 1010` [3Ah]
After Instruction
REG = `1100 0110` [C6h]

NOP	No Operation			
Syntax:	NOP			
Operands:	None			
Operation:	No operation			
Status Affected:	None			

Encoding:	`0000`
	`1111`
		`0000`
		`xxxx`
			`0000`
			`xxxx`
				`0000`
				`xxxx`

Description:	No operation.			
Words:	1			
Cycles:	1			

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

35.2.2 EXTENDED INSTRUCTION SET

ADDFSR — Add Literal to FSR

Syntax:	ADDFSR f, k
Operands:	0 ≤ k ≤ 63 f ∈ [0, 1, 2]
Operation:	FSR(f) + k → FSR(f)
Status Affected:	None
Encoding:	`1110` `1000` `ffkk` `kkkk`
Description:	The 6-bit literal ‘k’ is added to the contents of the FSR specified by ‘f’.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to FSR

Example: `ADDFSR 2, 23h`

Before Instruction
FSR2 = 03FFh

After Instruction
FSR2 = 0422h

ADDULNK — Add Literal to FSR2 and Return

Syntax:	ADDULNK k
Operands:	0 ≤ k ≤ 63
Operation:	FSR2 + k → FSR2, (TOS) → PC
Status Affected:	None
Encoding:	`1110` `1000` `11kk` `kkkk`
Description:	The 6-bit literal ‘k’ is added to the contents of FSR2. A `RETURN` is then executed by loading the PC with the TOS. The instruction takes two cycles to execute; a `NOP` is performed during the second cycle. This may be thought of as a special case of the `ADDFSR` instruction, where f = 3 (binary ‘`11`’); it operates only on FSR2.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to FSR
No Operation	No Operation	No Operation	No Operation

Example: `ADDULNK 23h`

Before Instruction
FSR2 = 03FFh
PC = 0100h

After Instruction
FSR2 = 0422h
PC = (TOS)

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction syntax then becomes: {label} instruction argument(s).

CALLW Subroutine Call Using WREG

Syntax:	CALLW
Operands:	None
Operation:	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU
Status Affected:	None
Encoding:	`0000` `0000` `0001` `0100`
Description	First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a `NOP` instruction while the new next instruction is fetched. Unlike `CALL`, there is no option to update W, Status or BSR.
Words:	1
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read WREG	PUSH PC to stack	No operation
No operation	No operation	No operation	No operation

Example: `HERE CALLW`

```
Before Instruction
    PC     =  address (HERE)
    PCLATH =  10h
    PCLATU =  00h
    W      =  06h
After Instruction
    PC     =  001006h
    TOS    =  address (HERE + 2)
    PCLATH =  10h
    PCLATU =  00h
    W      =  06h
```

MOVSF Move Indexed to f

Syntax:	MOVSF [z_s], f_d
Operands:	$0 \le z_s \le 127$ $0 \le f_d \le 4095$
Operation:	$((FSR2) + z_s) \rightarrow f_d$
Status Affected:	None

Encoding:				
1st word (source)	`1110`	`1011`	`0zzz`	$\texttt{zzzz}_s$
2nd word (destin.)	`1111`	`ffff`	`ffff`	$\texttt{ffff}_d$

Description:	The contents of the source register are moved to destination register 'f_d'. The actual address of the source register is determined by adding the 7-bit literal offset 'z_s' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f_d' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh). The `MOVSF` instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register. If the resultant source address points to an indirect addressing register, the value returned will be 00h.
Words:	2
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: `MOVSF [05h], REG2`

```
Before Instruction
    FSR2      =  80h
    Contents
    of 85h    =  33h
    REG2      =  11h
After Instruction
    FSR2      =  80h
    Contents
    of 85h    =  33h
    REG2      =  33h
```

TABLE 37-2: SUPPLY CURRENT (I_{DD})$^{(1,2,4)}$

PIC18LF24/25K40			Standard Operating Conditions (unless otherwise stated)					
PIC18F24/25K40								
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max.	Units	Conditions	
							V_{DD}	Note
D100	$I_{DD_{XT4}}$	XT = 4 MHz	—	450	650	μA	3.0V	
D100	$I_{DD_{XT4}}$	XT = 4 MHz	—	550	750	μA	3.0V	
D100A	$I_{DD_{XT4}}$	XT = 4 MHz	—	310	—	μA	3.0V	PMD's all 1's
D100A	$I_{DD_{XT4}}$	XT = 4 MHz	—	410	—	μA	3.0V	PMD's all 1's
D101	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	1.9	2.6	mA	3.0V	
D101	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	2.0	2.7	mA	3.0V	
D101A	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	1.4	—	mA	3.0V	PMD's all 1's
D101A	$I_{DD_{HFO16}}$	HFINTOSC = 16 MHz	—	1.5	—	mA	3.0V	PMD's all 1's
D102	$I_{DD_{HFOPLL}}$	HFINTOSC = 64 MHz	—	7.4	9.4	mA	3.0V	
D102	$I_{DD_{HFOPLL}}$	HFINTOSC = 64 MHz	—	7.5	9.5	mA	3.0V	
D102A	$I_{DD_{HFOPLL}}$	HFINTOSC = 64 MHz	—	5.2	—	mA	3.0V	PMD's all 1's
D102A	$I_{DD_{HFOPLL}}$	HFINTOSC = 64 MHz	—	5.3	—	mA	3.0V	PMD's all 1's
D103	$I_{DD_{HSPLL32}}$	HS+PLL = 64 MHz	—	6.9	8.9	mA	3.0V	
D103	$I_{DD_{HSPLL32}}$	HS+PLL = 64 MHz	—	7.0	9.0	mA	3.0V	
D103A	$I_{DD_{HSPLL32}}$	HS+PLL = 64 MHz	—	4.9	—	mA	3.0V	PMD's all 1's
D103A	$I_{DD_{HSPLL32}}$	HS+PLL = 64 MHz	—	5.0	—	mA	3.0V	PMD's all 1's
D104	$I_{DD_{IDLE}}$	IDLE mode, HFINTOSC = 16 MHz	—	1.05	—	mA	3.0V	
D104	$I_{DD_{IDLE}}$	IDLE mode, HFINTOSC = 16 MHz	—	1.15	—	mA	3.0V	
D105	$I_{DD_{DOZE}}$(3)	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.1	—	mA	3.0V	
D105	$I_{DD_{DOZE}}$(3)	DOZE mode, HFINTOSC = 16 MHz, Doze Ratio = 16	—	1.2	—	mA	3.0V	

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins are outputs driven low; $\overline{\text{MCLR}}$ = V_{DD}; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: $I_{DD_{DOZE}} = [I_{DD_{IDLE}}*(N-1)/N] + I_{DD_{HFO}}16/N$ where N = DOZE Ratio (Register 6-2).

4: PMD bits are all in the default state, no modules are disabled.

FIGURE 37-16: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

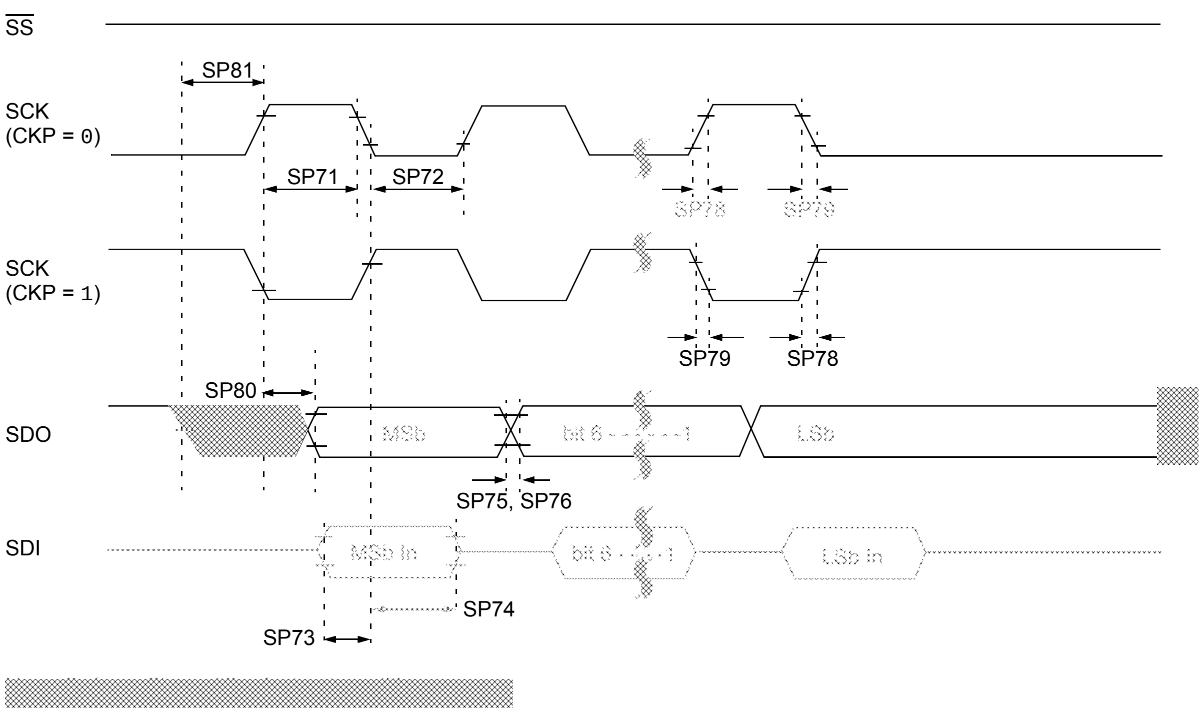
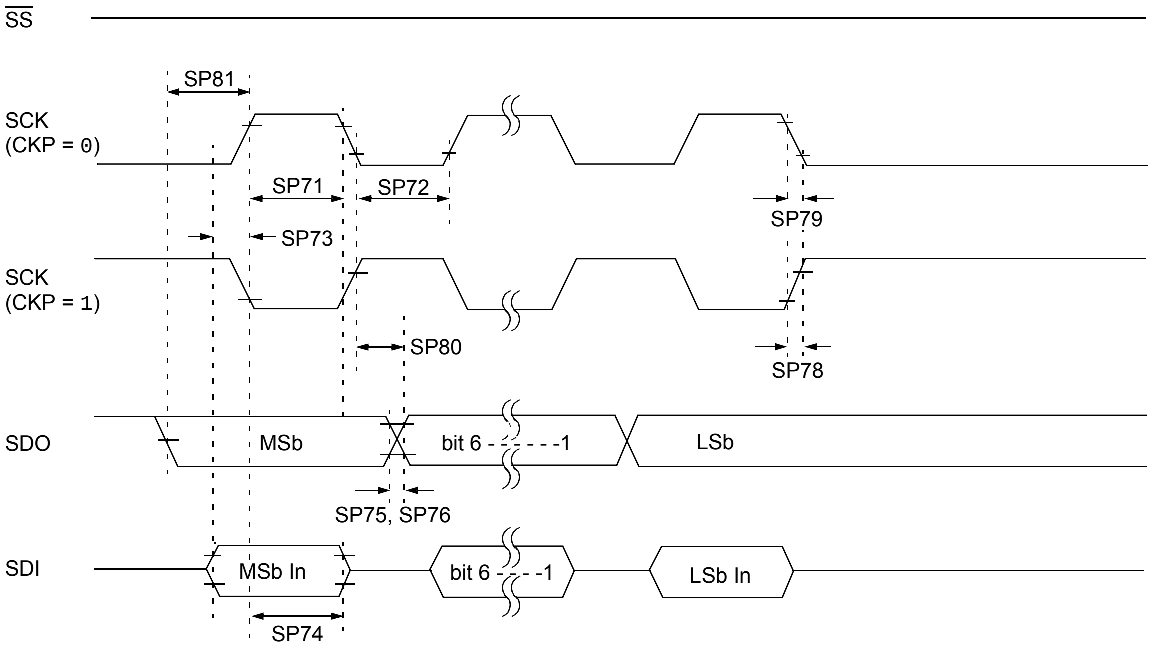


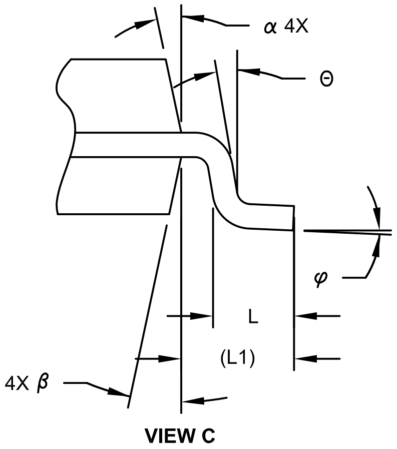
FIGURE 37-17: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)

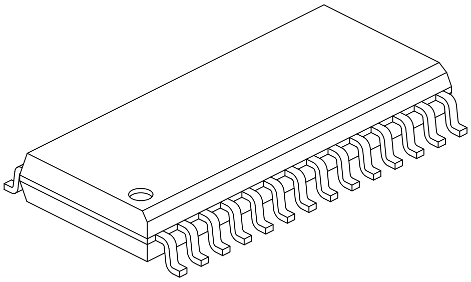


Note: Refer to Figure 37-4 for load conditions.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

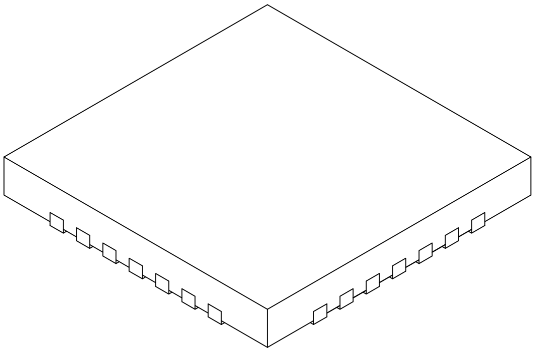
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-152A Sheet 2 of 2