



Welcome to [E-XFL.COM](https://www.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	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf24k40t-i-ml

1.2 Other Special Features

- **Memory Endurance:** The Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 10K for program memory and 100K for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-programmability:** These devices can write to their own program memory spaces under internal software control. By using a boot loader routine located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- **Extended Instruction Set:** The PIC18(L)F2x/4xK40 family introduces an optional extension to the PIC18 instruction set, which adds eight new instructions and an Indexed Addressing mode. This extension, enabled as a device configuration option, has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as C.
- **Enhanced Peripheral Pin Select:** The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins.
- **Enhanced Addressable EUSART:** This serial communication module is capable of standard RS-232 operation and provides support for the LIN bus protocol. Other enhancements include automatic baud rate detection and a 16-bit Baud Rate Generator for improved resolution. When the microcontroller is using the internal oscillator block, the EUSART provides stable operation for applications that talk to the outside world without using an external crystal (or its accompanying power requirement).
- **10-bit A/D Converter with Computation:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reduce code overhead. It has a new module called ADC² with computation features, which provides a digital filter and threshold interrupt functions.
- **Windowed Watchdog Timer (WWDT):**
 - Timer monitoring of overflow and underflow events
 - Variable prescaler selection
 - Variable window size selection
 - All sources configurable in hardware or software

1.3 Details on Individual Family Members

Devices in the PIC18(L)F2x/4xK40 family are available in 28-pin and 40/44-pin packages. The block diagram for this device is shown in Figure 1-1.

The devices have the following differences:

1. Program Flash Memory
2. Data Memory SRAM
3. Data Memory EEPROM
4. A/D channels
5. I/O ports
6. Enhanced USART
7. Input Voltage Range/Power Consumption

All other features for devices in this family are identical. These are summarized in Table 1-1.

The pinouts for all devices are listed in the pin summary tables ().

REGISTER 3-4: Configuration Word 2H (30 0003h): Supervisor

R/W-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{XINST}}$	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	$\overline{\text{ZCD}}$	BORV<1:0>	
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '1'
 -n = Value for blank device '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

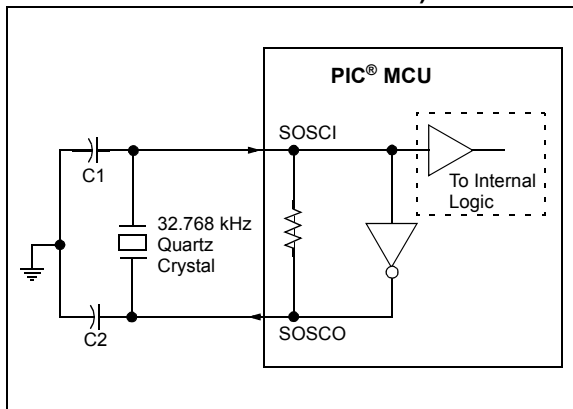
- bit 7 **$\overline{\text{XINST}}$** : Extended Instruction Set Enable bit
 1 = Extended Instruction Set and Indexed Addressing mode disabled (Legacy mode)
 0 = Extended Instruction Set and Indexed Addressing mode enabled
- bit 6 **Unimplemented**: Read as '1'
- bit 5 **$\overline{\text{DEBUG}}$** : Debugger Enable bit
 1 = Background debugger disabled
 0 = Background debugger enabled
- bit 4 **STVREN**: Stack Overflow/Underflow Reset Enable bit
 1 = Stack Overflow or Underflow will cause a Reset
 0 = Stack Overflow or Underflow will not cause a Reset
- bit 3 **PPS1WAY**: PPSLOCKED bit One-Way Set Enable bit
 1 = The PPSLOCKED bit can only be set once after an unlocking sequence is executed; once PPSLOCK is set, all future changes to PPS registers are prevented
 0 = The PPSLOCKED bit can be set and cleared as needed (provided an unlocking sequence is executed)
- bit 2 **$\overline{\text{ZCD}}$** : ZCD Disable bit
 1 = ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of ZCDCON
 0 = ZCD always enabled, ZCDMD bit is ignored
- bit 1-0 **BORV<1:0>**: Brown-out Reset Voltage Selection bit⁽¹⁾
 PIC18F2xK40 device:
 11 = Brown-out Reset Voltage (VBOR) set to 2.45V
 10 = Brown-out Reset Voltage (VBOR) set to 2.45V
 01 = Brown-out Reset Voltage (VBOR) set to 2.7V
 00 = Brown-out Reset Voltage (VBOR) set to 2.85V
 PIC18LF2xK40 device:
 11 = Brown-out Reset Voltage (VBOR) set to 1.90V
 10 = Brown-out Reset Voltage (VBOR) set to 2.45V
 01 = Brown-out Reset Voltage (VBOR) set to 2.7V
 00 = Brown-out Reset Voltage (VBOR) set to 2.85V

Note 1: The higher voltage setting is recommended for operation at or above 16 MHz.

4.3.1.5 Secondary Oscillator

The secondary oscillator is a separate oscillator block that can be used as an alternate system clock source. The secondary oscillator is optimized for 32.768 kHz, and can be used with an external crystal oscillator connected to the SOSC1 and SOSCO device pins, or an external clock source connected to the SOSCIN pin. The secondary oscillator can be selected during run-time using clock switching. Refer to **Section 4.4 “Clock Switching”** for more information.

FIGURE 4-5: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)



Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

- 2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.
- 3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, “Crystal Oscillator Basics and Crystal Selection for PIC® and PIC® Devices” (DS00826)
- AN849, “Basic PIC® Oscillator Design” (DS00849)
- AN943, “Practical PIC® Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)
- TB097, “Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS” (DS91097)
- AN1288, “Design Practices for Low-Power External Oscillators” (DS01288)

4.3.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> bits in the OSCCON1 register to switch the system clock source to the internal oscillator during run-time. See **Section 4.4 “Clock Switching”** for more information.

In INTOSC mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory-calibrated and operates from 1 to 64 MHz. The frequency of HFINTOSC can be selected through the OSCFRQ Frequency Selection register, and fine-tuning can be done via the OSCTUNE register.
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory-calibrated and operates at 31 kHz.

7.0 PERIPHERAL MODULE DISABLE (PMD)

Sleep, Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume some amount of power. There may be cases where the application needs what these modes do not provide: the ability to allocate limited power resources to the CPU while eliminating power consumption from the peripherals.

The PIC18(L)F2x/4xK40 family addresses this requirement by allowing peripheral modules to be selectively enabled or disabled, placing them into the lowest possible power mode.

For legacy reasons, all modules are ON by default following any Reset.

7.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
- Any SFR becomes “unimplemented”
 - Writing is disabled
 - Reading returns 00h
- I/O functionality is prioritized as per **Section 15.1, I/O Priorities**
- All associated Input Selection registers are also disabled

7.2 Enabling a Module

When the PMD register bit is cleared, the module is re-enabled and will be in its Reset state (Power-on Reset). SFR data will reflect the POR Reset values.

Depending on the module, it may take up to one full instruction cycle for the module to become active. There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

7.3 Effects of a Reset

Following any Reset, each control bit is set to ‘0’, enabling all modules.

7.4 System Clock Disable

Setting SYSCMD (PMD0, Register 7-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

REGISTER 9-2: WDTCON1: WATCHDOG TIMER CONTROL REGISTER 1

U-0	R/W ⁽³⁾ -q/q ⁽¹⁾	R/W ⁽³⁾ -q/q ⁽¹⁾	R/W ⁽³⁾ -q/q ⁽¹⁾	U-0	R/W ⁽⁴⁾ -q/q ⁽²⁾	R/W ⁽⁴⁾ -q/q ⁽²⁾	R/W ⁽⁴⁾ -q/q ⁽²⁾
—	WDTCS<2:0>			—	WINDOW<2:0>		
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

q = Value depends on condition

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **WDTCS<2:0>:** Watchdog Timer Clock Select bits

111 = Reserved

•

•

•

010 = Reserved

001 = MFINTOSC 31.25 kHz

000 = LFINTOSC 31 kHz

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **WINDOW<2:0>:** Watchdog Timer Window Select bits

WINDOW<2:0>	Window delay Percent of time	Window opening Percent of time
111	N/A	100
110	12.5	87.5
101	25	75
100	37.5	62.5
011	50	50
010	62.5	37.5
001	75	25
000	87.5	12.5

Note 1: If WDTCCS <2:0> in CONFIG3H = 111, the Reset value of WDTCS<2:0> is 000.

2: The Reset value of WINDOW<2:0> is determined by the value of WDTCCS<2:0> in the CONFIG3H register.

3: If WDTCCS<2:0> in CONFIG3H ≠ 111, these bits are read-only.

4: If WDTCCS<2:0> in CONFIG3H ≠ 111, these bits are read-only.

10.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate buses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Program Flash Memory and Data EEPROM Memory is provided in **Section 11.0 “Nonvolatile Memory (NVM) Control”**.

10.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2 Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2 Mbyte address will return all '0's (a NOP instruction).

These devices contains the following:

- PIC18(L)F24K40: 16 Kbytes of Flash memory, up to 8,192 single-word instructions
- PIC18(L)F25K40: 32 K bytes of Flash memory, up to 16,384 single-word instructions

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

Note: For memory information on this family of devices, see Table 10-1 and Table 10-2.

10.2 Register Definitions: Stack Pointer

REGISTER 10-1: STKPTR: STACK POINTER REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	STKPTR<4:0>				
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented

C = Clearable only bit

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **STKPTR<4:0>:** Stack Pointer Location bits

10.2.1 STACK OVERFLOW AND UNDERFLOW RESETS

Device Resets on Stack Overflow and Stack Underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. When STVREN is set, a Full or Underflow condition will set the appropriate STKOVF or STKUNF bit and then cause a device Reset. When STVREN is cleared, a Full or Underflow condition will set the appropriate STKOVF or STKUNF bit but not cause a device Reset. The STKOVF or STKUNF bits are cleared by the user software or a Power-on Reset.

10.2.2 FAST REGISTER STACK

A fast register stack is provided for the Status, WREG and BSR registers, to provide a "fast return" option for interrupts. The stack for each register is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the stack registers. The values in the registers are then loaded back into their associated registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high priority interrupts are enabled, the stack registers cannot be used reliably to return from low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten. In these cases, users must save the key registers by software during a low priority interrupt.

If interrupt priority is not used, all interrupts may use the fast register stack for returns from interrupt. If no interrupts are used, the fast register stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the fast register stack. A RETURN, FAST instruction is then executed to restore these registers from the fast register stack.

Example 10-1 shows a source code example that uses the fast register stack during a subroutine call and return.

EXAMPLE 10-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                     ;SAVED IN FAST REGISTER
                     ;STACK
                     .
                     .
SUB1                  .
                     .
                     .
RETURN, FAST          ;RESTORE VALUES SAVED
                     ;IN FAST REGISTER STACK
```


TABLE 10-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F24/25K40 DEVICES

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FD7h	PCON0	FAFh	T6TMR	F87h	—
FFEh	TOSH	FD6h	T0CON1	FAEh	CCPTMRS	F86h	—
FFDh	TOSL	FD5h	T0CON0	FADh	CCP1CAP	F85h	LATC
FFCh	STKPTR	FD4h	TMR0H	FACH	CCP1CON	F84h	LATB
FFBh	PCLATU	FD3h	TMR0L	FABh	CCP1H	F83h	LATA
FFAh	PCLATH	FD2h	T1CLK	FAAh	CCP1L	F82h	NVMCON2
FF9h	PCL	FD1h	T1GATE	FA9h	CCP2CAP	F81h	NVMCON1
FF8h	TBLPTRU	FD0h	T1GCON	FA8h	CCP2CON	F80h	NVMDAT
FF7h	TBLPTRH	FCFh	T1CON	FA7h	CCP2H	F7Fh	—
FF6h	TBLPTRL	FCEh	TMR1H	FA6h	CCP2L	F7Eh	NVMADRL
FF5h	TABLAT	FCDh	TMR1L	FA5h	PWM3CON	F7Dh	CRCCON1
FF4h	PRODH	FCCh	T3CLK	FA4h	PWM3DCH	F7Ch	CRCCON0
FF3h	PRODL	FCBh	T3GATE	FA3h	PWM3DCL	F7Bh	CRCXORH
FF2h	INTCON	FCAh	T3GCON	FA2h	PWM4CON	F7Ah	CRCXORL
FF1h	—	FC9h	T3CON	FA1h	PWM4DCH	F79h	CRCSHIFTH
FF0h	—	FC8h	TMR3H	FA0h	PWM4DCL	F78h	CRCSHIFTL
FEFh	INDF0 ⁽¹⁾	FC7h	TMR3L	F9Fh	BAUD1CON	F77h	CRCACCH
FEeh	POSTINC0 ⁽¹⁾	FC6h	T5CLK	F9Eh	TX1STA	F76h	CRCACCL
FEDh	POSTDEC0 ⁽¹⁾	FC5h	T5GATE	F9Dh	RC1STA	F75h	CRCDATH
FECh	PREINC0 ⁽¹⁾	FC4h	T5GCON	F9Ch	SP1BRGH	F74h	CRCDATL
FEbh	PLUSW0 ⁽¹⁾	FC3h	T5CON	F9Bh	SP1BRGL	F73h	ADFLTRH
FEAh	FSR0H	FC2h	TMR5H	F9Ah	TX1REG	F72h	ADFLTRL
FE9h	FSR0L	FC1h	TMR5L	F99h	RC1REG	F71h	ADACCH
FE8h	WREG	FC0h	T2RST	F98h	SSP1CON3	F70h	ADACCL
FE7h	INDF1 ⁽¹⁾	FBFh	T2CLKCON	F97h	SSP1CON2	F6Fh	ADERRH
FE6h	POSTINC1 ⁽¹⁾	FBEh	T2HLT	F96h	SSP1CON1	F6Eh	ADERRL
FE5h	POSTDEC1 ⁽¹⁾	FBDh	T2CON	F95h	SSP1STAT	F6Dh	ADUTHH
FE4h	PREINC1 ⁽¹⁾	FBCh	T2PR	F94h	SSP1MSK	F6Ch	ADUTHL
FE3h	PLUSW1 ⁽¹⁾	FBBh	T2TMR	F93h	SSP1ADD	F6Bh	ADLTHH
FE2h	FSR1H	FBAh	T4RST	F92h	SSP1BUF	F6Ah	ADLTHL
FE1h	FSR1L	FB9h	T4CLKCON	F91h	PORTE	F69h	ADSTPTH
FE0h	BSR	FB8h	T4HLT	F90h	—	F68h	ADSTPTL
FDFh	INDF2 ⁽¹⁾	FB7h	T4CON	F8Fh	PORTC	F67h	ADCNT
FDEh	POSTINC2 ⁽¹⁾	FB6h	T4PR	F8Eh	PORTB	F66h	ADRPT
FDDh	POSTDEC2 ⁽¹⁾	FB5h	T4TMR	F8Dh	PORTA	F65h	ADSTAT
FDCh	PREINC2 ⁽¹⁾	FB4h	T6RST	F8Ch	—	F64h	ADRESH
FDBh	PLUSW2 ⁽¹⁾	FB3h	T6CLKCON	F8Bh	—	F63h	ADRESL
FDAh	FSR2H	FB2h	T6HLT	F8Ah	TRISC	F62h	ADPREVH
FD9h	FSR2L	FB1h	T6CON	F89h	TRISB	F61h	ADPREVL
FD8h	STATUS	FB0h	T6PR	F88h	TRISA	F60h	ADCON0

Note 1: This is not a physical register.

13.6 CRC Check Value

The CRC check value will be located in the CRCACC registers after the CRC calculation has finished. The check value will depend on two mode settings of the CRCCON register: ACCM and SHIFTM.

When the ACCM bit is set, the CRC module augments the data with a number of zeros equal to the length of the polynomial to align the final check value. When the ACCM bit is not set, the CRC will stop at the end of the data. A number of zeros equal to the length of the polynomial can then be entered into CRCDAT to find the same check value as augmented mode. Alternatively, the expected check value can be entered at this point to make the final result equal 0.

When the CRC check value is computed with the SHIFTM bit set, selecting LSb first, and the ACCM bit is set then the final value in the CRCACC registers will be reversed such that the LSb will be in the MSb position and vice versa. This is the expected check value in bit reversed form. If you are creating a check value to be appended to a data stream then a bit reversal must be performed on the final value to achieve the correct checksum. You can use the CRC to do this reversal by the following method:

- Save CRCACC value in user RAM space
- Clear the CRCACC registers
- Clear the CRCXOR registers
- Write the saved CRCACC value to the CRCDAT input

The properly oriented check value will be in the CRCACC registers as the result.

13.7 CRC Interrupt

The CRC will generate an interrupt when the BUSY bit transitions from 1 to 0. The CRCIF Interrupt Flag bit of the PIR7 register is set every time the BUSY bit transitions, regardless of whether or not the CRC interrupt is enabled. The CRCIF bit can only be cleared in software. The CRC interrupt enable is the CRCIE bit of the PIE7 register.

13.8 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

1. Determine if the automatic program memory scan will be used with the scanner or manual calculation through the SFR interface and perform the actions specified in **Section 13.5 “CRC Data Sources”**, depending on which decision was made.
2. If desired, seed a starting CRC value into the CRCACCH/L registers.
3. Program the CRCXORH/L registers with the desired generator polynomial.
4. Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word - 1 (refer to Example 13-1). This determines how many times the shifter will shift into the accumulator for each data word.
5. Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial - 2 (refer to Example 13-1).
6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of the CRCCON0 register appropriately.
7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of the CRCCON0 register appropriately.
8. Write the CRCGO bit of the CRCCON0 register to begin the shifting process.
- 9a. If manual SFR entry is used, monitor the FULL bit of the CRCCON0 register. When FULL = 0, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has more than eight bits, as the shifter will begin upon the CRCDATL register being written.
- 9b. If the scanner is used, the scanner will automatically stuff words into the CRCDATH/L registers as needed, as long as the SCANGO bit is set.
- 10a. If using the Flash memory scanner, monitor the SCANIF (or the SCANGO bit) for the scanner to finish pushing information into the CRCDATA registers. After the scanner is completed, monitor the BUSY bit to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set (or both BUSY and SCANGO bits are cleared), the completed CRC calculation can be read from the CRCACCH/L registers.
- 10b. If manual entry is used, monitor the BUSY bit to determine when the CRCACC registers will hold the check value.

REGISTER 14-19: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1
OSCFIP	CSWIP	—	—	—	—	ADTIP	ADIP
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 **OSCFIP:** Oscillator Fail Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 6 **CSWIP:** Clock-Switch Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 5-2 **Unimplemented:** Read as '0'
- bit 1 **ADTIP:** ADC Threshold Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 0 **ADIP:** ADC Interrupt Priority bit
 1 = High priority
 0 = Low priority

19.0 TIMER1/3/5 MODULE WITH GATE CONTROL

Timer1/3/5 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMRxH:TMRxL)
- Programmable internal or external clock source
- 2-bit prescaler
- Dedicated Secondary 32 kHz oscillator circuit
- Optionally synchronized comparator out
- Multiple Timer1/3/5 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- 16-Bit Read/Write Operation
- Time base for the Capture/Compare function with the CCP modules
- Special Event Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 19-1 is a block diagram of the Timer1/3/5 module.

REGISTER 24-6: CWG1AS0: CWG AUTO-SHUTDOWN CONTROL REGISTER 0

R/W/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—
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	HS/HC = Bit is set/cleared by hardware
q = Value depends on condition		

bit 7 **SHUTDOWN:** Auto-Shutdown Event Status bit^(1,2)

- 1 = An auto-shutdown state is in effect
- 0 = No auto-shutdown event has occurred

bit 6 **REN:** Auto-Restart Enable bit

- 1 = Auto-restart is enabled
- 0 = Auto-restart is disabled

bit 5-4 **LSBD<1:0>:** CWG1B and CWG1D Auto-Shutdown State Control bits

- 11 = A logic '1' is placed on CWG1B/D when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWG1B/D when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWG1B/D when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWG1B/D after the required dead-band interval when an auto-shutdown event occurs.

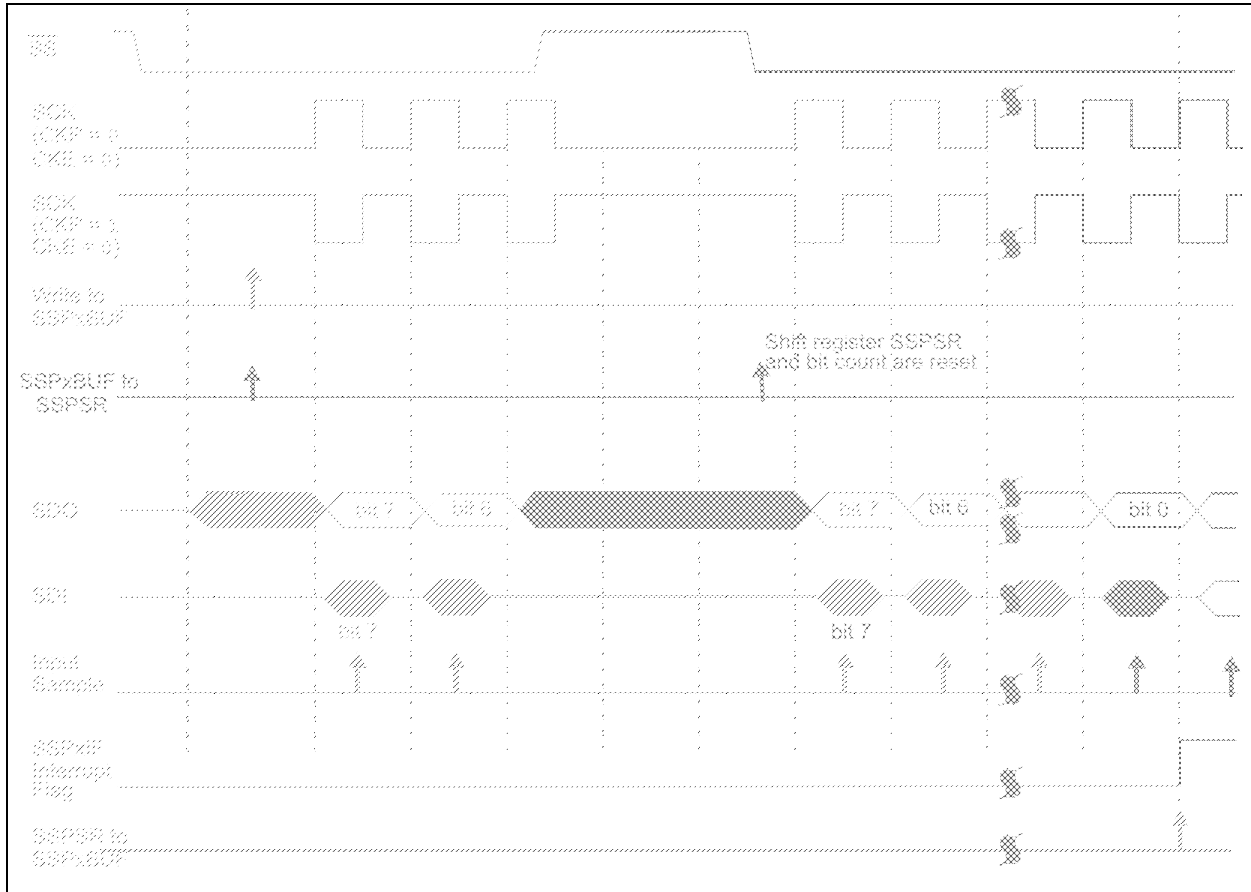
bit 3-2 **LSAC<1:0>:** CWG1A and CWG1C Auto-Shutdown State Control bits

- 11 = A logic '1' is placed on CWG1A/C when an auto-shutdown event occurs.
- 10 = A logic '0' is placed on CWG1A/C when an auto-shutdown event occurs.
- 01 = Pin is tri-stated on CWG1A/C when an auto-shutdown event occurs.
- 00 = The inactive state of the pin, including polarity, is placed on CWG1A/C after the required dead-band interval when an auto-shutdown event occurs.

bit 1-0 **Unimplemented:** Read as '0'

- Note 1:** This bit may be written while EN = 0 (Register 24-1), to place the outputs into the shutdown configuration.
- 2:** The outputs will remain in auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

FIGURE 26-6: SLAVE SELECT SYNCHRONOUS WAVEFORM



26.10.10 SLEEP OPERATION

While in Sleep mode, the I²C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

26.10.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

26.10.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCLxIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

26.10.13 MULTI-MASTER

COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF and reset the I²C port to its Idle state (Figure 26-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

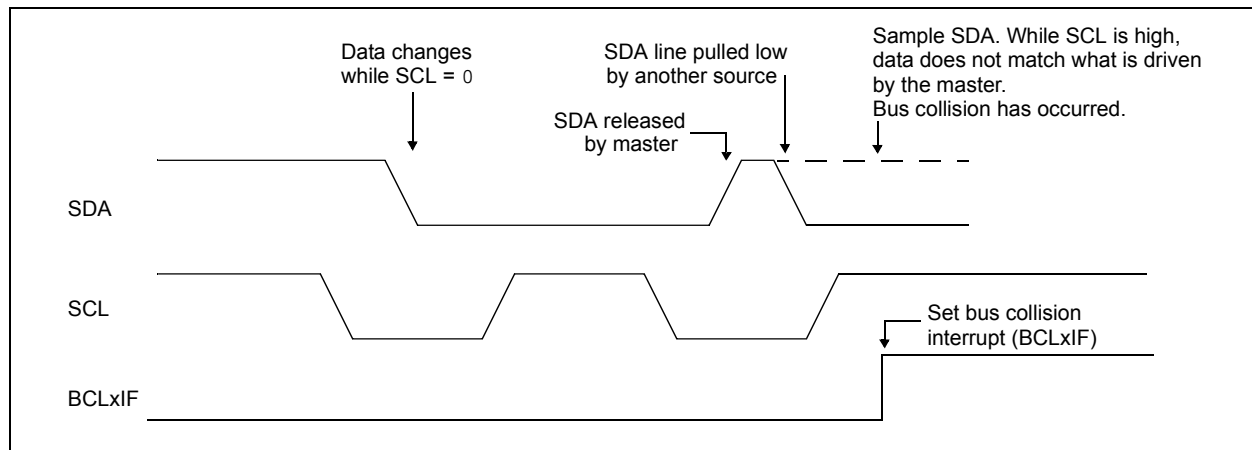
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I²C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 26-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



REGISTER 31-11: ADCAP: ADC ADDITIONAL SAMPLE CAPACITOR SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ADCAP<4:0>				
bit 7			bit 0				

ANDWF

AND W with f

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

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

0001	01da	ffff	ffff
------	------	------	------

Description: The contents of W are AND'ed with 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 $f \leq 95$ (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 to destination

Example: ANDWF REG, 0, 0

Before Instruction

W = 17h
 REG = C2h

After Instruction

W = 02h
 REG = C2h

BC

Branch if Carry

Syntax: BC n

Operands: $-128 \leq n \leq 127$

Operation: if CARRY bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0010	nnnn	nnnn
------	------	------	------

Description: If the CARRY bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC + 2 + 2n$. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BC 5

Before Instruction

PC = address (HERE)

After Instruction

If CARRY = 1;
 PC = address (HERE + 12)
 If CARRY = 0;
 PC = address (HERE + 2)

SUBWFB Subtract W from f with Borrow

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

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

Operation: $(f) - (W) - (\overline{C}) \rightarrow \text{dest}$

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

Encoding:

0101	10da	ffff	ffff
------	------	------	------

Description: Subtract W and the CARRY flag (borrow) from register 'f' (2's complement method). 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 $f \leq 95$ (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 to destination

Example 1: SUBWFB REG, 1, 0

Before Instruction

REG	=	19h	(0001 1001)
W	=	0Dh	(0000 1101)
C	=	1	

After Instruction

REG	=	0Ch	(0000 1100)
W	=	0Dh	(0000 1101)
C	=	1	
Z	=	0	
N	=	0	; result is positive

Example 2: SUBWFB REG, 0, 0

Before Instruction

REG	=	1Bh	(0001 1011)
W	=	1Ah	(0001 1010)
C	=	0	

After Instruction

REG	=	1Bh	(0001 1011)
W	=	00h	
C	=	1	
Z	=	1	; result is zero
N	=	0	

Example 3: SUBWFB REG, 1, 0

Before Instruction

REG	=	03h	(0000 0011)
W	=	0Eh	(0000 1110)
C	=	1	

After Instruction

REG	=	F5h	(1111 0101) ; [2's comp]
W	=	0Eh	(0000 1110)
C	=	0	
Z	=	0	
N	=	1	; result is negative

SWAPF Swap f

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

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

Operation: $(f<3:0>) \rightarrow \text{dest}<7:4>$,
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

Status Affected: None

Encoding:

0011	10da	ffff	ffff
------	------	------	------

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed 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 $f \leq 95$ (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 to destination

Example: SWAPF REG, 1, 0

Before Instruction

REG	=	53h
-----	---	-----

After Instruction

REG	=	35h
-----	---	-----

37.3 DC Characteristics

TABLE 37-1: SUPPLY VOLTAGE

PIC18LF24/25K40			Standard Operating Conditions (unless otherwise stated)				
PIC18F24/25K40							
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
Supply Voltage							
D002	VDD		1.8	—	3.6	V	FOSC ≤ 16 MHz
			2.5	—	3.6	V	FOSC > 16 MHz
			3.0	—	3.6	V	FOSC > 32 MHz
D002	VDD		2.3	—	5.5	V	FOSC ≤ 16 MHz
			2.5	—	5.5	V	FOSC > 16 MHz
			3.0	—	5.5	V	FOSC > 32 MHz
RAM Data Retention ⁽¹⁾							
D003	VDR		1.5	—	—	V	Device in Sleep mode
D003	VDR		1.7	—	—	V	Device in Sleep mode
Power-on Reset Release Voltage ⁽²⁾							
D004	VPOR		—	1.6	—	V	BOR or LPBOR disabled ⁽³⁾
D004	VPOR		—	1.6	—	V	BOR or LPBOR disabled ⁽³⁾
Power-on Reset Rearm Voltage ⁽²⁾							
D005	VPORR		—	0.8	—	V	BOR or LPBOR disabled ⁽³⁾
D005	VPORR		—	1.5	—	V	BOR or LPBOR disabled ⁽³⁾
VDD Rise Rate to ensure internal Power-on Reset signal ⁽²⁾							
D006	SVDD		0.05	—	—	V/ms	BOR or LPBOR disabled ⁽³⁾
D006	SVDD		0.05	—	—	V/ms	BOR or LPBOR disabled ⁽³⁾

† 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: This is the limit to which V_{DD} can be lowered in Sleep mode without losing RAM data.

2: See Figure 37-3, POR and POR REARM with Slow Rising V_{DD}.

3: Please see Table 37-11 for BOR and LPBOR trip point information.

FIGURE 37-7: CLKOUT AND I/O TIMING

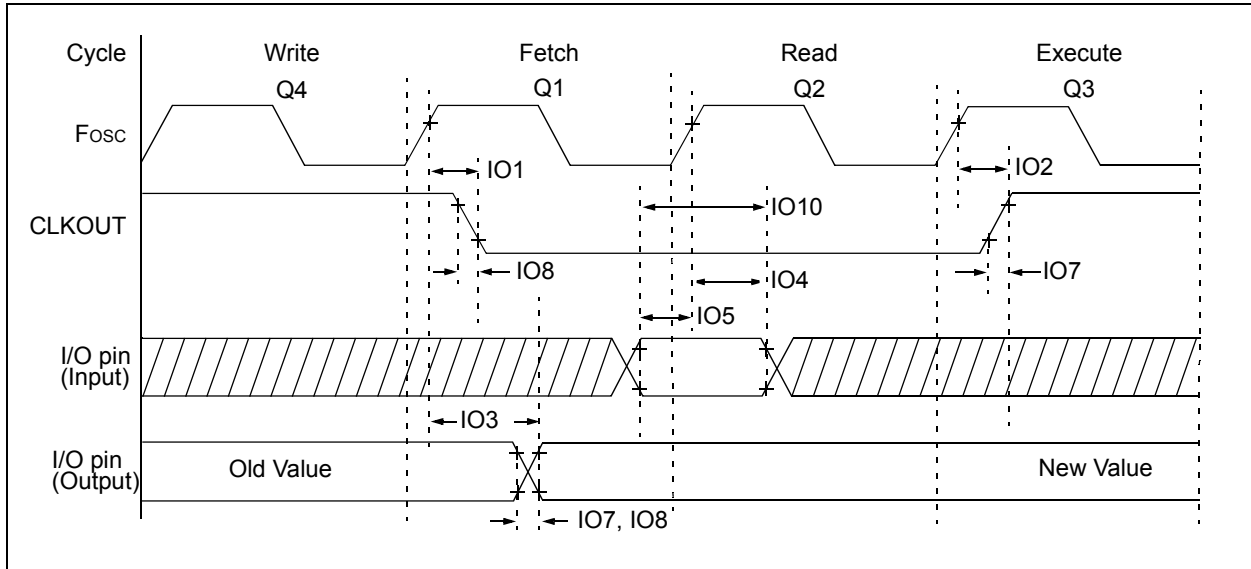


TABLE 37-10: I/O AND CLKOUT TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
IO1*	$T_{CLKOUTH}$	CLKOUT rising edge delay (rising edge Fosc (Q1 cycle) to falling edge CLKOUT)	—	—	70	ns	
IO2*	$T_{CLKOUTL}$	CLKOUT falling edge delay (rising edge Fosc (Q3 cycle) to rising edge CLKOUT)	—	—	72	ns	
IO3*	T_{IO_VALID}	Port output valid time (rising edge Fosc (Q1 cycle) to port valid)	—	50	70	ns	
IO4*	T_{IO_SETUP}	Port input setup time (Setup time before rising edge Fosc – Q2 cycle)	20	—	—	ns	
IO5*	T_{IO_HOLD}	Port input hold time (Hold time after rising edge Fosc – Q2 cycle)	50	—	—	ns	
IO6*	T_{IOR_SLREN}	Port I/O rise time, slew rate enabled	—	25	—	ns	$V_{DD} = 3.0V$
IO7*	T_{IOR_SLRDIS}	Port I/O rise time, slew rate disabled	—	5	—	ns	$V_{DD} = 3.0V$
IO8*	T_{IOF_SLREN}	Port I/O fall time, slew rate enabled	—	25	—	ns	$V_{DD} = 3.0V$
IO9*	T_{IOF_SLRDIS}	Port I/O fall time, slew rate disabled	—	5	—	ns	$V_{DD} = 3.0V$
IO10*	T_{INT}	INT pin high or low time to trigger an interrupt	25	—	—	ns	
IO11*	T_{IOC}	Interrupt-on-Change minimum high or low time to trigger interrupt	25	—	—	ns	

*These parameters are characterized but not tested.

FIGURE 37-18: SPI SLAVE MODE TIMING (CKE = 0)

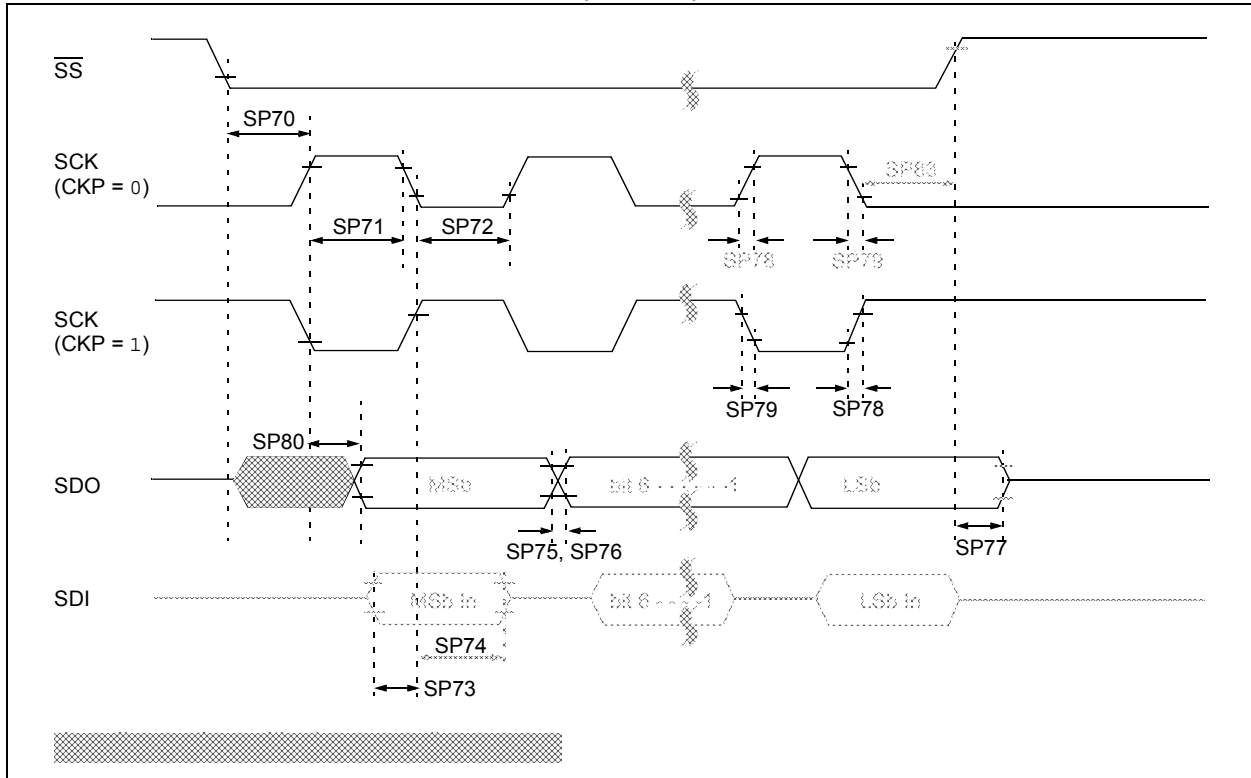


FIGURE 37-19: SPI SLAVE MODE TIMING (CKE = 1)

