

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.154", 3.90mm Width)
Supplier Device Package	16-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f835-gsr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f835-gsr</a>

---

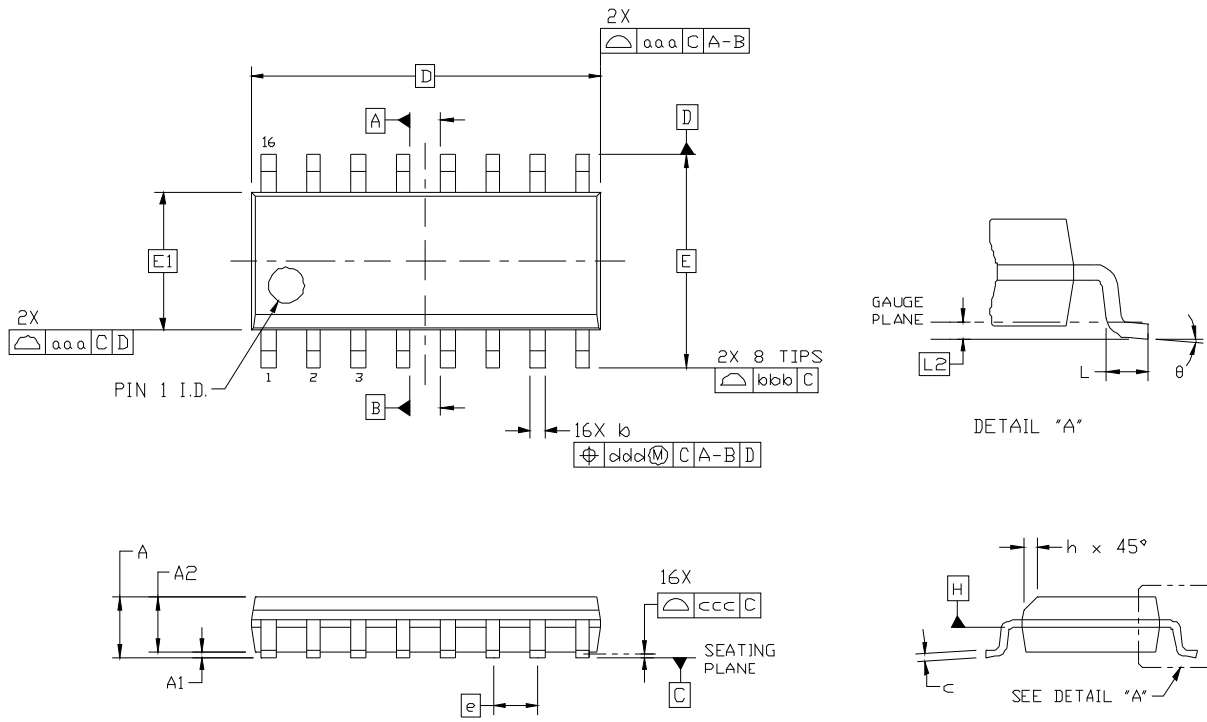
## Table of Contents

<b>1. System Overview .....</b>	<b>15</b>
<b>2. Ordering Information .....</b>	<b>25</b>
<b>3. Pin Definitions.....</b>	<b>28</b>
<b>4. QFN-20 Package Specifications .....</b>	<b>33</b>
<b>5. QSOP-24 Package Specifications .....</b>	<b>35</b>
<b>6. SOIC-16 Package Specifications .....</b>	<b>37</b>
<b>7. Electrical Characteristics .....</b>	<b>39</b>
7.1. Absolute Maximum Specifications .....	39
7.2. Electrical Characteristics .....	40
<b>8. 10-Bit ADC (ADC0).....</b>	<b>46</b>
8.1. Output Code Formatting .....	47
8.2. 8-Bit Mode .....	47
8.3. Modes of Operation .....	47
8.3.1. Starting a Conversion.....	47
8.3.2. Tracking Modes.....	48
8.3.3. Settling Time Requirements.....	49
8.4. Programmable Window Detector.....	53
8.4.1. Window Detector Example.....	55
8.5. ADC0 Analog Multiplexer .....	56
<b>9. Temperature Sensor .....</b>	<b>58</b>
9.1. Calibration .....	58
<b>10. Voltage and Ground Reference Options.....</b>	<b>60</b>
10.1. External Voltage References.....	61
10.2. Internal Voltage Reference Options .....	61
10.3. Analog Ground Reference.....	61
10.4. Temperature Sensor Enable .....	61
<b>11. Voltage Regulator (REG0).....</b>	<b>63</b>
<b>12. Comparator0.....</b>	<b>65</b>
12.1. Comparator Multiplexer .....	69
<b>13. Capacitive Sense (CS0).....</b>	<b>71</b>
13.1. Configuring Port Pins as Capacitive Sense Inputs .....	72
13.2. Capacitive Sense Start-Of-Conversion Sources .....	72
13.3. Automatic Scanning.....	72
13.4. CS0 Comparator.....	73
13.5. CS0 Conversion Accumulator .....	74
13.6. Capacitive Sense Multiplexer .....	80
<b>14. CIP-51 Microcontroller.....</b>	<b>82</b>
14.1. Instruction Set.....	83
14.1.1. Instruction and CPU Timing .....	83
14.2. CIP-51 Register Descriptions .....	88
<b>15. Memory Organization .....</b>	<b>92</b>
15.1. Program Memory.....	93
15.1.1. MOVX Instruction and Program Memory .....	93

## List of Registers

SFR Definition 8.1. ADC0CF: ADC0 Configuration .....	50
SFR Definition 8.2. ADC0H: ADC0 Data Word MSB .....	51
SFR Definition 8.3. ADC0L: ADC0 Data Word LSB .....	51
SFR Definition 8.4. ADC0CN: ADC0 Control .....	52
SFR Definition 8.5. ADC0GTH: ADC0 Greater-Than Data High Byte .....	53
SFR Definition 8.6. ADC0GTL: ADC0 Greater-Than Data Low Byte .....	53
SFR Definition 8.7. ADC0LTH: ADC0 Less-Than Data High Byte .....	54
SFR Definition 8.8. ADC0LTL: ADC0 Less-Than Data Low Byte .....	54
SFR Definition 8.9. ADC0MX: AMUX0 Channel Select .....	57
SFR Definition 10.1. REF0CN: Voltage Reference Control .....	62
SFR Definition 11.1. REG0CN: Voltage Regulator Control .....	64
SFR Definition 12.1. CPT0CN: Comparator0 Control .....	67
SFR Definition 12.2. CPT0MD: Comparator0 Mode Selection .....	68
SFR Definition 12.3. CPT0MX: Comparator0 MUX Selection .....	70
SFR Definition 13.1. CS0CN: Capacitive Sense Control .....	75
SFR Definition 13.2. CS0CF: Capacitive Sense Configuration .....	76
SFR Definition 13.3. CS0DH: Capacitive Sense Data High Byte .....	77
SFR Definition 13.4. CS0DL: Capacitive Sense Data Low Byte .....	77
SFR Definition 13.5. CS0SS: Capacitive Sense Auto-Scan Start Channel .....	78
SFR Definition 13.6. CS0SE: Capacitive Sense Auto-Scan End Channel .....	78
SFR Definition 13.7. CS0THH: Capacitive Sense Comparator Threshold High Byte ...	79
SFR Definition 13.8. CS0THL: Capacitive Sense Comparator Threshold Low Byte ....	79
SFR Definition 13.9. CS0MX: Capacitive Sense Mux Channel Select .....	81
SFR Definition 14.1. DPL: Data Pointer Low Byte .....	88
SFR Definition 14.2. DPH: Data Pointer High Byte .....	88
SFR Definition 14.3. SP: Stack Pointer .....	89
SFR Definition 14.4. ACC: Accumulator .....	89
SFR Definition 14.5. B: B Register .....	90
SFR Definition 14.6. PSW: Program Status Word .....	91
SFR Definition 16.1. HWID: Hardware Identification Byte .....	95
SFR Definition 16.2. DERIVID: Derivative Identification Byte .....	96
SFR Definition 16.3. REVID: Hardware Revision Identification Byte .....	96
SFR Definition 18.1. IE: Interrupt Enable .....	105
SFR Definition 18.2. IP: Interrupt Priority .....	106
SFR Definition 18.3. EIE1: Extended Interrupt Enable 1 .....	107
SFR Definition 18.4. EIE2: Extended Interrupt Enable 2 .....	108
SFR Definition 18.5. EIP1: Extended Interrupt Priority 1 .....	109
SFR Definition 18.6. EIP2: Extended Interrupt Priority 2 .....	110
SFR Definition 18.7. IT01CF: INT0/INT1 Configuration .....	112
SFR Definition 19.1. PSCTL: Program Store R/W Control .....	118
SFR Definition 19.2. FLKEY: Flash Lock and Key .....	119
SFR Definition 20.1. PCON: Power Control .....	122
SFR Definition 21.1. VDM0CN: VDD Monitor Control .....	126

## 6. SOIC-16 Package Specifications



**Figure 6.1. SOIC-16 Package Drawing**

**Table 6.1. SOIC-16 Package Dimensions**

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	—		1.75	L	0.40		1.27
A1	0.10		0.25	L2	0.25 BSC		
A2	1.25		—	h	0.25		0.50
b	0.31		0.51	θ	0°		8°
c	0.17		0.25	aaa	0.10		
D	9.90 BSC			bbb	0.20		
E	6.00 BSC			ccc	0.10		
E1	3.90 BSC			ddd	0.25		
e	1.27 BSC						

**Notes:**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MS-012, Variation AC.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

## SFR Definition 13.7. CS0THH: Capacitive Sense Comparator Threshold High Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0THH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x97

Bit	Name	Description
7:0	CS0THH[7:0]	<b>CS0 Comparator Threshold High Byte.</b> High byte of the 16-bit value compared to the Capacitive Sense conversion result.

## SFR Definition 13.8. CS0THL: Capacitive Sense Comparator Threshold Low Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0THL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x96

Bit	Name	Description
7:0	CS0THL[7:0]	<b>CS0 Comparator Threshold Low Byte.</b> Low byte of the 16-bit value compared to the Capacitive Sense conversion result.

**Table 14.1. CIP-51 Instruction Set Summary (Continued)**

Mnemonic	Description	Bytes	Clock Cycles
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
<b>Program Branching</b>			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	4/5
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

## Notes on Registers, Operands and Addressing Modes:

**Rn**—Register R0–R7 of the currently selected register bank.

**@Ri**—Data RAM location addressed indirectly through R0 or R1.

**rel**—8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

**direct**—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

**#data**—8-bit constant

**#data16**—16-bit constant

**bit**—Direct-accessed bit in Data RAM or SFR

**addr11**—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

**addr16**—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

All mnemonics copyrighted © Intel Corporation 1980.

---

## SFR Definition 14.5. B: B Register

---

Bit	7	6	5	4	3	2	1	0
Name	B[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF0; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	<b>B Register.</b> This register serves as a second accumulator for certain arithmetic operations.



**Table 17.2. Special Function Registers (Continued)**

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
<b>CS0CF</b>	0x9E	CS0 Configuration	76
<b>CS0MX</b>	0x9C	CS0 Mux	81
<b>CS0SE</b>	0xBA	Auto Scan End Channel	78
<b>CS0SS</b>	0xB9	Auto Scan Start Channel	78
<b>DERIVID</b>	0xAD	Derivative Identification	96
<b>DPH</b>	0x83	Data Pointer High	88
<b>DPL</b>	0x82	Data Pointer Low	88
<b>EIE1</b>	0xE6	Extended Interrupt Enable 1	107
<b>EIE2</b>	0xE7	Extended Interrupt Enable 2	108
<b>EIP1</b>	0xF3	Extended Interrupt Priority 1	109
<b>EIP2</b>	0xF4	Extended Interrupt Priority 2	110
<b>FLKEY</b>	0xB7	Flash Lock And Key	119
<b>HWID</b>	0xB5	Hardware Identification	95
<b>IE</b>	0xA8	Interrupt Enable	105
<b>IP</b>	0xB8	Interrupt Priority	106
<b>IT01CF</b>	0xE4	INT0/INT1 Configuration	112
<b>OSCICL</b>	0xB3	Internal Oscillator Calibration	131
<b>OSCICN</b>	0xB2	Internal Oscillator Control	132
<b>OSCXCN</b>	0xB1	External Oscillator Control	134
<b>P0</b>	0x80	Port 0 Latch	153
<b>P0MASK</b>	0xFE	Port 0 Mask	151
<b>P0MAT</b>	0xFD	Port 0 Match	151
<b>P0MDIN</b>	0xF1	Port 0 Input Mode Configuration	154
<b>P0MDOUT</b>	0xA4	Port 0 Output Mode Configuration	154
<b>P0SKIP</b>	0xD4	Port 0 Skip	155
<b>P1</b>	0x90	Port 1 Latch	155
<b>P1MASK</b>	0xEE	P0 Mask	152

# C8051F80x-83x

## SFR Definition 18.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	<b>Enable All Interrupts.</b> Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	<b>Enable Serial Peripheral Interface (SPI0) Interrupt.</b> This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	<b>Enable Timer 2 Interrupt.</b> This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	<b>Enable UART0 Interrupt.</b> This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	<b>Enable Timer 1 Interrupt.</b> This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	<b>Enable External Interrupt 1.</b> This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	<b>Enable Timer 0 Interrupt.</b> This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	<b>Enable External Interrupt 0.</b> This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

---

**SFR Definition 18.4. EIE2: Extended Interrupt Enable 2**


---

Bit	7	6	5	4	3	2	1	0
Name							ECSGRT	ECSCPT
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE7

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = don't care.
1	ECSGRT	<b>Enable Capacitive Sense Greater Than Comparator Interrupt.</b> 0: Disable Capacitive Sense Greater Than Comparator interrupt. 1: Enable interrupt requests generated by CS0CMPF.
0	ECSCPT	<b>Enable Capacitive Sense Conversion Complete Interrupt.</b> 0: Disable Capacitive Sense Conversion Complete interrupt. 1: Enable interrupt requests generated by CS0INT.

---

## 19.4.3. System Clock

1. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
2. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in "AN201: Writing to Flash from Firmware," available from the Silicon Laboratories website.

---

## 21.2. Power-Fail Reset / $V_{DD}$ Monitor

When a power-down transition or power irregularity causes  $V_{DD}$  to drop below  $V_{RST}$ , the power supply monitor will drive the  $\overline{RST}$  pin low and hold the CIP-51 in a reset state (see Figure 21.2). When  $V_{DD}$  returns to a level above  $V_{RST}$ , the CIP-51 will be released from the reset state. Even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if  $V_{DD}$  dropped below the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The  $V_{DD}$  monitor is enabled and selected as a reset source after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the  $V_{DD}$  monitor is disabled by code and a software reset is performed, the  $V_{DD}$  monitor will still be disabled after the reset.

**Important Note:** If the  $V_{DD}$  monitor is being turned on from a disabled state, it should be enabled before it is selected as a reset source. Selecting the  $V_{DD}$  monitor as a reset source before it is enabled and stabilized may cause a system reset. In some applications, this reset may be undesirable. If this is not desirable in the application, a delay should be introduced between enabling the monitor and selecting it as a reset source. The procedure for enabling the  $V_{DD}$  monitor and configuring it as a reset source from a disabled state is shown below:

1. Enable the  $V_{DD}$  monitor (VDMEN bit in VDM0CN = 1).
2. If necessary, wait for the  $V_{DD}$  monitor to stabilize.
3. Select the  $V_{DD}$  monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 21.2 for  $V_{DD}$  monitor timing; note that the power-on-reset delay is not incurred after a  $V_{DD}$  monitor reset. See Section “7. Electrical Characteristics” on page 39 for complete electrical characteristics of the  $V_{DD}$  monitor.

# C8051F80x-83x

## SFR Definition 24.2. CRC0IN: CRC Data Input

Bit	7	6	5	4	3	2	1	0
Name	CRC0IN[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDD

Bit	Name	Function
7:0	CRC0IN[7:0]	<b>CRC0 Data Input.</b> Each write to CRC0IN results in the written data being computed into the existing CRC result according to the CRC algorithm described in Section 24.1

## SFR Definition 24.3. CRC0DATA: CRC Data Output

Bit	7	6	5	4	3	2	1	0
Name	CRC0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDE

Bit	Name	Function
7:0	CRC0DAT[7:0]	<b>CRC0 Data Output.</b> Each read or write performed on CRC0DAT targets the CRC result bits pointed to by the CRC0 Result Pointer (CRC0PNT bits in CRC0CN).

## 25.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 25.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

### 25.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

### 25.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

### 25.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

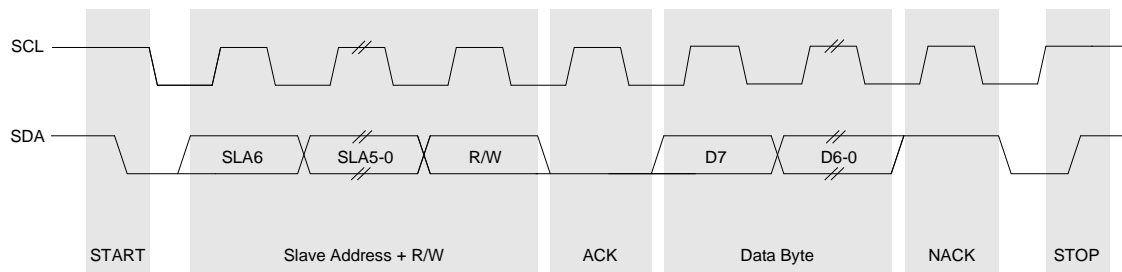
1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 25.2, Figure 25.3, and Figure 25.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “23. Port Input/Output” on page 138 for general purpose port I/O and crossbar information.

## 25.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 26.3 illustrates a typical SMBus transaction.



**Figure 26.3. SMBus Transaction**

## 26.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

## 26.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “26.3.5. SCL High (SMBus Free) Timeout” on page 183). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

## 26.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

## 26.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

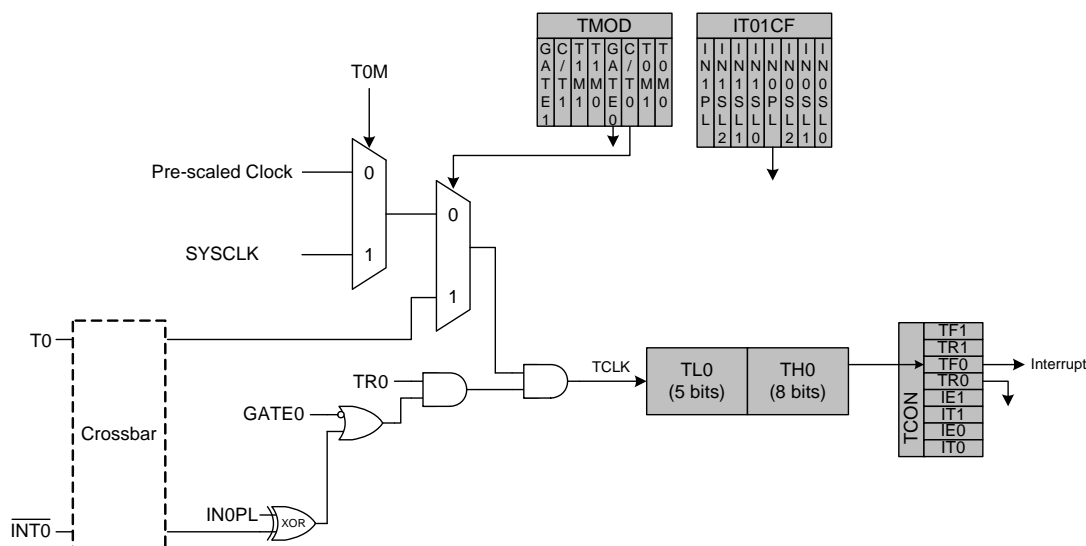


**SFR Definition 26.1. SMB0CF: SMBus Clock/Configuration**

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1

Bit	Name	Function
7	ENSMB	<b>SMBus Enable.</b> This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	<b>SMBus Slave Inhibit.</b> When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	<b>SMBus Busy Indicator.</b> This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	<b>SMBus Setup and Hold Time Extension Enable.</b> This bit controls the SDA setup and hold times according to Table 26.2. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	<b>SMBus SCL Timeout Detection Enable.</b> This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	<b>SMBus Free Timeout Detection Enable.</b> When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	<b>SMBus Clock Source Selection.</b> These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 26.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow



**Figure 28.1. T0 Mode 0 Block Diagram**

## 28.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

## 28.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “18.3. INT0 and INT1 External Interrupts” on page 111 for details on the external input signals INT0 and INT1).

# C8051F80x-83x

## SFR Definition 28.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8C

Bit	Name	Function
7:0	TH0[7:0]	<b>Timer 0 High Byte.</b> The TH0 register is the high byte of the 16-bit Timer 0.

## SFR Definition 28.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8D

Bit	Name	Function
7:0	TH1[7:0]	<b>Timer 1 High Byte.</b> The TH1 register is the high byte of the 16-bit Timer 1.

## 29.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 2. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

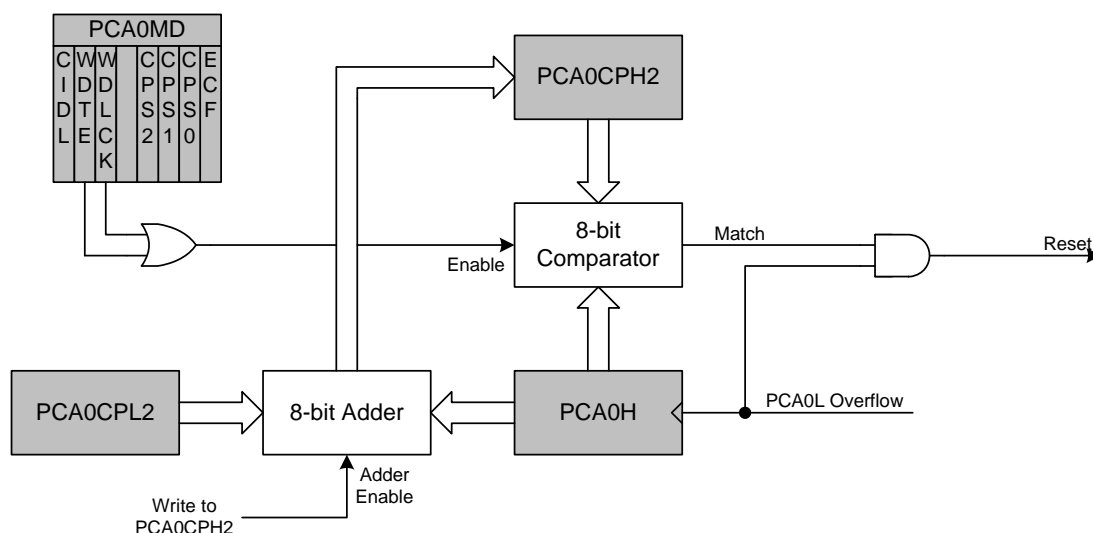
With the WDTE bit set in the PCA0MD register, Module 2 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA counter high byte; the Module 2 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

### 29.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2–CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 29.11).



**Figure 29.11. PCA Module 2 with Watchdog Timer Enabled**

The 8-bit offset held in PCA0CPH2 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total off-



## Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



**IoT Portfolio**  
[www.silabs.com/IoT](http://www.silabs.com/IoT)



**SW/HW**  
[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

### Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

### Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



**Silicon Laboratories Inc.**  
 400 West Cesar Chavez  
 Austin, TX 78701  
 USA

<http://www.silabs.com>