



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	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f854-c-gur

8.2.1. Internal RAM	53
8.2.2. External RAM	54
8.2.3. Special Function Registers	55
9. Special Function Register Memory Map.....	56
10. Flash Memory	61
10.1. Security Options	61
10.2. Programming the Flash Memory	63
10.2.1. Flash Lock and Key Functions	63
10.2.2. Flash Erase Procedure	63
10.2.3. Flash Write Procedure	63
10.3. Non-Volatile Data Storage.....	64
10.4. Flash Write and Erase Guidelines.....	64
10.4.1. Voltage Supply Maintenance and the Supply Monitor	64
10.4.2. PSWE Maintenance	65
10.4.3. System Clock	65
10.5. Flash Control Registers	66
11. Device Identification and Unique Identifier	68
11.1. Device Identification Registers	69
12. Interrupts	72
12.1. MCU Interrupt Sources and Vectors.....	72
12.1.1. Interrupt Priorities.....	72
12.1.2. Interrupt Latency	72
12.2. Interrupt Control Registers	75
13. Power Management and Internal Regulator	82
13.1. Power Modes.....	82
13.1.1. Idle Mode	82
13.1.2. Stop Mode.....	83
13.2. LDO Regulator.....	83
13.3. Power Control Registers.....	83
13.4. LDO Control Registers	84
14. Analog-to-Digital Converter (ADC0).....	85
14.1. ADC0 Analog Multiplexer	86
14.2. ADC Operation	88
14.2.1. Starting a Conversion.....	88
14.2.2. Tracking Modes.....	88
14.2.3. Burst Mode.....	89
14.2.4. Settling Time Requirements.....	90
14.2.5. Gain Setting	91
14.3. 8-Bit Mode	91
14.4. 12-Bit Mode	91
14.5. Power Considerations	92
14.6. Output Code Formatting	94
14.7. Programmable Window Detector.....	95
14.7.1. Window Detector In Single-Ended Mode	95
14.8. Voltage and Ground Reference Options	97

28. Revision-Specific Behavior.....	301
28.1. Revision Identification.....	301
28.2. Temperature Sensor Offset and Slope.....	303
28.3. Flash Endurance	303
28.4. Latch-Up Performance	303
28.5. Unique Identifier	303
29. C2 Interface	304
29.1. C2 Pin Sharing	304
29.2. C2 Interface Registers.....	305
Document Change List.....	310
Contact Information.....	311

Table 1.2. Power Consumption (Continued)

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Digital Core Supply Current (–I _x Devices, -40°C to +125°C)						
Normal Mode—Full speed with code executing from flash	I _{DD}	F _{SYSCLK} = 24.5 MHz ²	—	4.45	5.25	mA
		F _{SYSCLK} = 1.53 MHz ²	—	915	1600	μA
		F _{SYSCLK} = 80 kHz ³ , T _A = 25 °C	—	250	290	μA
		F _{SYSCLK} = 80 kHz ³	—	250	725	μA
Idle Mode—Core halted with peripherals running	I _{DD}	F _{SYSCLK} = 24.5 MHz ²	—	2.05	2.6	mA
		F _{SYSCLK} = 1.53 MHz ²	—	550	1000	μA
		F _{SYSCLK} = 80 kHz ³ , T _A = 25 °C	—	125	130	μA
		F _{SYSCLK} = 80 kHz ³	—	125	550	μA
Stop Mode—Core halted and all clocks stopped, Supply monitor off.	I _{DD}	Internal LDO ON, T _A = 25 °C	—	105	120	μA
		Internal LDO ON	—	105	270	μA
		Internal LDO OFF	—	0.2	—	μA
Analog Peripheral Supply Currents (Both –G _x and –I _x Devices)						
High-Frequency Oscillator	I _{HFOSC}	Operating at 24.5 MHz, T _A = 25 °C	—	155	—	μA
Low-Frequency Oscillator	I _{LFOSC}	Operating at 80 kHz, T _A = 25 °C	—	3.5	—	μA
ADC0 Always-on ⁴	I _{ADC}	800 ksps, 10-bit conversions or 200 ksps, 12-bit conversions Normal bias settings V _{DD} = 3.0 V	—	845	1200	μA
		250 ksps, 10-bit conversions or 62.5 ksps 12-bit conversions Low power bias settings V _{DD} = 3.0 V	—	425	580	μA
ADC0 Burst Mode, 10-bit single conversions, external reference	I _{ADC}	200 ksps, V _{DD} = 3.0 V	—	370	—	μA
		100 ksps, V _{DD} = 3.0 V	—	185	—	μA
		10 ksps, V _{DD} = 3.0 V	—	19	—	μA
Notes:						
1. Currents are additive. For example, where I _{DD} is specified and the mode is not mutually exclusive, enabling the functions increases supply current by the specified amount.						
2. Includes supply current from internal regulator, supply monitor, and High Frequency Oscillator.						
3. Includes supply current from internal regulator, supply monitor, and Low Frequency Oscillator.						
4. ADC0 always-on power excludes internal reference supply current.						
5. The internal reference is enabled as-needed when operating the ADC in burst mode to save power.						

Table 3.1. Pin Definitions for C8051F850/1/2/3/4/5-GU and C8051F850/1/2/3/4/5-IU

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P0.0	Standard I/O	4	Yes	P0MAT.0 INT0.0 INT1.0	ADC0.0 CP0P.0 CP0N.0 VREF
P0.1	Standard I/O	3	Yes	P0MAT.1 INT0.1 INT1.1	ADC0.1 CP0P.1 CP0N.1 AGND
P0.2	Standard I/O	2	Yes	P0MAT.2 INT0.2 INT1.2	ADC0.2 CP0P.2 CP0N.2
P0.3 / EXTCLK	Standard I/O / External CMOS Clock Input	23	Yes	P0MAT.3 EXTCLK INT0.3 INT1.3	ADC0.3 CP0P.3 CP0N.3
P0.4	Standard I/O	22	Yes	P0MAT.4 INT0.4 INT1.4	ADC0.4 CP0P.4 CP0N.4
P0.5	Standard I/O	21	Yes	P0MAT.5 INT0.5 INT1.5	ADC0.5 CP0P.5 CP0N.5
P0.6	Standard I/O	20	Yes	P0MAT.6 CNVSTR INT0.6 INT1.6	ADC0.6 CP0P.6 CP0N.6
P0.7	Standard I/O	19	Yes	P0MAT.7 INT0.7 INT1.7	ADC0.7 CP0P.7 CP0N.7

8. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F85x/86x device family is shown in Figure 8.1.

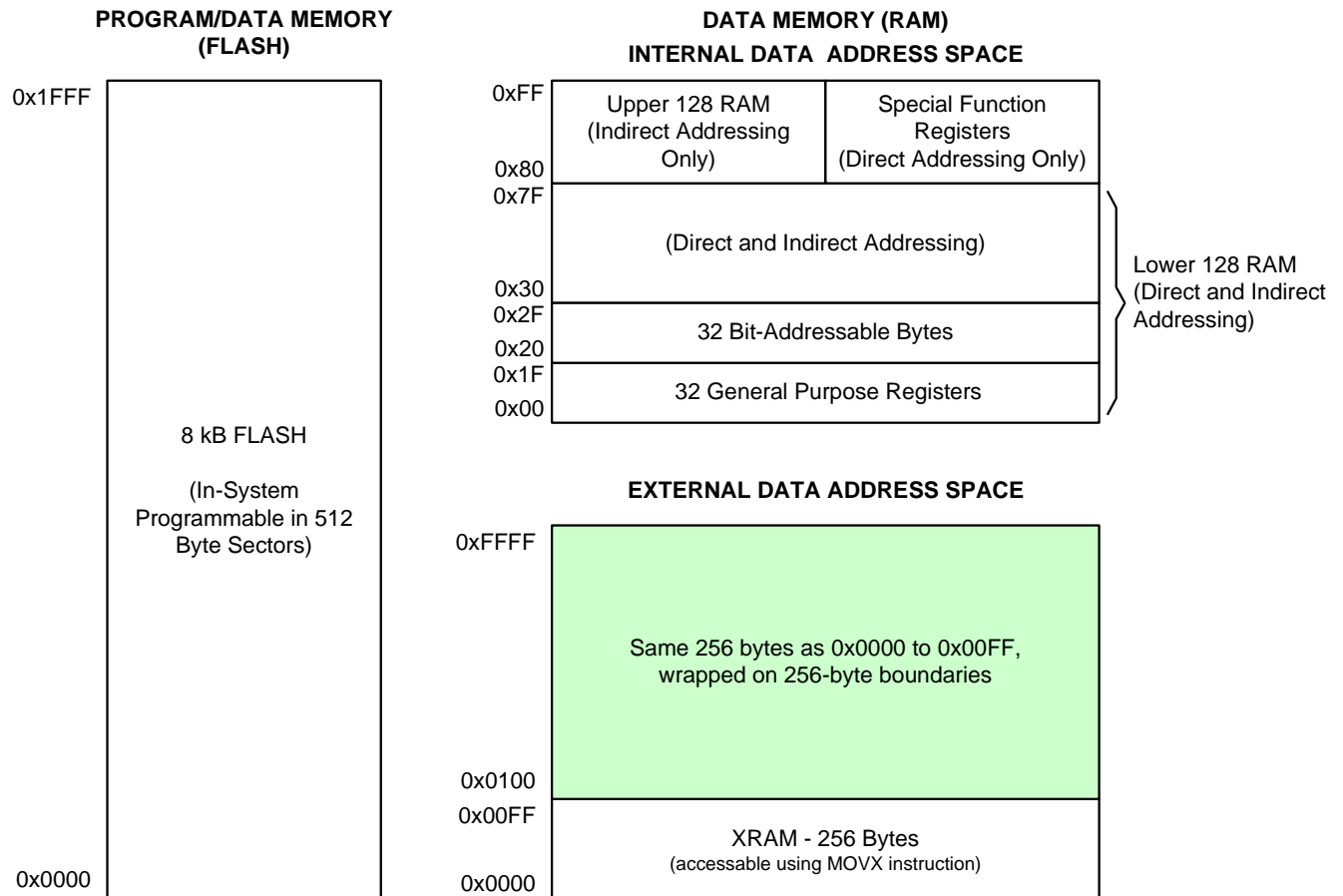


Figure 8.1. C8051F85x/86x Memory Map (8 kB flash version shown)

8.2.3. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided.

Register 17.2. CPT0MD: Comparator 0 Mode

Bit	7	6	5	4	3	2	1	0
Name	CPLOUT	Reserved	CPRIE	CPFIE	Reserved		CPMD	
Type	RW	R	RW	RW	R		RW	
Reset	0	0	0	0	0	0	1	0

SFR Address: 0x9D

Table 17.6. CPT0MD Register Bit Descriptions

Bit	Name	Function
7	CPLOUT	Comparator 0 Latched Output Flag. This bit represents the comparator output value at the most recent PCA counter overflow. 0: Comparator output was logic low at last PCA overflow. 1: Comparator output was logic high at last PCA overflow.
6	Reserved	Must write reset value.
5	CPRIE	Comparator 0 Rising-Edge Interrupt Enable. 0: Comparator Rising-Edge interrupt disabled. 1: Comparator Rising-Edge interrupt enabled.
4	CPFIE	Comparator 0 Falling-Edge Interrupt Enable. 0: Comparator Falling-Edge interrupt disabled. 1: Comparator Falling-Edge interrupt enabled.
3:2	Reserved	Must write reset value.
1:0	CPMD	Comparator 0 Mode Select. These bits affect the response time and power consumption of the comparator. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

Register 17.3. CPT0MX: Comparator 0 Multiplexer Selection

Bit	7	6	5	4	3	2	1	0
Name	CMXN				CMXP			
Type	RW				RW			
Reset	1	1	1	1	1	1	1	1

SFR Address: 0x9F

Table 17.7. CPT0MX Register Bit Descriptions

Bit	Name	Function
7:4	CMXN	Comparator 0 Negative Input MUX Selection. 0000: External pin CP0N.0 0001: External pin CP0N.1 0010: External pin CP0N.2 0011: External pin CP0N.3 0100: External pin CP0N.4 0101: External pin CP0N.5 0110: External pin CP0N.6 0111: External pin CP0N.7 1000: GND 1001-1111: Reserved.
3:0	CMXP	Comparator 0 Positive Input MUX Selection. 0000: External pin CP0P.0 0001: External pin CP0P.1 0010: External pin CP0P.2 0011: External pin CP0P.3 0100: External pin CP0P.4 0101: External pin CP0P.5 0110: External pin CP0P.6 0111: External pin CP0P.7 1000: Internal LDO output 1001-1111: Reserved.

The 16-bit CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
    unsigned char i;                                // loop counter

    #define POLY 0x1021

    // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
    // with no carries)
    CRC_acc = CRC_acc ^ (CRC_input << 8);

    // "Divide" the poly into the dividend using CRC XOR subtraction
    // CRC_acc holds the "remainder" of each divide
    //
    // Only complete this division for 8 bits since input is 1 byte
    for (i = 0; i < 8; i++)
    {
        // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
        // into the "dividend")
        if ((CRC_acc & 0x8000) == 0x8000)
        {
            // if so, shift the CRC value, and XOR "subtract" the poly
            CRC_acc = CRC_acc << 1;
            CRC_acc ^= POLY;
        }
        else
        {
            // if not, just shift the CRC value
            CRC_acc = CRC_acc << 1;
        }
    }

    // Return the final remainder (CRC value)
    return CRC_acc;
}
```

Table 18.1 lists several input values and the associated outputs using the 16-bit CRC algorithm:

Table 18.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

Register 20.4. PCA0CLR: PCA Comparator Clear Control

Bit	7	6	5	4	3	2	1	0
Name	CPCPOL	Reserved				CPCE2	CPCE1	CPCE0
Type	RW	R				RW	RW	RW
Reset	0	0	0	0	0	0	0	0

SFR Address: 0x9C

Table 20.6. PCA0CLR Register Bit Descriptions

Bit	Name	Function
7	CPCPOL	Comparator Clear Polarity. Selects the polarity of the comparator result that will clear the PCA channel(s). 0: PCA channel(s) will be cleared when comparator result goes logic low. 1: PCA channel(s) will be cleared when comparator result goes logic high.
6:3	Reserved	Must write reset value.
2	CPCE2	Comparator Clear Enable for CEX2. Enables the comparator clear function on PCA channel 2.
1	CPCE1	Comparator Clear Enable for CEX1. Enables the comparator clear function on PCA channel 1.
0	CPCE0	Comparator Clear Enable for CEX0. Enables the comparator clear function on PCA channel 0.

Register 21.5. P0MASK: Port 0 Mask

Bit	7	6	5	4	3	2	1	0
Name	P0MASK							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xFE								

Table 21.8. P0MASK Register Bit Descriptions

Bit	Name	Function
7:0	P0MASK	Port 0 Mask Value. Selects P0 pins to be compared to the corresponding bits in P0MAT. 0: P0.x pin logic value is ignored and will cause a port mismatch event. 1: P0.x pin logic value is compared to P0MAT.x.

23.1. Signal Descriptions

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

23.1.1. 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.

23.1.2. 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.

23.1.3. Serial Clock (SCK)

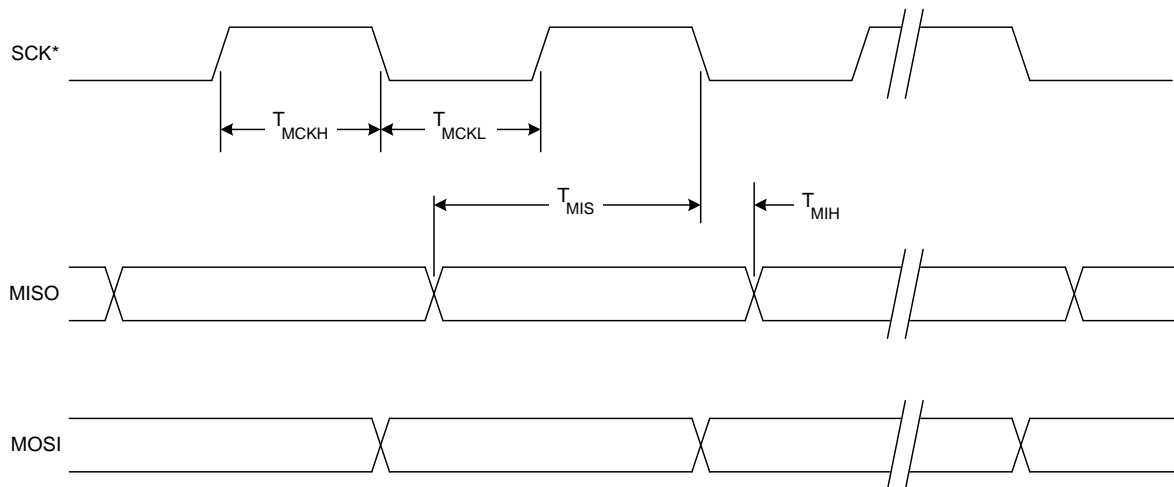
The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when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

23.1.4. 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:

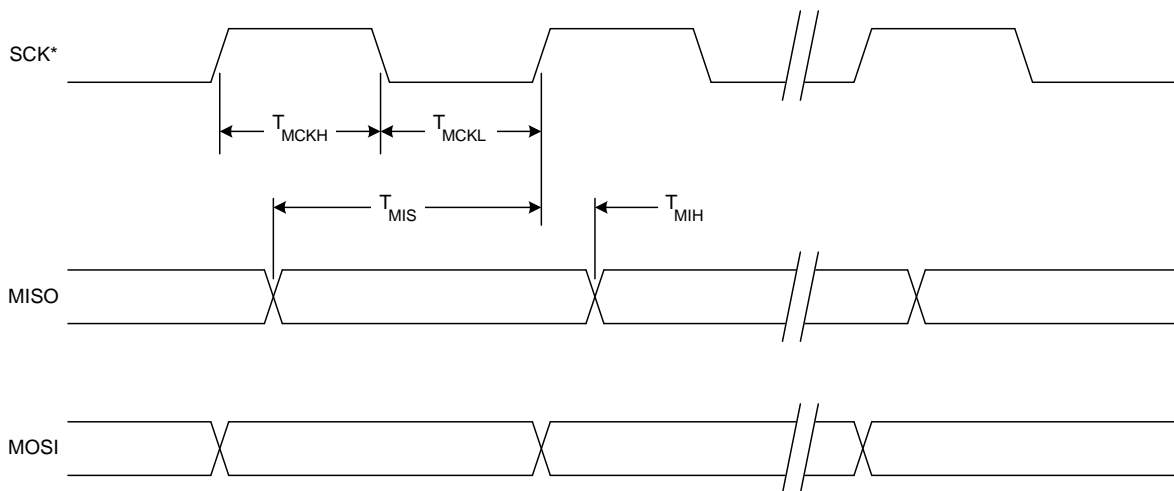
1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point 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 23.2, Figure 23.3, and Figure 23.4 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.



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 23.8. SPI Master Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 23.9. SPI Master Timing (CKPHA = 1)

Register 23.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xA3								

Table 23.5. SPI0DAT Register Bit Descriptions

Bit	Name	Function
7:0	SPI0DAT	SPI0 Transmit and Receive Data. The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0-DAT places the data into the transmit buffer and initiates a transfer when in master mode. A read of SPI0DAT returns the contents of the receive buffer.

Table 24.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

24.4.6. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

Register 24.2. SMB0TC: SMBus0 Timing and Pin Control

Bit	7	6	5	4	3	2	1	0
Name	SWAP	Reserved					SDD	
Type	RW	R					RW	
Reset	0	0	0	0	0	0	0	0

SFR Address: 0xAC

Table 24.8. SMB0TC Register Bit Descriptions

Bit	Name	Function
7	SWAP	SMBus0 Swap Pins. This bit swaps the order of the SMBus0 pins on the crossbar. 0: SDA is mapped to the lower-numbered port pin, and SCL is mapped to the higher-numbered port pin. 1: SCL is mapped to the lower-numbered port pin, and SDA is mapped to the higher-numbered port pin.
6:2	Reserved	Must write reset value.
1:0	SDD	SMBus0 Start Detection Window. These bits increase the hold time requirement between SDA falling and SCL falling for START detection. 00: No additional hold time window (0-1 SYSCLK). 01: Increase hold time window to 2-3 SYSCLKs. 10: Increase hold time window to 4-5 SYSCLKs. 11: Increase hold time window to 8-9 SYSCLKs.

Table 25.2. Timer Peripheral Clocking / Event Triggering

Function	T0 Overflow	T1 Overflow	T2 High Overflow	T2 Low Overflow	T3 High Overflow	T3 Low Overflow
ADC0 Conversion Start	X		X*	X*	X*	X*
*Note: The high-side overflow is used when the timer is in 16-bit mode. The low-side overflow is used in 8-bit mode.						

Register 25.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xCC								

Table 25.13. TMR2L Register Bit Descriptions

Bit	Name	Function
7:0	TMR2L	Timer 2 Low Byte. In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

26.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit in register SCON.

26.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX pin and received at the RX pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB8 in the SCON register.

Data transmission begins when software writes a data byte to the SBUF0 register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI must be logic 0, and if MCE is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB8 and the RI flag is set. If these conditions are not met, SBUF0 and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI is set.

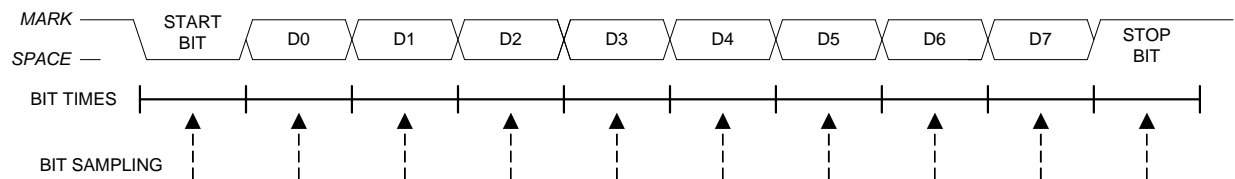


Figure 26.3. 8-Bit UART Timing Diagram

26.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB8, which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB8 and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI must be logic 0, and (2) if MCE is logic 1, the 9th bit must be logic 1 (when MCE is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB8, and the RI flag is set to 1. If the above conditions are not met, SBUF0 and RB8 will not be loaded and the RI flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI or RI is set to 1.

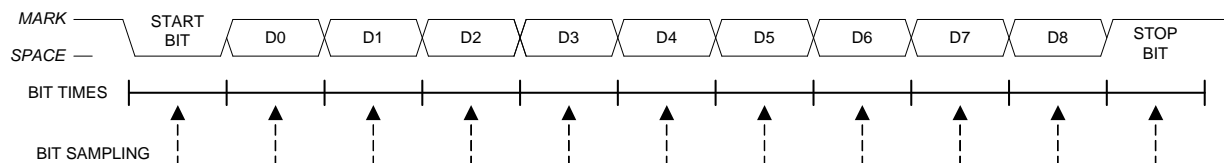


Figure 26.4. 9-Bit UART Timing Diagram