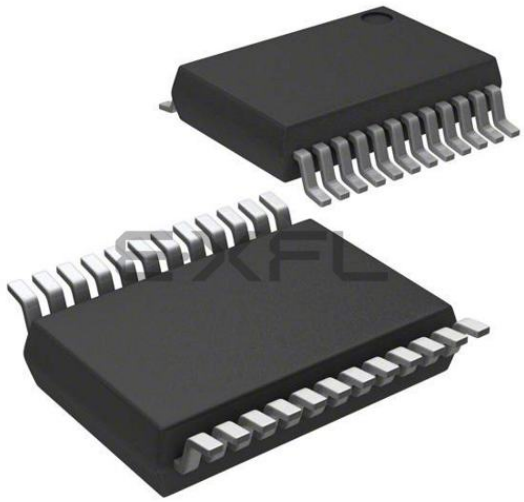

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Last Time Buy
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 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	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f820-gu

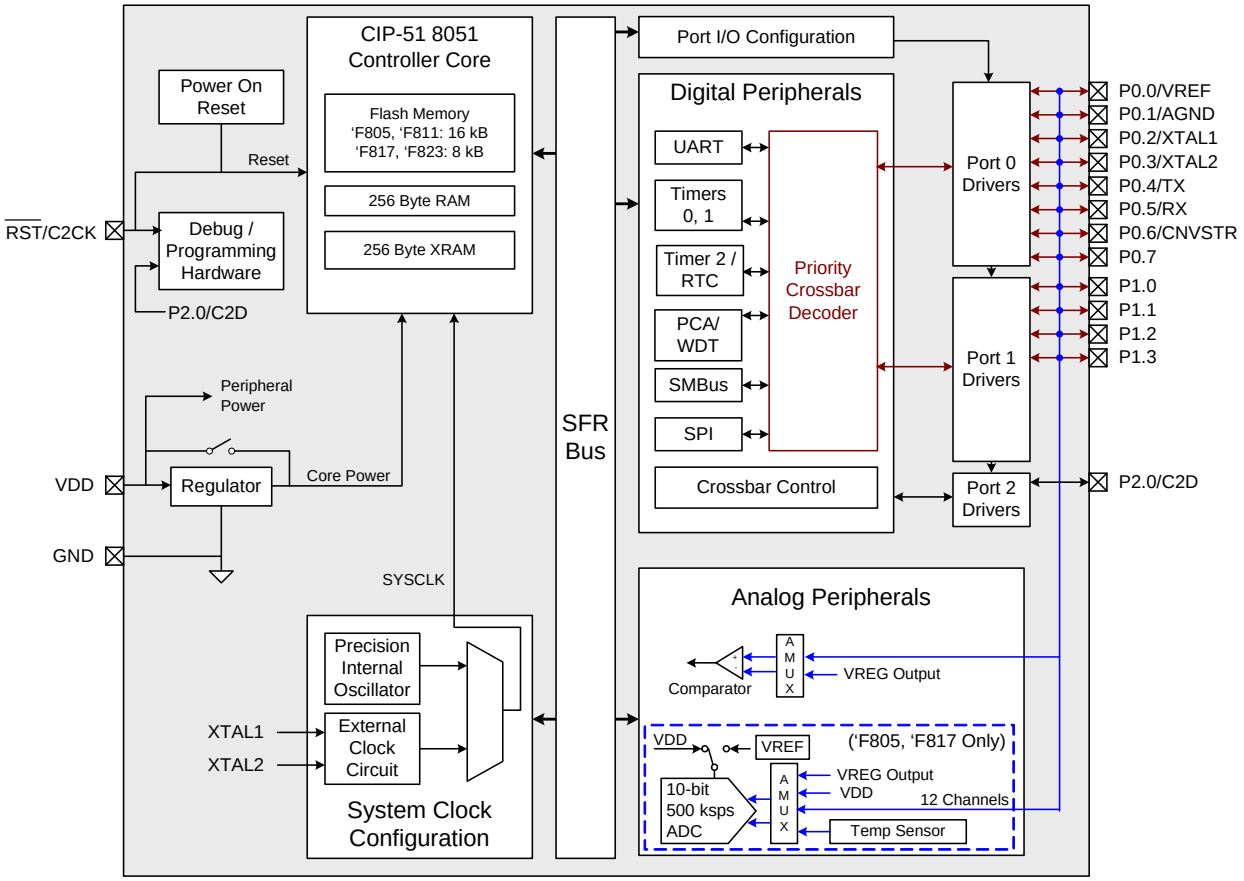

Figure 1.6. C8051F805, C8051F811, C8051F817, C8051F823 Block Diagram

SFR Definition 8.9. ADC0MX: AMUX0 Channel Select

Bit	7	6	5	4	3	2	1	0
Name				AMX0P[3:0]				
Type	R	R	R	R/W				
Reset	0	0	0	1	1	1	1	1

SFR Address = 0xBB

Bit	Name	Function		
7:5	Unused	Read = 000b; Write = Don't Care.		
4:0	AMX0P[4:0]	**AMUX0 Positive Input Selection.**		
			20-Pin and 24-Pin Devices	**16-Pin Devices**
		00000:	P0.0	P0.0
		00001:	P0.1	P0.1
		00010:	P0.2	P0.2
		00011:	P0.3	P0.3
		00100:	P0.4	P0.4
		00101:	P0.5	P0.5
		00110:	P0.6	P0.6
		00111:	P0.7	P0.7
		01000	P1.0	P1.0
		01001	P1.1	P1.1
		01010	P1.2	P1.2
		01011	P1.3	P1.3
		01100	P1.4	Reserved.
		01101	P1.5	Reserved.
		01110	P1.6	Reserved.
		01111	P1.7	Reserved.
		10000:	Temp Sensor	Temp Sensor
		10001:	VREG Output	VREG Output
		10010:	VDD	VDD
		10011:	GND	GND
		10100 – 11111:	no input selected	

SFR Definition 12.3. CPT0MX: Comparator0 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W				R/W			
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x9F

Bit	Name	Function		
7:4	CMX0N[3:0]	**Comparator0 Negative Input MUX Selection.**		
			20-Pin and 24-Pin Devices	**16-Pin Devices**
		0000	P0.1	P0.1
		0001	P0.3	P0.3
		0010	P0.5	P0.5
		0011	P0.7	P0.7
		0100	P1.1	P1.1
		0101	P1.3	P1.3
		0110	P1.5	Reserved.
		0111	P1.7	Reserved.
		1000	VREG Output.	VREG Output.
		1001–1111	No input selected.	No input selected.
3:0	CMX0P[3:0]	**Comparator0 Positive Input MUX Selection.**		
			20-Pin and 24-Pin Devices	**16-Pin Devices**
		0000	P0.0	P0.0
		0001	P0.2	P0.2
		0010	P0.4	P0.4
		0011	P0.6	P0.6
		0100	P1.0	P1.0
		0101	P1.2	P1.2
		0110	P1.4	Reserved.
		0111	P1.6	Reserved.
		1000	VREG Output.	VREG Output.
		1001–1111	No input selected.	No input selected.

13.5. CS0 Conversion Accumulator

CS0 can be configured to accumulate multiple conversions on an input channel. The number of samples to be accumulated is configured using the CS0ACU2:0 bits (CS0CF2:0). The accumulator can accumulate 1, 4, 8, 16, 32, or 64 samples. After the defined number of samples have been accumulated, the result is converted to a 16-bit value by dividing the 22-bit accumulator by either 1, 4, 8, 16, 32, or 64 (depending on the CS0ACU[2:0] setting) and copied to the CS0DH:CS0DL SFRs.

Table 13.1. Operation with Auto-scan and Accumulate

Auto-Scan Enabled	Accumulator Enabled	CS0 Conversion Complete Interrupt Behavior	CS0 Greater Than Interrupt Behavior	CS0MX Behavior
N	N	CS0INT Interrupt serviced after 1 conversion completes	Interrupt serviced after 1 conversion completes if value in CS0DH:CS0DL is greater than CS0THH:CS0THL	CS0MX unchanged.
N	Y	CS0INT Interrupt serviced after *M* conversions complete	Interrupt serviced after *M* conversions complete if value in 16-bit accumulator is greater than CS0THH:CS0THL	CS0MX unchanged.
Y	N	CS0INT Interrupt serviced after 1 conversion completes	Interrupt serviced after conversion completes if value in CS0DH:CS0DL is greater than CS0THH:CS0THL; Auto-Scan stopped	If greater-than comparator detects conversion value is greater than CS0THH:CS0THL, CMUX0 is left unchanged; otherwise, CMUX0 updates to the next channel (CS0MX + 1) and wraps back to CS0SS after passing CS0SE
Y	Y	CS0INT Interrupt serviced after *M* conversions complete	Interrupt serviced after *M* conversions complete if value in 16-bit accumulator is greater than CS0THH:CS0THL; Auto-Scan stopped	If greater-than comparator detects conversion value is greater than CS0THH:CS0THL, CS0MX is left unchanged; otherwise, CS0MX updates to the next channel (CS0MX + 1) and wraps back to CS0SS after passing CS0SE
M = Accumulator setting (1x, 4x, 8x, 16x, 32x, 64x)				

Table 17.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
P1MAT	0xED	P1 Match	152
P1MDIN	0xF2	Port 1 Input Mode Configuration	156
P1MDOUT	0xA5	Port 1 Output Mode Configuration	156
P1SKIP	0xD5	Port 1 Skip	157
P2	0xA0	Port 2 Latch	157
P2MDOUT	0xA6	Port 2 Output Mode Configuration	158
PCA0CN	0xD8	PCA Control	238
PCA0CPH0	0xFC	PCA Capture 0 High	243
PCA0CPH1	0xEA	PCA Capture 1 High	243
PCA0CPH2	0xEC	PCA Capture 2 High	243
PCA0CPL0	0xFB	PCA Capture 0 Low	243
PCA0CPL1	0xE9	PCA Capture 1 Low	243
PCA0CPL2	0xEB	PCA Capture 2 Low	243
PCA0CPM0	0xDA	PCA Module 0 Mode Register	241
PCA0CPM1	0xDB	PCA Module 1 Mode Register	241
PCA0CPM2	0xDC	PCA Module 2 Mode Register	241
PCA0H	0xFA	PCA Counter High	242
PCA0L	0xF9	PCA Counter Low	242
PCA0MD	0xD9	PCA Mode	239
PCA0PWM	0xF7	PCA PWM Configuration	240
PCON	0x87	Power Control	122
PSCTL	0x8F	Program Store R/W Control	118
PSW	0xD0	Program Status Word	91
REF0CN	0xD1	Voltage Reference Control	62
REG0CN	0xC9	Voltage Regulator Control	64
REVID	0xB6	Revision ID	96
RSTSRC	0xEF	Reset Source Configuration/Status	128

SFR Definition 18.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	ECP0	EADC0	EPCA0	EWADC0	EMAT	ESMB0
Type	W	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE6

Bit	Name	Function
7	Reserved	Must write 0.
6	Reserved	**Reserved.** Must write 0.
5	ECP0	**Enable Comparator0 (CP0) Interrupt.** This bit sets the masking of the CP0 rising edge or falling edge interrupt. 0: Disable CP0 interrupts. 1: Enable interrupt requests generated by the CP0RIF and CP0FIF flags.
4	EADC0	**Enable ADC0 Conversion Complete Interrupt.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the AD0INT flag.
3	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
2	EWADC0	**Enable Window Comparison ADC0 interrupt.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).
1	EMAT	**Enable Port Match Interrupts.** This bit sets the masking of the Port Match event interrupt. 0: Disable all Port Match interrupts. 1: Enable interrupt requests generated by a Port Match.
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

SFR Definition 21.1. VDM0CN: V_{DD} Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VDMEN	VDDSTAT						
Type	R/W	R	R	R	R	R	R	R
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xFF

Bit	Name	Function
7	VDMEN	**V_{DD} Monitor Enable.** This bit turns the V_{DD} monitor circuit on/off. The V_{DD} Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 21.2). Selecting the V_{DD} monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the V_{DD} Monitor and selecting it as a reset source. After a power-on reset, the VDD monitor is enabled, and this bit will read 1. The state of this bit is sticky through any other reset source. 0: V_{DD} Monitor Disabled. 1: V_{DD} Monitor Enabled.
6	VDDSTAT	**V_{DD} Status.** This bit indicates the current power supply status (V_{DD} Monitor output). 0: V_{DD} is at or below the V_{DD} monitor threshold. 1: V_{DD} is above the V_{DD} monitor threshold.
5:0	Unused	Read = Varies; Write = Don't care.

21.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Section "7. Electrical Characteristics" on page 39 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

21.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than the MCD timeout, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read 1, signifying the MCD as the reset source; otherwise, this bit reads 0. Writing a 1 to the MCDRSF bit enables the Missing Clock Detector; writing a 0 disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

23.1. Port I/O Modes of Operation

Port pins P0.0–P1.7 use the Port I/O cell shown in Figure 23.2. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN and PnMDOUT registers. Port pin P2.0 can be configured by software for digital I/O using the P2MDOUT register. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled. Until the crossbar is enabled (XBARE = 1), both the high and low port I/O drive circuits are explicitly disabled on all crossbar pins.

23.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC input, Capacitive Sense input, external oscillator input/output, VREF output, or AGND connection should be configured for analog I/O (PnMDIN.n = 0, Pn.n = 1). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. To prevent the low port I/o drive circuit from pulling the pin low, a '1' should be written to the corresponding port latch (Pn.n = 1). Port pins configured for analog I/O will always read back a value of 0 regardless of the actual voltage on the pin.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital I/O may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

23.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the VDD or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high and low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the VDD supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

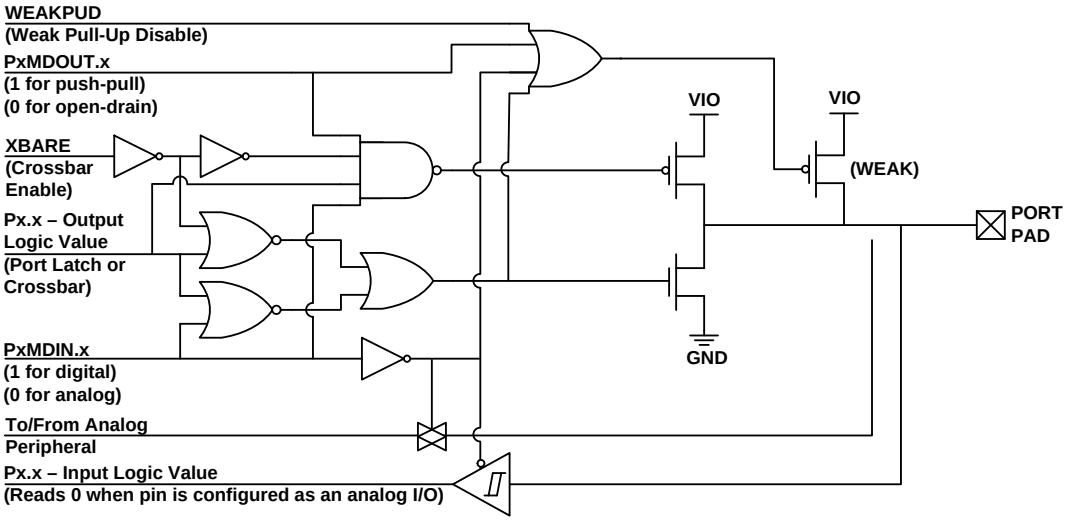

Figure 23.2. Port I/O Cell Block Diagram

Table 23.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, SYSCLK, PCA0 (CEX0-2 and ECI), T0, or T1.	Any Port pin available for assignment by the Crossbar. This includes P0.0 - P1.7[2] pins which have their PnSKIP bit set to 0.[1]	XBR0, XBR1
Any pin used for GPIO	P0.0–P2.0[2]	PnSKIP

Notes:
1. The Crossbar will always assign UART0 pins to P0.4 and P0.5.
2. Port pins P1.4–P1.7 are not available on the 16-pin packages.

23.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions

External digital event capture functions can be used to trigger an interrupt or wake the device from a low power mode when a transition occurs on a digital I/O pin. The digital event capture functions do not require dedicated pins and will function on both GPIO pins (PnSKIP = 1) and pins in use by the Crossbar (PnSKIP = 0). External digital event capture functions cannot be used on pins configured for analog I/O. Table 23.3 shows all available external digital event capture functions.

Table 23.3. Port I/O Assignment for External Digital Event Capture Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
External Interrupt 0	P0.0–P0.7	IT01CF
External Interrupt 1	P0.0–P0.7	IT01CF
Port Match	P0.0–P1.7*	P0MASK, P0MAT P1MASK, P1MAT

Note: Port pins P1.4–P1.7 are not available on the 16-pin packages.

Port	P0								P1								P2
Pin Number	0	1	2	3	4	5	6	7	0	1	2	3	4[1]	5[1]	6[1]	7[1]	0
Special Function Signals	VREF	AGND	XTAL1	XTAL2			CNVSTR										Signal Unavailable to Crossbar
TX0	P0.0 Skipped		P0.2 Skipped	P0.3 Skipped	■												
RX0						■											
SCK		■															
MISO							■										
MOSI								■									
NSS[2]									■								
SDA																	
SCL																	
CP0																	
CP0A																	
SYSCLK																	
CEX0										■							
CEX1											■						
CEX2												■					
ECI																	
T0																	
T1																	
Pin Skip Settings	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
	P0SKIP								P1SKIP								

In this example, the crossbar is configured to assign the UART TX0 and RX0 signals, the SPI signals, and the PCA signals. Note that the SPI signals are assigned as multiple signals. Additionally, pins P0.0, P0.2, and P0.3 are configured to be skipped using the P0SKIP register.

■ These boxes represent the port pins which are used by the peripherals in this configuration.

1st TX0 is assigned to P0.4
2nd RX0 is assigned to P0.5
3rd SCK, MISO, MOSI, and NSS are assigned to P0.1, P0.6, P0.7, and P1.0, respectively.
4th CEX0, CEX1, and CEX2 are assigned to P1.1, P1.2, and P1.3, respectively.

All unassigned pins, including those skipped by XBR0 can be used as GPIO or for other non-crossbar functions.

Notes:
1. P1.4-P1.7 are not available on 16-pin packages.
2. NSS is only pinned out when the SPI is in 4-wire mode.

Figure 23.6. Priority Crossbar Decoder Example 2—Skipping Pins

SFR Definition 23.8. P0MDIN: Port 0 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF1

Bit	Name	Function
7:0	P0MDIN[7:0]	**Analog Configuration Bits for P0.7–P0.0 (respectively).** Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. **In order for the P0.n pin to be in analog mode, there MUST be a '1' in the Port Latch register corresponding to that pin.** 0: Corresponding P0.n pin is configured for analog mode. 1: Corresponding P0.n pin is not configured for analog mode.

SFR Definition 23.9. P0MDOUT: Port 0 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA4

Bit	Name	Function
7:0	P0MDOUT[7:0]	**Output Configuration Bits for P0.7–P0.0 (respectively).** These bits are ignored if the corresponding bit in register P0MDIN is logic 0. 0: Corresponding P0.n Output is open-drain. 1: Corresponding P0.n Output is push-pull.

24.2. 32-bit CRC Algorithm

The C8051F80x-83x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
2. Right-shift the CRC result.
3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit C8051F80x-83x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input){
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ CRC_input;
   // "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 & 0x00000001) == 0x00000001)
      {
         // 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 CRC_acc; // Return the final remainder (CRC value)
}
```

Table 24.2 lists example input values and the associated outputs using the 32-bit C8051F80x-83x CRC algorithm (an initial value of 0xFFFFFFFF is used):

Table 24.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

25. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

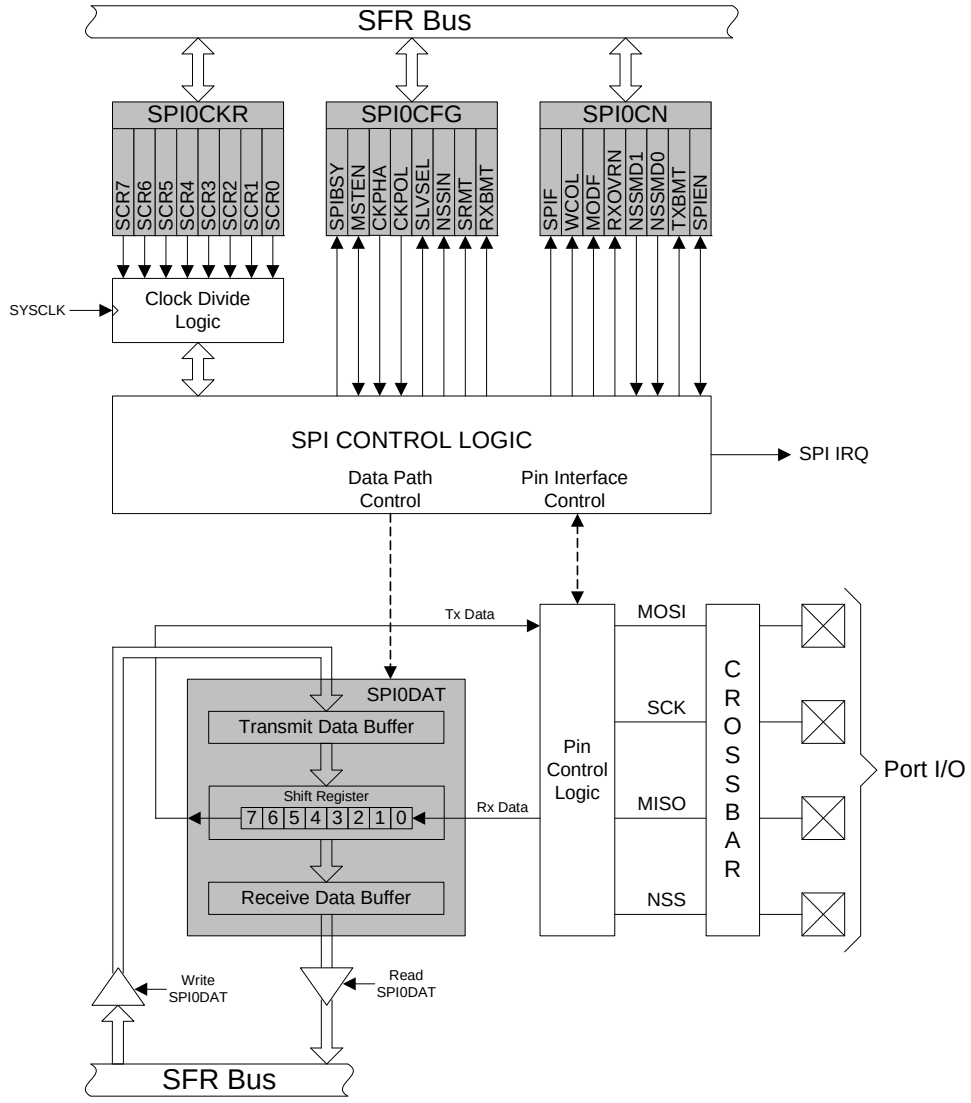

Figure 25.1. SPI Block Diagram

is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 25.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 25.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 25.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

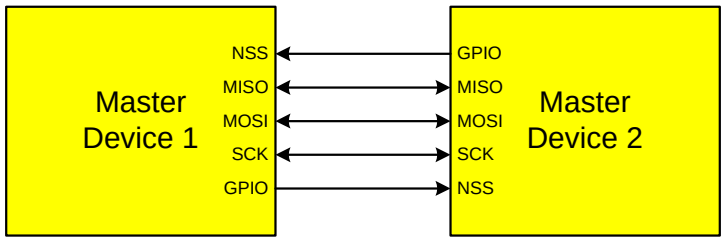

Figure 25.2. Multiple-Master Mode Connection Diagram

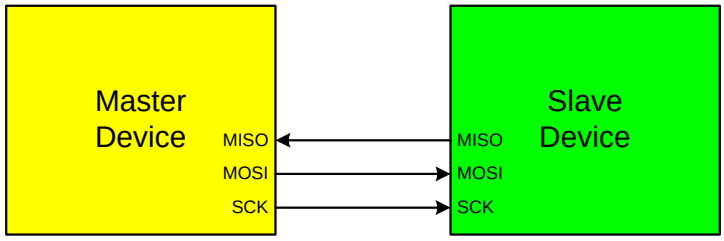

Figure 25.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

26.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 26.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the "data byte transferred" interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

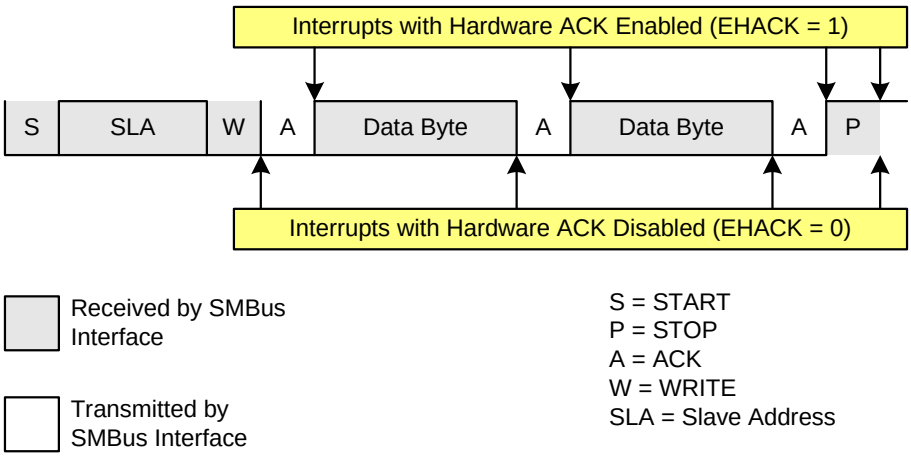

Figure 26.7. Typical Slave Write Sequence

SFR Definition 28.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8A

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 28.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8B

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

C2 Register Definition 30.2. DEVICEID: C2 Device ID

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

C2 Address: 0x00

Bit	Name	Function
7:0	DEVICEID[7:0]	**Device ID.** This read-only register returns the 8-bit device ID: 0x23 (C8051F80x-83x).

C2 Register Definition 30.3. REVID: C2 Revision ID

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

C2 Address: 0x01

Bit	Name	Function
7:0	REVID[7:0]	**Revision ID.** This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.