E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Last Time Buy
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K × 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 16x10b
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/c8051f813-gur

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

15.2 Data Memory	93
15.2.1 Internal RAM	00 Q3
15 2 1 1 General Purpose Registers	94
15 2 1 2 Bit Addressable Locations	94
15 2 1 3 Stack	94
16. In-System Device Identification	
17. Special Function Registers	
18 Interrunts	102
18.1 MCU Interrupt Sources and Vectors	103
18.1.1. Interrupt Priorities	. 103
18.1.2. Interrupt Latency	103
18.2. Interrupt Register Descriptions	104
18.3. INTO and INT1 External Interrupts	. 111
19. Flash Memory	. 113
19.1. Programming The Flash Memory	. 113
19.1.1. Flash Lock and Key Functions	. 113
19.1.2. Flash Erase Procedure	. 113
19.1.3. Flash Write Procedure	. 114
19.2. Non-volatile Data Storage	114
19.3. Security Options	114
19.4. Flash Write and Erase Guidelines	115
19.4.1. VDD Maintenance and the VDD Monitor	116
19.4.2. PSWE Maintenance	116
19.4.3. System Clock	. 117
20. Power Management Modes	. 120
20.1. Idle Mode	. 120
20.2. Stop Mode	. 121
20.3. Suspend Mode	. 121
21. Reset Sources	. 123
21.1. Power-On Reset	. 124
21.2. Power-Fail Reset / VDD Monitor	. 125
21.3. External Reset	. 126
21.4. Missing Clock Detector Reset	. 126
21.5. Comparator0 Reset	. 127
21.6. PCA Watchdog Timer Reset	. 127
21.7. Flash Error Reset	. 127
21.8. Software Reset	. 127
22. Oscillators and Clock Selection	. 129
22.1. System Clock Selection	. 129
22.2. Programmable Internal High-Frequency (H-F) Oscillator	. 131
22.3. External Oscillator Drive Circuit	. 133
22.3.1. External Crystal Example	. 135
22.3.2. External RC Example	. 136
22.3.3. External Capacitor Example	. 13/
23. Port input/Output	. 138









SFR Definition 8.7. ADC0LTH: ADC0 Less-Than Data High Byte

Bit	7	6	5	4	3	2	1	0
Nam	e	ADC0LTH[7:0]						
Туре	R/W							
Rese	et O	0	0	0	0	0	0	0
SFR A	SFR Address = 0xC6							
Bit	Name		Function					
7:0	ADC0LTH[7:0]	ADC0 Le	ADC0 Less-Than Data Word High-Order Bits.					

SFR Definition 8.8. ADC0LTL: ADC0 Less-Than Data Low Byte

Bit	7	6	5	4	3	2	1	0
Nam	e	ADC0LTL[7:0]						
Туре	e	R/W						
Rese	et O	0	0	0	0	0	0	0
SFR A	SFR Address = 0xC5							
Bit	Name		Function					
7:0	ADC0LTL[7:0]	ADC0 Le	ADC0 Less-Than Data Word Low-Order Bits.					



SFR Definition 12.2. CPT0MD: Comparator0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP0RIE	CP0FIE			CP0M	D[1:0]
Туре	R	R	R/W	R/W	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9D

Bit	Name	Function
7:6	Unused	Read = 00b, Write = Don't Care.
5	CP0RIE	Comparator0 Rising-Edge Interrupt Enable.
		0: Comparator0 Rising-edge interrupt disabled.
		1: Comparator0 Rising-edge interrupt enabled.
4	CP0FIE	Comparator0 Falling-Edge Interrupt Enable.
		0: Comparator0 Falling-edge interrupt disabled.
		1: Comparator0 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP0MD[1:0]	Comparator0 Mode Select.
		These bits affect the response time and power consumption for Comparator0.
		00: Mode 0 (Fastest Response Time, Highest Power Consumption)
		01: Mode 1
		10: Mode 2
		11: Mode 3 (Slowest Response Time, Lowest Power Consumption)



13.1. Configuring Port Pins as Capacitive Sense Inputs

In order for a port pin to be measured by CS0, that port pin must be configured as an analog input (see "23. Port Input/Output"). Configuring the input multiplexer to a port pin not configured as an analog input will cause the capacitive sense comparator to output incorrect measurements.

13.2. Capacitive Sense Start-Of-Conversion Sources

A capacitive sense conversion can be initiated in one of seven ways, depending on the programmed state of the CS0 start of conversion bits (CS0CF6:4). Conversions may be initiated by one of the following:

- 1. Writing a 1 to the CS0BUSY bit of register CS0CN
- 2. Timer 0 overflow
- 3. Timer 2 overflow
- 4. Timer 1 overflow
- 5. Convert continuously
- 6. Convert continuously with auto-scan enabled

Conversions can be configured to be initiated continuously through one of two methods. CS0 can be configured to convert at a single channel continuously or it can be configured to convert continuously with auto-scan enabled. When configured to convert continuously, conversions will begin after the CS0BUSY bit in CS0CF has been set.

An interrupt will be generated if CS0 conversion complete interrupts are enabled by setting the ECSCPT bit (EIE2.0).

Note: CS0 conversion complete interrupt behavior depends on the settings of the CS0 accumulator. If CS0 is configured to accumulate multiple conversions on an input channel, a CS0 conversion complete interrupt will be generated only after the last conversion completes.

13.3. Automatic Scanning

CS0 can be configured to automatically scan a sequence of contiguous CS0 input channels by configuring and enabling auto-scan. Using auto-scan with the CS0 comparator interrupt enabled allows a system to detect a change in measured capacitance without requiring any additional dedicated MCU resources.

Auto-scan is enabled by setting the CS0 start-of-conversion bits (CS0CF6:4) to 111b. After enabling autoscan, the starting and ending channels should be set to appropriate values in CS0SS and CS0SE, respectively. Writing to CS0SS when auto-scan is enabled will cause the value written to CS0SS to be copied into CS0MX. After being enabled, writing a 1 to CS0BUSY will start auto-scan conversions. When auto-scan completes the number of conversions defined in the CS0 accumulator bits (CS0CF1:0) (see "13.5. CS0 Conversion Accumulator"), auto-scan configures CS0MX to the next highest port pin configured as an analog input and begins a conversion on that channel. This scan sequence continues until CS0MX reaches the ending input channel value defined in CS0SE. After one or more conversions have been taken at this channel, auto-scan configures CS0MX back to the starting input channel. For an example system configured to use auto-scan, please see Figure "13.2 Auto-Scan Example" on page 73.

Note: Auto-scan attempts one conversion on a CS0MX channel regardless of whether that channel's port pin has been configured as an analog input.

If auto-scan is enabled when the device enters suspend mode, auto-scan will remain enabled and running. This feature allows the device to wake from suspend through CS0 greater-than comparator event on any configured capacitive sense input included in the auto-scan sequence of inputs.





Figure 13.2. Auto-Scan Example

13.4. CS0 Comparator

The CS0 comparator compares the latest capacitive sense conversion result with the value stored in CS0THH:CS0THL. If the result is less than or equal to the stored value, the CS0CMPF bit(CS0CN:0) is set to 0. If the result is greater than the stored value, CS0CMPF is set to 1.

If the CS0 conversion accumulator is configured to accumulate multiple conversions, a comparison will not be made until the last conversion has been accumulated.

An interrupt will be generated if CS0 greater-than comparator interrupts are enabled by setting the ECS-GRT bit (EIE2.1) when the comparator sets CS0CMPF to 1.

If auto-scan is running when the comparator sets the CS0CMPF bit, no further auto-scan initiated conversions will start until firmware sets CS0BUSY to 1.

A CS0 greater-than comparator event can wake a device from suspend mode. This feature is useful in systems configured to continuously sample one or more capacitive sense channels. The device will remain in the low-power suspend state until the captured value of one of the scanned channels causes a CS0 greater-than comparator event to occur. It is not necessary to have CS0 comparator interrupts enabled in order to wake a device from suspend with a greater-than event.

Note: On waking from suspend mode due to a CS0 greater-than comparator event, the CS0CN register should be accessed only after at least two system clock cycles have elapsed.

For a summary of behavior with different CS0 comparator, auto-scan, and auto accumulator settings, please see Table 13.1.



15. 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 C8051F80x-83x device family is shown in Figure 15.1



Figure 15.1. C8051F80x-83x Memory Map



Table 17.2. Special Function Registers

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

Register	Address	Description	Page
ACC	0xE0	Accumulator	89
ADC0CF	0xBC	ADC0 Configuration	50
ADC0CN	0xE8	ADC0 Control	52
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	53
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	53
ADC0H	0xBE	ADC0 High	51
ADC0L	0xBD	ADC0 Low	51
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	54
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	54
ADCOMX	0xBB	AMUX0 Multiplexer Channel Select	57
В	0xF0	B Register	90
CKCON	0x8E	Clock Control	210
CLKSEL	0xA9	Clock Select	210
CPT0CN	0x9B	Comparator0 Control	67
CPT0MD	0x9D	Comparator0 Mode Selection	68
СРТОМХ	0x9F	Comparator0 MUX Selection	70
CRC0AUTO	0xD2	CRC0 Automatic Control Register	165
CRC0CN	0xCE	CRC0 Control	163
CRC0CNT	0xD3	CRC0 Automatic Flash Sector Count	165
CRC0DATA	0xDE	CRC0 Data Output	164
CRC0FLIP	0xCF	CRC0 Bit Flip	166
CRCOIN	0xDD	CRC Data Input	164
CS0THH	0x97	CS0 Digital Compare Threshold High	79
CS0THL	0x96	CS0 Digital Compare Threshold High	79
CS0CN	0xB0	CS0 Control	75
CS0DH	0xAC	CS0 Data High	77
CSODL	0xAB	CS0 Data Low	77



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 18.2. IP: Interrupt Priority

r		1						
Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Туре	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	 Serial Peripheral Interface (SPI0) Interrupt Priority Control. This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	 Timer 2 Interrupt Priority Control. This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	 UART0 Interrupt Priority Control. This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	Timer 1 Interrupt Priority Control.This bit sets the priority of the Timer 1 interrupt.0: Timer 1 interrupt set to low priority level.1: Timer 1 interrupt set to high priority level.
2	PX1	External Interrupt 1 Priority Control. This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	Timer 0 Interrupt Priority Control.This bit sets the priority of the Timer 0 interrupt.0: Timer 0 interrupt set to low priority level.1: Timer 0 interrupt set to high priority level.
0	PX0	 External Interrupt 0 Priority Control. This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.



SFR Definition 18.6. EIP2: Extended Interrupt Priority 2

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PSCGRT	PSCCPT
Туре	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4

Bit	Name	Function
7:2	Reserved	
1	PSCGRT	Capacitive Sense Greater Than Comparator Priority Control.
		This bit sets the priority of the Capacitive Sense Greater Than Comparator interrupt. 0: CS0 Greater Than Comparator interrupt set to low priority level. 1: CS0 Greater Than Comparator set to high priority level.
0	PSCCPT	Capacitive Sense Conversion Complete Priority Control.
		This bit sets the priority of the Capacitive Sense Conversion Complete interrupt.
		0: CS0 Conversion Complete set to low priority level.
		1: CS0 Conversion Complete set to high priority level.



The following guidelines are recommended for any system that contains routines which write or erase Flash from code.

19.4.1. VDD Maintenance and the VDD Monitor

- 1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
- 2. Make certain that the minimum VDD rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external VDD brownout circuit to the RST pin of the device that holds the device in reset until VDD reaches the minimum device operating voltage and re-asserts RST if VDD drops below the minimum device operating voltage.
- 3. Keep the on-chip VDD Monitor enabled and enable the VDD Monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Code examples showing this can be found in "AN201: Writing to Flash from Firmware," available from the Silicon Laboratories website.
- **Note:** On C8051F80x-83x devices, both the VDD Monitor and the VDD Monitor reset source must be enabled to write or erase Flash without generating a Flash Error Device Reset.

On C8051F80x-83x devices, both the VDD Monitor and the VDD Monitor reset source are enabled by hardware after a power-on reset.

- 4. As an added precaution, explicitly enable the VDD Monitor and enable the VDD Monitor as a reset source inside the functions that write and erase Flash memory. The VDD Monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.
- Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct, but "RSTSRC |= 0x02" is incorrect.
- 6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

19.4.2. PSWE Maintenance

- 1. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write Flash bytes and one routine in code that sets both PSWE and PSEE both to a 1 to erase Flash pages.
- 2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in "AN201: Writing to Flash from Firmware," available from the Silicon Laboratories website.
- 3. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
- Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
- 5. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.



SFR Definition 19.1. PSCTL: Program Store R/W Control

Bit	7	6	5	4	3	2	1	0
Name							PSEE	PSWE
Туре	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address =0x8F

Bit	Name	Function
7:2	Unused	Read = 000000b, Write = don't care.
1	PSEE	Program Store Erase Enable.
		Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. 0: Flash program memory erasure disabled. 1: Flash program memory erasure enabled.
0	PSWE	Program Store Write Enable.
		 Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The Flash location should be erased before writing data. 0: Writes to Flash program memory disabled. 1: Writes to Flash program memory enabled; the MOVX write instruction targets Flash memory.



23. Port Input/Output

Digital and analog resources are available through 17 I/O pins (24-pin and 20-pin packages) or 13 I/O pins (16-pin packages). Port pins P0.0–P1.7 can be defined as general-purpose I/O (GPIO) or assigned to one of the internal digital resources as shown in Figure 23.4. Port pin P2.0 can be used as GPIO and is shared with the C2 Interface Data signal (C2D). The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 23.5). The registers XBR0 and XBR1, defined in SFR Definition 23.1 and SFR Definition 23.2, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 23.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1). Complete Electrical Specifications for Port I/O are given in Section "7. Electrical Characteristics" on page 39.



Figure 23.1. Port I/O Functional Block Diagram



Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P1.7	ADC0MX, PnSKIP, PnMDIN
Comparator0 Input	P0.0-P1.7	CPT0MX, PnSKIP, PnMDIN
CS0 Input	P0.0–P1.7	CS0MX, CS0SS, CS0SE, PnMDIN
Voltage Reference (VREF0)	P0.0	REF0CN, P0SKIP, PnMDIN
Ground Reference (AGND)	P0.1	REF0CN, P0SKIP
External Oscillator in Crystal Mode (XTAL1)	P0.2	OSCXCN, POSKIP, POMDIN
External Oscillator in RC, C, or Crystal Mode (XTAL2)	P0.3	OSCXCN, POSKIP, POMDIN

Table 23.1. Port I/O Assignment for Analog Functions

23.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital func-tions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 23.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.



24.6. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 24.1. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

SFR Definition 24.6. CRC0FLIP: CRC Bit Flip

Bit	7	6	5	4	3	2	1	0
Name	CRC0FLIP[7:0]							
Туре	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCF

Bit	Name	Function
7:0	CRC0FLIP[7:0]	CRC0 Bit Flip.
		Any byte written to CRC0FLIP is read back in a bit-reversed order, i.e. the written LSB becomes the MSB. For example:
		If 0xC0 is written to CRC0FLIP, the data read back will be 0x03.
		If 0x05 is written to CRC0FLIP, the data read back will be 0xA0.



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



28. Timers

Each MCU includes three counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and one is a 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 2 offers the ability to be clocked from the external oscillator while the device is in Suspend mode, and can be used as a wake-up source. This allows for implementation of a very low-power system, including RTC capability.

Timer 0 and Timer 1 Modes	Timer 2 Modes	
13-bit counter/timer	16-bit timer with auto-reload	
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 28.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.









NOTES:

