

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	13
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 12x12b
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	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f860-c-gsr

Table 1.2. Power Consumption (Continued)

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
ADC0 Burst Mode, 10-bit single conversions, internal reference, Low power bias settings	I_{ADC}	200 ksps, $V_{DD} = 3.0\text{ V}$	—	490	—	μA
		100 ksps, $V_{DD} = 3.0\text{ V}$	—	245	—	μA
		10 ksps, $V_{DD} = 3.0\text{ V}$	—	23	—	μA
ADC0 Burst Mode, 12-bit single conversions, external reference	I_{ADC}	100 ksps, $V_{DD} = 3.0\text{ V}$	—	530	—	μA
		50 ksps, $V_{DD} = 3.0\text{ V}$	—	265	—	μA
		10 ksps, $V_{DD} = 3.0\text{ V}$	—	53	—	μA
ADC0 Burst Mode, 12-bit single conversions, internal reference	I_{ADC}	100 ksps, $V_{DD} = 3.0\text{ V}$, Normal bias	—	950	—	μA
		50 ksps, $V_{DD} = 3.0\text{ V}$, Low power bias	—	420	—	μA
		10 ksps, $V_{DD} = 3.0\text{ V}$, Low power bias	—	85	—	μA
Internal ADC0 Reference, Always-on ⁵	I_{IREF}	Normal Power Mode	—	680	790	μA
		Low Power Mode	—	160	210	μA
Temperature Sensor	I_{TSENSE}		—	75	120	μA
Comparator 0 (CMP0), Comparator 1 (CMP1)	I_{CMP}	CPnMD = 11	—	0.5	—	μA
		CPnMD = 10	—	3	—	μA
		CPnMD = 01	—	10	—	μA
		CPnMD = 00	—	25	—	μA
Voltage Supply Monitor (VMON0)	I_{VMON}		—	15	20	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.

1.4. Absolute Maximum Ratings

Stresses above those listed under Table 1.13 may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

For more information on the available quality and reliability data, see the Quality and Reliability Monitor Report at <http://www.silabs.com/support/quality/pages/default.aspx>.

Table 1.13. Absolute Maximum Ratings

Parameter	Symbol	Test Condition	Min	Max	Unit
Ambient Temperature Under Bias	T _{BIAS}		−55	125	°C
Storage Temperature	T _{STG}		−65	150	°C
Voltage on V _{DD}	V _{DD}		GND−0.3	4.2	V
Voltage on I/O pins or $\overline{\text{RST}}$	V _{IN}	V _{DD} ≥ 3.3 V	GND−0.3	5.8	V
		V _{DD} < 3.3 V	GND−0.3	V _{DD} +2.5	V
Total Current Sunk into Supply Pin	I _{VDD}		—	400	mA
Total Current Sourced out of Ground Pin	I _{GND}		400	—	mA
Current Sourced or Sunk by Any I/O Pin or $\overline{\text{RST}}$	I _{PIO}		-100	100	mA
Operating Junction Temperature	T _J	Commercial Grade Devices (-GM, -GS, -GU)	−40	105	°C
		Industrial Grade Devices (-IM, -IS, -IU)	−40	125	°C
Note: Exposure to maximum rating conditions for extended periods may affect device reliability.					

12. Interrupts

The C8051F85x/86x includes an extended interrupt system supporting multiple interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE and EIE1). However, interrupts must first be globally enabled by setting the EA bit in the IE register to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

12.1. MCU Interrupt Sources and Vectors

The C8051F85x/86x MCUs support interrupt sources for each peripheral on the device. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 12.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

12.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 12.1.

12.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock

Table 12.2. IE Register Bit Descriptions

Bit	Name	Function
0	EX0	Enable External Interrupt 0. This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the INT0 input.

17. Comparators (CMP0 and CMP1)

C8051F85x/86x devices include two on-chip programmable voltage comparators, CMP0 and CMP1. The two comparators are functionally identical, but have different connectivity within the device. A functional block diagram is shown in Figure 17.1.

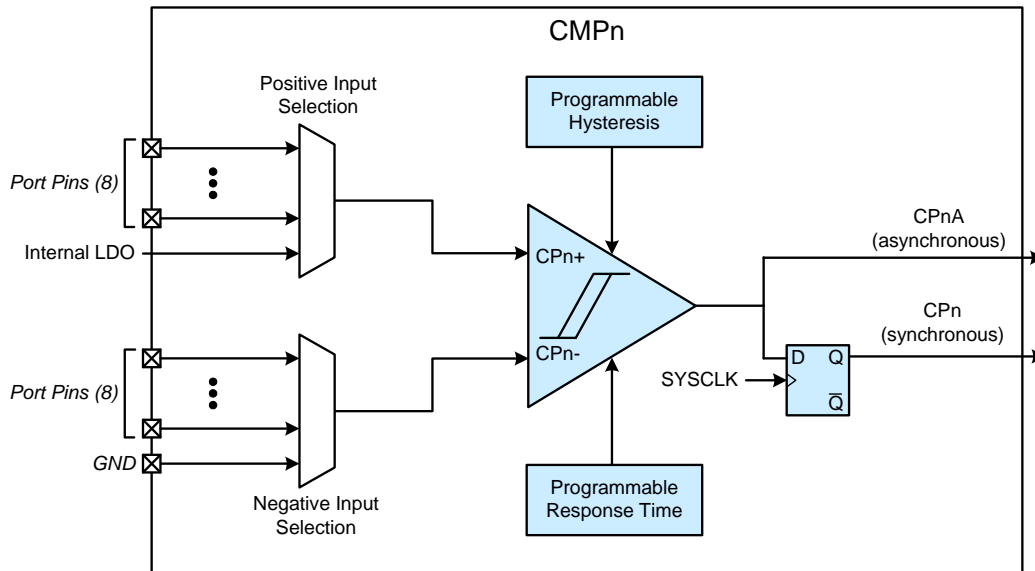


Figure 17.1. Comparator Functional Block Diagram

17.1. System Connectivity

Comparator inputs are routed to port I/O pins or internal signals using the comparator mux registers. The comparator's synchronous and asynchronous outputs can optionally be routed to port I/O pins through the port I/O crossbar. The output of either comparator may also be configured to generate a system interrupt. CMP0 may also be used as a reset source, or as a trigger to kill a PCA output channel.

The CMP0 inputs are selected in the CPT0MX register, while CPT1MX selects the CMP1 inputs. The CMXP field selects the comparator's positive input (CPnP.x); the CMXN field selects the comparator's negative input (CPnN.x). Table 17.1 through Table 17.4 detail the comparator input multiplexer options on the C8051F85x/86x family. See the port I/O crossbar sections for details on configuring comparator outputs via the digital crossbar. Comparator inputs can be externally driven from -0.25 V to $(V_{DD}) + 0.25\text{ V}$ without damage or upset.

Important Note About Comparator Inputs: The port pins selected as comparator inputs should be configured as analog inputs in their associated port configuration register, and configured to be skipped by the crossbar.

20.4.3. 8 to 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8 through 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9 through 11-bit PWM modes. It is important to note that all channels configured for 8 to 11-bit PWM mode will use the same cycle length. It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently. Each channel configured for a PWM mode can be individually selected to operate in edge-aligned or center-aligned mode.

20.4.3.1. 8-bit Pulse Width Modulator Mode

In 8-bit PWM mode, the duty cycle is determined by the value of the low byte of the PCA0CPn register (PCA0CPLn). To adjust the duty cycle, PCA0CPLn should not normally be written directly. Instead, it is recommended to adjust the duty cycle using the high byte of the PCA0CPn register (register PCA0CPHn). This allows seamless updating of the PWM waveform, as PCA0CPLn is reloaded automatically with the value stored in PCA0CPHn during the overflow edge (in edge-aligned mode) or the up edge (in center-aligned mode).

Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a match edge or up edge occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles.

20.4.3.2. 9 to 11-bit Pulse Width Modulator Mode

In 9 to 11-bit PWM mode, the duty cycle is determined by the value of the least significant N bits of the PCA0CPn register, where N is the selected PWM resolution.

To adjust the duty cycle, PCA0CPn should not normally be written directly. Instead, it is recommended to adjust the duty cycle by writing to an “Auto-Reload” register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0. This allows seamless updating of the PWM waveform, as the PCA0CPn register is reloaded automatically with the value stored in the auto-reload registers during the overflow edge (in edge-aligned mode) or the up edge (in center-aligned mode).

Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a match edge or up edge occurs. The COVF flag in PCA0PWM can be used to detect the overflow or down edge.

The 9 to 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a match edge or up edge occurs. The COVF flag in PCA0PWM can be used to detect the overflow or down edge.

Important Note About PCA0CPHn and PCA0CPLn Registers: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

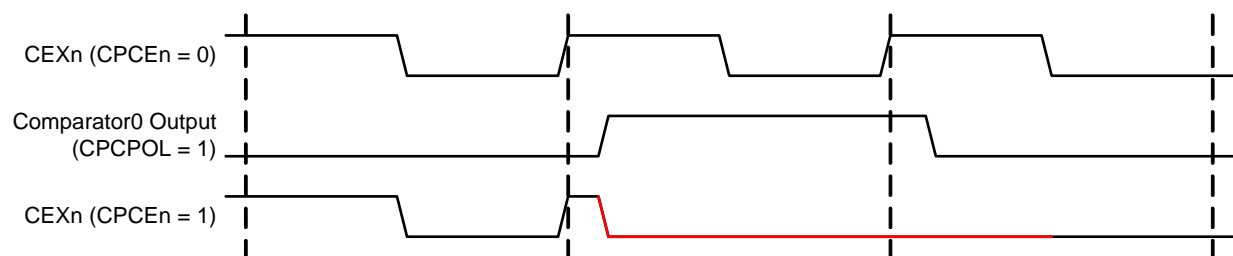


Figure 20.11. CEXn with CPCEn = 1, CPCPOL = 1

20.6. PCA Control Registers

Register 20.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	Reserved			CCF2	CCF1	CCF0
Type	RW	RW	R			RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD8 (bit-addressable)								

Table 20.3. PCA0CN Register Bit Descriptions

Bit	Name	Function
7	CF	PCA Counter/Timer Overflow Flag. Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	PCA Counter/Timer Run Control. This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled. 1: PCA Counter/Timer enabled.
5:3	Reserved	Must write reset value.
2	CCF2	PCA Module 2 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	PCA Module 1 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	PCA Module 0 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Register 20.13. PCA0CPM2: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

SFR Address: 0xDC

Table 20.15. PCA0CPM2 Register Bit Descriptions

Bit	Name	Function
7	PWM16	16-bit Pulse Width Modulation Enable. This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	Comparator Function Enable. This bit enables the comparator function.
5	CAPP	Capture Positive Function Enable. This bit enables the positive edge capture capability.
4	CAPN	Capture Negative Function Enable. This bit enables the negative edge capture capability.
3	MAT	Match Function Enable. This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF2 bit in the PCA0MD register to be set to logic 1.
2	TOG	Toggle Function Enable. This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX2 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	Pulse Width Modulation Mode Enable. This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX2 pin. 8 to 11-bit PWM is used if PWM16 is cleared; 16-bit mode is used if PWM16 is set to logic 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	Capture/Compare Flag Interrupt Enable. This bit sets the masking of the Capture/Compare Flag (CCF2) interrupt. 0: Disable CCF2 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF2 is set.

21.3. Priority Crossbar Decoder

The priority crossbar decoder assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned port pin is assigned to that resource (excluding UART0, which is always at pins P0.4 and P0.5). If a port pin is assigned, the crossbar skips that pin when assigning the next selected resource. Additionally, the crossbar will skip port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a port pin is claimed by a peripheral without use of the crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.1 if AGND is used, P0.3 if the EXTCLK input is enabled, P0.6 if the ADC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or comparator inputs. The crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

Figure 21.2 shows all of the potential peripheral-to-pin assignments available to the crossbar. Note that this does not mean any peripheral can always be assigned to the highlighted pins. The actual pin assignments are determined by the priority of the enabled peripherals.

Port	P0							P1							P2			
Pin Number	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1
SOIC-16 Package	VREF													N/A	N/A	N/A		
QFN-20 Package					EXTCLK												C2D	N/A
QSOP-24 Package								CNVSTR										
UART0-TX																		
UART0-RX																		
SPI0-SCK																		
SPI0-MISO																		
SPI0-MOSI																		
SPI0-NSS*																		
SMB0-SDA																		
SMB0-SCL																		
CMP0-CP0																		
CMP0-CP0A																		
CMP1-CP1																		
CMP1-CP1A																		
SYSCLK																		
PCA0-CEX0																		
PCA0-CEX1																		
PCA0-CEX2																		
PCA0-ECI																		
Timer0-T0																		
Timer1-T1																		
Timer2-T2																		
Pin Skip Settings	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	P0SKIP								P1SKIP									

The crossbar peripherals are assigned in priority order from top to bottom.

■ These boxes represent Port pins which can potentially be assigned to a peripheral.

□ Special Function Signals are not assigned by the crossbar. When these signals are enabled, the Crossbar should be manually configured to skip the corresponding port pins.

□ Pins can be "skipped" by setting the corresponding bit in PnSKIP to 1.

* NSS is only pinned out when the SPI is in 4-wire mode.

Figure 21.2. Crossbar Priority Decoder - Possible Pin Assignments

Register 21.3. XBR2: Port I/O Crossbar 2

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved					
Type	RW	RW	R					
Reset	0	0	0	0	0	0	0	0

SFR Address: 0xE3**Table 21.6. XBR2 Register Bit Descriptions**

Bit	Name	Function
7	WEAKPUD	Port I/O Weak Pullup Disable. 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	Crossbar Enable. 0: Crossbar disabled. 1: Crossbar enabled.
5:0	Reserved	Must write reset value.

23.7. SPI Control Registers

Register 23.1. SPI0CFG: SPI0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	RW	RW	RW	R	R	R	R
Reset	0	0	0	0	0	1	1	1
SFR Address: 0xA1								

Table 23.2. SPI0CFG Register Bit Descriptions

Bit	Name	Function
7	SPIBSY	SPI Busy. This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	Master Mode Enable. 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	SPI0 Clock Phase. 0: Data centered on first edge of SCK period. 1: Data centered on second edge of SCK period.
4	CKPOL	SPI0 Clock Polarity. 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	Slave Selected Flag. This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	NSS Instantaneous Pin Input. This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	Shift Register Empty (valid in slave mode only). This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device.		

Table 23.2. SPI0CFG Register Bit Descriptions

Bit	Name	Function
0	RXBMT	Receive Buffer Empty (valid in slave mode only). This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.
Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device.		

Table 23.3. SPI0CN Register Bit Descriptions

Bit	Name	Function
0	SPIEN	SPI0 Enable. 0: SPI disabled. 1: SPI enabled.

24.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data.

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.**

Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 24.6 shows a typical master read 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.

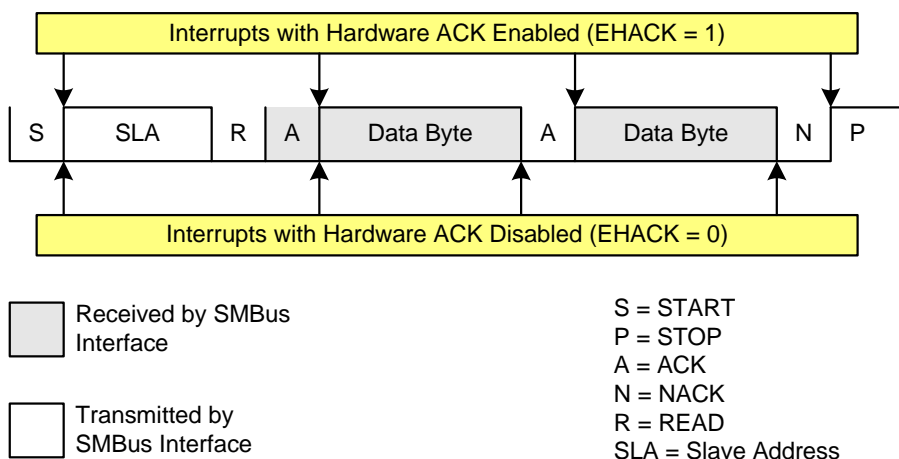


Figure 24.6. Typical Master Read Sequence

24.5.3. 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. The interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 24.7 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.

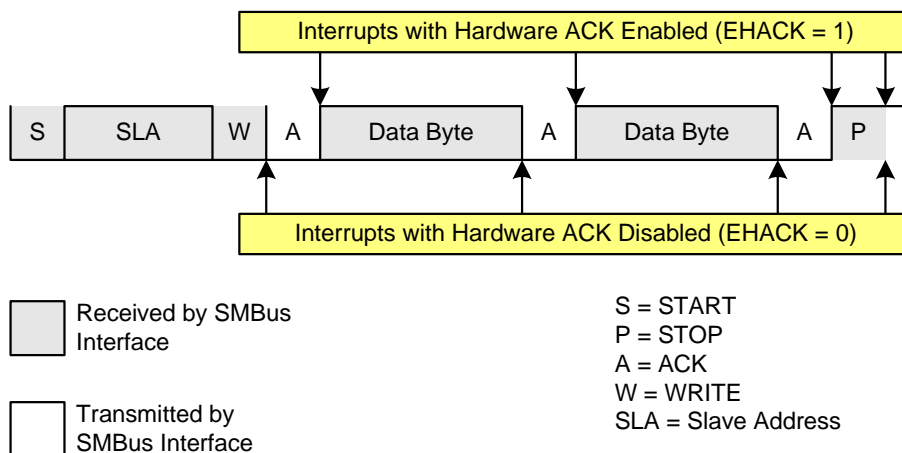


Figure 24.7. Typical Slave Write Sequence

Table 24.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (Continued)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	—
						Reschedule failed transfer.	1	0	0	1110

Table 24.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0-DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000

Register 25.9. TMR2RLL: Timer 2 Reload Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xCA								

Table 25.11. TMR2RLL Register Bit Descriptions

Bit	Name	Function
7:0	TMR2RLL	Timer 2 Reload Low Byte. When operating in one of the auto-reload modes, TMR2RLL holds the reload value for the low byte of Timer 2 (TMR2L). When operating in capture mode, TMR2RLL is the captured value of TMR2L.

Table 26.1. Timer Settings for Standard Baud Rates Using the Internal 24.5 MHz Oscillator

Frequency: 49 MHz							
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select) ¹	T1M ¹	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX ²	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B
Notes: <ol style="list-style-type: none"> 1. SCA1–SCA0 and T1M bit definitions can be found in Timer1 chapter. 2. X = Don't care. 							

27. Watchdog Timer (WDT0)

The C8051F85x/86x family includes a programmable Watchdog Timer (WDT) running off the low-frequency oscillator. A WDT overflow will force the MCU into the reset state. To prevent the reset, the WDT must be restarted by application software before overflow. If the system experiences a software or hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset.

Following a reset the WDT is automatically enabled and running with the default maximum time interval. If desired the WDT can be disabled by system software or locked on to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the RST pin is unaffected by this reset.

The WDT consists of an internal timer running from the low-frequency oscillator. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. When the WDT is active, the low-frequency oscillator is forced on. All watchdog features are controlled via the Watchdog Timer Control Register (WDTCN).

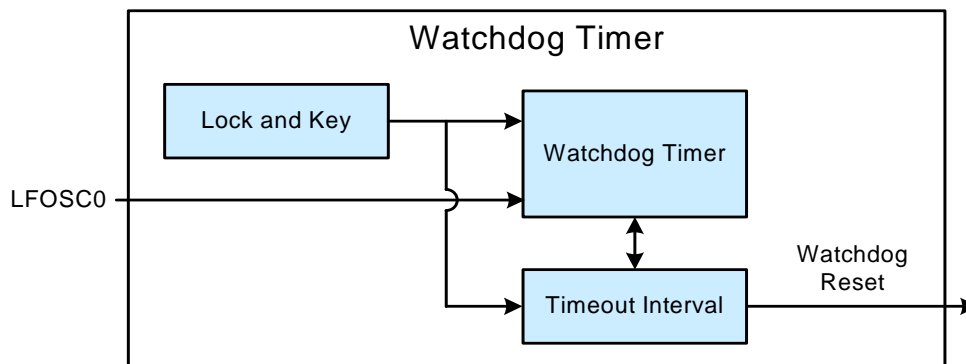


Figure 27.1. Watchdog Timer Block Diagram