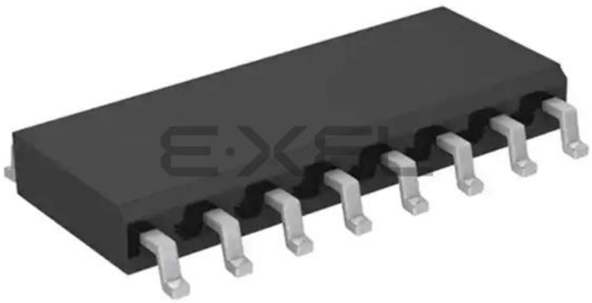


What is "Embedded - Microcontrollers"?

"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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
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	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/c8051f861-c-gs

2.1.3.1. Normal Mode

Normal mode encompasses the typical full-speed operation. The power consumption of the device in this mode will vary depending on the system clock speed and any analog peripherals that are enabled.

2.1.3.2. Idle Mode

Setting the IDLE bit in PCON causes the hardware to halt the CPU and enter idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the IDLE bit to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

2.1.3.3. Stop Mode (Regulator On)

Setting the STOP bit in PCON when STOPCF in REG0CN is clear causes the controller core to enter stop mode as soon as the instruction that sets the bit completes execution. In stop mode the internal oscillator, CPU, and all digital peripherals are stopped. Each analog peripheral may be shut down individually prior to entering stop mode. Stop mode can only be terminated by an internal or external reset.

2.1.3.4. Shutdown Mode (Regulator Off)

Shutdown mode is an extension of the normal stop mode operation. Setting the STOP bit in PCON when STOPCF in REG0CN is also set causes the controller core to enter shutdown mode as soon as the instruction that sets the bit completes execution, and then the internal regulator is powered down. In shutdown mode, all core functions, memories and peripherals are powered off. An external pin reset or power-on reset is required to exit shutdown mode.

2.2. I/O

2.2.1. General Features

The C8051F85x/86x ports have the following features:

- Push-pull or open-drain output modes and analog or digital modes.
- Port Match allows the device to recognize a change on a port pin value and wake from idle mode or generate an interrupt.
- Internal pull-up resistors can be globally enabled or disabled.
- Two external interrupts provide unique interrupt vectors for monitoring time-critical events.
- Above-rail tolerance allows 5 V interface when device is powered.

2.2.2. Crossbar

The C8051F85x/86x devices have a digital peripheral crossbar with the following features:

- Flexible peripheral assignment to port pins.
- Pins can be individually skipped to move peripherals as needed for design or layout considerations.

The crossbar has a fixed priority for each I/O function and assigns these functions to the port pins. When a digital resource is selected, the least-significant unassigned port pin is assigned to that resource. 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. This provides some flexibility when designing a system: pins involved with sensitive analog measurements can be moved away from digital I/O and peripherals can be moved around the chip as needed to ease layout constraints.

2.6. Analog Peripherals

2.6.1. 12-Bit Analog-to-Digital Converter (ADC0)

The ADC0 module on C8051F85x/86x devices is a Successive Approximation Register (SAR) Analog to Digital Converter (ADC). The key features of the ADC module are:

- Single-ended 12-bit and 10-bit modes.
- Supports an output update rate of 200 ksps samples per second in 12-bit mode or 800 ksps samples per second in 10-bit mode.
- Operation in low power modes at lower conversion speeds.
- Selectable asynchronous hardware conversion trigger.
- Output data window comparator allows automatic range checking.
- Support for Burst Mode, which produces one set of accumulated data per conversion-start trigger with programmable power-on settling and tracking time.
- Conversion complete and window compare interrupts supported.
- Flexible output data formatting.
- Includes an internal fast-settling reference with two levels (1.65 V and 2.4 V) and support for external reference and signal ground.

2.6.2. Low Current Comparators (CMP0, CMP1)

The comparators take two analog input voltages and output the relationship between these voltages (less than or greater than) as a digital signal. The Low Power Comparator module includes the following features:

- Multiple sources for the positive and negative poles, including VDD, VREF, and I/O pins.
- Two outputs are available: a digital synchronous latched output and a digital asynchronous raw output.
- Programmable hysteresis and response time.
- Falling or rising edge interrupt options on the comparator output.
- Provide “kill” signal to PCA module.
- Comparator 0 can be used to reset the device.

10.2. Programming the Flash Memory

Writes to flash memory clear bits from logic 1 to logic 0, and can be performed on single byte locations. Flash erasures set bits back to logic 1, and occur only on full pages. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a flash write/erase operation.

The simplest means of programming the flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device.

To ensure the integrity of flash contents, it is strongly recommended that the on-chip supply monitor be enabled in any system that includes code that writes and/or erases flash memory from software.

10.2.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a flash write or erase is attempted before the key codes have been written properly. The flash lock resets after each write or erase; the key codes must be written again before a following flash operation can be performed.

10.2.2. Flash Erase Procedure

The flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to flash memory using MOVX, flash write operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit in the PSCTL register to logic 1 (this directs the MOVX writes to target flash memory); and (2) Writing the flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

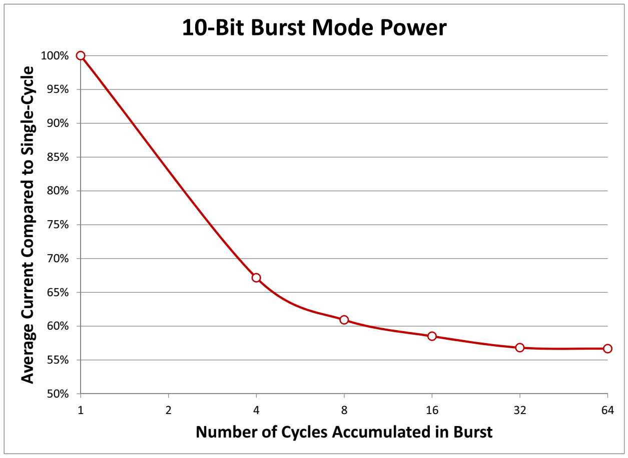
A write to flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in flash. **A byte location to be programmed should be erased before a new value is written.** Erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire page, perform the following steps:

1. Disable interrupts (recommended).
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.
6. Using the MOVX instruction, write a data byte to any location within the page to be erased.
7. Clear the PSWE and PSEE bits.

10.2.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the flash page containing the target location, as described in Section 10.2.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the desired



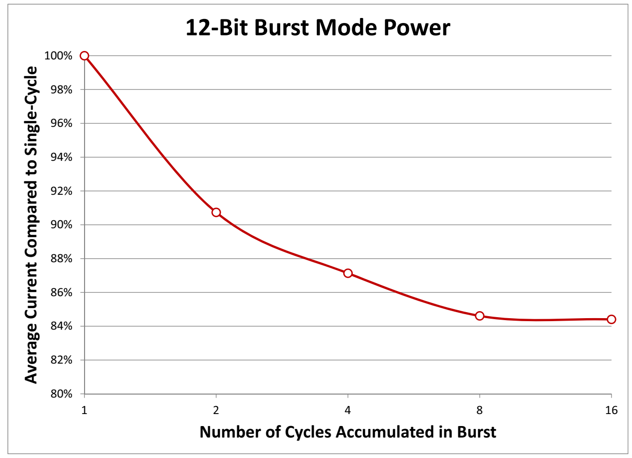


Figure 14.5. Burst Mode Accumulation Power Savings

14.6. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the ADSJST field. When the repeat count is set to 1 in 10-bit mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	**Right-Justified** **ADC0H:ADC0L (ADSJST = 000)**	**Left-Justified** **ADC0H:ADC0L (ADSJST = 100)**
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When the repeat count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, 16, 32, or 64 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the ADRPT bits in the ADC0AC register. When a repeat count is higher than 1, the ADC output must be right-justified (ADSJST = 0xx); unused bits in the ADC0H and ADC0L registers are set to 0. The example below shows the right-justified result for various input voltages and repeat counts. Notice that accumulating 2^n samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

Input Voltage	**Repeat Count = 4**	**Repeat Count = 16**	**Repeat Count = 64**
V_{REF} x 1023/1024	0x0FFC	0x3FF0	0xFFC0
V_{REF} x 512/1024	0x0800	0x2000	0x8000
V_{REF} x 511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

SILICON LABS

15.4. CPU Core Registers

Register 15.1. DPL: Data Pointer Low

Bit	7	6	5	4	3	2	1	0
Name	DPL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x82								

Table 15.2. DPL Register Bit Descriptions

Bit	Name	Function
7:0	DPL	**Data Pointer Low.** The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.

Register 15.5. B: B Register

Bit	7	6	5	4	3	2	1	0
Name	B							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xF0 (bit-addressable)								

Table 15.6. B Register Bit Descriptions

Bit	Name	Function
7:0	B	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

17.2. Functional Description

The comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the port pins: a synchronous “latched” output (CPn), or an asynchronous “raw” output (CPnA). The asynchronous CPnA signal is available even when the system clock is not active. This allows the comparator to operate and generate an output with the device in STOP mode.

When disabled, the comparator output (if assigned to a port I/O pin via the crossbar) defaults to the logic low state, and the power supply to the comparator is turned off.

The comparator response time may be configured in software via the CPTnMD register. Selecting a longer response time reduces the comparator supply current.

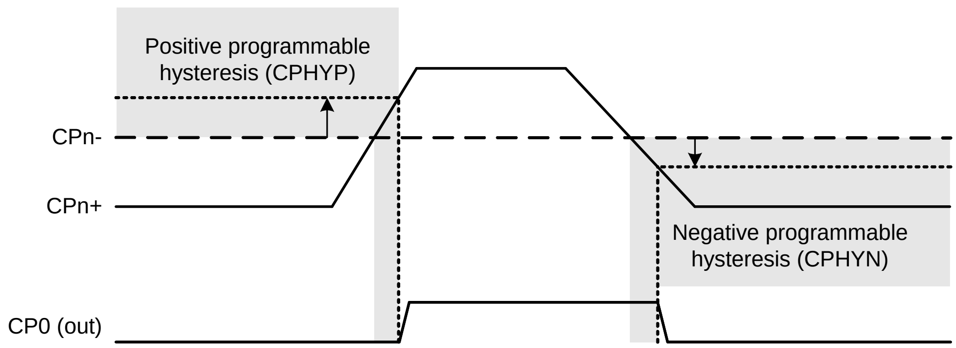


Figure 17.2. Comparator Hysteresis Plot

The comparator hysteresis is software-programmable via its Comparator Control register CPTnCN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The comparator hysteresis is programmable using the CPHYN and CPHYP fields in the Comparator Control Register CPTnCN. The amount of negative hysteresis voltage is determined by the settings of the CPHYN bits. As shown in Figure 17.2, settings of 20, 10, or 5 mV (nominal) of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CPHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. The CPFIF flag is set to logic 1 upon a comparator falling-edge occurrence, and the CPRIF flag is set to logic 1 upon the comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The comparator rising-edge interrupt mask is enabled by setting CPRIE to a logic 1. The comparator falling-edge interrupt mask is enabled by setting CPFIE to a logic 1.

The output state of the comparator can be obtained at any time by reading the CPOUT bit. The comparator is enabled by setting the CPEN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed, before enabling comparator interrupts.

17.3. Comparator Control Registers

Register 17.1. CPT0CN: Comparator 0 Control

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CPEN	CPOUT	CPRIF	CPFIF	CPHYP		CPHYN	
Type	RW	R	RW	RW	RW		RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x9B								

Table 17.5. CPT0CN Register Bit Descriptions

Bit	Name	Function
7	CPEN	**Comparator 0 Enable Bit.** 0: Comparator Disabled. 1: Comparator Enabled.
6	CPOUT	**Comparator 0 Output State Flag.** 0: Voltage on CP0P < CP0N. 1: Voltage on CP0P > CP0N.
5	CPRIF	**Comparator 0 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator Rising Edge has occurred since this flag was last cleared. 1: Comparator Rising Edge has occurred.
4	CPFIF	**Comparator 0 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator Falling-Edge has occurred since this flag was last cleared. 1: Comparator Falling-Edge has occurred.
3:2	CPHYP	**Comparator 0 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CPHYN	**Comparator 0 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

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 20.15. PCA0CPH1: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPH1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xEA								

Table 20.17. PCA0CPH1 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPH1	**PCA Capture Module High Byte.** The PCA0CPH1 register holds the high byte (MSB) of the 16-bit capture module.This register address also allows access to the high byte of the corresponding PCA channels auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the modules ECOM bit to a 1.		

Register 21.7. P0: Port 0 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	P0							
Type	RW							
Reset	1	1	1	1	1	1	1	1
SFR Address: 0x80 (bit-addressable)								

Table 21.10. P0 Register Bit Descriptions

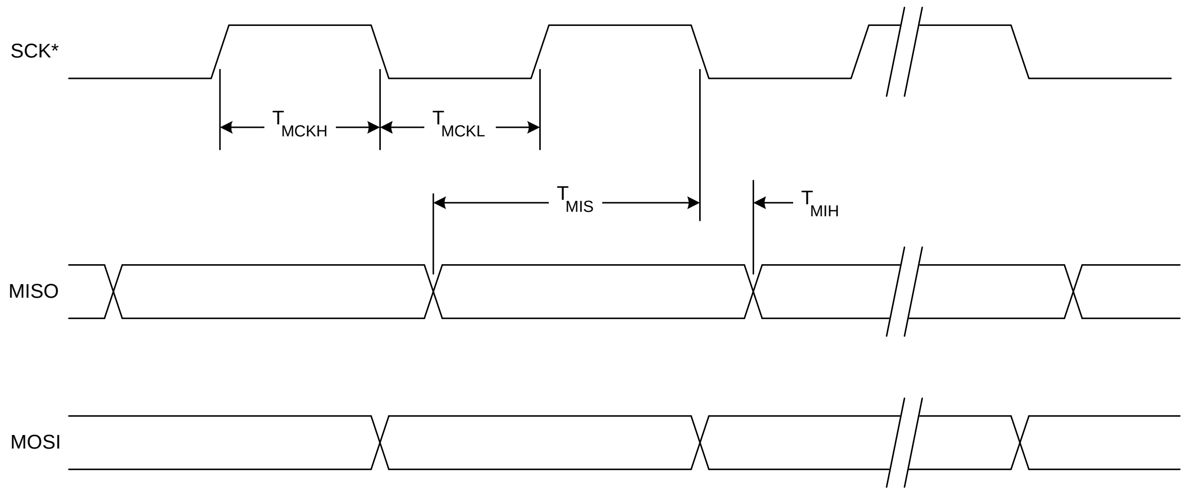
Bit	Name	Function
7:0	P0	**Port 0 Data.** Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 21.12. P1MAT: Port 1 Match

Bit	7	6	5	4	3	2	1	0
Name	P1MAT							
Type	RW							
Reset	1	1	1	1	1	1	1	1
SFR Address: 0xED								

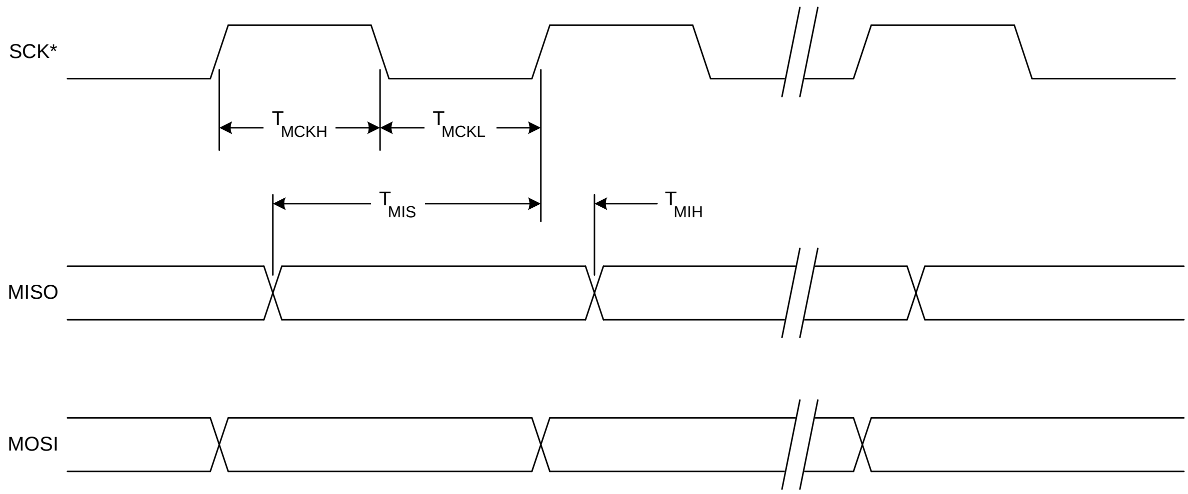
Table 21.15. P1MAT Register Bit Descriptions

Bit	Name	Function
7:0	P1MAT	**Port 1 Match Value.** Match comparison value used on P1 pins for bits in P1MASK which are set to 1. 0: P1.x pin logic value is compared with logic LOW. 1: P1.x pin logic value is compared with logic HIGH.
Note: Port 1 consists of 8 bits (P1.0-P1.7) on QSOP24 packages and 7 bits (P1.0-P1.6) on QFN20 packages and 4 bits (P1.0-P1.3) on SOIC16 packages.		



* 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.

24.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

24.2. SMBus Configuration

Figure 24.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. However, the maximum voltage on any port pin must conform to the electrical characteristics specifications. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

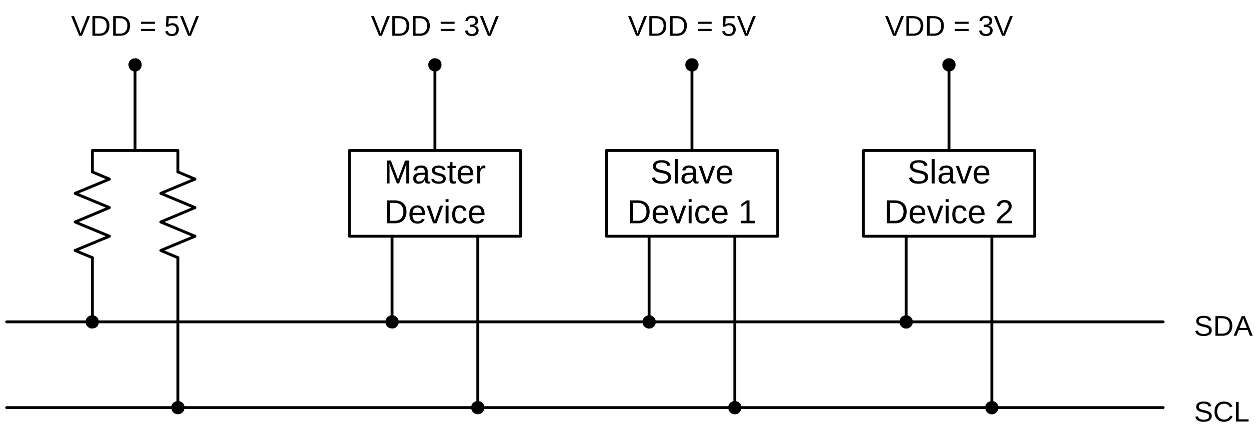


Figure 24.2. Typical SMBus Configuration

24.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. It is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 24.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary for SMBus compliance when SYSCLK is above 10 MHz.

Table 24.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} – 4 system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks
Note: Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgment, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI0 is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “24.3.4. SCL Low Timeout” on page 235). The SMBus interface will force the associated timer to reload while SCL is high, and allow the timer to count when SCL is low. The timer interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 24.4).

24.4.2. SMBus Pin Swap

The SMBus peripheral is assigned to pins using the priority crossbar decoder. By default, the SMBus signals are assigned to port pins starting with SDA on the lower-numbered pin, and SCL on the next available pin. The SWAP bit in the SMBTC register can be set to 1 to reverse the order in which the SMBus signals are assigned.

24.4.3. SMBus Timing Control

The SDD field in the SMBTC register is used to restrict the detection of a START condition under certain circumstances. In some systems where there is significant mismatch between the impedance or the capacitance on the SDA and SCL lines, it may be possible for SCL to fall after SDA during an address or data transfer. Such an event can cause a false START detection on the bus. These kind of events are not expected in a standard SMBus or I2C-compliant system. **In most systems this parameter should not be adjusted, and it is recommended that it be left at its default value.**

By default, if the SCL falling edge is detected after the falling edge of SDA (i.e. one SYSCLK cycle or more), the device will detect this as a START condition. The SDD field is used to increase the amount of hold time that is required between SDA and SCL falling before a START is recognized. An additional 2, 4, or 8 SYSCLKs can be added to prevent false START detection in systems where the bus conditions warrant this.

24.4.4. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information. The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

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.

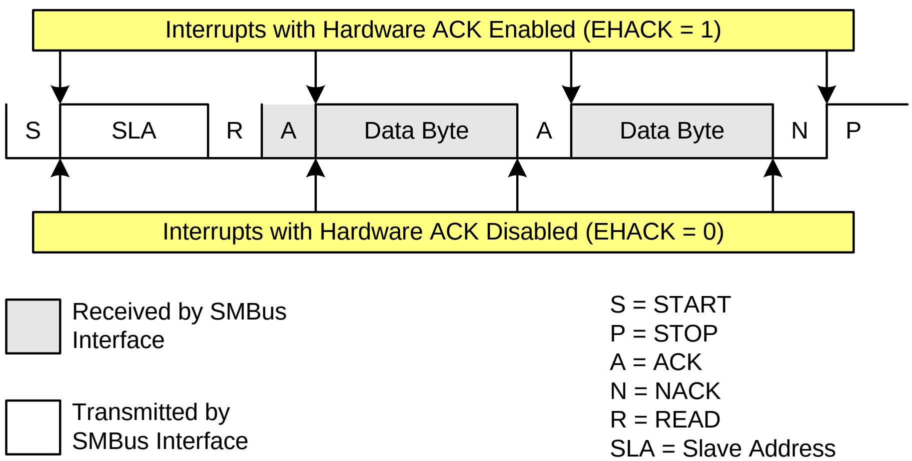


Figure 24.6. Typical Master Read Sequence

25. Timers (Timer0, Timer1, Timer2 and Timer3)

Each MCU in the C8051F85x/86x family includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timers for timing peripherals 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 and Timer 3 are also identical and offer both 16-bit and split 8-bit timer functionality with auto-reload capabilities. Timer 2 and Timer 3 both offer a capture function, but are different in their system-level connections. Timer 2 is capable of performing a capture function on an external signal input routed through the crossbar, while the Timer 3 capture is dedicated to the low-frequency oscillator output. Table 25.1 summarizes the modes available to each timer.

Table 25.1. Timer Modes

Timer 0 and Timer 1 Modes	Timer 2 Modes	Timer 3 Modes
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
8-bit counter/timer with auto-reload	Input pin capture	Low-frequency oscillator capture
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.

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 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 must be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

All four timers are capable of clocking other peripherals and triggering events in the system. The individual peripherals select which timer to use for their respective functions. Table 25.2 summarizes the peripheral connections for each timer. Note that the Timer 2 and Timer 3 high overflows apply to the full timer when operating in 16-bit mode or the high-byte timer when operating in 8-bit split mode.

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
UART0 Baud Rate		X				
SMBus0 Clock Rate	X	X	X	X		
SMBus0 SCL Low Timeout					X	
PCA0 Clock	X					

25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, CT0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

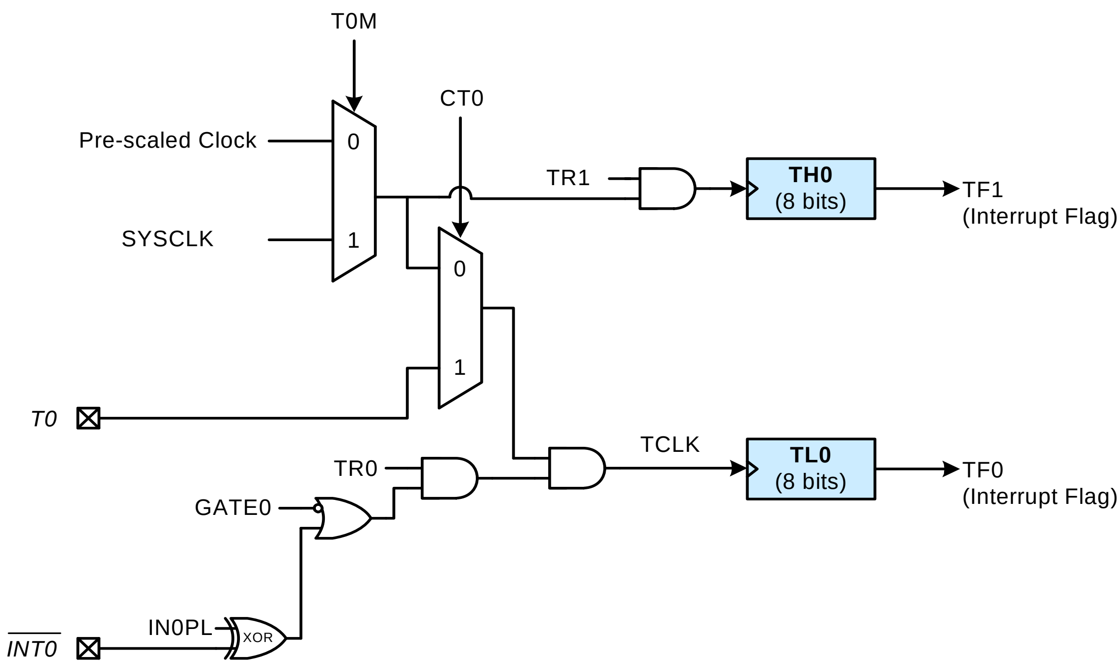


Figure 25.3. T0 Mode 3 Block Diagram

Register 25.3. TMOD: Timer 0/1 Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	CT1	T1M		GATE0	CT0	T0M	
Type	RW	RW	RW		RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x89								

Table 25.5. TMOD Register Bit Descriptions

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of INT1 logic level. 1: Timer 1 enabled only when TR1 = 1 and INT1 is active as defined by bit IN1PL in register IT01CF.
6	CT1	**Counter/Timer 1 Select.** 0: Timer Mode. Timer 1 increments on the clock defined by T1M in the CKCON register. 1: Counter Mode. Timer 1 increments on high-to-low transitions of an external pin (T1).
5:4	T1M	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of INT0 logic level. 1: Timer 0 enabled only when TR0 = 1 and INT0 is active as defined by bit IN0PL in register IT01CF.
2	CT0	**Counter/Timer 0 Select.** 0: Timer Mode. Timer 0 increments on the clock defined by T0M in the CKCON register. 1: Counter Mode. Timer 0 increments on high-to-low transitions of an external pin (T0).
1:0	T0M	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers