

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	Obsolete
Core Processor	CIP-51™
Core Size	8-Bit
Speed	25MHz
Connectivity	I ² C, SMBus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
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 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f850-b-gu

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

1.2. Typical Performance Curves

1.2.1. Operating Supply Current

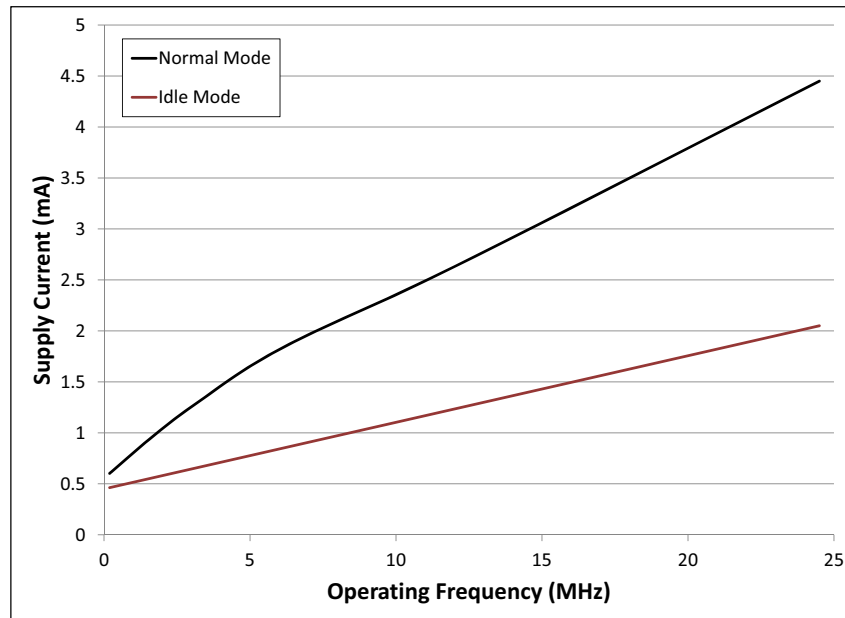


Figure 1.1. Typical Operating Current Running From 24.5 MHz Internal Oscillator

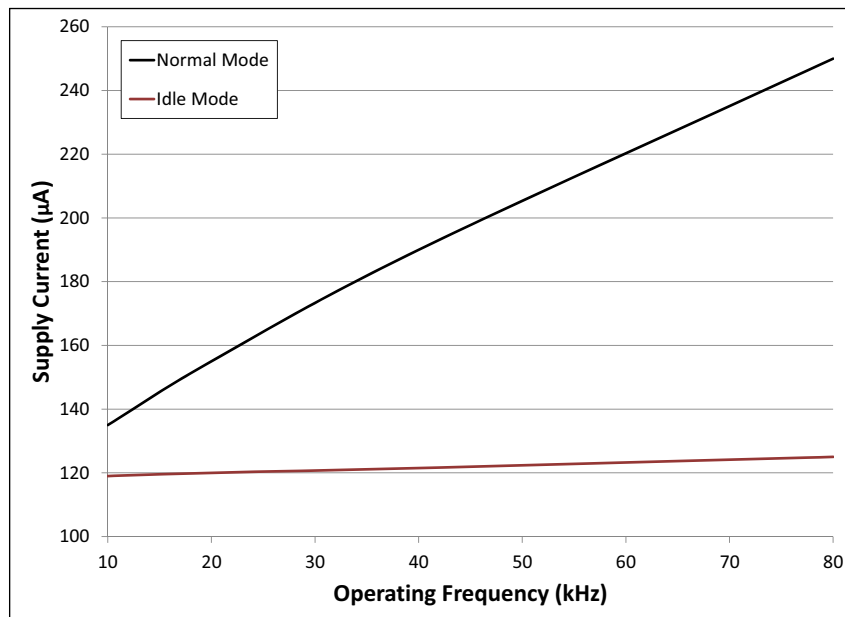


Figure 1.2. Typical Operating Current Running From 80 kHz Internal Oscillator

2.3. Clocking

The C8051F85x/86x devices have two internal oscillators and the option to use an external CMOS input at a pin as the system clock. A programmable divider allows the user to internally run the system clock at a slower rate than the selected oscillator if desired.

2.4. Counters/Timers and PWM

2.4.1. Programmable Counter Array (PCA0)

The C8051F85x/86x devices include a three-channel, 16-bit Programmable Counter Array with the following features:

- 16-bit time base.
- Programmable clock divisor and clock source selection.
- Three independently-configurable channels.
- 8, 9, 10, 11 and 16-bit PWM modes (center or edge-aligned operation).
- Output polarity control.
- Frequency output mode.
- Capture on rising, falling or any edge.
- Compare function for arbitrary waveform generation.
- Software timer (internal compare) mode.
- Can accept hardware “kill” signal from comparator 0.

2.4.2. Timers (Timer 0, Timer 1, Timer 2 and Timer 3)

Timers include the following features:

- Timer 0 and Timer 1 are standard 8051 timers, supporting backwards-compatibility with firmware and hardware.
- Timer 2 and Timer 3 can each operate as 16-bit auto-reload or two independent 8-bit auto-reload timers, and include pin or LFO clock capture capabilities.

2.4.3. Watchdog Timer (WDT0)

The watchdog timer includes a 16-bit timer with a programmable reset period. The registers are protected from inadvertent access by an independent lock and key interface.

The Watchdog Timer has the following features:

- Programmable timeout interval.
- Runs from the low frequency oscillator.
- Lock-out feature to prevent any modification until a system reset.

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

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
N/C	No Connection	1 13 24			

Table 3.2. Pin Definitions for C8051F850/1/2/3/4/5-GM and C8051F850/1/2/3/4/5-IM

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P0.0	Standard I/O	2	Yes	P0MAT.0 INT0.0 INT1.0	ADC0.0 CP0P.0 CP0N.0 VREF
P0.1	Standard I/O	1	Yes	P0MAT.1 INT0.1 INT1.1	ADC0.1 CP0P.1 CP0N.1 AGND
P0.2	Standard I/O	20	Yes	P0MAT.2 INT0.2 INT1.2	ADC0.2 CP0P.2 CP0N.2
P0.3	Standard I/O	19	Yes	P0MAT.3 EXTCLK INT0.3 INT1.3	ADC0.3 CP0P.3 CP0N.3
P0.4	Standard I/O	18	Yes	P0MAT.4 INT0.4 INT1.4	ADC0.4 CP0P.4 CP0N.4
P0.5	Standard I/O	17	Yes	P0MAT.5 INT0.5 INT1.5	ADC0.5 CP0P.5 CP0N.5
P0.6	Standard I/O	16	Yes	P0MAT.6 CNVSTR INT0.6 INT1.6	ADC0.6 CP0P.6 CP0N.6
P0.7	Standard I/O	15	Yes	P0MAT.7 INT0.7 INT1.7	ADC0.7 CP0P.7 CP0N.7

6. QFN-20 Package Specifications

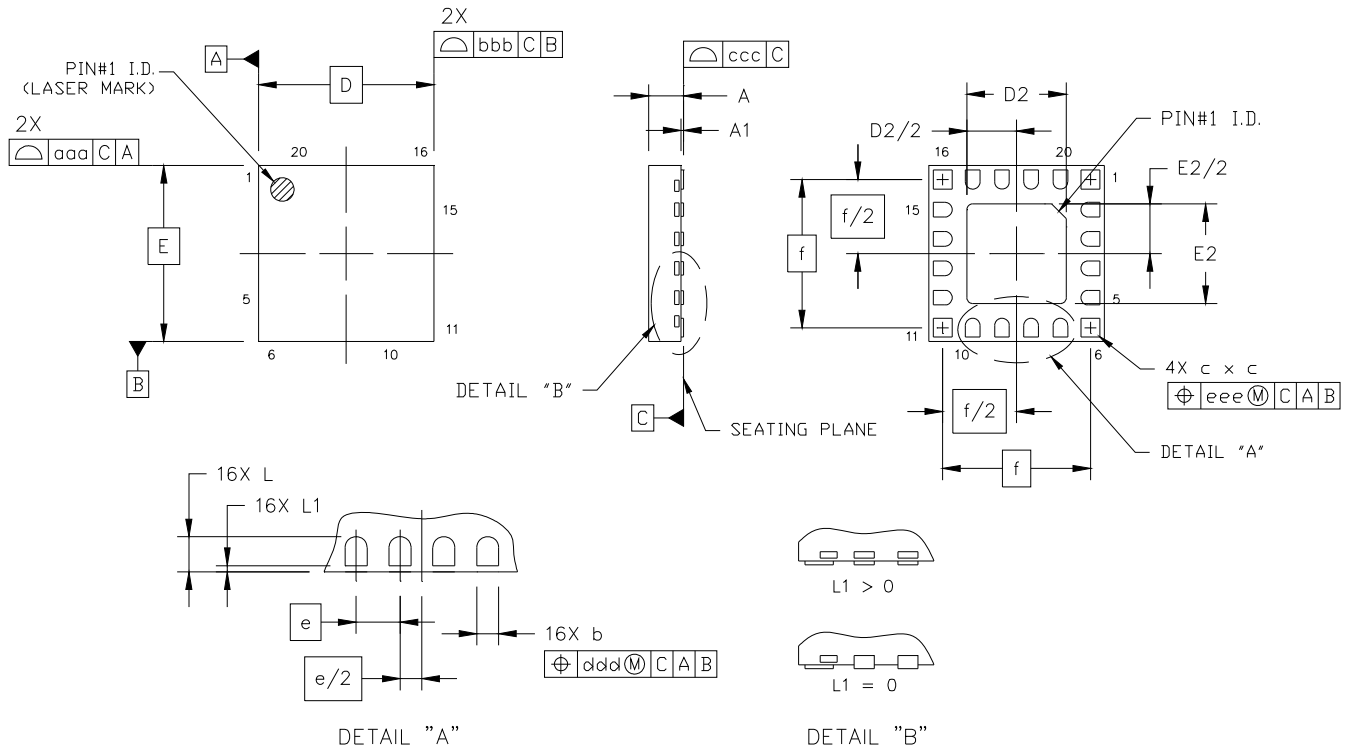


Figure 6.1. QFN-20 Package Drawing

Table 6.1. QFN-20 Package Dimensions

Symbol	Millimeters		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.20	0.25	0.30
c	0.25	0.30	0.35
D	3.00 BSC		
D2	1.6	1.70	1.8
e	0.50 BSC		
E	3.00 BSC		
E2	1.6	1.70	1.8
f	2.53 BSC		
L	0.3	0.40	0.5
L1	0.00	—	0.10
aaa	—	—	0.05
bbb	—	—	0.05
ccc	—	—	0.08
ddd	—	—	0.10
eee	—	—	0.10

Notes:

1. All dimensions are shown in millimeters unless otherwise noted.
2. Dimensioning and tolerancing per ANSI Y14.5M-1994.

8.2.3. Special Function Registers

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

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

10.5. Flash Control Registers

Register 10.1. PSCTL: Program Store Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved						PSEE	PSWE
Type	R						RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x8F								

Table 10.2. PSCTL Register Bit Descriptions

Bit	Name	Function
7:2	Reserved	Must write reset value.
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.

Table 14.3. ADC0 Optimal Power Configuration (12-bit Mode)

Required Throughput	Reference Source	Mode Configuration	SAR Clock Speed	Other Register Field Settings
180-200 ksps	Any	Always-On + Burst Mode (ADEN = 1 ADBMEN = 1)	12.25 MHz (ADSC = 1)	ADC0PWR = 0x40 ADC0TK = 0xBF ADRPT = 1
125-180 ksps	Any	Always-On + Burst Mode (ADEN = 1 ADBMEN = 1)	12.25 MHz (ADSC = 1)	ADC0PWR = 0x40 ADC0TK = 0x3A ADRPT = 1
0-125 ksps	External	Burst Mode (ADEN = 0 ADBMEN = 1)	12.25 MHz (ADSC = 1)	ADC0PWR = 0x44 ADC0TK = 0x3A ADRPT = 1
50-125 ksps	Internal	Burst Mode (ADEN = 0 ADBMEN = 1)	12.25 MHz (ADSC = 1)	ADC0PWR = 0x44 ADC0TK = 0x3A ADRPT = 1
0-50 ksps	Internal	Burst Mode (ADEN = 0 ADBMEN = 1)	4.08 MHz (ADSC = 5)	ADC0PWR = 0xF4 ADC0TK = 0x34 ADRPT = 1
Note: ADRPT reflects the minimum setting for this bit field. When using the ADC in Burst Mode, up to 64 samples may be auto-accumulated per conversion trigger by adjusting ADRPT.				

For applications where burst mode is used to automatically accumulate multiple results, additional supply current savings can be realized. The length of time the ADC is active during each burst contains power-up time at the beginning of the burst as well as the conversion time required for each conversion in the burst. The power-on time is only required at the beginning of each burst. When compared with single-sample bursts to collect the same number of conversions, multi-sample bursts will consume significantly less power. For example, performing an eight-cycle burst of 10-bit conversions consumes about 61% of the power required to perform those same eight samples in single-cycle bursts. For 12-bit conversions, an eight-cycle burst results in about 85% of the equivalent single-cycle bursts. Figure 14.5 shows this relationship for the different burst cycle lengths.

See the Electrical Characteristics chapter for details on power consumption and the maximum clock frequencies allowed in each mode.

16. Clock Sources and Selection (HFOSC0, LFOSC0, and EXTCLK)

The C8051F85x/86x devices can be clocked from the internal low power 24.5 MHz oscillator, the internal low-frequency 80 kHz oscillator, or an external CMOS clock signal at the EXTCLK pin. An adjustable clock divider allows the selected clock source to be post-scaled by powers of 2, up to a factor of 128. By default, the system clock comes up as the 24.5 MHz oscillator divided by 8.

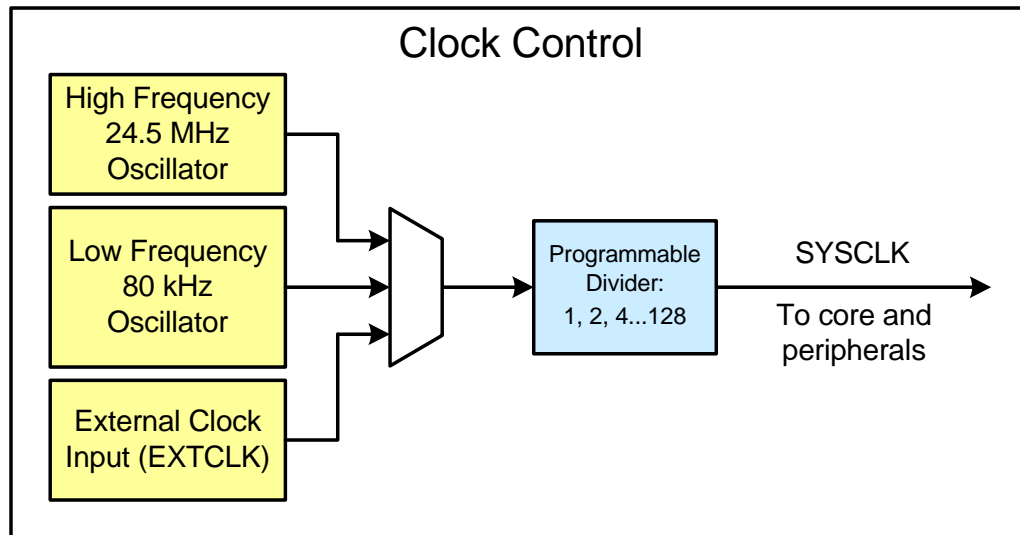


Figure 16.1. Clocking Options

16.1. Programmable High-Frequency Oscillator

All C8051F85x/86x devices include a programmable internal high-frequency oscillator that defaults as the system clock after a system reset. The oscillator is automatically enabled when it is requested. The internal oscillator period can be adjusted via the OSCICL register. On C8051F85x/86x devices, OSCICL is factory calibrated to obtain a 24.5 MHz base frequency.

16.2. Programmable Low-Frequency Oscillator

A programmable low-frequency internal oscillator is also included. The low-frequency oscillator is calibrated to a nominal frequency of 80 kHz. A divider at the oscillator output is capable of dividing the output clock of the module by 1, 2, 4, or 8, using the OSCLD bits in the OSCLCN register. Additionally, the OSCLF bits can be used to coarsely adjust the oscillator's output frequency.

16.2.1. Calibrating the Internal L-F Oscillator

Timer 3 includes a capture function that can be used to capture the oscillator frequency, when running from a known time base. When Timer 3 is configured for L-F Oscillator Capture Mode, a rising edge of the low-frequency oscillator's output will cause a capture event on the corresponding timer. As a capture event occurs, the current timer value (TMR3H:TMR3L) is copied into the timer reload registers (TMR3RLH:TMR3RLL). By recording the difference between two successive timer capture values, the low-frequency oscillator's period can be calculated. The OSCLF bits can then be adjusted to produce the desired oscillator frequency.

16.3. External Clock

An external CMOS clock source is also supported by the C8051F85x/86x family. The EXTCLK pin on the device serves as the external clock input when running in this mode. The EXTCLK input may also be used to clock some of the digital peripherals (e.g., Timers, PCA, etc.) while SYSCLK runs from one of the internal oscillator sources. When not selected as the SYSCLK source, the EXTCLK input is always re-synchronized to SYSCLK.

16.4. Clock Selection

The CLKSEL register is used to select the clock source for the system. The CLKSL field selects which oscillator source is used as the system clock, while CLKDIV controls the programmable divider. CLKSL must be set to 01b for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA) when the internal oscillator is selected as the system clock. In these cases, the external oscillator source is synchronized to the SYSCLK source. The system clock may be switched on-the-fly between any of the oscillator sources so long as the selected clock source is enabled and has settled, and CLKDIV may be changed at any time.

The internal high-frequency and low-frequency oscillators require little start-up time and may be selected as the system clock immediately following the register write which enables the oscillator. When selecting the EXTCLK pin as a clock input source, the pin should be skipped in the crossbar and configured as a digital input. Firmware should ensure that the external clock source is present or enable the missing clock detector before switching the CLKSL field.

20.5. Comparator Clear Function

In 8/9/10/11/16-bit PWM modes, the comparator clear function utilizes the Comparator0 output synchronized to the system clock to clear CEXn to logic low for the current PWM cycle. This comparator clear function can be enabled for each PWM channel by setting the CPCEn bits to 1 in the PCA0CLR SFR. When the comparator clear function is disabled, CEXn is unaffected.

The asynchronous Comparator 0 output is logic high when the voltage of CP0+ is greater than CP0- and logic low when the voltage of CP0+ is less than CP0-. The polarity of the Comparator 0 output is used to clear CEXn as follows: when CPCPOL = 0, CEXn is cleared on the falling edge of the Comparator0 output (see Figure 20.8); when CPCPOL = 1, CEXn is cleared on the rising edge of the Comparator0 output (see Figure 20.9).

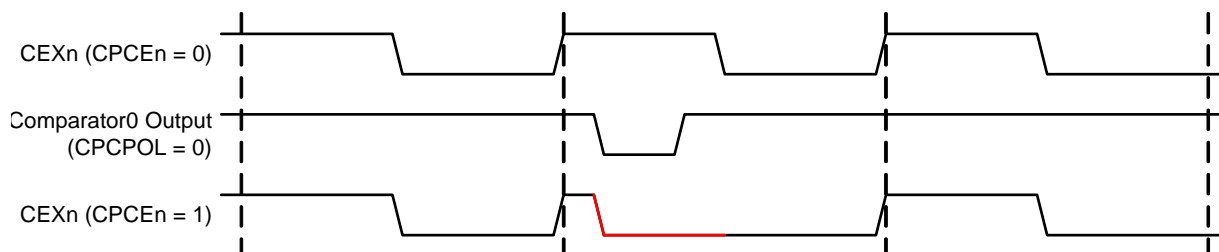


Figure 20.8. CEXn with CPCEn = 1, CPCPOL = 0

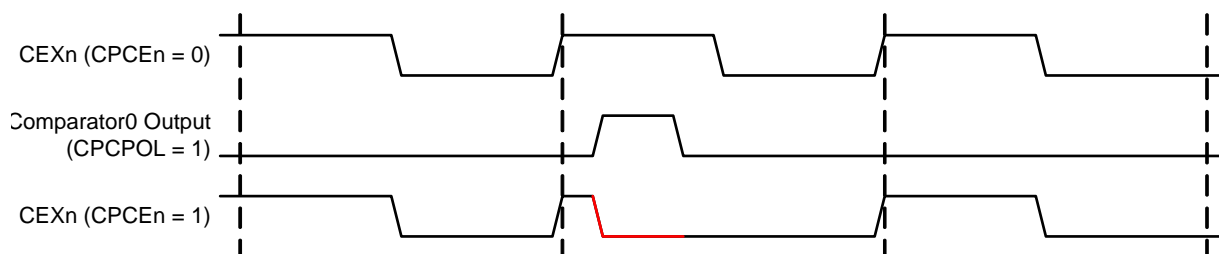


Figure 20.9. CEXn with CPCEn = 1, CPCPOL = 1

In the PWM cycle following the current cycle, should the Comparator 0 output remain logic low when CPCPOL = 0 or logic high when CPCPOL = 1, CEXn will continue to be cleared. See Figure 20.10 and Figure 20.11.

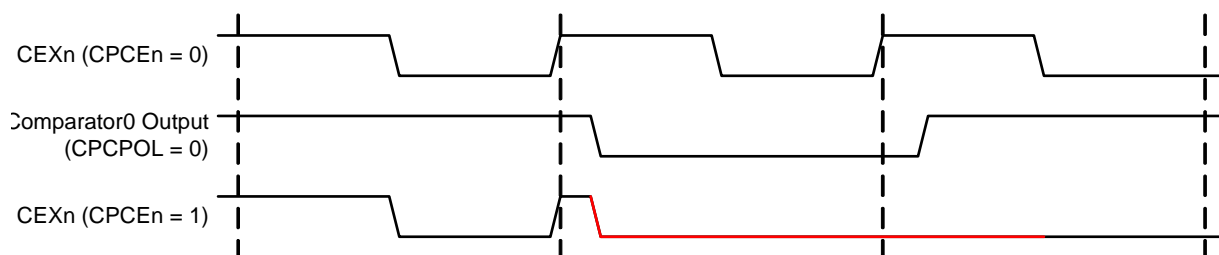


Figure 20.10. CEXn with CPCEn = 1, CPCPOL = 0

Register 20.9. PCA0CPH0: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPH0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xFC								

Table 20.11. PCA0CPH0 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPH0	PCA Capture Module High Byte. The PCA0CPH0 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 module's ECOM bit to a 1.		

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 23.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD		TXBMT	SPIEN
Type	RW	RW	RW	RW	RW		R	RW
Reset	0	0	0	0	0	1	1	0

SFR Address: 0xF8 (bit-addressable)

Table 23.3. SPI0CN Register Bit Descriptions

Bit	Name	Function
7	SPIF	SPI0 Interrupt Flag. This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	Write Collision Flag. This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	Mode Fault Flag. This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	Receive Overrun Flag (valid in slave mode only). This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD	Slave Select Mode. Selects between the following NSS operation modes: 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 10: 4-Wire Single-Master Mode. NSS is an output and logic low. 11: 4-Wire Single-Master Mode. NSS is an output and logic high.
1	TXBMT	Transmit Buffer Empty. This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.

Table 24.3. Sources for Hardware Changes to SMB0CN (Continued)

Bit	Set by Hardware When:	Cleared by Hardware When:
ACKRQ	<ul style="list-style-type: none"> ■ A byte has been received and an ACK response value is needed (only when hardware ACK is not enabled). 	<ul style="list-style-type: none"> ■ After each ACK cycle.
ARBLOST	<ul style="list-style-type: none"> ■ A repeated START is detected as a MASTER when STA is low (unwanted repeated START). ■ SCL is sensed low while attempting to generate a STOP or repeated START condition. ■ SDA is sensed low while transmitting a 1 (excluding ACK bits). 	<ul style="list-style-type: none"> ■ Each time SIn is cleared.
ACK	<ul style="list-style-type: none"> ■ The incoming ACK value is low (ACKNOWLEDGE). 	<ul style="list-style-type: none"> ■ The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	<ul style="list-style-type: none"> ■ A START has been generated. ■ Lost arbitration. ■ A byte has been transmitted and an ACK/ NACK received. ■ A byte has been received. ■ A START or repeated START followed by a slave address + R/W has been received. ■ A STOP has been received. 	<ul style="list-style-type: none"> ■ Must be cleared by software.

24.4.5. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 24.4.4.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register and the SMBus Slave Address Mask register. A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in a bit of the slave address mask SLVM enables a comparison between the received slave address and the hardware's slave address SLV for that bit. A 0 in a bit of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address. Additionally, if the GC bit in register SMB0ADR is set to 1, hardware will recognize the General Call Address (0x00). Table 24.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

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

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35

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

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).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	—
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

24.7. I2C / SMBus Control Registers

Register 24.1. SMB0CF: SMBus0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS	
Type	RW	RW	R	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xC1								

Table 24.7. SMB0CF Register Bit Descriptions

Bit	Name	Function
7	ENSMB	SMBus0 Enable. This bit enables the SMBus0 interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	SMBus0 Slave Inhibit. When this bit is set to logic 1, the SMBus0 does not generate an interrupt when slave events occur. This effectively removes the SMBus0 slave from the bus. Master Mode interrupts are not affected.
5	BUSY	SMBus0 Busy Indicator. This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	SMBus0 Setup and Hold Time Extension Enable. This bit controls the SDA setup and hold times. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	SMBus0 SCL Timeout Detection Enable. This bit enables SCL low timeout detection. If set to logic 1, the SMBus0 forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus0 communication.
2	SMBFTE	SMBus0 Free Timeout Detection Enable. When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.

26.2.2. 9-Bit UART

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

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

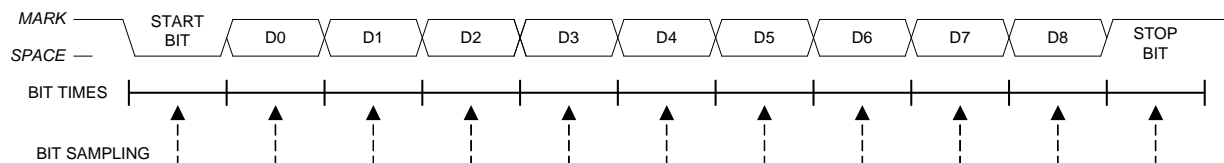


Figure 26.4. 9-Bit UART Timing Diagram

28. Revision-Specific Behavior

C8051F85x/86x Revision B devices have differences from Revision C devices:

- Temperature Sensor offset and slope
- Flash endurance
- Latch-up performance
- Unique Identifier

28.1. Revision Identification

The Lot ID Code on the top side of the device package can be used for decoding device revision information. Figure 28.1, Figure 28.2, and Figure 28.3 show how to find the Lot ID Code on the top side of the device package.

Firmware can distinguish between a Revision B and Revision C device using the value of the REVID register described in “Device Identification and Unique Identifier” on page 68.

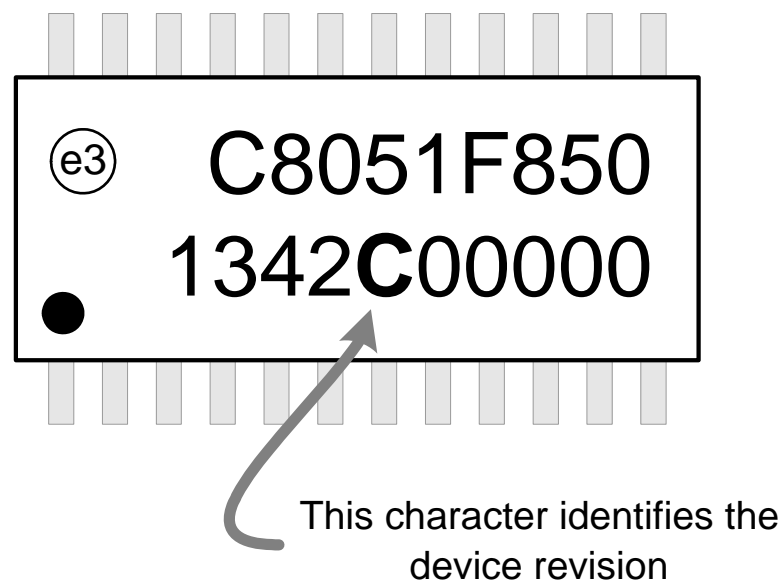


Figure 28.1. QSOP-24 Package Revision Marking