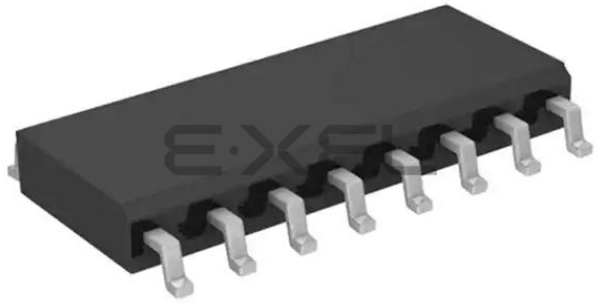

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	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	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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/c8051f863-c-is

Table of Contents

Table 1.7. ADC

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Resolution	N_{bits}	12 Bit Mode		12		Bits
		10 Bit Mode		10		Bits
Throughput Rate (High Speed Mode)	f_S	12 Bit Mode	—	—	200	ksps
		10 Bit Mode	—	—	800	ksps
Throughput Rate (Low Power Mode)	f_S	12 Bit Mode	—	—	62.5	ksps
		10 Bit Mode	—	—	250	ksps
Tracking Time	t_{TRK}	High Speed Mode	230	—	—	ns
		Low Power Mode	450	—	—	ns
Power-On Time	t_{PWR}		1.2	—	—	µs
SAR Clock Frequency	f_{SAR}	High Speed Mode, Reference is 2.4 V internal	—	—	6.25	MHz
		High Speed Mode, Reference is not 2.4 V internal	—	—	12.5	MHz
		Low Power Mode	—	—	4	MHz
Conversion Time	t_{CNV}	10-Bit Conversion, SAR Clock = 12.25 MHz, System Clock = 24.5 MHz.		1.1		µs
Sample/Hold Capacitor	C_{SAR}	Gain = 1	—	5	—	pF
		Gain = 0.5	—	2.5	—	pF
Input Pin Capacitance	C_{IN}		—	20	—	pF
Input Mux Impedance	R_{MUX}		—	550	—	Ω
Voltage Reference Range	V_{REF}		1	—	V_{DD}	V
Input Voltage Range*	V_{IN}	Gain = 1	0	—	V_{REF}	V
		Gain = 0.5	0	—	$2xV_{REF}$	V
Power Supply Rejection Ratio	$PSRR_{ADC}$		—	70	—	dB
DC Performance						
Integral Nonlinearity	INL	12 Bit Mode	—	±1	±2.3	LSB
		10 Bit Mode	—	±0.2	±0.6	LSB
Differential Nonlinearity (Guaranteed Monotonic)	DNL	12 Bit Mode	–1	±0.7	1.9	LSB
		10 Bit Mode	—	±0.2	±0.6	LSB
***Note:** Absolute input pin voltage is limited by the V_{DD} supply.						

2. System Overview

The C8051F85x/86x device family are fully integrated, mixed-signal system-on-a-chip MCUs. Highlighted features are listed below. Refer to Table 4.1 for specific product feature selection and part ordering numbers.

- **Core:**
  - Pipelined CIP-51 Core
  - Fully compatible with standard 8051 instruction set
  - 70% of instructions execute in 1-2 clock cycles
  - 25 MHz maximum operating frequency
- **Memory:**
  - 2-8 kB flash; in-system programmable in 512-byte sectors
  - 512 bytes RAM (including 256 bytes standard 8051 RAM and 256 bytes on-chip XRAM)
- **Power:**
  - Internal low drop-out (LDO) regulator for CPU core voltage
  - Power-on reset circuit and brownout detectors
- **I/O: Up to 18 total multifunction I/O pins:**
  - All pins 5 V tolerant under bias
  - Flexible peripheral crossbar for peripheral routing
  - 5 mA source, 12.5 mA sink allows direct drive of LEDs
- **Clock Sources:**
  - Low-power internal oscillator: 24.5 MHz ±2%
  - Low-frequency internal oscillator: 80 kHz
  - External CMOS clock option
- **Timers/Counters and PWM:**
  - 3-channel Programmable Counter Array (PCA) supporting PWM, capture/compare and frequency output modes
  - 4x 16-bit general-purpose timers
  - Independent watchdog timer, clocked from low frequency oscillator
- **Communications and Other Digital Peripherals:**
  - UART
  - SPI™
  - I^2C / SMBus™
  - 16-bit CRC Unit, supporting automatic CRC of flash at 256-byte boundaries
- **Analog:**
  - 12-Bit Analog-to-Digital Converter (ADC)
  - 2 x Low-Current Comparators
- **On-Chip Debugging**

With on-chip power-on reset, voltage supply monitor, watchdog timer, and clock oscillator, the C8051F85x/86x devices are truly standalone system-on-a-chip solutions. The flash memory is reprogrammable in-circuit, providing non-volatile data storage and allowing field upgrades of the firmware.

The on-chip debugging interface (C2) allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, and run and halt commands. All analog and digital peripherals are fully functional while debugging.

Each device is specified for 2.2 to 3.6 V operation, and are available in 20-pin QFN, 16-pin SOIC or 24-pin QSOP packages. All package options are lead-free and RoHS compliant. The device is available in two temperature grades: -40 to +85 °C or –40 to +125 °C. See Table 4.1 for ordering information. A block diagram is included in Figure 2.1.

11. Device Identification and Unique Identifier

The C8051F85x/86x has SFRs that identify the device family, derivative, and revision. These SFRs can be read by firmware at runtime to determine the capabilities of the MCU that is executing code. This allows the same firmware image to run on MCUs with different memory sizes and peripherals, and dynamically change functionality to suit the capabilities of that MCU.

In addition to the device identification registers, a 32-bit unique identifier (UID) is pre-programmed into all Revision C and later devices. The UID resides in the last four bytes of XRAM (C8051F850/1/3/4 and C8051F860/1/3/4) or RAM (C8051F852/5 and C8051F862/5). For devices with the UID in RAM, the UID can be read by firmware using indirect data accesses. For devices with the UID in XRAM, the UID can be read by firmware using MOVX instructions. The UID can also be read through the debug port for all devices.

Firmware can overwrite the UID during normal operation, and the bytes in memory will be automatically reinitialized with the UID value after any device reset. Firmware using this area of memory should always initialize the memory to a known value, as any previous data stored at these locations will be overwritten and not retained through a reset.

Table 11.1. UID Implementation Information

Device	Memory Segment	Addresses
C8051F850 C8051F851 C8051F853 C8051F854 C8051F860 C8051F861 C8051F863 C8051F864	XRAM	**(MSB)** 0x00FF, 0x00FE, 0x00FD, 0x00FC **(LSB)**
C8051F852 C8051F855 C8051F862 C8051F865	RAM (indirect)	**(MSB)** 0xFF, 0xFE, 0xFD, 0xFC **(LSB)**

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.

Register 14.5. ADC0PWR: ADC0 Power Control

Bit	7	6	5	4	3	2	1	0
Name	ADBIAS		ADMXLP	ADLPM	ADPWR			
Type	RW		RW	RW	RW			
Reset	0	0	0	0	1	1	1	1
SFR Address: 0xDF								

Table 14.8. ADC0PWR Register Bit Descriptions

Bit	Name	Function
7:6	ADBIAS	**Bias Power Select.** This field can be used to adjust the ADC's power consumption based on the conversion speed. Higher bias currents allow for faster conversion times. 00: Select bias current mode 0. Recommended to use modes 1, 2, or 3. 01: Select bias current mode 1 (SARCLK <= 16 MHz). 10: Select bias current mode 2. 11: Select bias current mode 3 (SARCLK <= 4 MHz).
5	ADMXLP	**Mux and Reference Low Power Mode Enable.** Enables low power mode operation for the multiplexer and voltage reference buffers. 0: Low power mode disabled. 1: Low power mode enabled (SAR clock < 4 MHz).
4	ADLPM	**Low Power Mode Enable.** This bit can be used to reduce power to the ADC's internal common mode buffer. It can be set to 1 to reduce power when tracking times in the application are longer (slower sample rates). 0: Disable low power mode. 1: Enable low power mode (requires extended tracking time).
3:0	ADPWR	**Burst Mode Power Up Time.** This field sets the time delay allowed for the ADC to power up from a low power state. When ADTM is set, an additional 4 SARCLKs are added to this time. $T_{PWRTIME} = \frac{8 \times ADPWR}{F_{HFOSC}}$

Register 14.14. REF0CN: Voltage Reference Control

Bit	7	6	5	4	3	2	1	0
Name	IREFLVL	Reserved	GNDSL	REFSL		TEMPE	Reserved	
Type	RW	R	RW	RW		RW	R	
Reset	0	0	0	1	1	0	0	0
SFR Address: 0xD1								

Table 14.17. REF0CN Register Bit Descriptions

Bit	Name	Function
7	IREFLVL	**Internal Voltage Reference Level.** Sets the voltage level for the internal reference source. 0: The internal reference operates at 1.65 V nominal. 1: The internal reference operates at 2.4 V nominal.
6	Reserved	Must write reset value.
5	GNDSL	**Analog Ground Reference.** Selects the ADC0 ground reference. 0: The ADC0 ground reference is the GND pin. 1: The ADC0 ground reference is the AGND pin.
4:3	REFSL	**Voltage Reference Select.** Selects the ADC0 voltage reference. 00: The ADC0 voltage reference is the VREF pin. 01: The ADC0 voltage reference is the VDD pin. 10: The ADC0 voltage reference is the internal 1.8 V digital supply voltage. 11: The ADC0 voltage reference is the internal voltage reference.
2	TEMPE	**Temperature Sensor Enable.** Enables/Disables the internal temperature sensor. 0: Temperature Sensor Disabled. 1: Temperature Sensor Enabled.
1:0	Reserved	Must write reset value.

18.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRCVAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRCINIT).

18.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks read from flash. The following steps can be used to automatically perform a CRC on flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit to 1 in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRCCNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes. See the note in the CRC0CN register definition for more information on how to properly initiate a CRC calculation.
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result.

18.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRCPNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

18.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 18.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

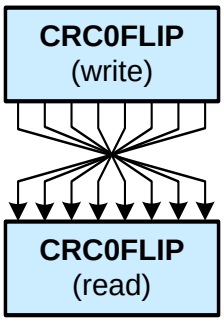

Figure 18.2. Bit Reversal

Register 18.4. CRC0AUTO: CRC0 Automatic Control

Bit	7	6	5	4	3	2	1	0
Name	AUTOEN	Reserved	CRCST					
Type	RW	R	RW					
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD2								

Table 18.5. CRC0AUTO Register Bit Descriptions

Bit	Name	Function
7	AUTOEN	**Automatic CRC Calculation Enable.** When AUTOEN is set to 1, any write to CRC0CN will initiate an automatic CRC starting at flash sector CRCST and continuing for CRCCNT sectors.
6	Reserved	Must write reset value.
5:0	CRCST	**Automatic CRC Calculation Starting Block.** These bits specify the flash block to start the automatic CRC calculation. The starting address of the first flash block included in the automatic CRC calculation is CRCST x block_size, where block_size is 256 bytes.

19.1. External Interrupt Control Registers

Register 19.1. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL			IN0PL	IN0SL		
Type	RW	RW			RW	RW		
Reset	0	0	0	0	0	0	0	1
SFR Address: 0xE4								

Table 19.1. IT01CF Register Bit Descriptions

Bit	Name	Function
7	IN1PL	**INT1 Polarity.** 0: INT1 input is active low. 1: INT1 input is active high.
6:4	IN1SL	**INT1 Port Pin Selection Bits.** These bits select which Port pin is assigned to INT1. This pin assignment is independent of the Crossbar; INT1 will monitor the assigned port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**INT0 Polarity.** 0: INT0 input is active low. 1: INT0 input is active high.

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).

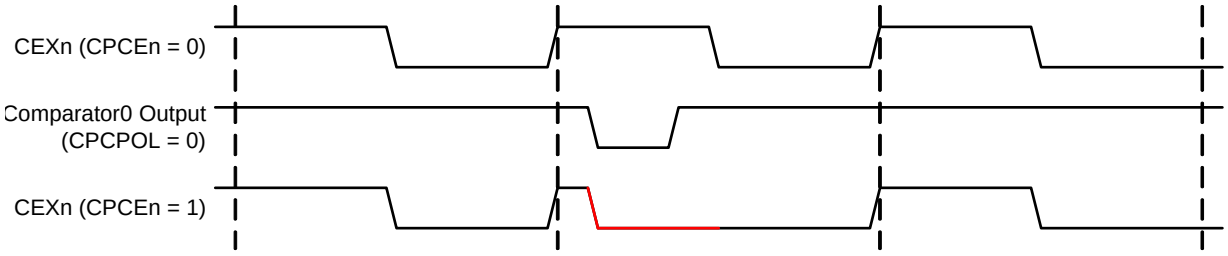

Figure 20.8. CEXn with CPCEn = 1, CPCPOL = 0

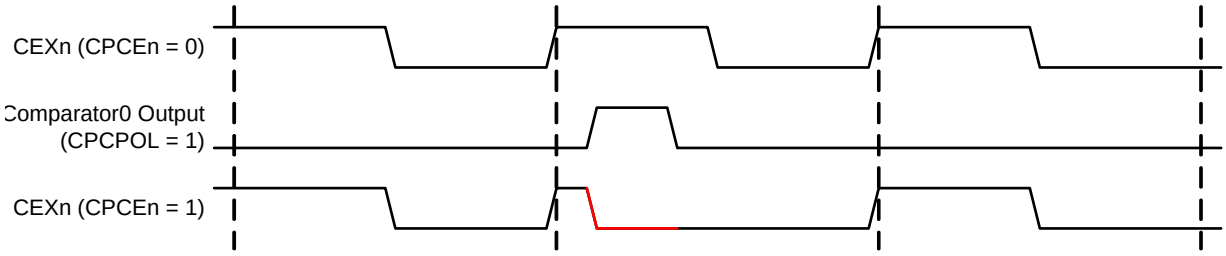

Figure 20.9. CEXn with CPCEn = 1, CPCPOL = 1

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.

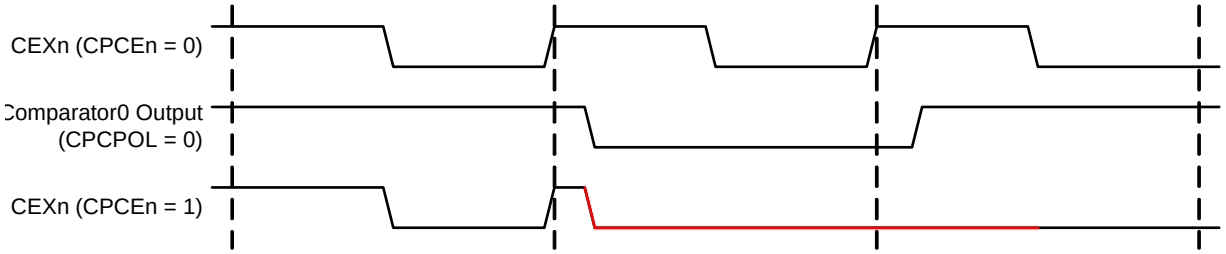

Figure 20.10. CEXn with CPCEn = 1, CPCPOL = 0

Register 20.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF	Reserved		CLSEL		
Type	RW	RW	RW	R		RW		
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xF7								

Table 20.5. PCA0PWM Register Bit Descriptions

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9 to 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th to 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:3	Reserved	Must write reset value.
2:0	CLSEL	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode. 000: 8 bits. 001: 9 bits. 010: 10 bits. 011: 11 bits. 100-111: Reserved.

Register 20.4. PCA0CLR: PCA Comparator Clear Control

Bit	7	6	5	4	3	2	1	0
Name	CPCPOL	Reserved				CPCE2	CPCE1	CPCE0
Type	RW	R				RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x9C								

Table 20.6. PCA0CLR Register Bit Descriptions

Bit	Name	Function
7	CPCPOL	**Comparator Clear Polarity.** Selects the polarity of the comparator result that will clear the PCA channel(s). 0: PCA channel(s) will be cleared when comparator result goes logic low. 1: PCA channel(s) will be cleared when comparator result goes logic high.
6:3	Reserved	Must write reset value.
2	CPCE2	**Comparator Clear Enable for CEX2.** Enables the comparator clear function on PCA channel 2.
1	CPCE1	**Comparator Clear Enable for CEX1.** Enables the comparator clear function on PCA channel 1.
0	CPCE0	**Comparator Clear Enable for CEX0.** Enables the comparator clear function on PCA channel 0.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 24.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

24.4.4.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

24.4.4.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 24.4.5. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 24.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 24.5 for SMBus status decoding using the SMB0CN register.

Table 24.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	■ A START is generated.	■ A STOP is generated. ■ Arbitration is lost.
TXMODE	■ START is generated. ■ SMB0DAT is written before the start of an SMBus frame.	■ A START is detected. ■ Arbitration is lost. ■ SMB0DAT is not written before the start of an SMBus frame.
STA	■ A START followed by an address byte is received.	■ Must be cleared by software.
STO	■ A STOP is detected while addressed as a slave. ■ Arbitration is lost due to a detected STOP.	■ A pending STOP is generated.

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	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
		1	1	X	Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—

25.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are each implemented as a16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register. Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently for the operating modes described below.

25.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal $\overline{INT0}$ is active as defined by bit IN0PL in register IT01CF.

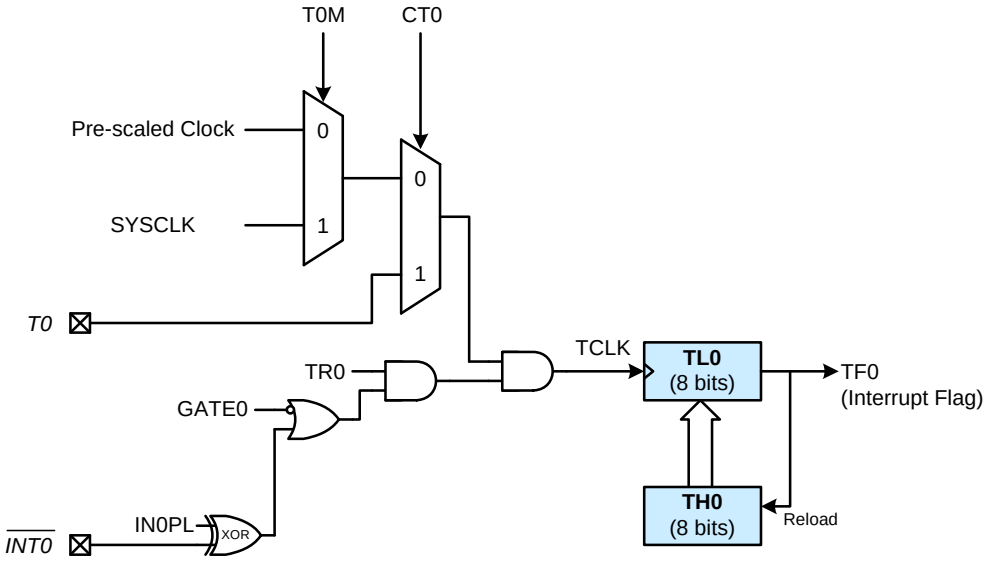

Figure 25.2. T0 Mode 2 Block Diagram

Table 25.3. CKCON Register Bit Descriptions

Bit	Name	Function
1:0	SCA	**Timer 0/1 Prescale Bits.** These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)

Register 25.13. TMR3CN: Timer 3 Control

Bit	7	6	5	4	3	2	1	0
Name	TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	Reserved	T3XCLK
Type	RW	RW	RW	RW	RW	RW	R	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x91								

Table 25.15. TMR3CN Register Bit Descriptions

Bit	Name	Function
7	TF3H	**Timer 3 High Byte Overflow Flag.** Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16-bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF3L	**Timer 3 Low Byte Overflow Flag.** Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. TF3L will be set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.
5	TF3LEN	**Timer 3 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 3 Low Byte interrupts. If Timer 3 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
4	TF3CEN	**Timer 3 Capture Enable.** When set to 1, this bit enables Timer 3 Capture Mode. If TF3CEN is set and Timer 3 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR3H:TMR3L will be copied to TMR3RLH:TMR3RLL.
3	T3SPLIT	**Timer 3 Split Mode Enable.** When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.
2	TR3	**Timer 3 Run Control.** Timer 3 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in split mode.
1	Reserved	Must write reset value.

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.

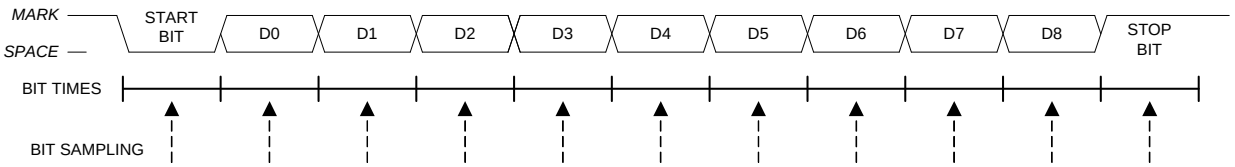

Figure 26.4. 9-Bit UART Timing Diagram