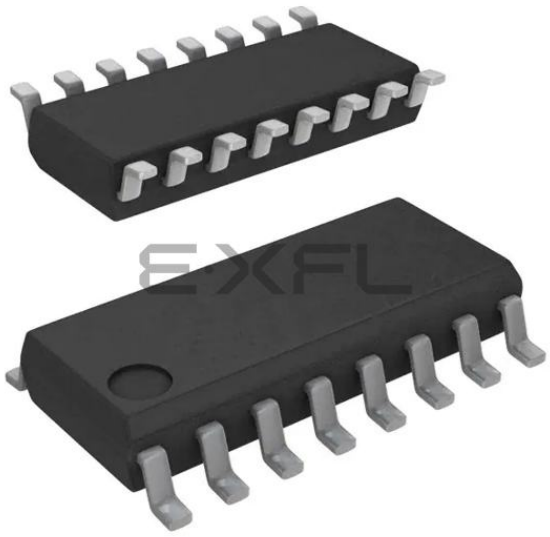

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Cap Sense, 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)	1.8V ~ 3.6V
Data Converters	-
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	16-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f822-gs

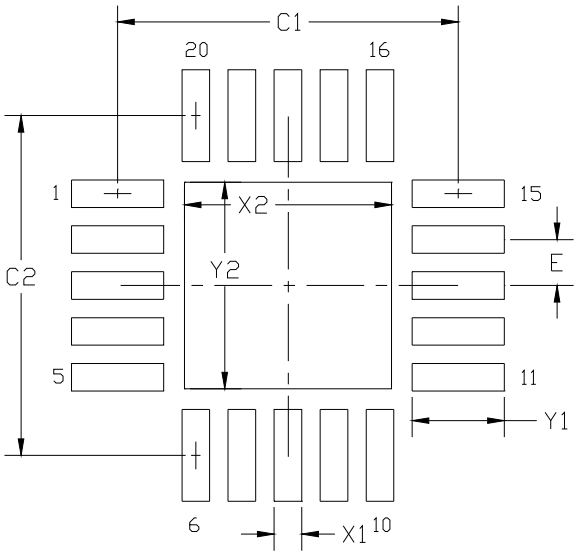

Figure 4.2. QFN-20 Recommended PCB Land Pattern

Table 4.2. QFN-20 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	3.70	
C2	3.70	
E	0.50	
X1	0.20	0.30

Dimension	Min	Max
X2	2.15	2.25
Y1	0.90	1.00
Y2	2.15	2.25

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

4. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

Stencil Design

5. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
6. The stencil thickness should be 0.125 mm (5 mils).
7. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pins.
8. A 2x2 array of 0.95 mm openings on a 1.1 mm pitch should be used for the center pad to assure the proper paste volume.

Card Assembly

9. A No-Clean, Type-3 solder paste is recommended.
10. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Table 7.9. ADC0 Electrical Characteristics

V_{DD} = 3.0 V, VREF = 2.40 V (REFSL=0), –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			10		bits
Integral Nonlinearity		—	±0.5	±1	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Offset Error		–2	0	2	LSB
Full Scale Error		–2	0	2	LSB
Offset Temperature Coefficient		—	45	—	ppm/°C
Dynamic performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		54	60	—	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	75	—	dB
Spurious-Free Dynamic Range		—	–90	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	8.33	MHz
Conversion Time in SAR Clocks	10-bit Mode 8-bit Mode	13 11	— —	— —	clocks clocks
Track/Hold Acquisition Time	V_{DD} >= 2.0 V V_{DD} < 2.0 V	300 2.0	— —	— —	ns µs
Throughput Rate		—	—	500	ksps
Analog Inputs					
ADC Input Voltage Range		0	—	VREF	V
Sampling Capacitance	1x Gain 0.5x Gain	— —	5 3	— —	pF pF
Input Multiplexer Impedance		—	5	—	kΩ
Power Specifications					
Power Supply Current	Operating Mode, 500 ksps	—	630	1000	µA
Power Supply Rejection		—	–70	—	dB

12. Comparator0

C8051F80x-83x devices include an on-chip programmable voltage comparator, Comparator0, shown in Figure 12.1.

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 (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A 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 assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see Section “23.4. Port I/O Initialization” on page 147). Comparator0 may also be used as a reset source (see Section “21.5. Comparator0 Reset” on page 127).

The Comparator0 inputs are selected by the comparator input multiplexer, as detailed in Section “12.1. Comparator Multiplexer” on page 69.

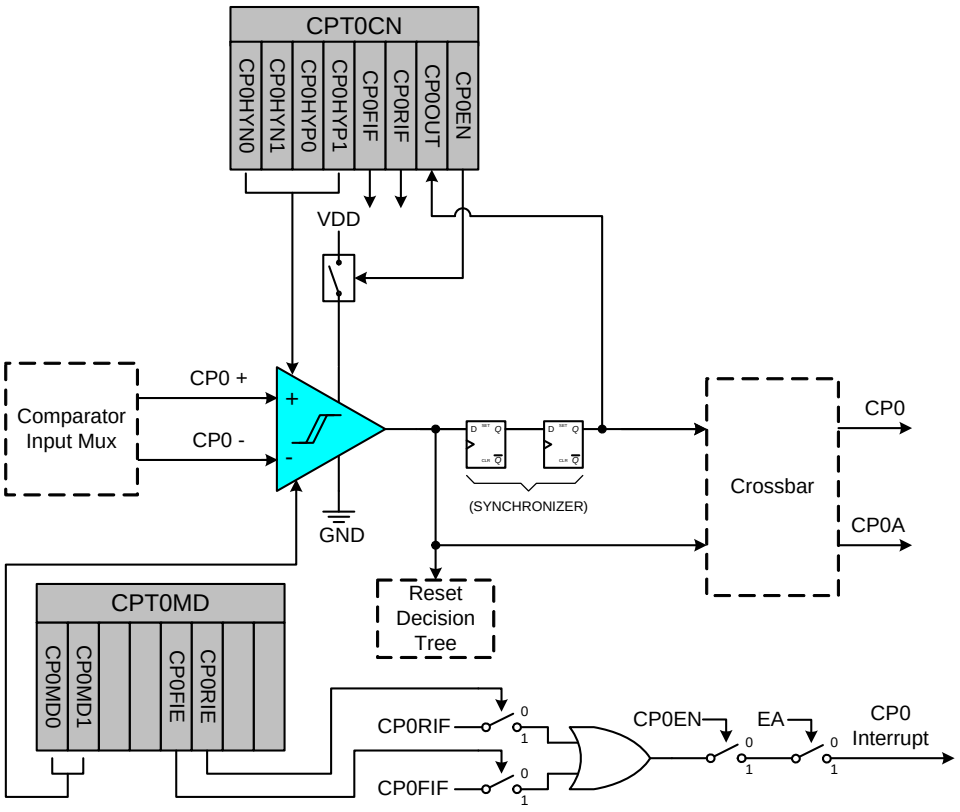

Figure 12.1. Comparator0 Functional Block Diagram

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). 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. See Section “23.3. Priority Crossbar Decoder” on page 143 for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Section “7. Electrical Characteristics” on page 39.

SFR Definition 12.2. CPT0MD: Comparator0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP0RIE	CP0FIE			CP0MD[1:0]	
Type	R	R	R/W	R/W	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9D

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

13. Capacitive Sense (CS0)

The Capacitive Sense subsystem included on the C8051F800/1/3/4/6/7/9, C8051F810/2/3/5/6/8/9, C8051F821/2/4/5/7/8, C8051F830/1/3/4 uses a capacitance-to-digital circuit to determine the capacitance on a port pin. The module can take measurements from different port pins using the module's analog multiplexer. The multiplexer supports up to 16 channels. See SFR Definition 13.9. "CS0MX: Capacitive Sense Mux Channel Select" on page 81 for channel availability for specific part numbers. The module is enabled only when the CS0EN bit (CS0CN) is set to 1. Otherwise the module is in a low-power shutdown state. The module can be configured to take measurements on one port pin or a group of port pins, using auto-scan. An accumulator can be configured to accumulate multiple conversions on an input channel. Interrupts can be generated when CS0 completes a conversion or when the measured value crosses a threshold defined in CS0THH:L.

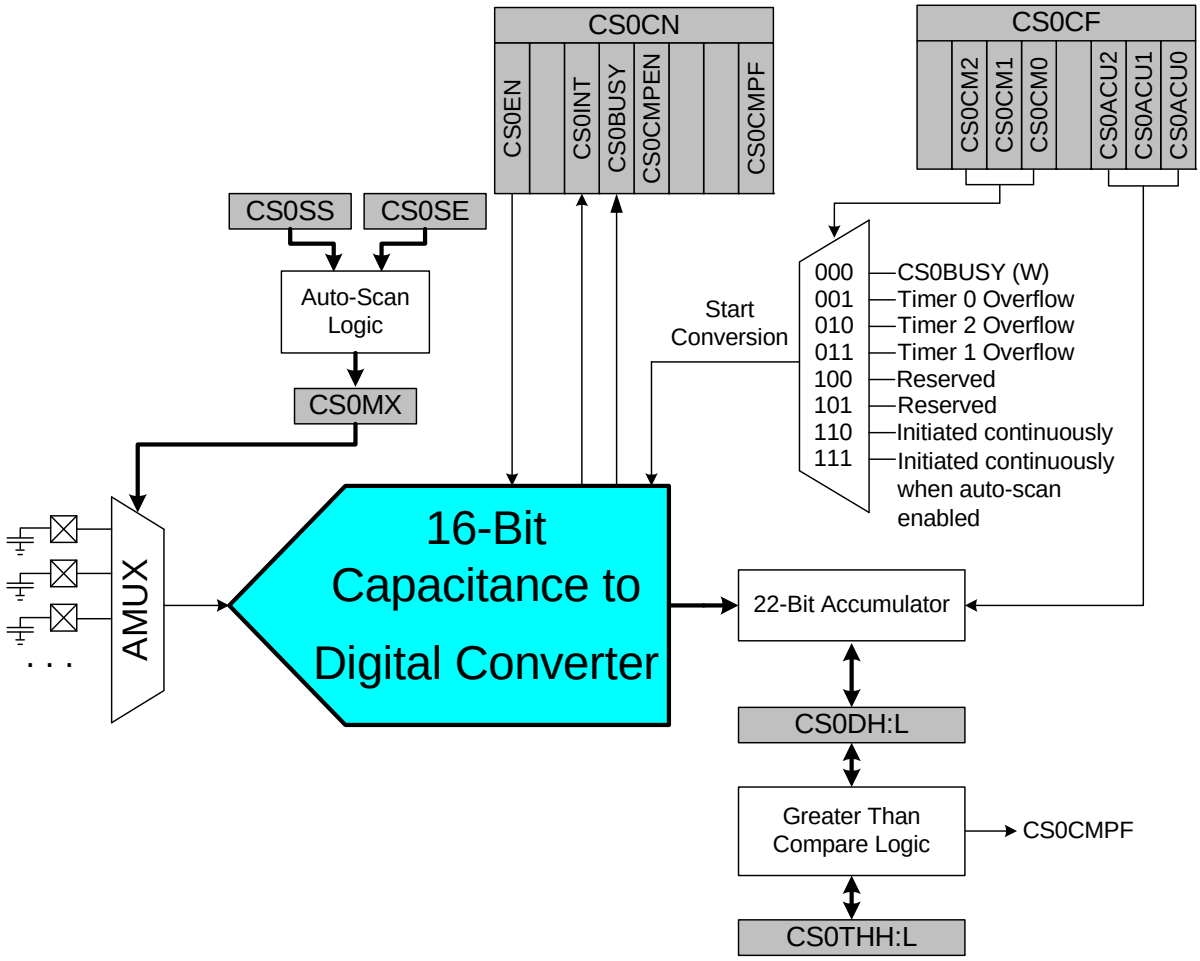

Figure 13.1. CS0 Block Diagram

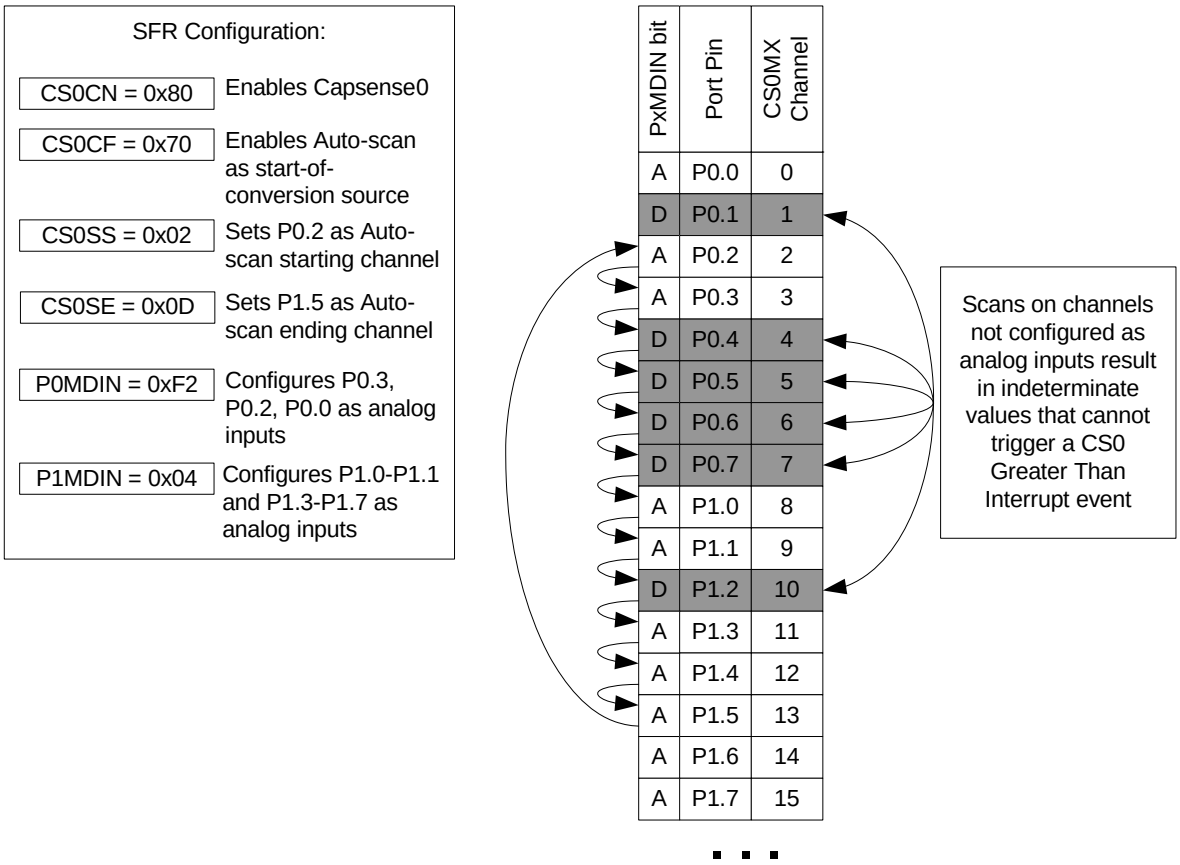

Figure 13.2. Auto-Scan Example

13.4. CS0 Comparator

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

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

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

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

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

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

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

SFR Definition 18.6. EIP2: Extended Interrupt Priority 2

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

SFR Address = 0xF4

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

20. Power Management Modes

The C8051F80x-83x devices have three software programmable power management modes: Idle, Stop, and Suspend. Idle mode and Stop mode are part of the standard 8051 architecture, while Suspend mode is an enhanced power-saving mode implemented by the high-speed oscillator peripheral.

Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Suspend mode is similar to Stop mode in that the internal oscillator and CPU are halted, but the device can wake on events such as a Port Mismatch, Comparator low output, or a Timer 3 overflow. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode and Suspend mode consume the least power because the majority of the device is shut down with no clocks active. SFR Definition 20.1 describes the Power Control Register (PCON) used to control the C8051F80x-83x's Stop and Idle power management modes. Suspend mode is controlled by the SUSPEND bit in the OSCICN register (SFR Definition 22.3).

Although the C8051F80x-83x has Idle, Stop, and Suspend modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off oscillators lowers power consumption considerably, at the expense of reduced functionality.

20.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) 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 Mode Selection bit (PCON.0) 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.

Note: If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in 'C':
PCON |= 0x01;                  // set IDLE bit
PCON = PCON;                   // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h                 ; set IDLE bit
MOV PCON, PCON                 ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section "29.4. Watchdog Timer Mode" on page 236 for more information on the use and configuration of the WDT.

24.2. 32-bit CRC Algorithm

The C8051F80x-83x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
2. Right-shift the CRC result.
3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit C8051F80x-83x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input){
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ CRC_input;
   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x00000001) == 0x00000001)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc >> 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc >> 1;
      }
   }
   return CRC_acc; // Return the final remainder (CRC value)
}
```

Table 24.2 lists example input values and the associated outputs using the 32-bit C8051F80x-83x CRC algorithm (an initial value of 0xFFFFFFFF is used):

Table 24.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

25. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

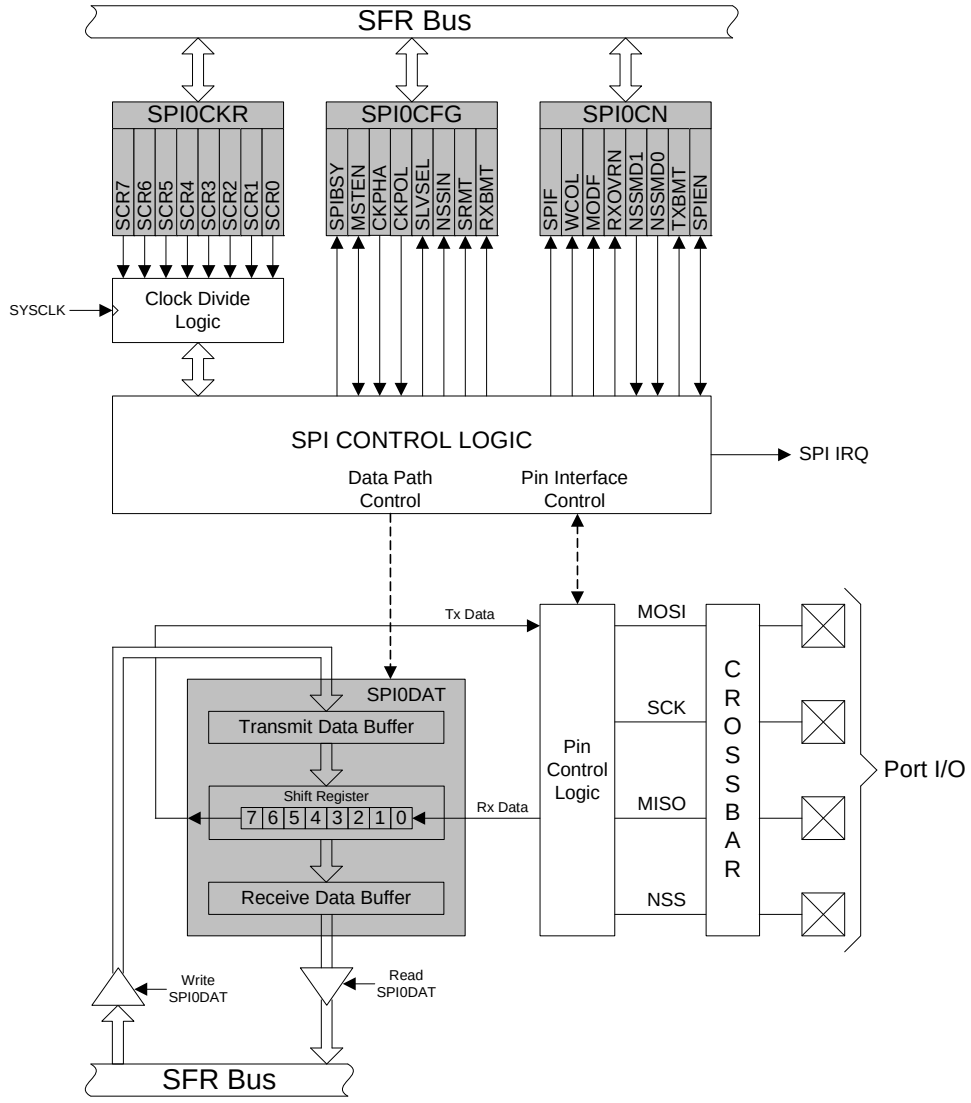

Figure 25.1. SPI Block Diagram

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 26.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

Table 26.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV[6:0]	Slave Address Mask SLVM[6:0]	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

27. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “27.1. Enhanced Baud Rate Generation” on page 202). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

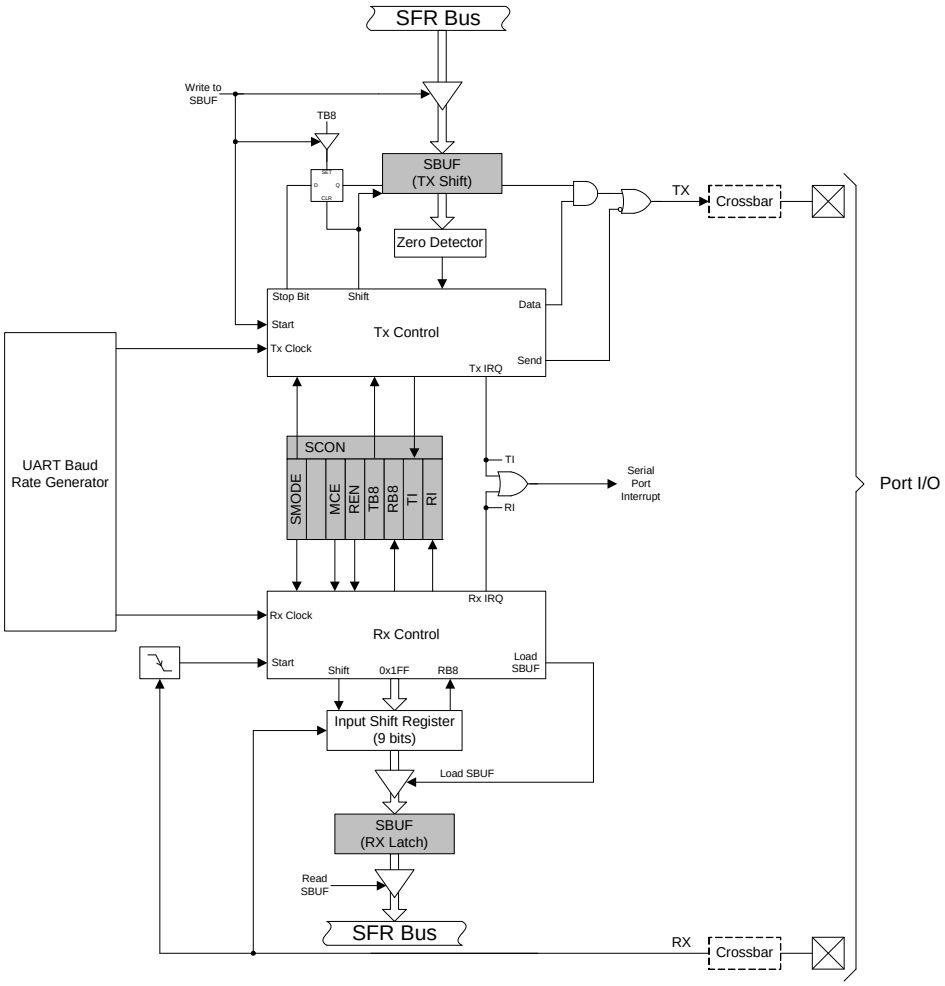

Figure 27.1. UART0 Block Diagram

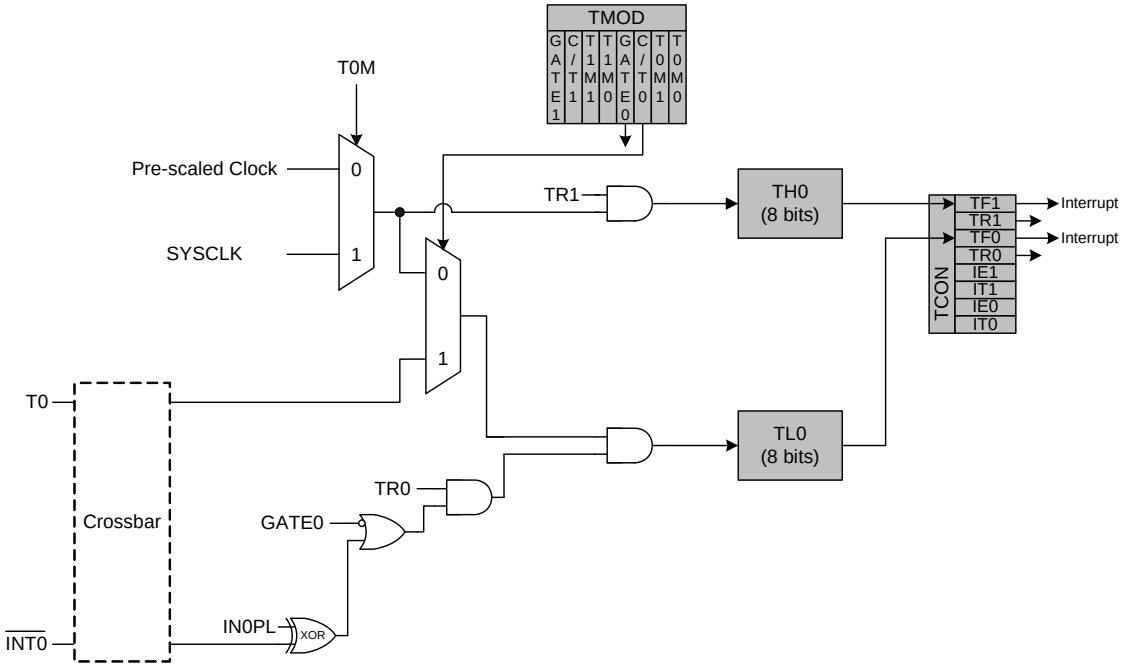

Figure 28.3. T0 Mode 3 Block Diagram

SFR Definition 28.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 18.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**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 $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 18.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**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

SFR Definition 28.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 28.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

29. programmable Counter Array

The programmable counter array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and three 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled. The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflows, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8 to 15-Bit PWM, or 16-Bit PWM (each mode is described in Section “29.3. Capture/Compare Modules” on page 228). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 29.1

Important Note: The PCA Module 2 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See Section 29.4 for details.

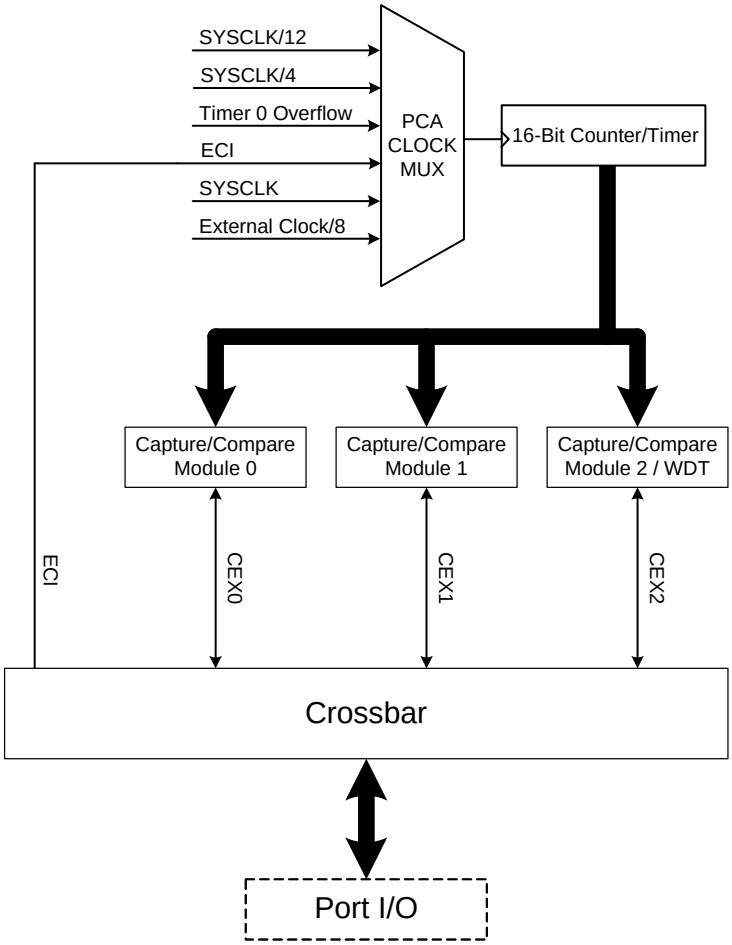

Figure 29.1. PCA Block Diagram

C2 Register Definition 30.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**C2 Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 30.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xBF

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase