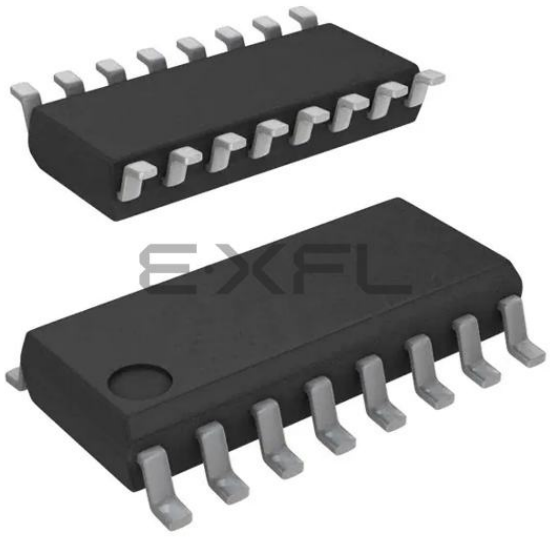

Welcome to E-XFL.COM

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	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/c8051f823-gsr

Table of Contents

1. System Overview 15
2. Ordering Information 25
3. Pin Definitions 28
4. QFN-20 Package Specifications 33
5. QSOP-24 Package Specifications 35
6. SOIC-16 Package Specifications 37
7. Electrical Characteristics 39
7.1. Absolute Maximum Specifications 39
7.2. Electrical Characteristics 40
8. 10-Bit ADC (ADC0) 46
8.1. Output Code Formatting 47
8.2. 8-Bit Mode 47
8.3. Modes of Operation 47
8.3.1. Starting a Conversion 47
8.3.2. Tracking Modes 48
8.3.3. Settling Time Requirements 49
8.4. Programmable Window Detector 53
8.4.1. Window Detector Example 55
8.5. ADC0 Analog Multiplexer 56
9. Temperature Sensor 58
9.1. Calibration 58
10. Voltage and Ground Reference Options 60
10.1. External Voltage References 61
10.2. Internal Voltage Reference Options 61
10.3. Analog Ground Reference 61
10.4. Temperature Sensor Enable 61
11. Voltage Regulator (REG0) 63
12. Comparator0 65
12.1. Comparator Multiplexer 69
13. Capacitive Sense (CS0) 71
13.1. Configuring Port Pins as Capacitive Sense Inputs 72
13.2. Capacitive Sense Start-Of-Conversion Sources 72
13.3. Automatic Scanning 72
13.4. CS0 Comparator 73
13.5. CS0 Conversion Accumulator 74
13.6. Capacitive Sense Multiplexer 80
14. CIP-51 Microcontroller 82
14.1. Instruction Set 83
14.1.1. Instruction and CPU Timing 83
14.2. CIP-51 Register Descriptions 88
15. Memory Organization 92
15.1. Program Memory 93
15.1.1. MOVX Instruction and Program Memory 93

15.2. Data Memory 93
15.2.1. Internal RAM 93
15.2.1.1. General Purpose Registers 94
15.2.1.2. Bit Addressable Locations 94
15.2.1.3. Stack 94
16. In-System Device Identification 95
17. Special Function Registers 97
18. Interrupts 102
18.1. MCU Interrupt Sources and Vectors 103
18.1.1. Interrupt Priorities 103
18.1.2. Interrupt Latency 103
18.2. Interrupt Register Descriptions 104
18.3. INT0 and INT1 External Interrupts 111
19. Flash Memory 113
19.1. Programming The Flash Memory 113
19.1.1. Flash Lock and Key Functions 113
19.1.2. Flash Erase Procedure 113
19.1.3. Flash Write Procedure 114
19.2. Non-volatile Data Storage 114
19.3. Security Options 114
19.4. Flash Write and Erase Guidelines 115
19.4.1. VDD Maintenance and the VDD Monitor 116
19.4.2. PSWE Maintenance 116
19.4.3. System Clock 117
20. Power Management Modes 120
20.1. Idle Mode 120
20.2. Stop Mode 121
20.3. Suspend Mode 121
21. Reset Sources 123
21.1. Power-On Reset 124
21.2. Power-Fail Reset / VDD Monitor 125
21.3. External Reset 126
21.4. Missing Clock Detector Reset 126
21.5. Comparator0 Reset 127
21.6. PCA Watchdog Timer Reset 127
21.7. Flash Error Reset 127
21.8. Software Reset 127
22. Oscillators and Clock Selection 129
22.1. System Clock Selection 129
22.2. Programmable Internal High-Frequency (H-F) Oscillator 131
22.3. External Oscillator Drive Circuit 133
22.3.1. External Crystal Example 135
22.3.2. External RC Example 136
22.3.3. External Capacitor Example 137
23. Port Input/Output 138

Table 3.1. Pin Definitions for the C8051F80x-83x (Continued)

Name	Pin QSOP-24	Pin QFN-20	Pin SOIC-16	Type	Description
P0.5	21	16	14	D I/O or A In	Port 0.5.
P0.6/ CNVSTR	20	15	13	D I/O or A In D In	Port 0.6. ADC0 External Convert Start or IDA0 Update Source Input.
P0.7	19	14	12	D I/O or A In	Port 0.7.
P1.0	18	13	11	D I/O or A In	Port 1.0.
P1.1	17	12	10	D I/O or A In	Port 1.1.
P1.2	16	11	9	D I/O or A In	Port 1.2.
P1.3	15	10	8	D I/O or A In	Port 1.3.
P1.4	14	9		D I/O or A In	Port 1.4.
P1.5	11	8		D I/O or A In	Port 1.5.
P1.6	10	7		D I/O or A In	Port 1.6.
P1.7	9	6		D I/O or A In	Port 1.7.
NC	1, 12, 13, 24				No Connection.

5. QSOP-24 Package Specifications

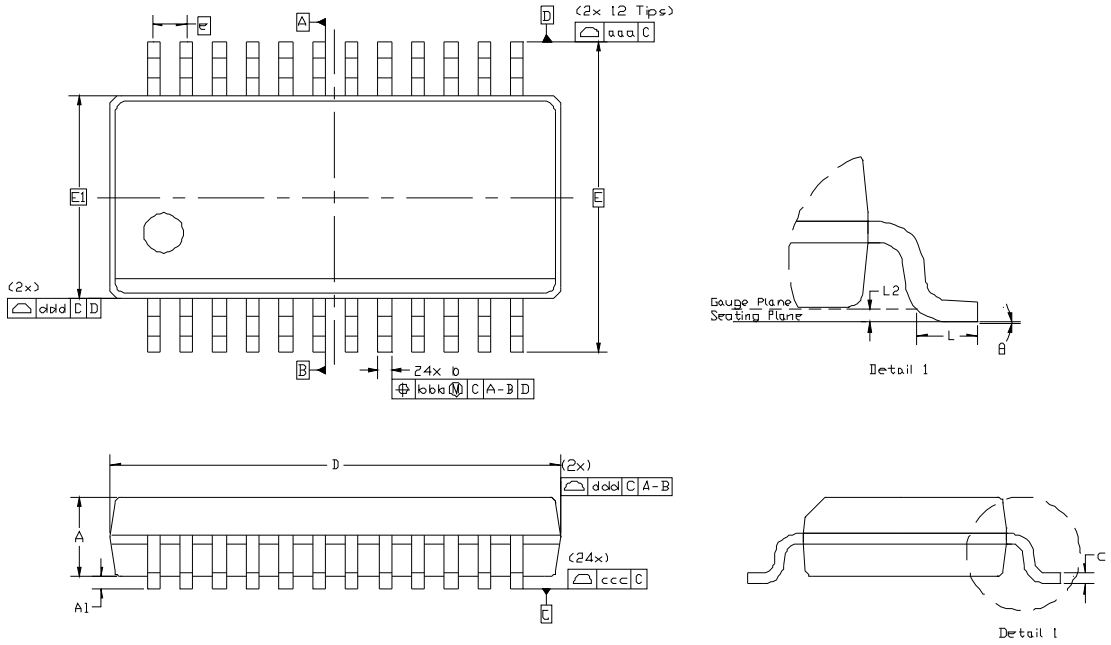

Figure 5.1. QSOP-24 Package Drawing

Table 5.1. QSOP-24 Package Dimensions

Dimension	Min	Nom	Max
A	—	—	1.75
A1	0.10	—	0.25
b	0.20	—	0.30
c	0.10	—	0.25
D	8.65 BSC		
E	6.00 BSC		
E1	3.90 BSC		
e	0.635 BSC		

Dimension	Min	Nom	Max
L	0.40	—	1.27
L2	0.25 BSC		
θ	0º	—	8º
aaa	0.20		
bbb	0.18		
ccc	0.10		
ddd	0.10		

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MO-137, variation AE.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

7. Electrical Characteristics

7.1. Absolute Maximum Specifications

Table 7.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient temperature under bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on $\overline{\text{RST}}$ or any Port I/O Pin with respect to GND		–0.3	—	5.8	V
Voltage on V_{DD} with respect to GND		–0.3	—	4.2	V
Maximum Total current through V_{DD} and GND		—	—	500	mA
Maximum output current sunk by $\overline{\text{RST}}$ or any Port pin		—	—	100	mA

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

Table 7.13. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	220	—	ns
	CP0+ – CP0– = –100 mV	—	225	—	ns
Response Time: Mode 1, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	340	—	ns
	CP0+ – CP0– = –100 mV	—	380	—	ns
Response Time: Mode 2, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	510	—	ns
	CP0+ – CP0– = –100 mV	—	945	—	ns
Response Time: Mode 3, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	1500	—	ns
	CP0+ – CP0– = –100 mV	—	5000	—	ns
Common-Mode Rejection Ratio		—	1	4	mV/V
Positive Hysteresis 1	Mode 2, CP0HYP1–0 = 00b	—	0	1	mV
Positive Hysteresis 2	Mode 2, CP0HYP1–0 = 01b	2	5	10	mV
Positive Hysteresis 3	Mode 2, CP0HYP1–0 = 10b	7	10	20	mV
Positive Hysteresis 4	Mode 2, CP0HYP1–0 = 11b	10	20	30	mV
Negative Hysteresis 1	Mode 2, CP0HYN1–0 = 00b	—	0	1	mV
Negative Hysteresis 2	Mode 2, CP0HYN1–0 = 01b	2	5	10	mV
Negative Hysteresis 3	Mode 2, CP0HYN1–0 = 10b	7	10	20	mV
Negative Hysteresis 4	Mode 2, CP0HYN1–0 = 11b	10	20	30	mV
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{DD} + 0.25	V
Input Offset Voltage		–7.5	—	7.5	mV
Power Specifications					
Power Supply Rejection		—	0.1	—	mV/V
Powerup Time		—	10	—	µs
Supply Current at DC	Mode 0	—	20	—	µA
	Mode 1	—	8	—	µA
	Mode 2	—	3	—	µA
	Mode 3	—	0.5	—	µA

Note: Vcm is the common-mode voltage on CP0+ and CP0–.

9. Temperature Sensor

An on-chip temperature sensor is included on the C8051F800/1/2/3/4/5, C8051F812/3/4/5/6/7, C8051F824/5/6, and C8051F830/1/2 which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 9.1. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly.

The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 10.1. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 7.11 for the slope and offset parameters of the temperature sensor.

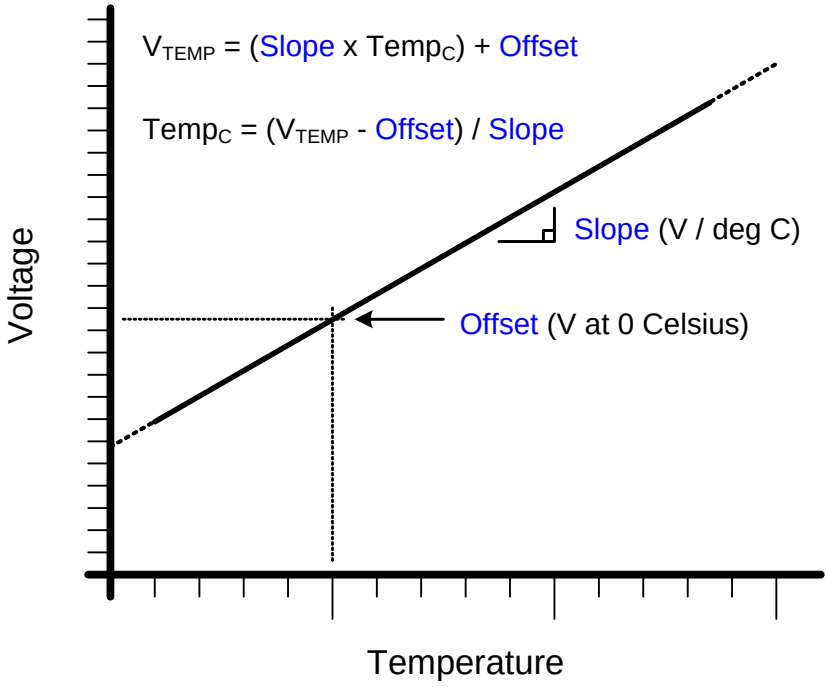

Figure 9.1. Temperature Sensor Transfer Function

9.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the ADC's input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 0 °C.

Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.

SFR Definition 10.1. REF0CN: Voltage Reference Control

Bit	7	6	5	4	3	2	1	0
Name			REFGND	REFSL		TEMPE	BIASE	
Type	R	R	R/W	R/W	R/W	R/W	R/W	R
Reset	0	0	0	1	0	0	0	0

SFR Address = 0xD1

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5	REFGND	**Analog Ground Reference.** Selects the ADC0 ground reference. 0: The ADC0 ground reference is the GND pin. 1: The ADC0 ground reference is the P0.1/AGND pin.
4:3	REFSL	**Voltage Reference Select.** Selects the ADC0 voltage reference. 00: The ADC0 voltage reference is the P0.0/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 1.65 V high speed voltage reference.
2	TEMPE	**Temperature Sensor Enable.** Enables/Disables the internal temperature sensor. 0: Temperature Sensor Disabled. 1: Temperature Sensor Enabled.
1	BIASE	**Internal Analog Bias Generator Enable Bit.** 0: Internal Bias Generator off. 1: Internal Bias Generator on.
0	Unused	Read = 0b; Write = Don't Care.

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.

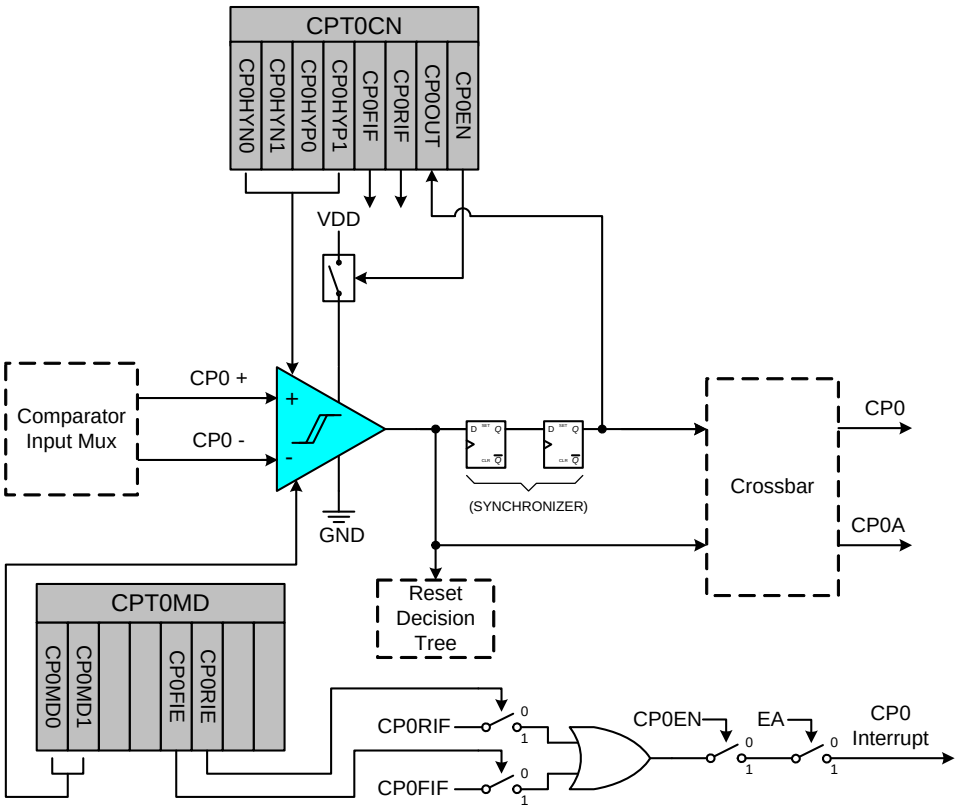

Figure 12.1. Comparator0 Functional Block Diagram

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.

Notes on Registers, Operands and Addressing Modes:
Rn—Register R0–R7 of the currently selected register bank.
@Ri—Data RAM location addressed indirectly through R0 or R1.
rel—8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.
direct—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).
#data—8-bit constant
#data16—16-bit constant
bit—Direct-accessed bit in Data RAM or SFR
addr11—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.
addr16—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.
There is one unused opcode (0xA5) that performs the same function as NOP. All mnemonics copyrighted © Intel Corporation 1980.

SFR Definition 19.2. FLKEY: Flash Lock and Key

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

SFR Address = 0xB7

Bit	Name	Function
7:0	FLKEY[7:0]	**Flash Lock and Key Register.** Write: This register provides a lock and key function for Flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a Flash write or erase operation is attempted while these operations are disabled, the Flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to Flash, it can intentionally lock the Flash by writing a non-0xA5 value to FLKEY from software. Read: When read, bits 1–0 indicate the current Flash lock state. 00: Flash is write/erase locked. 01: The first key code has been written (0xA5). 10: Flash is unlocked (writes/erases allowed). 11: Flash writes/erases disabled until the next reset.

22.3. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 22.1. A 10 MΩ resistor also must be wired across the XTAL2 and XTAL1 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 22.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 22.4).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “23.3. Priority Crossbar Decoder” on page 143 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “23.4. Port I/O Initialization” on page 147 for details on Port input mode selection.

Table 23.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P1.7	ADC0MX, PnSKIP, PnMDIN
Comparator0 Input	P0.0–P1.7	CPT0MX, PnSKIP, PnMDIN
CS0 Input	P0.0–P1.7	CS0MX, CS0SS, CS0SE, PnMDIN
Voltage Reference (VREF0)	P0.0	REF0CN, P0SKIP, PnMDIN
Ground Reference (AGND)	P0.1	REF0CN, P0SKIP
External Oscillator in Crystal Mode (XTAL1)	P0.2	OSCXCN, P0SKIP, P0MDIN
External Oscillator in RC, C, or Crystal Mode (XTAL2)	P0.3	OSCXCN, P0SKIP, P0MDIN

23.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 23.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

24.1. 16-bit CRC Algorithm

The C8051F80x-83x CRC unit calculates the 16-bit CRC MSB-first, using a poly of 0x1021. The following describes the 16-bit CRC algorithm performed by the hardware:

1. XOR the most-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 (0x0000 or 0xFFFF).
2. If the MSB of the CRC result is set, left-shift the CRC result, and then XOR the CRC result with the polynomial (0x1021).
3. If the MSB of the CRC result is not set, left-shift the CRC result.
4. Repeat at Step 2 for the number of input bits (8).

For example, the 16-bit C8051F80x-83x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input){
   unsigned char i;                    // loop counter
   #define POLY 0x1021
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);
   // "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 & 0x8000) == 0x8000)
      {
         // 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 24.1 lists example input values and the associated outputs using the 16-bit C8051F80x-83x CRC algorithm (an initial value of 0xFFFF is used):

Table 24.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

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

Table 25.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 25.8 and Figure 25.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing (See Figure 25.10 and Figure 25.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns
Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

Table 27.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

Table 27.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX[2]	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

28.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode. Timer 2 can also be used in capture mode to capture rising edges of the Comparator 0 output.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external oscillator source. The external oscillator source divided by 8 is synchronized with the system clock when in all operating modes except suspend. When the internal oscillator is placed in suspend mode, The external clock/8 signal can directly drive the timer. This allows the use of an external clock to wake up the device from suspend mode. The timer will continue to run in suspend mode and count up. When the timer overflow occurs, the device will wake from suspend mode, and begin executing code again. The timer value may be set prior to entering suspend, to overflow in the desired amount of time (number of clocks) to wake the device. If a wake-up source other than the timer wakes the device from suspend mode, it may take up to three timer clocks before the timer registers can be read or written. During this time, the STSYNC bit in register OSCICN will be set to 1, to indicate that it is not safe to read or write the timer registers.

28.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 28.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

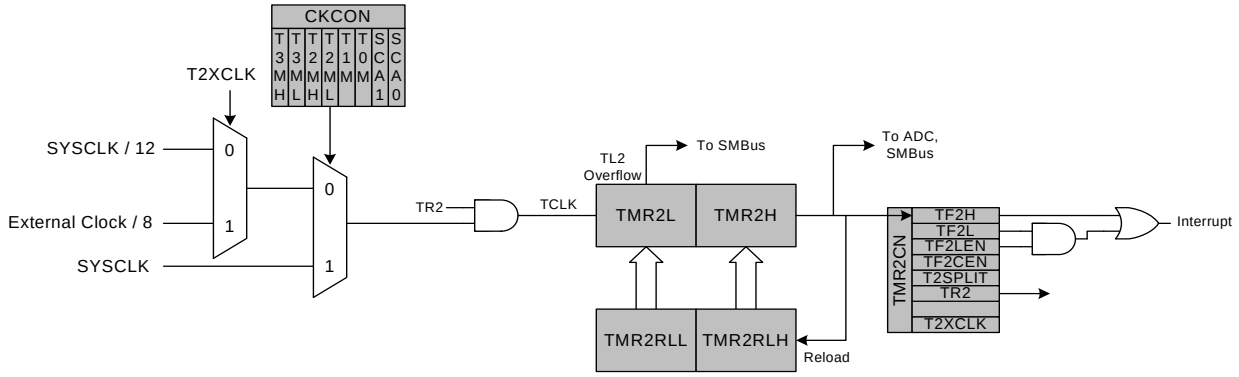

Figure 28.4. Timer 2 16-Bit Mode Block Diagram

SFR Definition 28.9. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCA

Bit	Name	Function
7:0	TMR2RLL[7:0]	**Timer 2 Reload Register Low Byte.** TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 28.10. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCB

Bit	Name	Function
7:0	TMR2RLH[7:0]	**Timer 2 Reload Register High Byte.** TMR2RLH holds the high byte of the reload value for Timer 2.

29.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 29.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 29.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. The MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

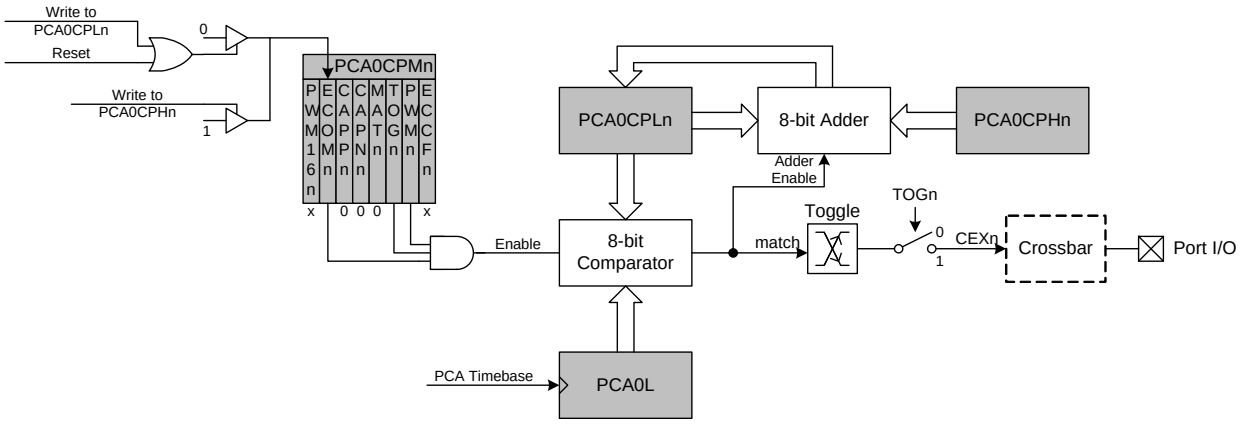

Figure 29.7. PCA Frequency Output Mode

29.3.5. 8-bit through 15-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10, 11, 12, 13, 14, or 15-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10, 11, 12, 13, 14, and 15-bit PWM modes. **It is important to note that all channels configured for 8-bit through 15-bit PWM mode will use the same cycle length.** For example, it is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode. However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.