

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	13
Program Memory Size	16KB (16K 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/c8051f810-gs

List of Figures

1. System Overview

Figure 1.1. C8051F800, C8051F806, C8051F812, C8051F818 Block Diagram	16
Figure 1.2. C8051F801, C8051F807, C8051F813, C8051F819 Block Diagram	17
Figure 1.3. C8051F802, C8051F808, C8051F814, C8051F820 Block Diagram	18
Figure 1.4. C8051F803, C8051F809, C8051F815, C8051F821 Block Diagram	19
Figure 1.5. C8051F804, C8051F810, C8051F816, C8051F822 Block Diagram	20
Figure 1.6. C8051F805, C8051F811, C8051F817, C8051F823 Block Diagram	21
Figure 1.7. C8051F824, C8051F827, C8051F830, C8051F833 Block Diagram	22
Figure 1.8. C8051F825, C8051F828, C8051F831, C8051F834 Block Diagram	23
Figure 1.9. C8051F826, C8051F829, C8051F832, C8051F835 Block Diagram	24

2. Ordering Information

3. Pin Definitions

Figure 3.1. QFN-20 Pinout Diagram (Top View)	30
Figure 3.2. QSOP-24 Pinout Diagram (Top View)	31
Figure 3.3. SOIC-16 Pinout Diagram (Top View)	32

4. QFN-20 Package Specifications

Figure 4.1. QFN-20 Package Drawing	33
Figure 4.2. QFN-20 Recommended PCB Land Pattern	34

5. QSOP-24 Package Specifications

Figure 5.1. QSOP-24 Package Drawing	35
Figure 5.2. QSOP-24 PCB Land Pattern	36

6. SOIC-16 Package Specifications

Figure 6.1. SOIC-16 Package Drawing	37
Figure 6.2. SOIC-16 PCB Land Pattern	38

7. Electrical Characteristics

8. 10-Bit ADC (ADC0)

Figure 8.1. ADC0 Functional Block Diagram	46
Figure 8.2. 10-Bit ADC Track and Conversion Example Timing	48
Figure 8.3. ADC0 Equivalent Input Circuits	49
Figure 8.4. ADC Window Compare Example: Right-Justified Data	55
Figure 8.5. ADC Window Compare Example: Left-Justified Data	55
Figure 8.6. ADC0 Multiplexer Block Diagram	56

9. Temperature Sensor

Figure 9.1. Temperature Sensor Transfer Function	58
Figure 9.2. Temperature Sensor Error with 1-Point Calibration at 0 °C	59

10. Voltage and Ground Reference Options

Figure 10.1. Voltage Reference Functional Block Diagram	60
---	----

11. Voltage Regulator (REG0)

12. Comparator0

Figure 12.1. Comparator0 Functional Block Diagram	65
Figure 12.2. Comparator Hysteresis Plot	66
Figure 12.3. Comparator Input Multiplexer Block Diagram	69

13. Capacitive Sense (CS0)

C8051F80x-83x

Figure 13.1. CS0 Block Diagram	71
Figure 13.2. Auto-Scan Example	73
Figure 13.3. CS0 Multiplexer Block Diagram	80
14. CIP-51 Microcontroller	
Figure 14.1. CIP-51 Block Diagram	82
15. Memory Organization	
Figure 15.1. C8051F80x-83x Memory Map	92
Figure 15.2. Flash Program Memory Map	93
16. In-System Device Identification	
17. Special Function Registers	
18. Interrupts	
19. Flash Memory	
20. Power Management Modes	
21. Reset Sources	
Figure 21.1. Reset Sources	123
Figure 21.2. Power-On and VDD Monitor Reset Timing	124
22. Oscillators and Clock Selection	
Figure 22.1. Oscillator Options	129
Figure 22.2. External 32.768 kHz Quartz Crystal Oscillator Connection Diagram	136
23. Port Input/Output	
Figure 23.1. Port I/O Functional Block Diagram	138
Figure 23.2. Port I/O Cell Block Diagram	139
Figure 23.3. Port I/O Overdrive Current	140
Figure 23.4. Priority Crossbar Decoder Potential Pin Assignments	144
Figure 23.5. Priority Crossbar Decoder Example 1—No Skipped Pins	145
Figure 23.6. Priority Crossbar Decoder Example 2—Skipping Pins	146
24. Cyclic Redundancy Check Unit (CRC0)	
Figure 24.1. CRC0 Block Diagram	159
25. Enhanced Serial Peripheral Interface (SPI0)	
Figure 25.1. SPI Block Diagram	167
Figure 25.2. Multiple-Master Mode Connection Diagram	169
Figure 25.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram	169
Figure 25.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram	170
Figure 25.5. Master Mode Data/Clock Timing	172
Figure 25.6. Slave Mode Data/Clock Timing (CKPHA = 0)	172
Figure 25.7. Slave Mode Data/Clock Timing (CKPHA = 1)	173
Figure 25.8. SPI Master Timing (CKPHA = 0)	177
Figure 25.9. SPI Master Timing (CKPHA = 1)	177
Figure 25.10. SPI Slave Timing (CKPHA = 0)	178
Figure 25.11. SPI Slave Timing (CKPHA = 1)	178
26. SMBus	
Figure 26.1. SMBus Block Diagram	180
Figure 26.2. Typical SMBus Configuration	181

Figure 26.3. SMBus Transaction	182
Figure 26.4. Typical SMBus SCL Generation	184
Figure 26.5. Typical Master Write Sequence	193
Figure 26.6. Typical Master Read Sequence	194
Figure 26.7. Typical Slave Write Sequence	195
Figure 26.8. Typical Slave Read Sequence	196
27. UART0	
Figure 27.1. UART0 Block Diagram	201
Figure 27.2. UART0 Baud Rate Logic	202
Figure 27.3. UART Interconnect Diagram	203
Figure 27.4. 8-Bit UART Timing Diagram	203
Figure 27.5. 9-Bit UART Timing Diagram	204
Figure 27.6. UART Multi-Processor Mode Interconnect Diagram	205
28. Timers	
Figure 28.1. T0 Mode 0 Block Diagram	212
Figure 28.2. T0 Mode 2 Block Diagram	213
Figure 28.3. T0 Mode 3 Block Diagram	214
Figure 28.4. Timer 2 16-Bit Mode Block Diagram	219
Figure 28.5. Timer 2 8-Bit Mode Block Diagram	220
29. Programmable Counter Array	
Figure 29.1. PCA Block Diagram	225
Figure 29.2. PCA Counter/Timer Block Diagram	226
Figure 29.3. PCA Interrupt Block Diagram	227
Figure 29.4. PCA Capture Mode Diagram	229
Figure 29.5. PCA Software Timer Mode Diagram	230
Figure 29.6. PCA High-Speed Output Mode Diagram	231
Figure 29.7. PCA Frequency Output Mode	232
Figure 29.8. PCA 8-Bit PWM Mode Diagram	233
Figure 29.9. PCA 9-bit through 15-Bit PWM Mode Diagram	234
Figure 29.10. PCA 16-Bit PWM Mode	235
Figure 29.11. PCA Module 2 with Watchdog Timer Enabled	236
30. C2 Interface	
Figure 30.1. Typical C2 Pin Sharing	247

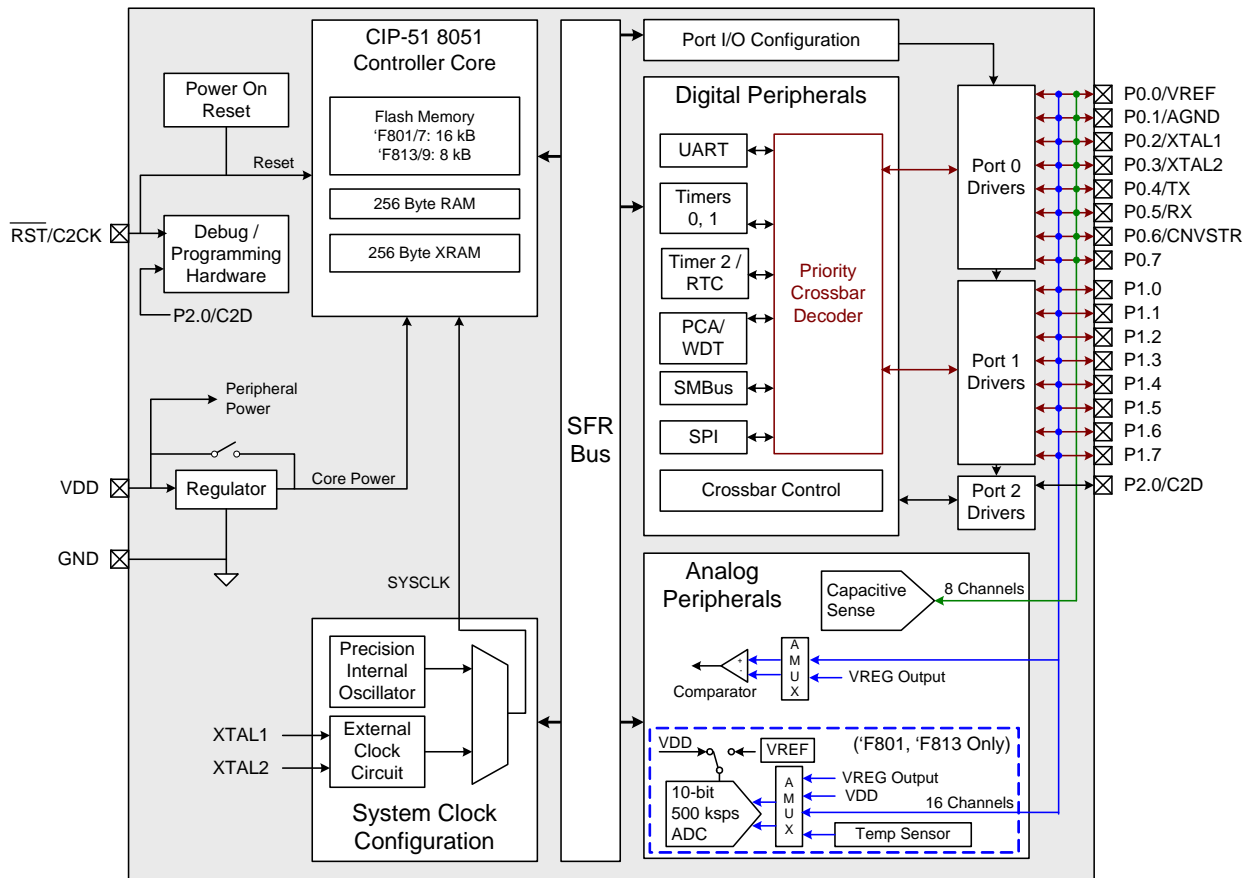


Figure 1.2. C8051F801, C8051F807, C8051F813, C8051F819 Block Diagram

C8051F80x-83x

SFR Definition 8.9. ADC0MX: AMUX0 Channel Select

Bit	7	6	5	4	3	2	1	0
Name				AMX0P[3:0]				
Type	R	R	R	R/W				
Reset	0	0	0	1	1	1	1	1

SFR Address = 0xBB

Bit	Name	Function																																																																		
7:5	Unused	Read = 000b; Write = Don't Care.																																																																		
4:0	AMX0P[4:0]	AMUX0 Positive Input Selection. <table> <tr> <th></th><th>20-Pin and 24-Pin Devices</th><th>16-Pin Devices</th></tr> <tr><td>00000:</td><td>P0.0</td><td>P0.0</td></tr> <tr><td>00001:</td><td>P0.1</td><td>P0.1</td></tr> <tr><td>00010:</td><td>P0.2</td><td>P0.2</td></tr> <tr><td>00011:</td><td>P0.3</td><td>P0.3</td></tr> <tr><td>00100:</td><td>P0.4</td><td>P0.4</td></tr> <tr><td>00101:</td><td>P0.5</td><td>P0.5</td></tr> <tr><td>00110:</td><td>P0.6</td><td>P0.6</td></tr> <tr><td>00111:</td><td>P0.7</td><td>P0.7</td></tr> <tr><td>01000:</td><td>P1.0</td><td>P1.0</td></tr> <tr><td>01001:</td><td>P1.1</td><td>P1.1</td></tr> <tr><td>01010:</td><td>P1.2</td><td>P1.2</td></tr> <tr><td>01011:</td><td>P1.3</td><td>P1.3</td></tr> <tr><td>01100:</td><td>P1.4</td><td>Reserved.</td></tr> <tr><td>01101:</td><td>P1.5</td><td>Reserved.</td></tr> <tr><td>01110:</td><td>P1.6</td><td>Reserved.</td></tr> <tr><td>01111:</td><td>P1.7</td><td>Reserved.</td></tr> <tr><td>10000:</td><td>Temp Sensor</td><td>Temp Sensor</td></tr> <tr><td>10001:</td><td>VREG Output</td><td>VREG Output</td></tr> <tr><td>10010:</td><td>VDD</td><td>VDD</td></tr> <tr><td>10011:</td><td>GND</td><td>GND</td></tr> <tr><td>10100 – 11111:</td><td colspan="2">no input selected</td></tr> </table>		20-Pin and 24-Pin Devices	16-Pin Devices	00000:	P0.0	P0.0	00001:	P0.1	P0.1	00010:	P0.2	P0.2	00011:	P0.3	P0.3	00100:	P0.4	P0.4	00101:	P0.5	P0.5	00110:	P0.6	P0.6	00111:	P0.7	P0.7	01000:	P1.0	P1.0	01001:	P1.1	P1.1	01010:	P1.2	P1.2	01011:	P1.3	P1.3	01100:	P1.4	Reserved.	01101:	P1.5	Reserved.	01110:	P1.6	Reserved.	01111:	P1.7	Reserved.	10000:	Temp Sensor	Temp Sensor	10001:	VREG Output	VREG Output	10010:	VDD	VDD	10011:	GND	GND	10100 – 11111:	no input selected	
	20-Pin and 24-Pin Devices	16-Pin Devices																																																																		
00000:	P0.0	P0.0																																																																		
00001:	P0.1	P0.1																																																																		
00010:	P0.2	P0.2																																																																		
00011:	P0.3	P0.3																																																																		
00100:	P0.4	P0.4																																																																		
00101:	P0.5	P0.5																																																																		
00110:	P0.6	P0.6																																																																		
00111:	P0.7	P0.7																																																																		
01000:	P1.0	P1.0																																																																		
01001:	P1.1	P1.1																																																																		
01010:	P1.2	P1.2																																																																		
01011:	P1.3	P1.3																																																																		
01100:	P1.4	Reserved.																																																																		
01101:	P1.5	Reserved.																																																																		
01110:	P1.6	Reserved.																																																																		
01111:	P1.7	Reserved.																																																																		
10000:	Temp Sensor	Temp Sensor																																																																		
10001:	VREG Output	VREG Output																																																																		
10010:	VDD	VDD																																																																		
10011:	GND	GND																																																																		
10100 – 11111:	no input selected																																																																			

C8051F80x-83x

SFR Definition 11.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0
Name	STOPCF							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC9

Bit	Name	Function
7	STOPCF	Stop Mode Configuration. This bit configures the regulator's behavior when the device enters STOP mode. 0: Regulator is still active in STOP mode. Any enabled reset source will reset the device. 1: Regulator is shut down in STOP mode. Only the $\overline{\text{RST}}$ pin or power cycle can reset the device.
6:0	Reserved	Must write to 0000000b.

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.

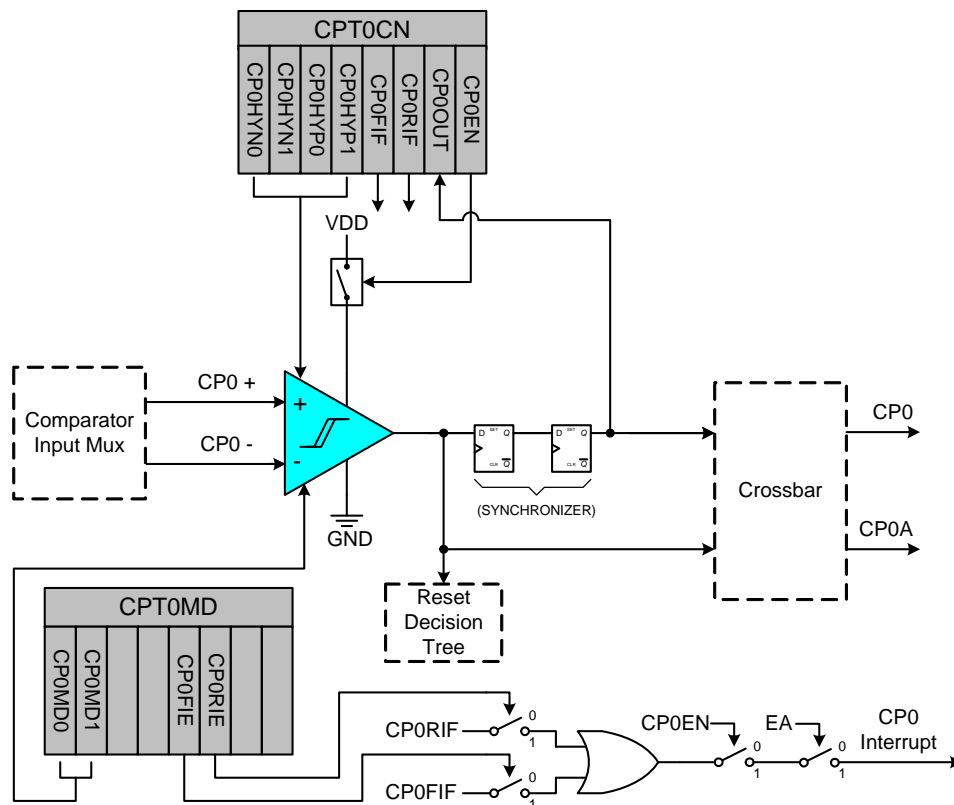


Figure 12.1. Comparator0 Functional Block Diagram

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 digital 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 $(V_{DD}) + 0.25\text{ V}$ without damage or upset. The complete Comparator electrical specifications are given in Section “7. Electrical Characteristics” on page 39.

Table 14.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2

Table 17.2. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
ACC	0xE0	Accumulator	89
ADC0CF	0xBC	ADC0 Configuration	50
ADC0CN	0xE8	ADC0 Control	52
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	53
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	53
ADC0H	0xBE	ADC0 High	51
ADC0L	0xBD	ADC0 Low	51
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	54
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	54
ADC0MX	0xBB	AMUX0 Multiplexer Channel Select	57
B	0xF0	B Register	90
CKCON	0x8E	Clock Control	210
CLKSEL	0xA9	Clock Select	210
CPT0CN	0x9B	Comparator0 Control	67
CPT0MD	0x9D	Comparator0 Mode Selection	68
CPT0MX	0x9F	Comparator0 MUX Selection	70
CRC0AUTO	0xD2	CRC0 Automatic Control Register	165
CRC0CN	0xCE	CRC0 Control	163
CRC0CNT	0xD3	CRC0 Automatic Flash Sector Count	165
CRC0DATA	0xDE	CRC0 Data Output	164
CRC0FLIP	0xCF	CRC0 Bit Flip	166
CRC0IN	0xDD	CRC Data Input	164
CS0THH	0x97	CS0 Digital Compare Threshold High	79
CS0THL	0x96	CS0 Digital Compare Threshold High	79
CS0CN	0xB0	CS0 Control	75
CS0DH	0xAC	CS0 Data High	77
CS0DL	0xAB	CS0 Data Low	77

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.

22.3.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 22.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation according to Equation 22.2, where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and V_{DD} = the MCU power supply in volts.

Equation 22.2. C Mode Oscillator Frequency

$$f = (KF) / (R \times V_{DD})$$

For example: Assume $V_{DD} = 3.0$ V and $f = 150$ kHz:

$$f = KF / (C \times V_{DD})$$

$$0.150 \text{ MHz} = KF / (C \times 3.0)$$

Since the frequency of roughly 150 kHz is desired, select the K Factor from the table in SFR Definition 22.4 (OSCXCN) as $KF = 22$:

$$0.150 \text{ MHz} = 22 / (C \times 3.0)$$

$$C \times 3.0 = 22 / 0.150 \text{ MHz}$$

$$C = 146.6 / 3.0 \text{ pF} = 48.8 \text{ pF}$$

Therefore, the XFCN value to use in this example is 011b and $C = 50$ pF.

23.4. Port I/O Initialization

Port I/O initialization consists of the following steps:

1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN). **If the pin is in analog mode, a '1' must also be written to the corresponding Port Latch (Pn).**
2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
4. Assign Port pins to desired peripherals (XBR0, XBR1).
5. Enable the Crossbar (XBARE = 1).

All Port pins must be configured as either analog or digital inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a 1 indicates a digital input, and a 0 indicates an analog input. All port pins in analog mode must have a '1' set in the corresponding Port Latch register. All pins default to digital inputs on reset. See SFR Definition 23.8 and SFR Definition 23.12 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a 0 to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to 1 enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. Port output drivers are disabled while the Crossbar is disabled.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 26.3 illustrates a typical SMBus transaction.

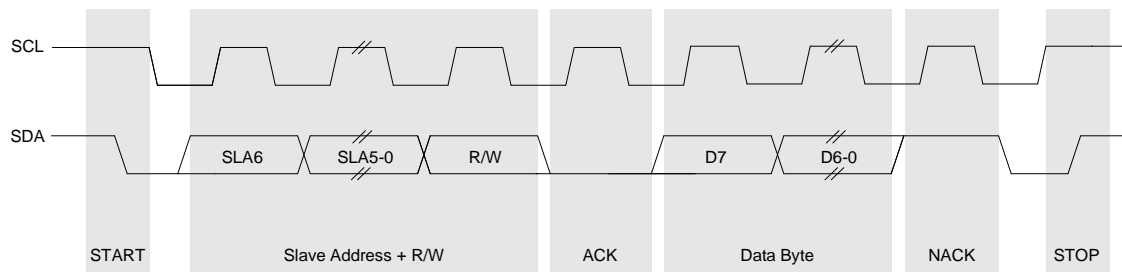


Figure 26.3. SMBus Transaction

26.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

26.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “26.3.5. SCL High (SMBus Free) Timeout” on page 183). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

26.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

26.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

26.4.4. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 26.5. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC2

Bit	Name	Function
7:0	SMB0DAT[7:0]	SMBus Data. The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

26.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. Note that the position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

26.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 26.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

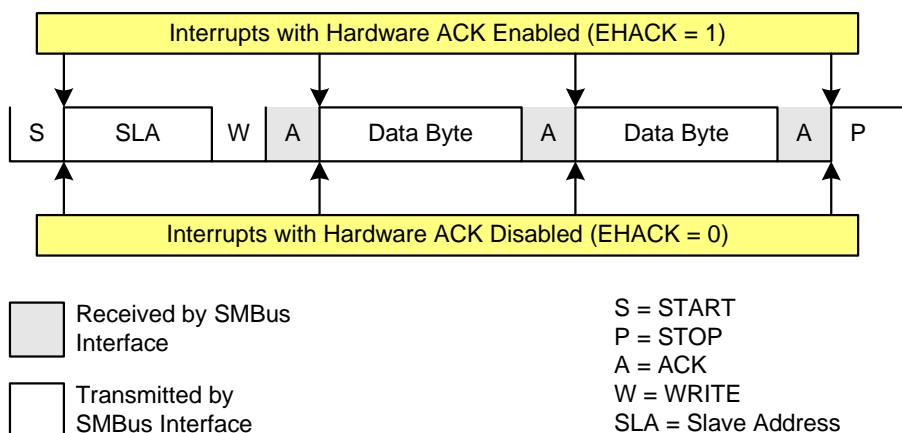


Figure 26.5. Typical Master Write Sequence

28.1. Timer 0 and Timer 1

Each timer is implemented as a 16-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 (Section “18.2. Interrupt Register Descriptions” on page 104); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section “18.2. Interrupt Register Descriptions” on page 104). 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. Each operating mode is described below.

28.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 in TCON is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit in the TMOD register selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “23.3. Priority Crossbar Decoder” on page 143 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit in register CKCON. When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 28.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 18.7). Setting GATE0 to 1 allows the timer to be controlled by the external input signal INT0 (see Section “18.2. Interrupt Register Descriptions” on page 104), facilitating pulse width measurements

TR0	GATE0	INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal INT1 is used with Timer 1; the INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 18.7).

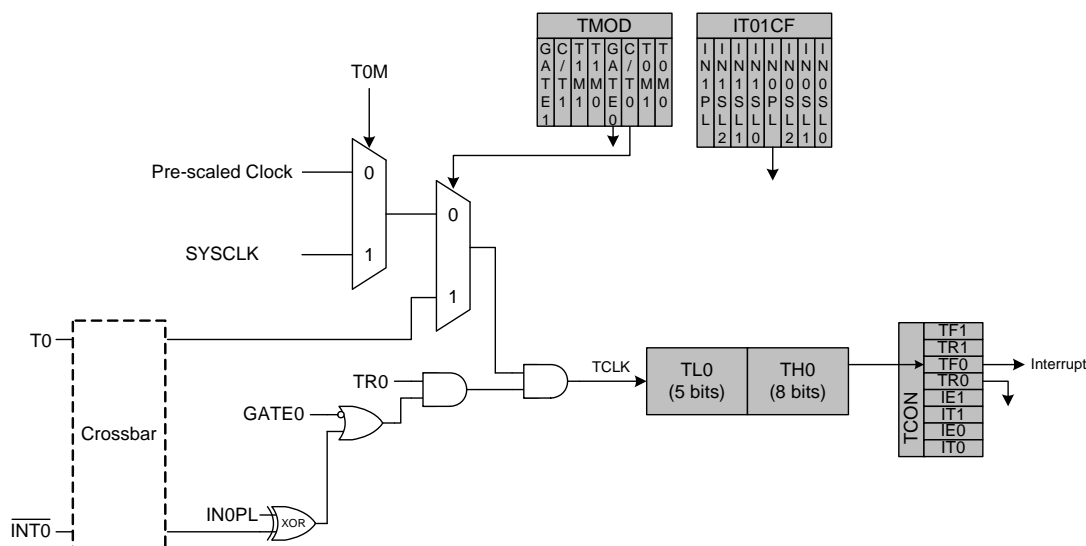


Figure 28.1. T0 Mode 0 Block Diagram

28.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

28.1.3. 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 (**TCON.4**) enables the timer when either **GATE0** in the **TMOD** register is logic 0 or when the input signal **INT0** is active as defined by bit **IN0PL** in register **IT01CF** (see Section “18.3. INT0 and INT1 External Interrupts” on page 111 for details on the external input signals **INT0** and **INT1**).

C8051F80x-83x

28.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 28.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode. Timer 2 can also be used in capture mode to capture rising edges of the Comparator 0 output.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

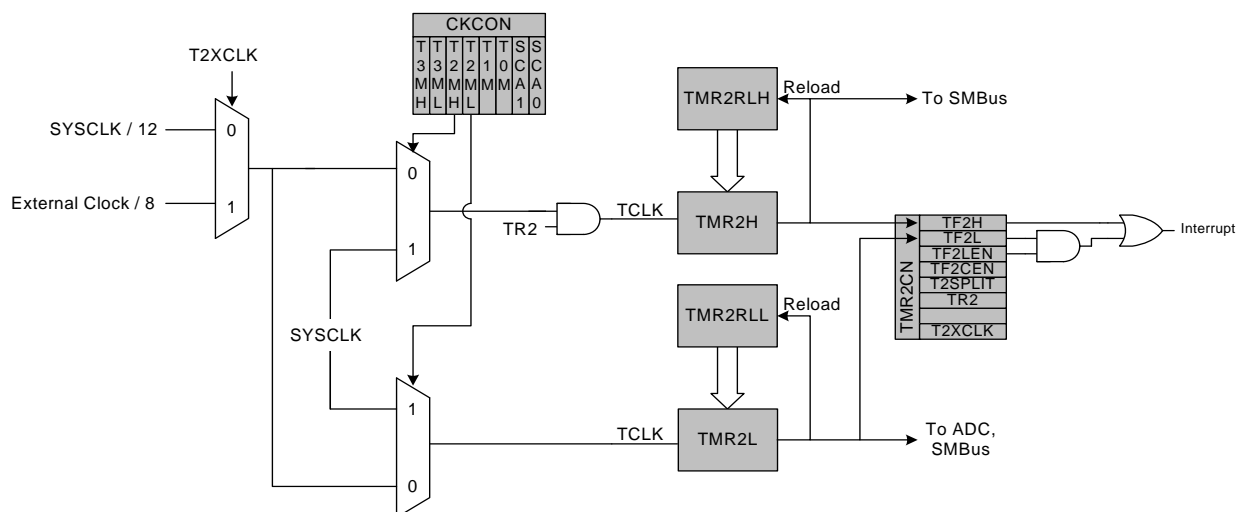


Figure 28.5. Timer 2 8-Bit Mode Block Diagram

SFR Definition 29.2. PCA0MD: PCA0 Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS2	CPS1	CPS0	ECF
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Address = 0xD9

Bit	Name	Function
7	CIDL	PCA Counter/Timer Idle Control. Specifies PCA behavior when CPU is in idle mode. 0: PCA continues to function normally while the system controller is in Idle mode. 1: PCA operation is suspended while the system controller is in idle mode.
6	WDTE	Watchdog Timer Enable. If this bit is set, PCA Module 2 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 2 enabled as Watchdog Timer.
5	WDLCK	Watchdog Timer Lock. This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = Don't care.
3:1	CPS[2:0]	PCA Counter/Timer Pulse Select. These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 11x: Reserved
0	ECF	PCA Counter/Timer Overflow Interrupt Enable. This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

Note: When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

C8051F80x-83x

C2 Register Definition 30.2. DEVICEID: C2 Device ID

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

C2 Address: 0x00

Bit	Name	Function
7:0	DEVICEID[7:0]	Device ID. This read-only register returns the 8-bit device ID: 0x23 (C8051F80x-83x).

C2 Register Definition 30.3. REVID: C2 Revision ID

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

C2 Address: 0x01

Bit	Name	Function
7:0	REVID[7:0]	Revision ID. This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.