

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 <sup>2</sup> C), SPI, UART/USART
Peripherals	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	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f811-gs">https://www.e-xfl.com/product-detail/silicon-labs/c8051f811-gs</a>

# C8051F80x-83x

---

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
<b>16. In-System Device Identification .....</b>	<b>95</b>
<b>17. Special Function Registers .....</b>	<b>97</b>
<b>18. Interrupts .....</b>	<b>102</b>
18.1. MCU Interrupt Sources and Vectors .....	103
18.1.1. Interrupt Priorities .....	103
18.1.2. Interrupt Latency .....	103
18.2. <u>Interrupt Register</u> Descriptions .....	104
18.3. INT0 and INT1 External Interrupts .....	111
<b>19. Flash Memory .....</b>	<b>113</b>
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
<b>20. Power Management Modes .....</b>	<b>120</b>
20.1. Idle Mode .....	120
20.2. Stop Mode .....	121
20.3. Suspend Mode .....	121
<b>21. Reset Sources .....</b>	<b>123</b>
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
<b>22. Oscillators and Clock Selection .....</b>	<b>129</b>
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
<b>23. Port Input/Output .....</b>	<b>138</b>

---

## List of Registers

SFR Definition 8.1. ADC0CF: ADC0 Configuration .....	50
SFR Definition 8.2. ADC0H: ADC0 Data Word MSB .....	51
SFR Definition 8.3. ADC0L: ADC0 Data Word LSB .....	51
SFR Definition 8.4. ADC0CN: ADC0 Control .....	52
SFR Definition 8.5. ADC0GTH: ADC0 Greater-Than Data High Byte .....	53
SFR Definition 8.6. ADC0GTL: ADC0 Greater-Than Data Low Byte .....	53
SFR Definition 8.7. ADC0LTH: ADC0 Less-Than Data High Byte .....	54
SFR Definition 8.8. ADC0LTL: ADC0 Less-Than Data Low Byte .....	54
SFR Definition 8.9. ADC0MX: AMUX0 Channel Select .....	57
SFR Definition 10.1. REF0CN: Voltage Reference Control .....	62
SFR Definition 11.1. REG0CN: Voltage Regulator Control .....	64
SFR Definition 12.1. CPT0CN: Comparator0 Control .....	67
SFR Definition 12.2. CPT0MD: Comparator0 Mode Selection .....	68
SFR Definition 12.3. CPT0MX: Comparator0 MUX Selection .....	70
SFR Definition 13.1. CS0CN: Capacitive Sense Control .....	75
SFR Definition 13.2. CS0CF: Capacitive Sense Configuration .....	76
SFR Definition 13.3. CS0DH: Capacitive Sense Data High Byte .....	77
SFR Definition 13.4. CS0DL: Capacitive Sense Data Low Byte .....	77
SFR Definition 13.5. CS0SS: Capacitive Sense Auto-Scan Start Channel .....	78
SFR Definition 13.6. CS0SE: Capacitive Sense Auto-Scan End Channel .....	78
SFR Definition 13.7. CS0THH: Capacitive Sense Comparator Threshold High Byte ...	79
SFR Definition 13.8. CS0THL: Capacitive Sense Comparator Threshold Low Byte ....	79
SFR Definition 13.9. CS0MX: Capacitive Sense Mux Channel Select .....	81
SFR Definition 14.1. DPL: Data Pointer Low Byte .....	88
SFR Definition 14.2. DPH: Data Pointer High Byte .....	88
SFR Definition 14.3. SP: Stack Pointer .....	89
SFR Definition 14.4. ACC: Accumulator .....	89
SFR Definition 14.5. B: B Register .....	90
SFR Definition 14.6. PSW: Program Status Word .....	91
SFR Definition 16.1. HWID: Hardware Identification Byte .....	95
SFR Definition 16.2. DERIVID: Derivative Identification Byte .....	96
SFR Definition 16.3. REVID: Hardware Revision Identification Byte .....	96
SFR Definition 18.1. IE: Interrupt Enable .....	105
SFR Definition 18.2. IP: Interrupt Priority .....	106
SFR Definition 18.3. EIE1: Extended Interrupt Enable 1 .....	107
SFR Definition 18.4. EIE2: Extended Interrupt Enable 2 .....	108
SFR Definition 18.5. EIP1: Extended Interrupt Priority 1 .....	109
SFR Definition 18.6. EIP2: Extended Interrupt Priority 2 .....	110
SFR Definition 18.7. IT01CF: INT0/INT1 Configuration .....	112
SFR Definition 19.1. PSCTL: Program Store R/W Control .....	118
SFR Definition 19.2. FLKEY: Flash Lock and Key .....	119
SFR Definition 20.1. PCON: Power Control .....	122
SFR Definition 21.1. VDM0CN: VDD Monitor Control .....	126

---

## 1. System Overview

C8051F80x-83x devices are fully integrated, mixed-signal, system-on-a-chip capacitive sensing MCUs. Highlighted features are listed below. Refer to Table 2.1 for specific product feature selection and part ordering numbers.

- High-speed pipelined 8051-compatible microcontroller core (up to 25 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Capacitive sense interface with 16 input channels
- 10-bit 500 ksp/s single-ended ADC with 16-channel analog multiplexer and integrated temperature sensor
- Precision calibrated 24.5 MHz internal oscillator
- 16 kb of on-chip Flash memory
- 512 bytes of on-chip RAM
- SMBus/I<sup>2</sup>C, Enhanced UART, and Enhanced SPI serial interfaces implemented in hardware
- Three general-purpose 16-bit timers
- Programmable counter/timer array (PCA) with three capture/compare modules
- On-chip internal voltage reference
- On-chip Watchdog timer
- On-chip power-on reset and supply monitor
- On-chip voltage comparator
- 17 general purpose I/O

With on-chip power-on reset,  $V_{DD}$  monitor, watchdog timer, and clock oscillator, the C8051F80x-83x devices are truly stand-alone, system-on-a-chip solutions. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The C8051F80x-83x processors include Silicon Laboratories' 2-Wire C2 Debug and Programming interface, which allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection of memory, viewing and modification of special function registers, setting breakpoints, single stepping, and run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

Each device is specified for 1.8–3.6 V operation over the industrial temperature range (–45 to +85 °C). An internal LDO regulator is used to supply the processor core voltage at 1.8 V. The Port I/O and  $\overline{RST}$  pins are tolerant of input signals up to 5 V. See Table 2.1 for ordering information. Block diagrams of the devices in the C8051F80x-83x family are shown in Figure 1.1 through Figure 1.9.

## 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 $\overline{RST}$ 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 $\overline{RST}$ or any Port pin		—	—	100	mA
<p><b>Note:</b> 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.</p>					

8. 10-Bit ADC (ADC0)

ADC0 on the C8051F800/1/2/3/4/5, C8051F812/3/4/5/6/7, C8051F824/5/6, and C8051F830/1/2 is a 500 ksps, 10-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, a gain stage programmable to 1x or 0.5x, and a programmable window detector. The ADC is fully configurable under software control via Special Function Registers. The ADC may be configured to measure various different signals using the analog multiplexer described in Section “8.5. ADC0 Analog Multiplexer” on page 56. The voltage reference for the ADC is selected as described in Section “9. Temperature Sensor” on page 58. The ADC0 subsystem is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

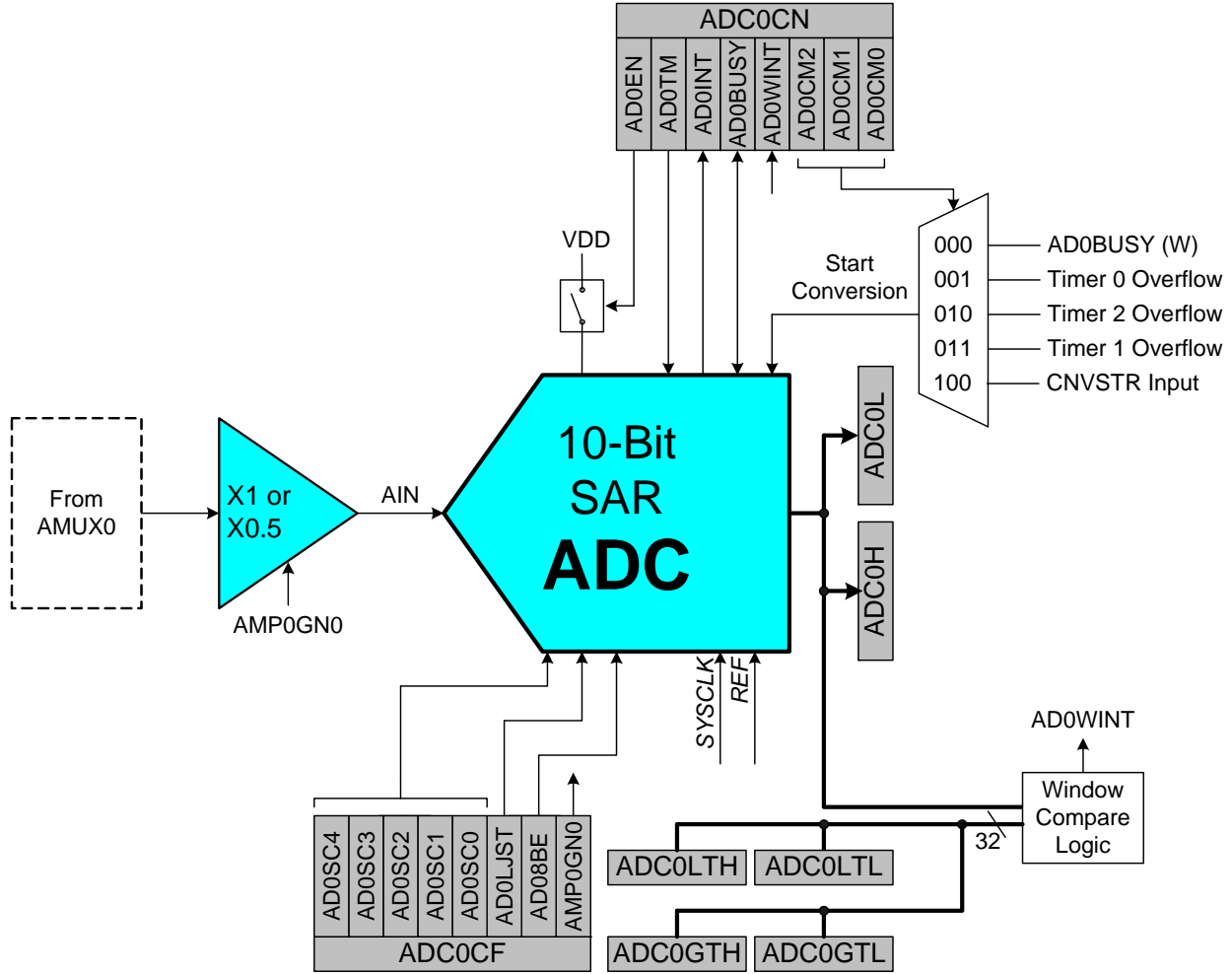
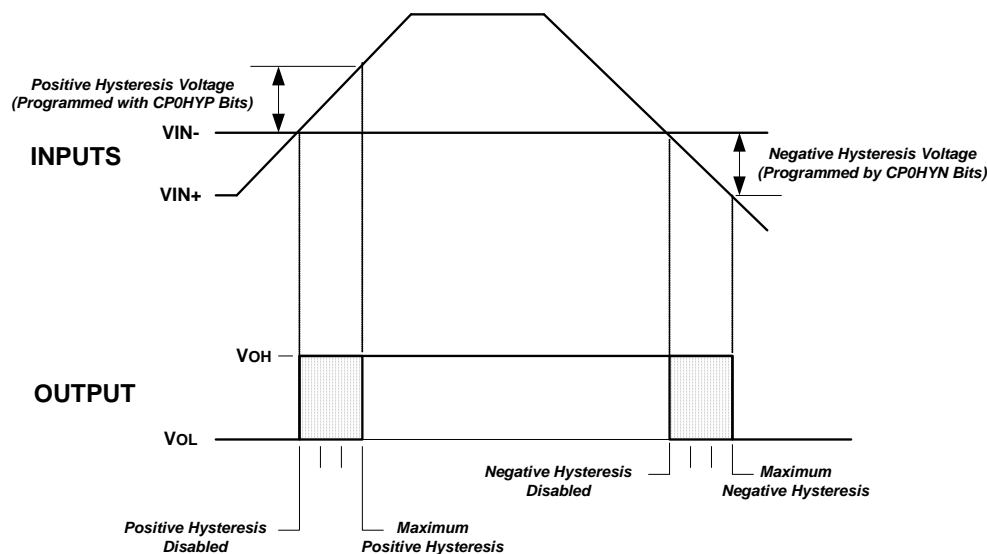
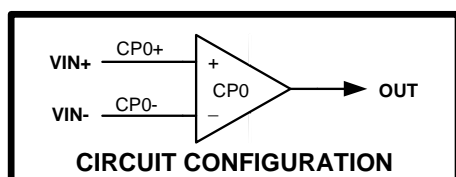


Figure 8.1. ADC0 Functional Block Diagram

# C8051F80x-83x

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 12.2). Selecting a longer response time reduces the Comparator supply current.



**Figure 12.2. Comparator Hysteresis Plot**

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using bits 3:0 in the Comparator Control Register CPT0CN (shown in SFR Definition 12.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 12.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section "18.1. MCU Interrupt Sources and Vectors" on page 103). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CP0RIF flag is set to logic 1 upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE to a logic 1. The Comparator0 falling-edge interrupt mask is enabled by setting CP0FIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed.

**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 18.4. EIE2: Extended Interrupt Enable 2

Bit	7	6	5	4	3	2	1	0
<b>Name</b>							ECSGRT	ECSCPT
<b>Type</b>	R	R	R	R	R	R	R/W	R/W
<b>Reset</b>	0	0	0	0	0	0	0	0

SFR Address = 0xE7

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = don't care.
1	ECSGRT	<b>Enable Capacitive Sense Greater Than Comparator Interrupt.</b> 0: Disable Capacitive Sense Greater Than Comparator interrupt. 1: Enable interrupt requests generated by CS0CMPF.
0	ECSCPT	<b>Enable Capacitive Sense Conversion Complete Interrupt.</b> 0: Disable Capacitive Sense Conversion Complete interrupt. 1: Enable interrupt requests generated by CS0INT.

---

## SFR Definition 20.1. PCON: Power Control

---

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

SFR Address = 0x87

Bit	Name	Function
7:2	GF[5:0]	<b>General Purpose Flags 5–0.</b> These are general purpose flags for use under software control.
1	STOP	<b>Stop Mode Select.</b> Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0. 1: CPU goes into Stop mode (internal oscillator stopped).
0	IDLE	<b>IDLE: Idle Mode Select.</b> Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0. 1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)

## SFR Definition 23.1. XBR0: Port I/O Crossbar Register 0

Bit	7	6	5	4	3	2	1	0
<b>Name</b>			CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E
<b>Type</b>	R	R	R/W	R/W	R/W	R/W	R/W	R/W
<b>Reset</b>	0	0	0	0	0	0	0	0

SFR Address = 0xE1

Bit	Name	Function
7:6	Unused	Read = 00b. Write = don't care.
5	CP0AE	<b>Comparator0 Asynchronous Output Enable.</b> 0: Asynchronous CP0 unavailable at Port pin. 1: Asynchronous CP0 routed to Port pin.
4	CP0E	<b>Comparator0 Output Enable.</b> 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.
3	SYSCKE	<b>SYSCLK Output Enable.</b> 0: $\overline{\text{SYSCLK}}$ unavailable at Port pin. 1: $\overline{\text{SYSCLK}}$ output routed to Port pin.
2	SMB0E	<b>SMBus I/O Enable.</b> 0: SMBus I/O unavailable at Port pins. 1: SMBus I/O routed to Port pins.
1	SPI0E	<b>SPI I/O Enable.</b> 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins. Note that the SPI can be assigned either 3 or 4 GPIO pins.
0	URT0E	<b>UART I/O Output Enable.</b> 0: UART I/O unavailable at Port pin. 1: UART TX0, RX0 routed to Port pins P0.4 and P0.5.

## SFR Definition 23.16. P2MDOUT: Port 2 Output Mode

Bit	7	6	5	4	3	2	1	0
Name								P2MDOUT[0]
Type	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA6

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don't Care
0	P2MDOUT[0]	<b>Output Configuration Bits for P2.0.</b> 0: P2.0 Output is open-drain. 1: P2.0 Output is push-pull.

## 25.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 25.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

### 25.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

### 25.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

### 25.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 25.2, Figure 25.3, and Figure 25.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “23. Port Input/Output” on page 138 for general purpose port I/O and crossbar information.

## 25.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag

## 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).

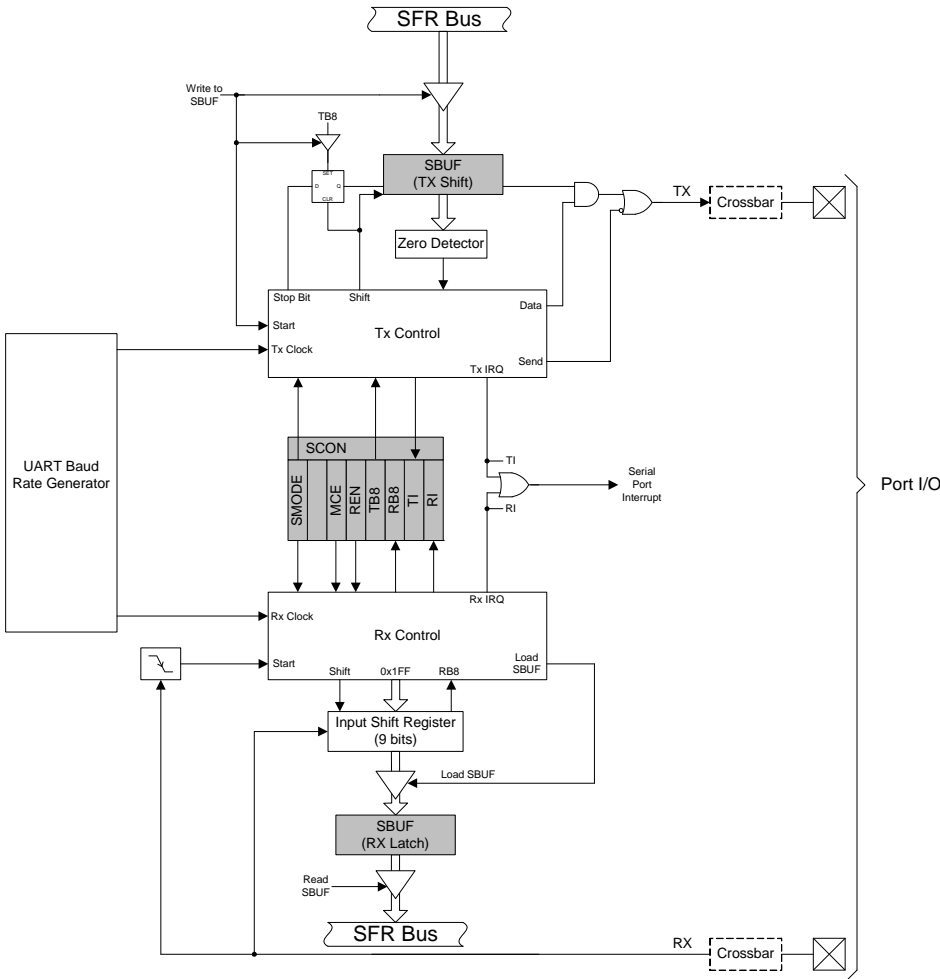


Figure 27.1. UART0 Block Diagram

## 27.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown in Figure 27.3.

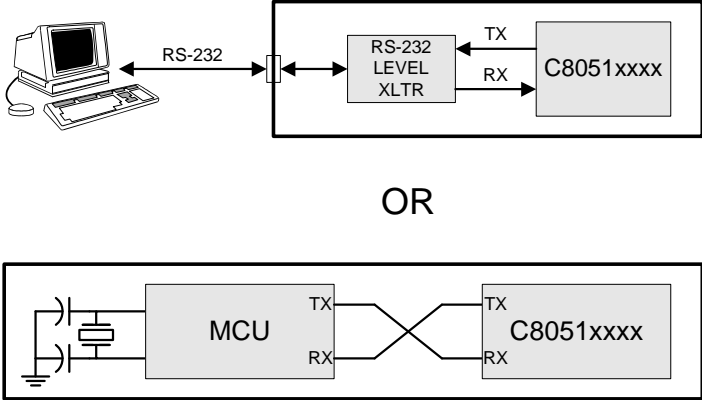


Figure 27.3. UART Interconnect Diagram

### 27.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The T10 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either T10 or RI0 is set.

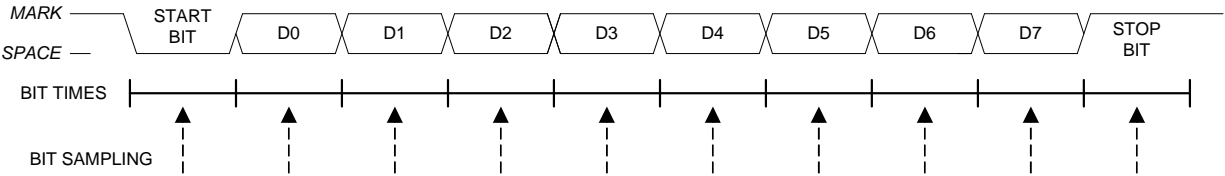


Figure 27.4. 8-Bit UART Timing Diagram

## SFR Definition 27.2. SBUF0: Serial (UART0) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x99

Bit	Name	Function
7:0	SBUF0[7:0]	<p><b>Serial Data Buffer Bits 7–0 (MSB–LSB).</b></p> <p>This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.</p>



## 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.

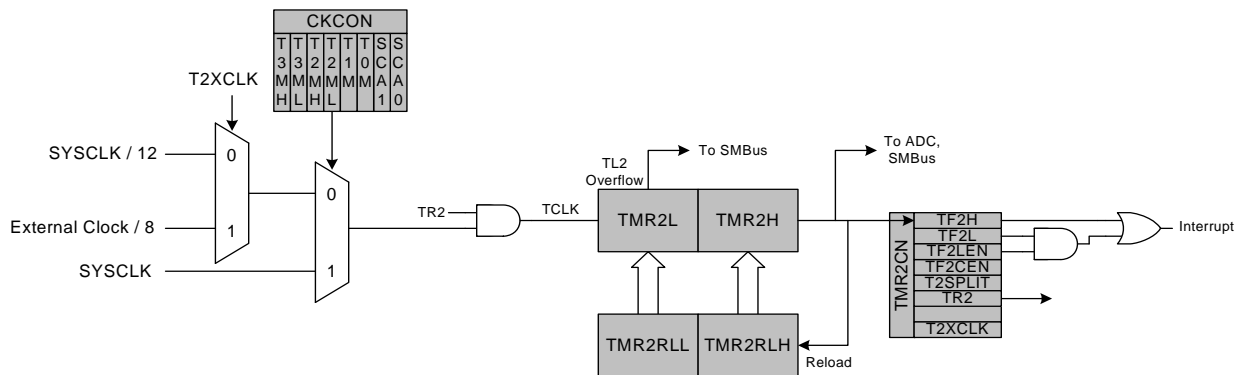


Figure 28.4. Timer 2 16-Bit Mode Block Diagram

### 28.2.3. Comparator 0 Capture Mode

The capture mode in Timer 2 allows Comparator 0 rising edges to be captured with the timer clocking from the system clock or the system clock divided by 12. Timer 2 capture mode is enabled by setting TF2CEN to 1 and T2SPLIT to 0.

When capture mode is enabled, a capture event will be generated on every Comparator 0 rising edge. When the capture event occurs, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set (triggering an interrupt if Timer 2 interrupts are enabled). By recording the difference between two successive timer capture values, the Comparator 0 period can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading.

This mode allows software to determine the time between consecutive Comparator 0 rising edges, which can be used for detecting changes in the capacitance of a capacitive switch, or measuring the frequency of a low-level analog signal.

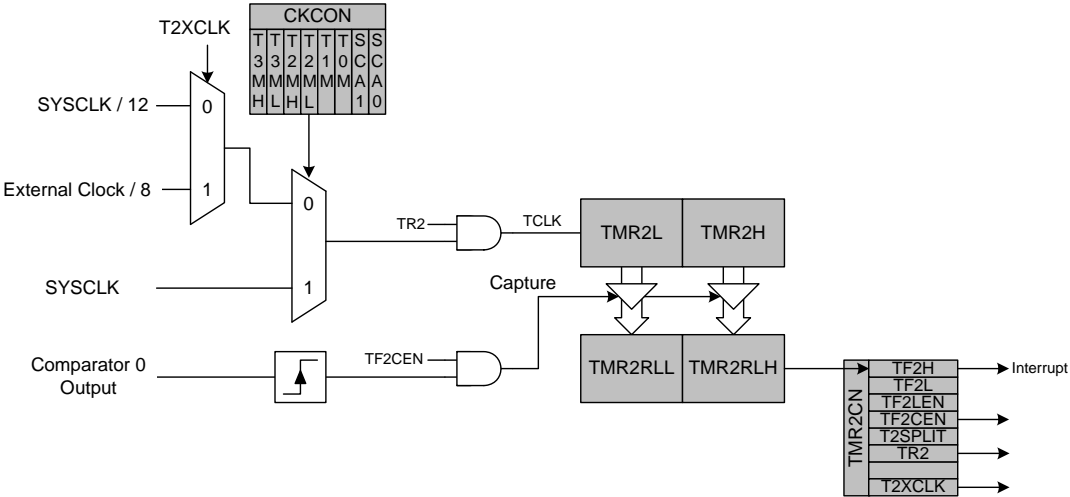


Figure 28.6. Timer 2 Capture Mode Block Diagram

## 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.

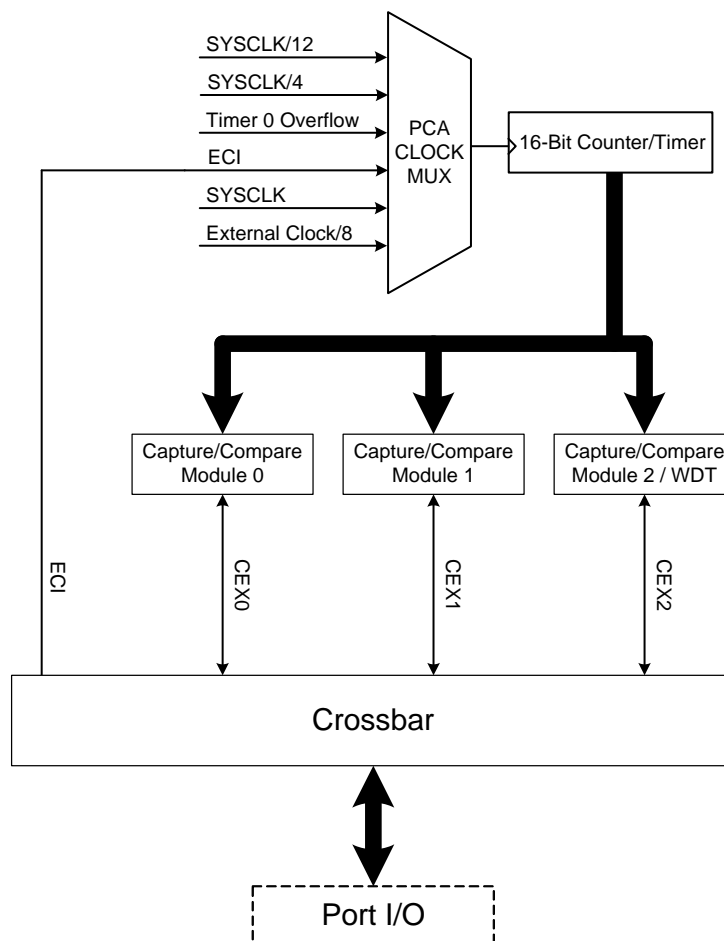


Figure 29.1. PCA Block Diagram

# C8051F80x-83x

## 29.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

**Important Note About Capture/Compare Registers:** When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

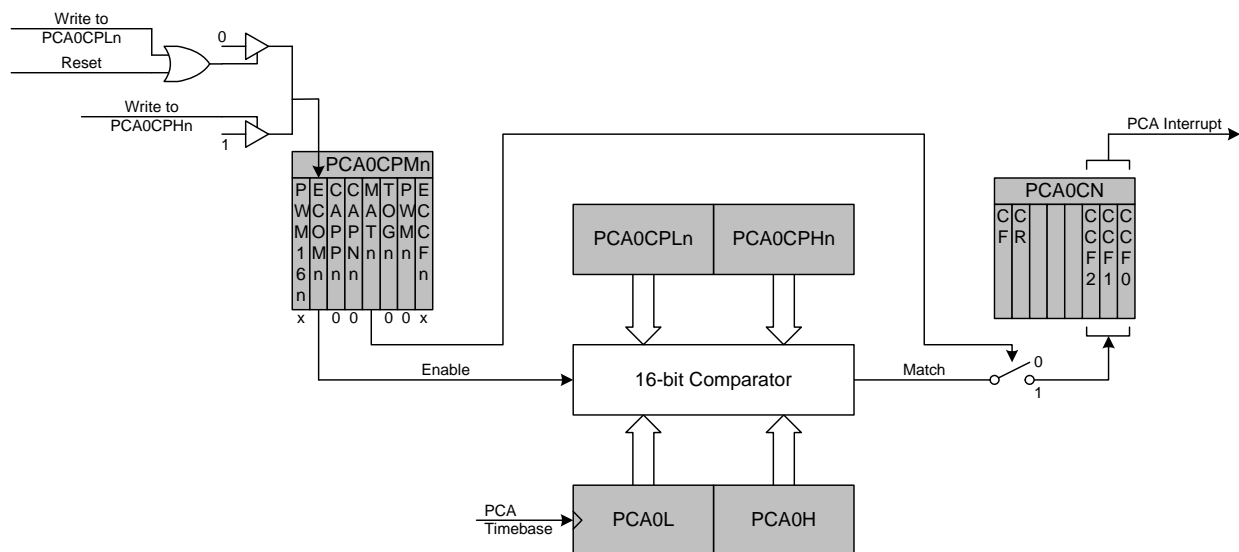


Figure 29.5. PCA Software Timer Mode Diagram

## SFR Definition 29.2. PCA0MD: PCA0 Mode

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

SFR Address = 0xD9

Bit	Name	Function
7	CIDL	<b>PCA Counter/Timer Idle Control.</b> 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	<b>Watchdog Timer Enable.</b> 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	<b>Watchdog Timer Lock.</b> 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]	<b>PCA Counter/Timer Pulse Select.</b> 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	<b>PCA Counter/Timer Overflow Interrupt Enable.</b> 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.