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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	8
Program Memory Size	2KB (2K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	256 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	10-VFDFN Exposed Pad
Supplier Device Package	11-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t605-gm

Table of Contents

List of Registers

9. 10-Bit ADC (ADC0, C8051T600/2/4 only)

ADC0 on the C8051T600/2/4 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 “9.5. ADC0 Analog Multiplexer (C8051T600/2/4 only)” on page 50. The voltage reference for the ADC is selected as described in Section “11. Voltage Reference Options” on page 55. 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.

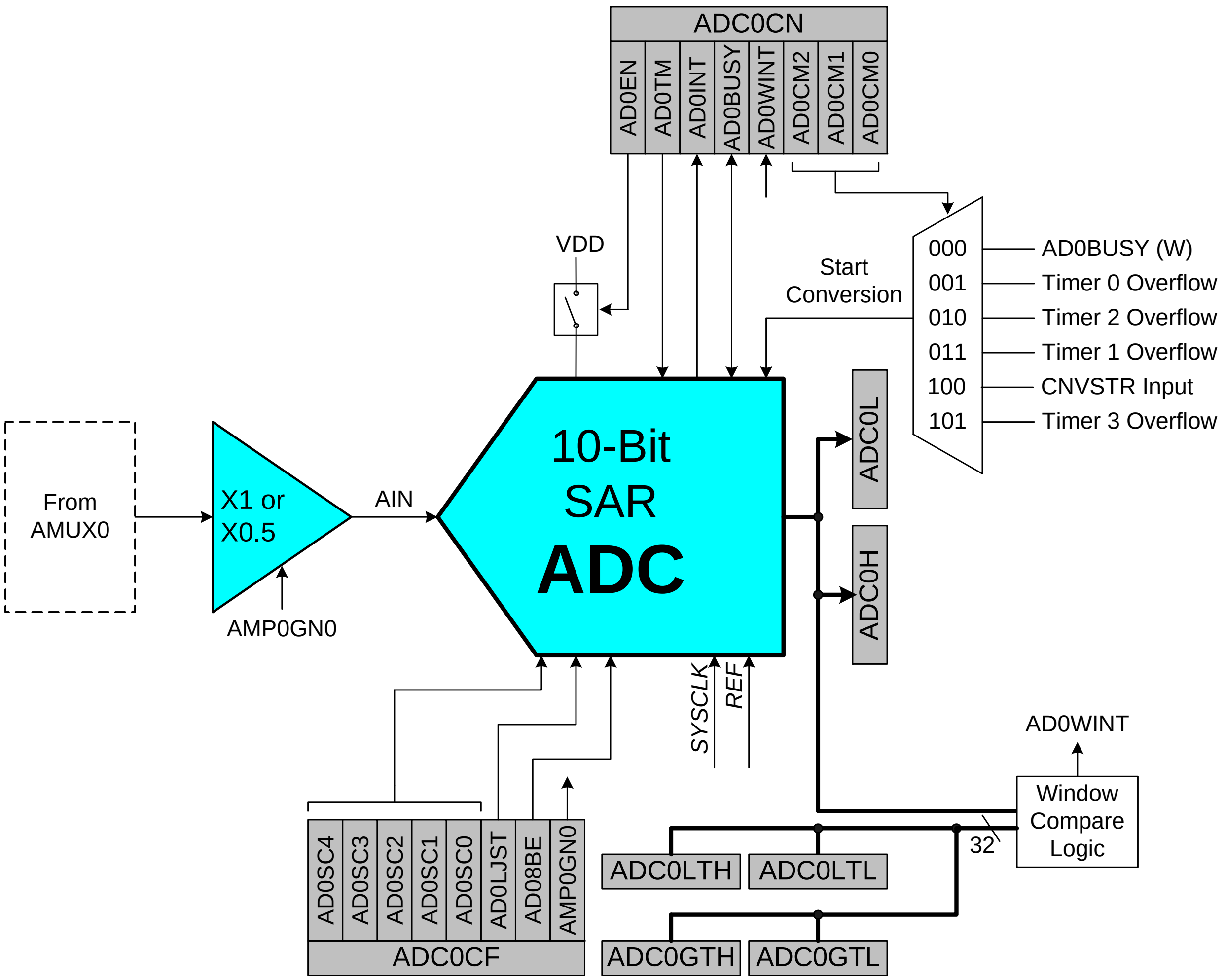

Figure 9.1. ADC0 Functional Block Diagram

9.4.1. Window Detector Example

Figure 9.4 shows two example window comparisons for right-justified data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). The input voltage can range from 0 to VREF x (1023/1024) with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0040 < ADC0H:ADC0L < 0x0080). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0040 or ADC0H:ADC0L > 0x0080). Figure 9.5 shows an example using left-justified data with the same comparison values.

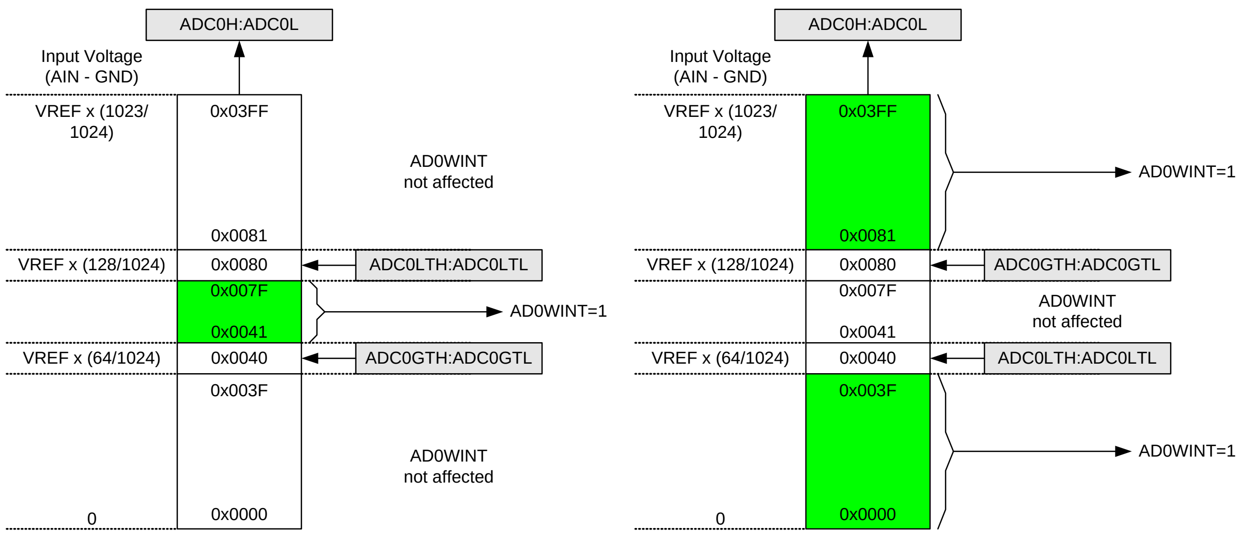

Figure 9.4. ADC Window Compare Example: Right-Justified Data

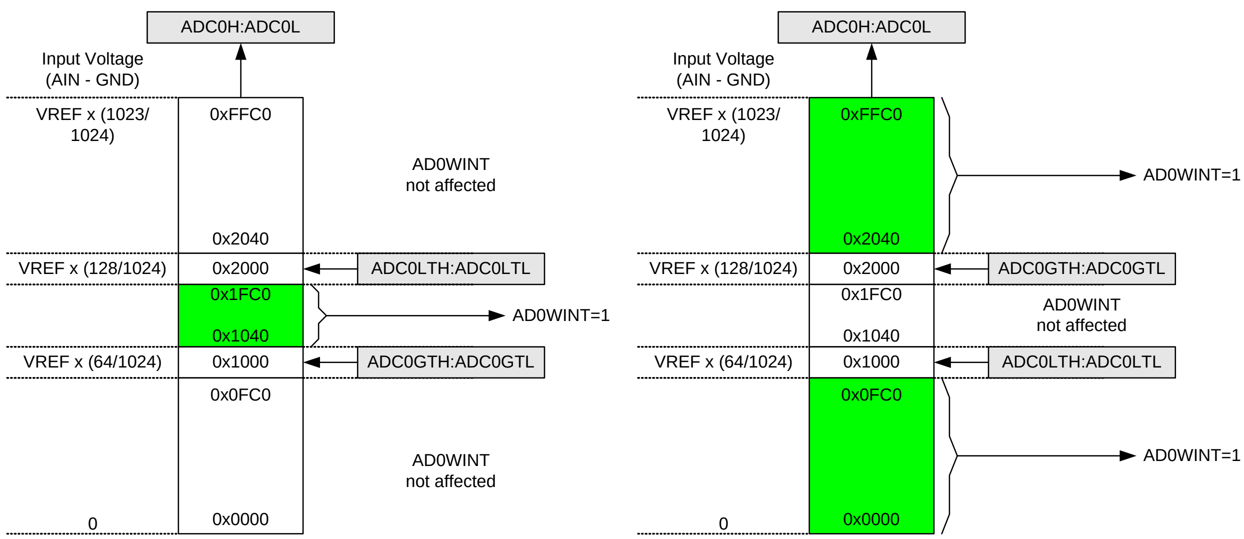

Figure 9.5. ADC Window Compare Example: Left-Justified Data

13.1. Comparator Multiplexer

C8051T600/1/2/3/4/5/6 devices include an analog input multiplexer to connect Port I/O pins to the comparator inputs. The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 13.3). The CMX0P1–CMX0P0 bits select the Comparator0 positive input; the CMX0N1–CMX0N0 bits select the Comparator0 negative input.

Important Note About Comparator Inputs: The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “22.5. Special Function Registers for Accessing and Configuring Port I/O” on page 118).

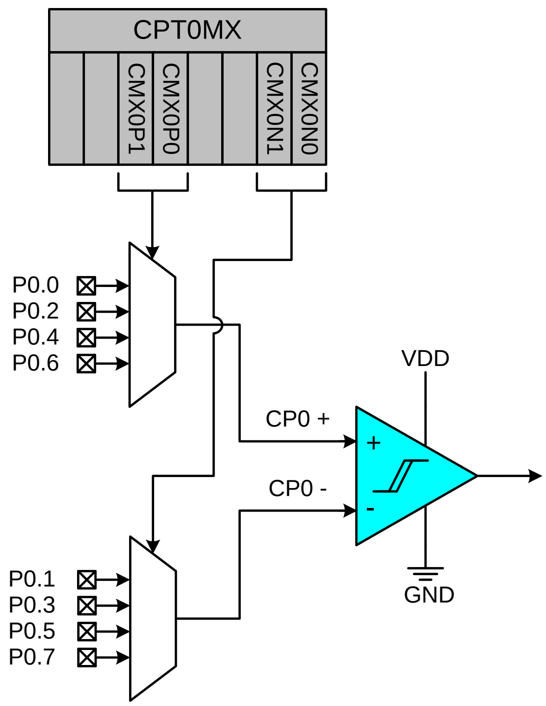

Figure 13.3. Comparator Input Multiplexer Block Diagram

Table 16.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
PCA0H	0xFA	PCA Counter High	176
PCA0L	0xF9	PCA Counter Low	176
PCA0MD	0xD9	PCA Mode	174
PCON	0x87	Power Control	91
PSW	0xD0	Program Status Word	73
REF0CN	0xD1	Voltage Reference Control	56
REG0CN	0xC7	Voltage Regulator Control	58
RSTSRC	0xEF	Reset Source Configuration/Status	96
SBUF0	0x99	UART0 Data Buffer	143
SCON0	0x98	UART0 Control	142
SMB0CF	0xC1	SMBus Configuration	126
SMB0CN	0xC0	SMBus Control	128
SMB0DAT	0xC2	SMBus Data	130
SP	0x81	Stack Pointer	72
TCON	0x88	Timer/Counter Control	151
TH0	0x8C	Timer/Counter 0 High	154
TH1	0x8D	Timer/Counter 1 High	154
TL0	0x8A	Timer/Counter 0 Low	153
TL1	0x8B	Timer/Counter 1 Low	153
TMOD	0x89	Timer/Counter Mode	152
TMR2CN	0xC8	Timer/Counter 2 Control	157
TMR2H	0xCD	Timer/Counter 2 High	159
TMR2L	0xCC	Timer/Counter 2 Low	158
TMR2RLH	0xCB	Timer/Counter 2 Reload High	158
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	158
TOFFH	0xA3	Temperature Sensor Offset Measurement High	54
TOFFL	0xA2	Temperature Sensor Offset Measurement Low	54
XBR0	0xE1	Port I/O Crossbar Control 0	115
XBR1	0xE2	Port I/O Crossbar Control 1	116
XBR2	0xE3	Port I/O Crossbar Control 2	117
All other SFR Locations		Reserved	

21.3. External Oscillator Drive Circuit

The external oscillator circuit may drive an external capacitor or RC network. A CMOS clock may also provide a clock input. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the EXTCLK pin as shown in Figure 21.1. 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 21.3).

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 capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as EXTCLK. The Port I/O Crossbar should be configured to skip the Port pin used by the oscillator circuit; see Section “22.3. Priority Crossbar Decoder” on page 111 for Crossbar configuration. Additionally, when using the external oscillator circuit in capacitor or RC mode, the associated Port pin should be configured as an **analog input**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “22.4. Port I/O Initialization” on page 114 for details on Port input mode selection.

SFR Definition 21.3. OSCXCN: External Oscillator Control

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

SFR Address = 0xB1

Bit	Name	Function
7	Unused	Read = 0b; Write = Don’t Care
6:4	XOSCMD[2:0]	**External Oscillator Mode Select.** 00x: External Oscillator circuit off. 010: External CMOS Clock Mode. 011: External CMOS Clock Mode with divide by 2 stage. 100: RC Oscillator Mode with divide by 2 stage. 101: Capacitor Oscillator Mode with divide by 2 stage. 11x: Reserved.
3	Unused	Read = 0b; Write = Don’t Care
2:0	XFCN[2:0]	**External Oscillator Frequency Control Bits.** Set according to the desired frequency range for RC mode. Set according to the desired K Factor for C mode.

XFCN	RC Mode	C Mode
000	$f \leq 25$ kHz	K Factor = 0.87
001	25 kHz $< f \leq$ 50 kHz	K Factor = 2.6
010	50 kHz $< f \leq$ 100 kHz	K Factor = 7.7
011	100 kHz $< f \leq$ 200 kHz	K Factor = 22
100	200 kHz $< f \leq$ 400 kHz	K Factor = 65
101	400 kHz $< f \leq$ 800 kHz	K Factor = 180
110	800 kHz $< f \leq$ 1.6 MHz	K Factor = 664
111	1.6 MHz $< f \leq$ 3.2 MHz	K Factor = 1590

22.1.3. Interfacing Port I/O to 5V Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage higher than VDD and less than 5.25 V. An external pullup resistor to the higher supply voltage is typically required for most systems.

Important Note: In a multi-voltage interface, the external pullup resistor should be sized to allow a current of at least 150 µA to flow into the Port pin when the supply voltage is between (VDD + 0.6 V) and (VDD + 1.0 V). Once the Port pin voltage increases beyond this range, the current flowing into the Port pin is minimal.

Table 23.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	■ A START is generated.	■ A STOP is generated. ■ Arbitration is lost.
TXMODE	■ START is generated. ■ SMB0DAT is written before the start of an SMBus frame.	■ A START is detected. ■ Arbitration is lost. ■ SMB0DAT is not written before the start of an SMBus frame.
STA	■ A START followed by an address byte is received.	■ Must be cleared by software.
STO	■ A STOP is detected while addressed as a slave. ■ Arbitration is lost due to a detected STOP.	■ A pending STOP is generated.
ACKRQ	■ A byte has been received and an ACK response value is needed (only when hardware ACK is not enabled).	■ After each ACK cycle.
ARBLOST	■ A repeated START is detected as a MASTER when STA is low (unwanted repeated START). ■ SCL is sensed low while attempting to generate a STOP or repeated START condition. ■ SDA is sensed low while transmitting a 1 (excluding ACK bits).	■ Each time SI is cleared.
ACK	■ The incoming ACK value is low (ACKNOWLEDGE).	■ The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	■ A START has been generated. ■ Lost arbitration. ■ A byte has been transmitted and an ACK/NACK received. ■ A byte has been received. ■ A START or repeated START followed by a slave address + R/W has been received. ■ A STOP has been received.	■ Must be cleared by software.

23.4.3. 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 23.3. 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.

24. 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 “24.1. Enhanced Baud Rate Generation” on page 138). 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).

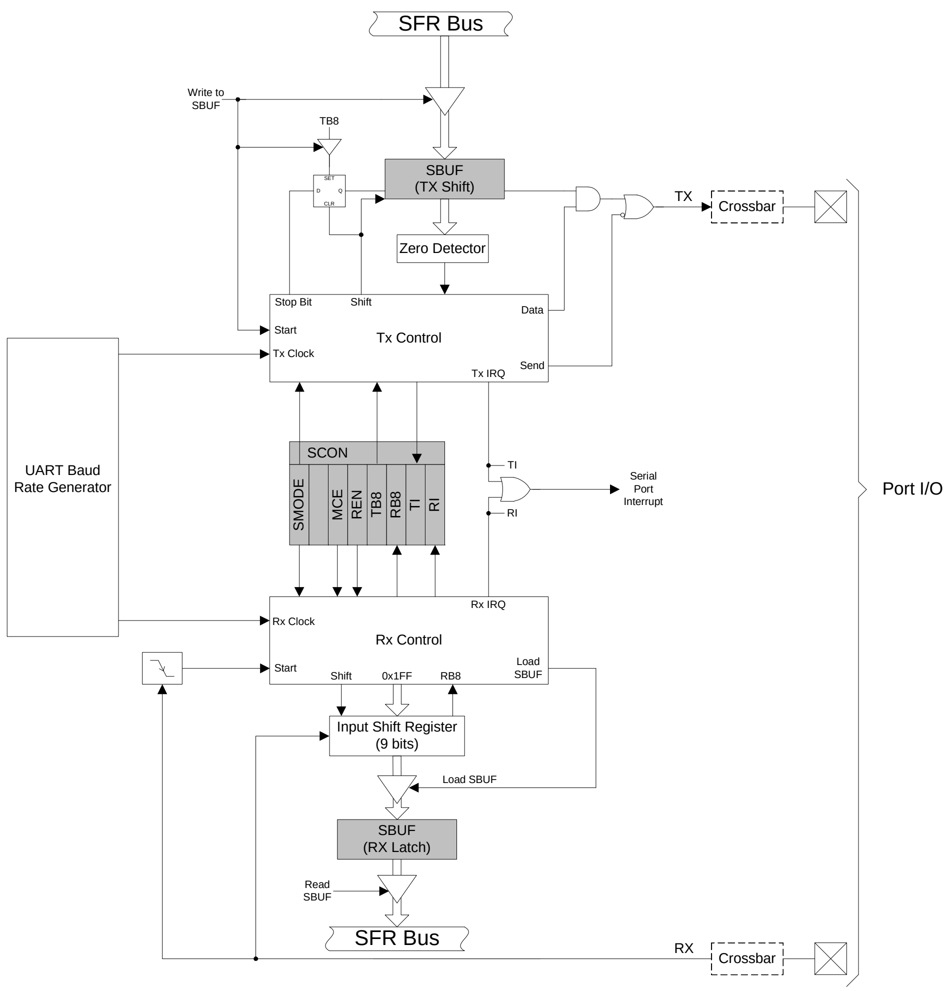

Figure 24.1. UART0 Block Diagram

24.2.2. 9-Bit UART

The 9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 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 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

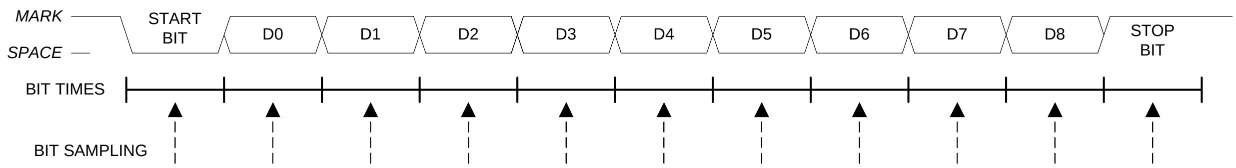

Figure 24.5. 9-Bit UART Timing Diagram

SFR Definition 25.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8A

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 25.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8B

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

26.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and 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 TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

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.

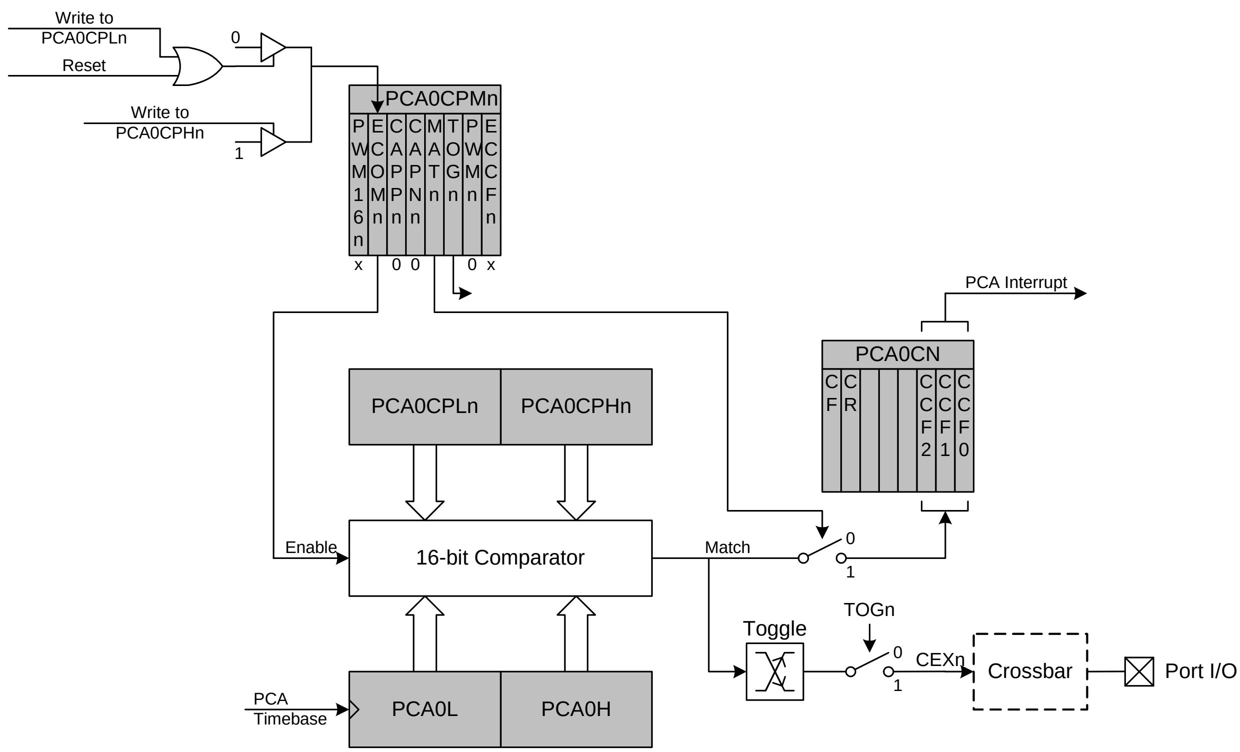

Figure 26.6. PCA High-Speed Output Mode Diagram

26.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 26.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR				CCF2	CCF1	CCF0
Type	R/W	R/W	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD8; Bit-Addressable

Bit	Name	Function
7	CF	**PCA Counter/Timer Overflow Flag.** Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA Counter/Timer Run Control.** This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled 1: PCA Counter/Timer enabled.
5:3	Unused	Unused. Read = 000b, Write = Don't care.
2	CCF2	**PCA Module 2 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	**PCA Module 1 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	**PCA Module 0 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

27. C2 Interface

C8051T600/1/2/3/4/5/6 devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow EPROM programming and in-system debugging with the production part installed in the end application. The C2 interface operates using only two pins: a bi-directional data signal (C2D), and a clock input (C2CK). See the C2 Interface Specification for details on the C2 protocol.

27.1. C2 Interface Registers

The following describes the C2 registers necessary to perform EPROM programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 27.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function		
7:0	C2ADD[7:0]	**Write: C2 Address.** Selects the target Data register for C2 Data Read and Data Write commands according to the following list.		
		Address	**Name**	**Description**
		0x00	DEVICEID	Selects the Device ID Register (read only)
		0x01	REVID	Selects the Revision ID Register (read only)
		0x02	DEVCTL	Selects the C2 Device Control Register
		0xDF	EPCTL	Selects the C2 EPROM Programming Control Register
		0xBF	EPDAT	Selects the C2 EPROM Data Register
		0xB7	EPSTAT	Selects the C2 EPROM Status Register
		0xAF	EPADDRH	Selects the C2 EPROM Address High Byte Register
		0xAE	EPADDRL	Selects the C2 EPROM Address Low Byte Register
		0xA9	CRC0	Selects the CRC0 Register
		0xAA	CRC1	Selects the CRC1 Register
		0xAB	CRC2	Selects the CRC2 Register
		0xAC	CRC3	Selects the CRC3 Register
		Read: C2 Status Returns status information on the current programming operation. When the MSB (bit 7) is set to ‘1’, a read or write operation is in progress. All other bits can be ignored by the programming tools.		

C2 Register Definition 27.4. DEVCTL: C2 Device Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	DEVCTL[7:0]	**Device Control Register.** This register is used to halt the device for EPROM operations via the C2 interface. Refer to the EPROM chapter for more information.

C2 Register Definition 27.5. EPCTL: EPROM Programming Control Register

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

C2 Address: 0xDF

Bit	Name	Function
7:0	EPCTL[7:0]	**EPROM Programming Control Register.** This register is used to enable EPROM programming via the C2 interface. Refer to the EPROM chapter for more information.

C2 Register Definition 27.6. EPDAT: C2 EPROM Data

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

C2 Address: 0xBF

Bit	Name	Function
7:0	EPDAT[7:0]	**C2 EPROM Data Register.** This register is used to pass EPROM data during C2 EPROM operations.

C2 Register Definition 27.7. EPSTAT: C2 EPROM Status

Bit	7	6	5	4	3	2	1	0
Name	WRLOCK	RDLOCK						ERROR
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xB7

Bit	Name	Function
7	WRLOCK	**Write Lock Indicator.** Set to 1 if EPADDR currently points to a write-locked address.
6	RDLOCK	**Read Lock Indicator.** Set to 1 if EPADDR currently points to a read-locked address.
5:1	Unused	Unused. Read = Varies; Write = Don’t Care.
0	ERROR	**Error Indicator.** Set to 1 if last EPROM read or write operation failed due to a security restriction.

C2 Register Definition 27.8. EPADDRH: C2 EPROM Address High Byte

Bit	7	6	5	4	3	2	1	0
Name	EPADDR[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xAF

Bit	Name	Function
7:0	EPADDR[15:8]	**C2 EPROM Address High Byte.** This register is used to set the EPROM address location during C2 EPROM operations.

C2 Register Definition 27.9. EPADDRL: C2 EPROM Address Low Byte

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

C2 Address: 0xAE

Bit	Name	Function
7:0	EPADDR[15:8]	**C2 EPROM Address Low Byte.** This register is used to set the EPROM address location during C2 EPROM operations.