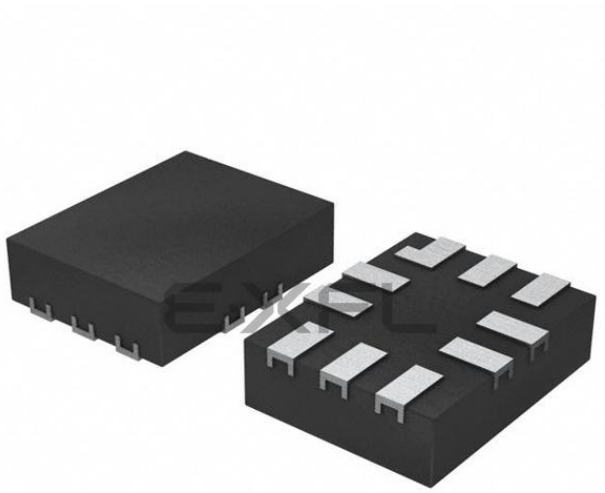

Welcome to E-XFL.COM

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	6
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 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-WFQFN
Supplier Device Package	10-QFN (2x2)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t606-zm

2. Ordering Information

Table 2.1. Product Selection Guide

Part Number	MIPS (Peak)	OTP EPROM (Bytes)	RAM (Bytes)	Calibrated Internal Oscillator	SMBus/I²C	UART	Timers (16-bit)	Programmable Counter Array	Digital Port I/Os	10-bit 500ksps ADC	Temperature Sensor	Analog Comparators	Lead-Free (ROHS Compliant)[2]	Package
C8051T600-GM	25	8k[1]	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	QFN-11
C8051T600-GS	25	8k[1]	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	SOIC-14
C8051T601-GM	25	8k[1]	256	Y	Y	Y	3	Y	8	—	—	1	Y	QFN-11
C8051T601-GS	25	8k[1]	256	Y	Y	Y	3	Y	8	—	—	1	Y	SOIC-14
C8051T602-GM	25	4k	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	QFN-11
C8051T602-GS	25	4k	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	SOIC-14
C8051T603-GM	25	4k	256	Y	Y	Y	3	Y	8	—	—	1	Y	QFN-11
C8051T603-GS	25	4k	256	Y	Y	Y	3	Y	8	—	—	1	Y	SOIC-14
C8051T604-GM	25	2k	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	QFN-11
C8051T604-GS	25	2k	256	Y	Y	Y	3	Y	8	Y	Y	1	Y	SOIC-14
C8051T605-GM	25	2k	256	Y	Y	Y	3	Y	8	—	—	1	Y	QFN-11
C8051T605-GS	25	2k	256	Y	Y	Y	3	Y	8	—	—	1	Y	SOIC-14
C8051T606-GM	25	1.5k	128	Y	Y	Y	3	Y	6	—	—	1	Y	QFN-11
C8051T606-GT	25	1.5k	128	Y	Y	Y	3	Y	6	—	—	1	Y	MSOP-10
C8051T606-ZM	25	1.5k	128	Y	Y	Y	3	Y	6	—	—	1	Y	QFN-10

Notes:
1. 512 Bytes Reserved
2. Lead Finish is 100% Matte Tin (Sn)

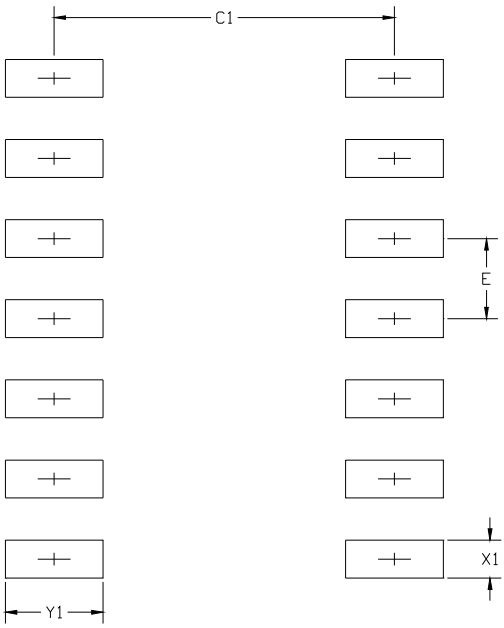

Figure 5.2. SOIC-14 Recommended PCB Land Pattern

Table 5.2. SOIC-14 PCB Land Pattern Dimensions

Dimension	Min	Max	Dimension	Min	Max
C1	5.30	5.40	X1	0.50	0.60
E	1.27 BSC		Y1	1.45	1.55

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

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

Stencil Design

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.

Card Assembly

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

SFR Definition 9.1. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD0LJST	AD08BE	AMP0GN0
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	1

SFR Address = 0xBC

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Period Bits.** SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification table. $AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$ **Note:** If the Memory Power Controller is enabled (MPCE = '1'), AD0SC must be set to at least "00001" for proper ADC operation.
2	AD0LJST	**ADC0 Left Justify Select.** 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified. **Note:** The AD0LJST bit is only valid for 10-bit mode (AD08BE = 0).
1	AD08BE	**8-Bit Mode Enable.** 0: ADC operates in 10-bit mode (normal). 1: ADC operates in 8-bit mode. **Note:** When AD08BE is set to 1, the AD0LJST bit is ignored.
0	AMP0GN0	**ADC Gain Control Bit.** 0: Gain = 0.5 1: Gain = 1

12. Voltage Regulator (REG0)

C8051T600/1/2/3/4/5/6 devices include an internal voltage regulator (REG0) to regulate the internal core supply to 1.8 V from a V_{DD} supply of 1.8 to 3.6 V. Two power-saving modes are built into the regulator to help reduce current consumption in low-power applications. These modes are accessed through the REG0CN register (SFR Definition 12.1). Electrical characteristics for the on-chip regulator are specified in Table 8.5 on page 34.

If an external regulator is used to power the device, the internal regulator may be put into bypass mode using the BYPASS bit. **The internal regulator should never be placed in bypass mode unless an external 1.8 V regulator is used to supply V_{DD}. Doing so could cause permanent damage to the device.**

Under default conditions, when the device enters STOP mode the internal regulator will remain on. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the $\overline{\text{RST}}$ pin or a full power cycle of the device are the only methods of generating a reset.

externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Section “8. Electrical Characteristics” on page 30.

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 13.2). Selecting a longer response time reduces the Comparator supply current.

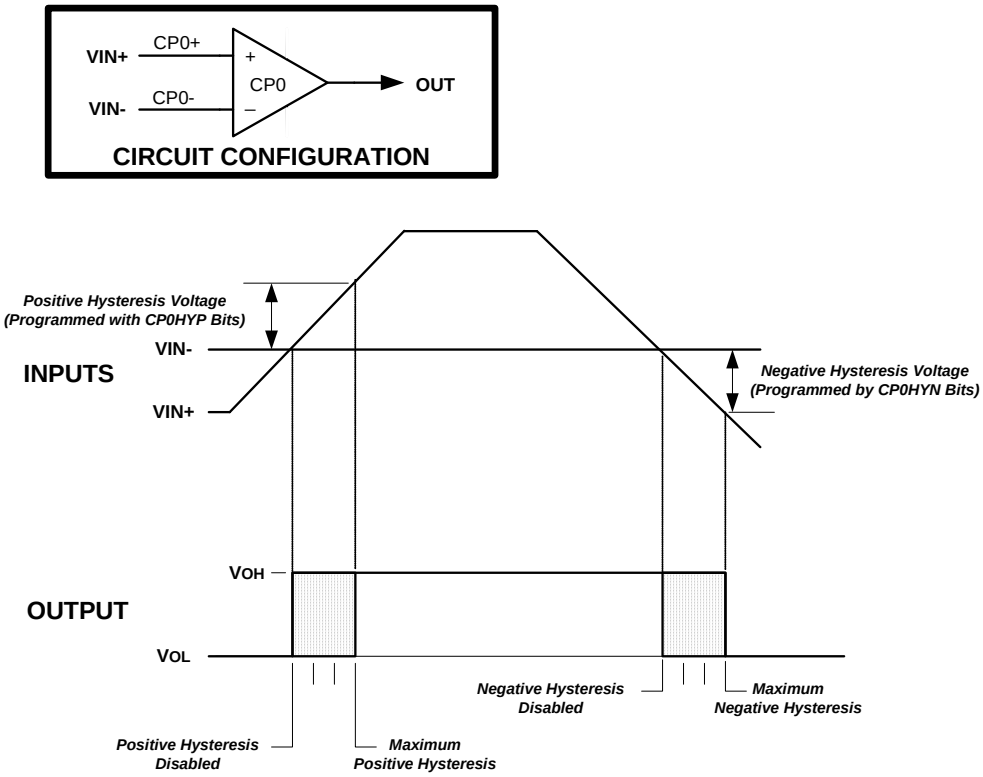

Figure 13.2. Comparator Hysteresis Plot

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

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 13.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 13.2, 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 “17.1. MCU Interrupt Sources and Vectors” on page 81). 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 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.

15. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types.

15.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051T600/1 implements 8192 bytes of this program memory space as in-system, Byte-Programmable EPROM, organized in a contiguous block from addresses 0x0000 to 0x1FFF. Note that 512 bytes (0x1E00 – 0x1FFF) of this memory are reserved for factory use and are not available for user program storage. The C8051T602/3 implements 4096 bytes of EPROM program memory space; the C8051T604/5 implements 2048 bytes of EPROM program memory space, and the C8051T606 implement 1536 bytes of EPROM program memory space. C2 Register Definition 15.1 shows the program memory maps for C8051T600/1/2/3/4/5/6 devices.

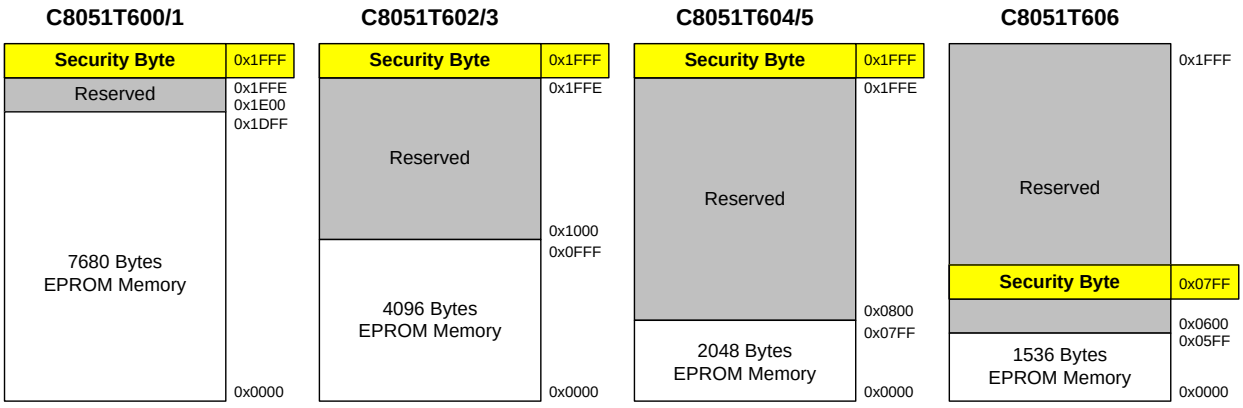

Figure 15.1. Program Memory Map

Program memory is read-only from within firmware. Individual program memory bytes can be read using the MOVC instruction. This facilitates the use of EPROM space for constant storage.

20. EPROM Memory

Electrically programmable read-only memory (EPROM) is included on-chip for program code storage. The EPROM memory can be programmed via the C2 debug and programming interface when a special programming voltage is applied to the V_{PP} pin. Each location in EPROM memory is programmable only once (i.e., non-erasable). Table 8.6 on page 34 shows the EPROM specifications.

20.1. Programming and Reading the EPROM Memory

Reading and writing the EPROM memory is accomplished through the C2 programming and debug interface. When creating hardware to program the EPROM, it is necessary to follow the programming steps listed below. Refer to the “C2 Interface Specification” available at http://www.silabs.com for details on communicating via the C2 interface. Section “27. C2 Interface” on page 178 has information about C2 register addresses for the C8051T600/1/2/3/4/5/6.

20.1.1. EPROM Write Procedure

1. Reset the device using the $\overline{\text{RST}}$ pin.
2. Wait at least 20 µs before sending the first C2 command.
3. Place the device in core reset: Write 0x04 to the DEVCTL register.
4. Set the device to program mode (1st step): Write 0x40 to the EPCTL register.
5. Set the device to program mode (2nd step): Write 0x58 to the EPCTL register.
6. Apply the VPP programming Voltage.
7. Write the first EPROM address for programming to EPADDRH and EPADDRL.
8. Write a data byte to EPDAT. EPADDRH:L will increment by 1 after this write.
9. Use a C2 Address Read command to poll for write completion.
10. (Optional) Check the ERROR bit in register EPSTAT and abort the programming operation if necessary.
11. If programming is not finished, return to Step 8 to write the next address in sequence, or return to Step 7 to program a new address.
12. Remove the VPP programming Voltage.
13. Remove program mode (1st step): Write 0x40 to the EPCTL register.
14. Remove program mode (2nd step): Write 0x00 to the EPCTL register.
15. Reset the device: Write 0x02 and then 0x00 to the DEVCTL register.

Important Note: There is a finite amount of time which V_{PP} can be applied without damaging the device, which is cumulative over the life of the device. Refer to Table 8.1 on page 30 for the V_{PP} timing specification.

22.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins can be assigned to various analog, digital, and external interrupt functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

22.2.1. Assigning Port I/O Pins to Analog Functions

Table 22.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in XBR0 set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the crossbar. Table 22.1 shows the potential mapping of Port I/O to each analog function.

Table 22.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P0.7	AMX0SL, XBR0
Comparator0 Input	P0.0–P0.7	CPT0MX, XBR0
Voltage Reference Input for ADC (VREF)	P0.0	REF0CN, XBR0
External Oscillator in RC or C Mode (EXTCLK)	P0.3	OSCXCN, XBR0

22.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the crossbar for pin assignment; however, some digital functions bypass the crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in XBR0 set to 1.** Table 22.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 22.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SMBus, CP0, CP0A, SYSCLK, PCA0 (CEX0-2 and ECI), T0 or T1.	Any Port pin available for assignment by the crossbar. This includes P0.0 - P0.7 pins which have their XBR0 bit set to 0. **Note:** The crossbar will always assign UART0 pins to P0.4 and P0.5.	XBR1, XBR2
Any pin used for GPIO	P0.0–P0.7	XBR0

SFR Definition 24.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	S0MODE		MCE0	REN0	TB80	RB80	TI0	RI0
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Address = 0x98; Bit-Addressable

Bit	Name	Function
7	S0MODE	**Serial Port 0 Operation Mode.** Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate. 1: 9-bit UART with Variable Baud Rate.
6	Unused	Unused. Read = 1b, Write = Don't Care.
5	MCE0	**Multiprocessor Communication Enable.** The function of this bit is dependent on the Serial Port 0 Operation Mode: **Mode 0: Checks for valid stop bit.** 0: Logic level of stop bit is ignored. 1: RI0 will only be activated if stop bit is logic level 1. **Mode 1: Multiprocessor Communications Enable.** 0: Logic level of ninth bit is ignored. 1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.
4	REN0	**Receive Enable.** 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB80	**Ninth Transmission Bit.** The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB80	**Ninth Receive Bit.** RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI0	**Transmit Interrupt Flag.** Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	**Receive Interrupt Flag.** Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 24.2. 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]	**Serial Data Buffer Bits 7–0 (MSB–LSB).** 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.

SFR Definition 25.1. CKCON: Clock Control

Bit	7	6	5	4	3	2	1	0
Name		T2MH	T2ML	T1M	T0M		SCA[1:0]	
Type	R	R/W	R/W	R/W	R/W	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8E

Bit	Name	Function
7	Unused	Unused. Read = 0b, Write = Don't Care
6	T2MH	**Timer 2 High Byte Clock Select.** Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
5	T2ML	**Timer 2 Low Byte Clock Select.** Selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 low byte uses the system clock.
4	T1M	**Timer 1 Clock Select.** Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale bits SCA[1:0]. 1: Timer 1 uses the system clock.
3	T0M	**Timer 0 Clock Select.** Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale bits SCA[1:0]. 1: Counter/Timer 0 uses the system clock.
2	Unused	Unused. Read = 0b, Write = Don't Care
1:0	SCA[1:0]	**Timer 0/1 Prescale Bits.** These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)

SFR Definition 25.2. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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 = 0x88; Bit-Addressable

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured /INT1 interrupt will be edge or level sensitive. /INT1 is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 17.5). 0: /INT1 is level triggered. 1: /INT1 is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 17.5). 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

SFR Definition 25.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN		T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 low byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	Unused	Unused. Read = 0b; Write = Don't Care
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Unused. Read = 0b; Write = Don't Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

26.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. **Reading the PCA0L register first guarantees an accurate reading of the entire 16-bit PCA0 counter.** Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 26.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle Mode.

Table 26.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8*
1	1	x	Reserved
Note: External oscillator source divided by 8 is synchronized with the system clock.			

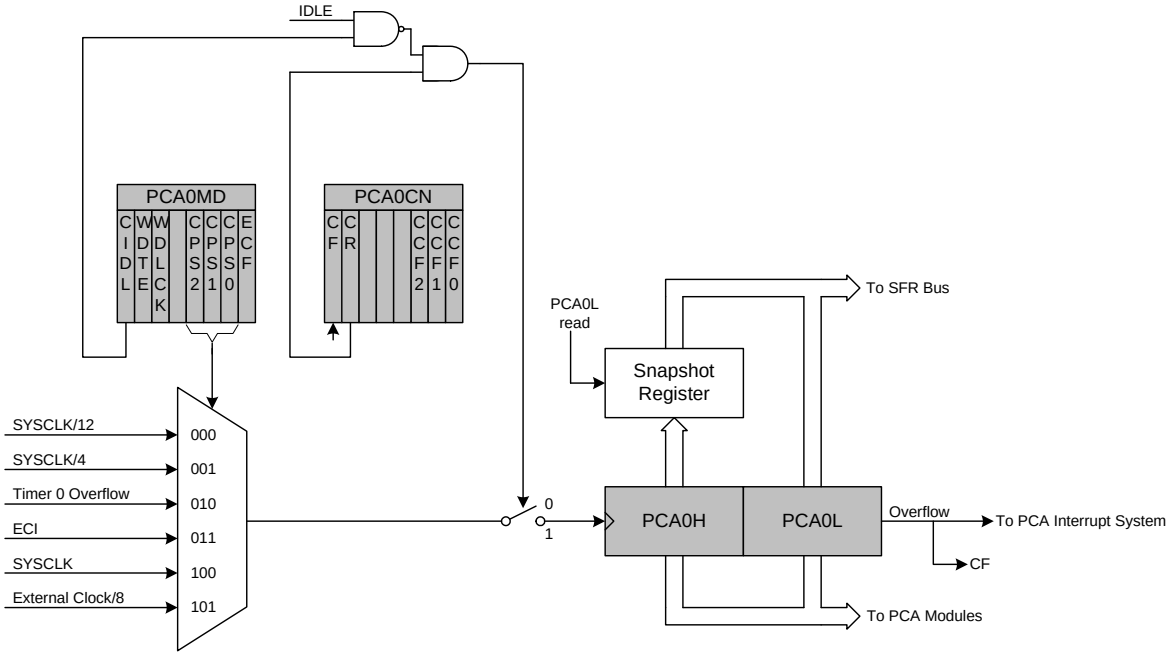

Figure 26.2. PCA Counter/Timer Block Diagram

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.

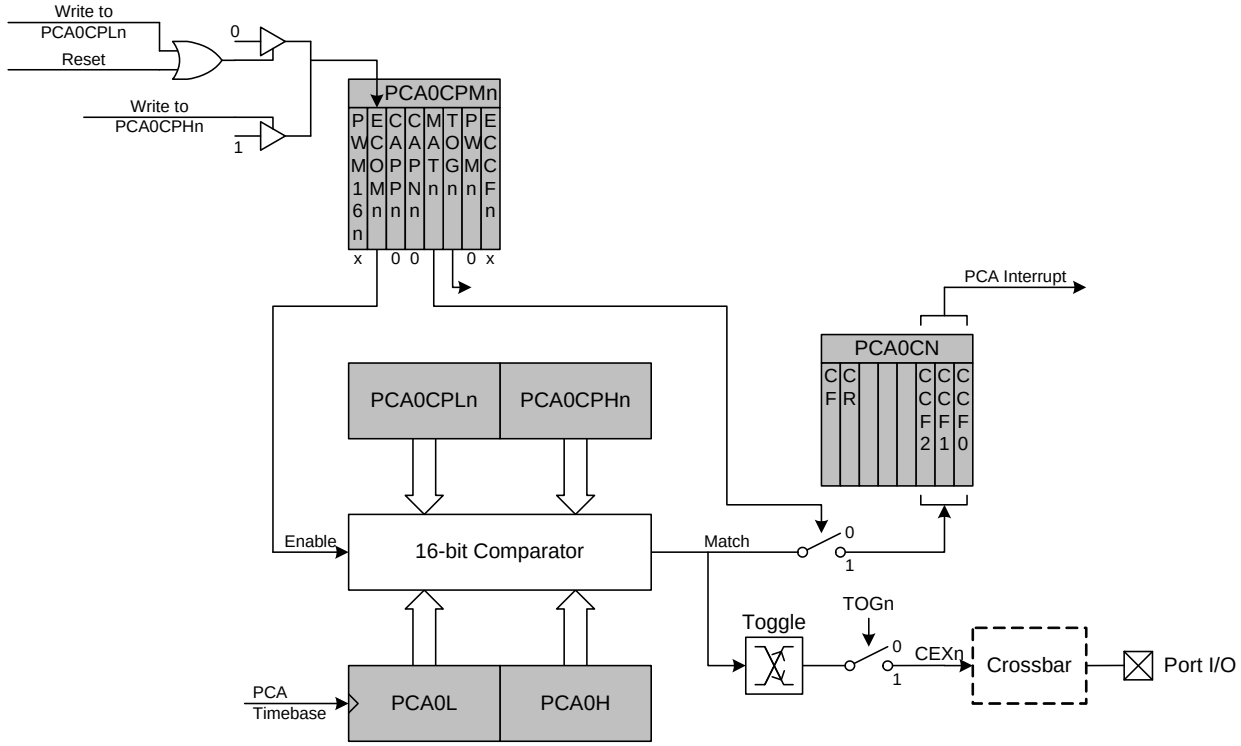

Figure 26.6. PCA High-Speed Output Mode Diagram

SFR Definition 26.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS[2:0]			ECF
Type	R/W	R/W	R/W	R	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	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.		

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.

27.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and EPROM programming functions may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (normally $\overline{\text{RST}}$) and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application when performing debug functions. These external resistors are not necessary for production boards. A typical isolation configuration is shown in Figure 27.1.

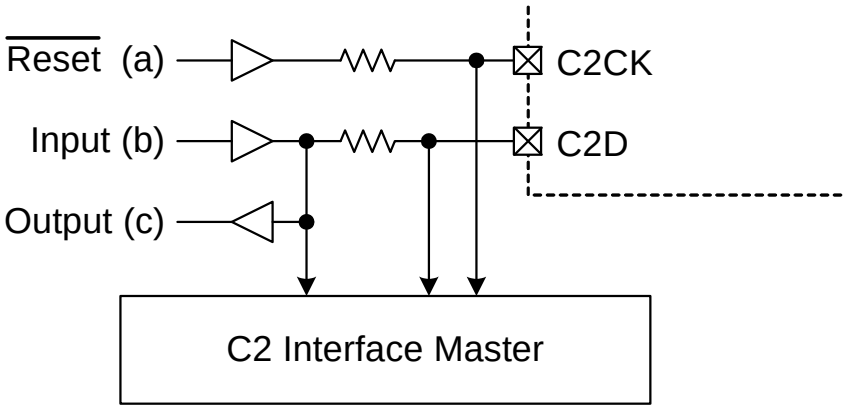

Figure 27.1. Typical C2 Pin Sharing

The configuration in Figure 27.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The $\overline{\text{RST}}$ pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.