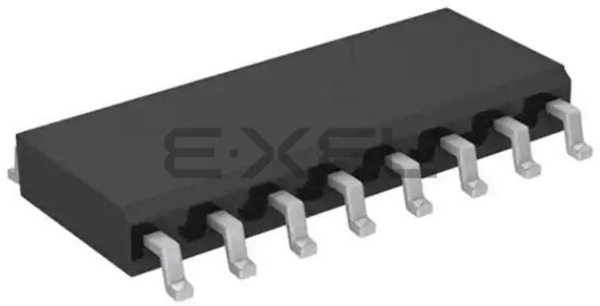

Welcome to E-XFL.COM

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f862-c-is

Table of Contents

Table 1.2. Power Consumption (Continued)

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
ADC0 Burst Mode, 10-bit single conversions, internal reference, Low power bias settings	I_{ADC}	200 ksps, V_{DD} = 3.0 V	—	490	—	µA
		100 ksps, V_{DD} = 3.0 V	—	245	—	µA
		10 ksps, V_{DD} = 3.0 V	—	23	—	µA
ADC0 Burst Mode, 12-bit single conversions, external reference	I_{ADC}	100 ksps, V_{DD} = 3.0 V	—	530	—	µA
		50 ksps, V_{DD} = 3.0 V	—	265	—	µA
		10 ksps, V_{DD} = 3.0 V	—	53	—	µA
ADC0 Burst Mode, 12-bit single conversions, internal reference	I_{ADC}	100 ksps, V_{DD} = 3.0 V, Normal bias	—	950	—	µA
		50 ksps, V_{DD} = 3.0 V, Low power bias	—	420	—	µA
		10 ksps, V_{DD} = 3.0 V, Low power bias	—	85	—	µA
Internal ADC0 Reference, Always-on[5]	I_{IREF}	Normal Power Mode	—	680	790	µA
		Low Power Mode	—	160	210	µA
Temperature Sensor	I_{TSENSE}		—	75	120	µA
Comparator 0 (CMP0), Comparator 1 (CMP1)	I_{CMP}	CPnMD = 11	—	0.5	—	µA
		CPnMD = 10	—	3	—	µA
		CPnMD = 01	—	10	—	µA
		CPnMD = 00	—	25	—	µA
Voltage Supply Monitor (VMON0)	I_{VMON}		—	15	20	µA

Notes:

1. Currents are additive. For example, where I_{DD} is specified and the mode is not mutually exclusive, enabling the functions increases supply current by the specified amount.
2. Includes supply current from internal regulator, supply monitor, and High Frequency Oscillator.
3. Includes supply current from internal regulator, supply monitor, and Low Frequency Oscillator.
4. ADC0 always-on power excludes internal reference supply current.
5. The internal reference is enabled as-needed when operating the ADC in burst mode to save power.

Table 1.11. Port I/O

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Output High Voltage (High Drive)	V_{OH}	I_{OH} = –3 mA	V_{DD} – 0.7	—	—	V
Output Low Voltage (High Drive)	V_{OL}	I_{OL} = 8.5 mA	—	—	0.6	V
Output High Voltage (Low Drive)	V_{OH}	I_{OH} = –1 mA	V_{DD} – 0.7	—	—	V
Output Low Voltage (Low Drive)	V_{OL}	I_{OL} = 1.4 mA	—	—	0.6	V
Input High Voltage	V_{IH}		V_{DD} – 0.6	—	—	V
Input Low Voltage	V_{IL}		—	—	0.6	V
Pin Capacitance	C_{IO}		—	7	—	pF
Weak Pull-Up Current (V_{IN} = 0 V)	I_{PU}	V_{DD} = 3.6	–30	–20	–10	µA
Input Leakage (Pullups off or Analog)	I_{LK}	GND ≤ V_{IN} ≤ V_{DD}	–1.1	—	1.1	µA
Input Leakage Current with V_{IN} above V_{DD}	I_{LK}	V_{DD} < V_{IN} < V_{DD}+2.0 V	0	5	150	µA

- Byte-level bit reversal.
- Automatic CRC of flash contents on one or more 256-byte blocks.
- Initial seed selection of 0x0000 or 0xFFFF.

Table 3.2. Pin Definitions for C8051F850/1/2/3/4/5-GM and C8051F850/1/2/3/4/5-IM

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P1.0	Standard I/O	14	Yes	P1MAT.0	ADC0.8 CP1P.0 CP1N.0
P1.1	Standard I/O	13	Yes	P1MAT.1	ADC0.9 CP1P.1 CP1N.1
P1.2	Standard I/O	11	Yes	P1MAT.2	ADC0.10 CP1P.2 CP1N.2
P1.3	Standard I/O	10	Yes	P1MAT.3	ADC0.11 CP1P.3 CP1N.3
P1.4	Standard I/O	9	Yes	P1MAT.4	ADC0.12 CP1P.4 CP1N.4
P1.5	Standard I/O	8	Yes	P1MAT.5	ADC0.13 CP1P.5 CP1N.5
P1.6	Standard I/O	7	Yes	P1MAT.6	ADC0.14 CP1P.6 CP1N.6
P2.0 / C2D	Standard I/O / C2 Debug Data	6			

Table 12.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flags	Bit addressable?	Cleared by HW?	Enable Flag
Reset	0x0000	Top	None	N/A	N/A	Always Enabled
External Interrupt 0 ($\overline{INT0}$)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)
External Interrupt 1 ($\overline{INT1}$)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)
UART0	0x0023	4	RI (SCON0.0) TI (SCON0.1)	Y	N	ES0 (IE.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)
Port Match	0x0043	8	None	N/A	N/A	EMAT (EIE1.1)
ADC0 Window Compare	0x004B	9	ADWINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)
ADC0 Conversion Complete	0x0053	10	ADINT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n) COVF (PCA0PWM.6)	Y	N	EPCA0 (EIE1.4)
Comparator0	0x0063	12	CPFIF (CPT0CN.4) CPRIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)
Comparator1	0x006B	13	CPFIF (CPT1CN.4) CPRIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)

In Burst Mode, tracking is determined by the settings in ADPWR and ADTK. Settling time requirements may need adjustment in some applications. Refer to “14.2.4. Settling Time Requirements” on page 90 for more details.

Notes:

- Setting ADTM to 1 will insert an additional 4 SAR clocks of tracking before each conversion, regardless of the settings of ADPWR and ADTK.
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals. The ADC will ignore convert start signals which arrive before a burst is finished.

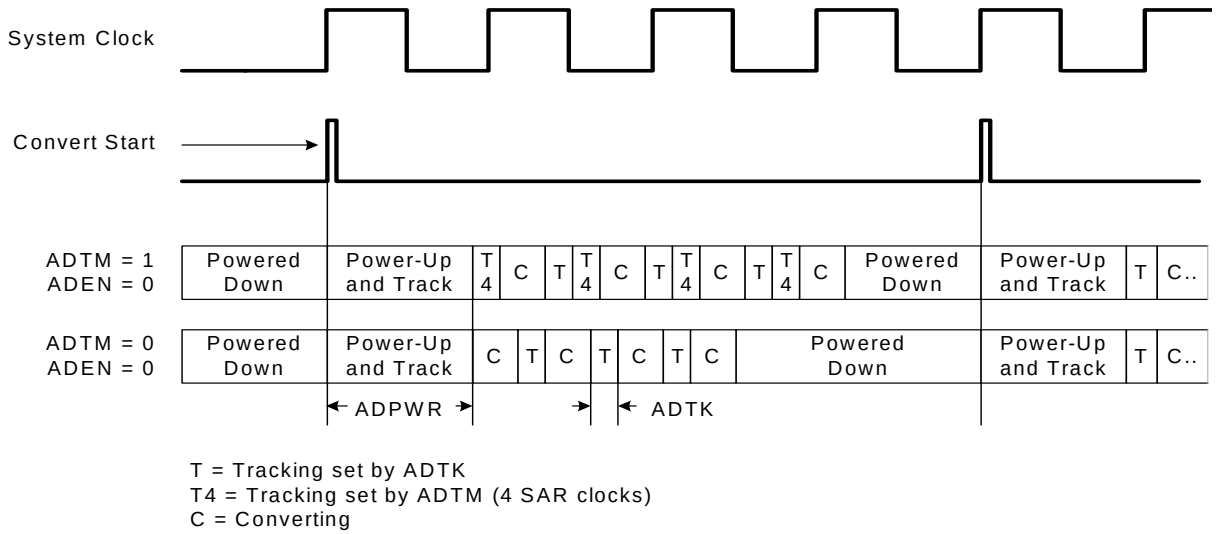

Figure 14.3. Burst Mode Tracking Example with Repeat Count Set to 4

14.2.4. Settling Time Requirements

A minimum amount of tracking time is required before each conversion can be performed, to allow the sampling capacitor voltage to settle. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that when ADTM is set to 1, four SAR clocks are used for tracking at the start of every conversion. Large external source impedance will increase the required tracking time.

Figure 14.4 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 14.1. When measuring any internal source, R_{TOTAL} reduces to R_{MUX}. See the electrical specification tables for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 14.1. ADC0 Settling Time Requirements

Where:

SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)

t is the required settling time in seconds

R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.

16.7. Clock Selection Control Registers

Register 16.3. CLKSEL: Clock Select

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CLKDIV			Reserved		CLKSL	
Type	R	RW			R		RW	
Reset	0	0	1	1	0	0	0	0
SFR Address: 0xA9								

Table 16.3. CLKSEL Register Bit Descriptions

Bit	Name	Function
7	Reserved	Must write reset value.
6:4	CLKDIV	**Clock Source Divider.** This field controls the divider applied to the clock source selected by CLKSL. The output of this divider is the system clock (SYSCLK). 000: SYSCLK is equal to selected clock source divided by 1. 001: SYSCLK is equal to selected clock source divided by 2. 010: SYSCLK is equal to selected clock source divided by 4. 011: SYSCLK is equal to selected clock source divided by 8. 100: SYSCLK is equal to selected clock source divided by 16. 101: SYSCLK is equal to selected clock source divided by 32. 110: SYSCLK is equal to selected clock source divided by 64. 111: SYSCLK is equal to selected clock source divided by 128.
3:2	Reserved	Must write reset value.
1:0	CLKSL	**Clock Source Select.** Selects the system clock source. 00: Clock derived from the Internal High-Frequency Oscillator. 01: Clock derived from the External Oscillator circuit. 10: Clock derived from the Internal Low-Frequency Oscillator. 11: Reserved.

18.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRCVAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRCINIT).

18.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks read from flash. The following steps can be used to automatically perform a CRC on flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit to 1 in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRCCNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes. See the note in the CRC0CN register definition for more information on how to properly initiate a CRC calculation.
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result.

18.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRCPNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

18.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 18.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

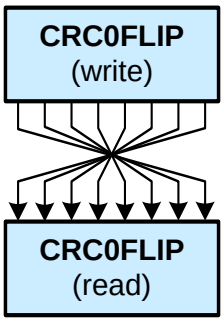

Figure 18.2. Bit Reversal

19.1. External Interrupt Control Registers

Register 19.1. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL			IN0PL	IN0SL		
Type	RW	RW			RW	RW		
Reset	0	0	0	0	0	0	0	1
SFR Address: 0xE4								

Table 19.1. IT01CF Register Bit Descriptions

Bit	Name	Function
7	IN1PL	**INT1 Polarity.** 0: INT1 input is active low. 1: INT1 input is active high.
6:4	IN1SL	**INT1 Port Pin Selection Bits.** These bits select which Port pin is assigned to INT1. This pin assignment is independent of the Crossbar; INT1 will monitor the assigned port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**INT0 Polarity.** 0: INT0 input is active low. 1: INT0 input is active high.

20.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte of the 16-bit counter/timer and PCA0L is the low byte. 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 20.1.

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 20.1. 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	0	Low frequency oscillator divided by 8*
1	1	1	Reserved

***Note:** Synchronized with the system clock.

20.2. PCA0 Interrupt Sources

The PCA0 module shares one interrupt vector among all of its modules. There are are several event flags that can be used to generate a PCA0 interrupt. They are: the main PCA counter overflow flag (CF), which is set upon a 16-bit overflow of the PCA0 counter, an intermediate overflow flag (COVF), which can be set on an overflow from the 8th - 11th bit of the PCA0 counter, and the individual flags for each PCA channel (CCFn), which are set according to the operation mode of that module. These event flags are always set when the trigger condition occurs. Each of these flags can be individually selected to generate a PCA0 interrupt, using the corresponding interrupt enable flag (ECF for CF, ECOV for COVF, and ECCFn for each CCFn). PCA0 interrupts must be globally enabled before any individual interrupt sources are recognized by the processor. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1.

20.3.3. 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.

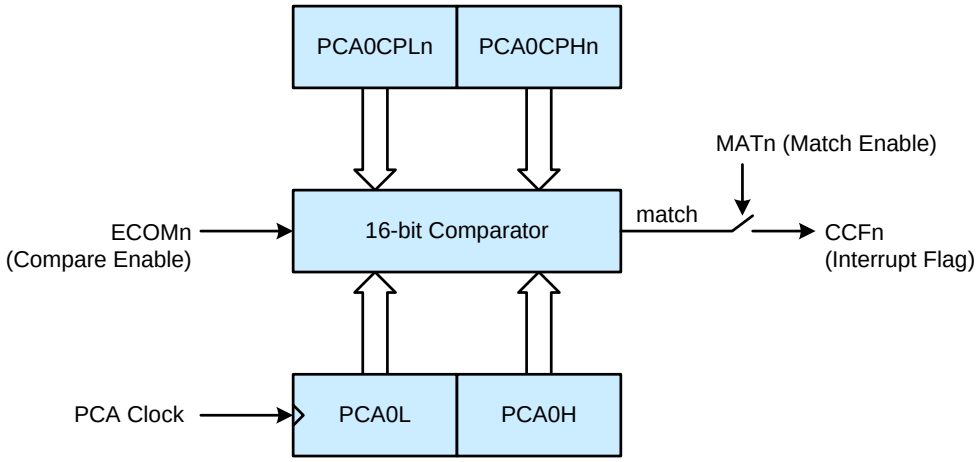

Figure 20.3. PCA Software Timer Mode Diagram

20.4.2. Center Aligned PWM

When configured for center-aligned mode, a module will generate an edge transition at two points for every $2^{(N+1)}$ PCA clock cycles, where N is the selected PWM resolution in bits. In center-aligned mode, these two edges are referred to as the “up” and “down” edges. The polarity at the output pin is selectable, and can be inverted by setting the appropriate channel bit to ‘1’ in the PCA0POL register.

The generated waveforms are centered about the points where the lower N bits of the PCA0 counter are zero. The $(N+1)^{th}$ bit in the PCA0 counter acts as a selection between up and down edges. In 16-bit mode, a special 17th bit is implemented internally for this purpose. At the center point, the (non-inverted) channel output will be low when the $(N+1)^{th}$ bit is ‘0’ and high when the $(N+1)^{th}$ bit is ‘1’, except for cases of 0% and 100% duty cycle. Prior to inversion, an up edge sets the channel to logic high, and a down edge clears the channel to logic low.

Down edges occur when the $(N+1)^{th}$ bit in the PCA0 counter is one, and a logical inversion of the value in the module’s PCA0CPn register matches the main PCA0 counter register for the lowest N bits. For example, with 10-bit PWM, the down edge will occur when the one’s complement of bits 9-0 of the PCA0CPn register match bits 9-0 of the PCA0 counter, and bit 10 of the PCA0 counter is ‘1’.

Up edges occur when the $(N+1)^{th}$ bit in the PCA0 counter is zero, and the lowest N bits of the module’s PCA0CPn register match the value of (PCA0 - 1). For example, with 10-bit PWM, the up edge will occur when bits 9-0 of the PCA0CPn register are one less than bits 9-0 of the PCA0 counter, and bit 10 of the PCA0 counter is ‘0’.

An example of the PWM timing in center-aligned mode for two channels is shown in Figure 20.7. In this example, the CEX0POL and CEX1POL bits are cleared to 0.

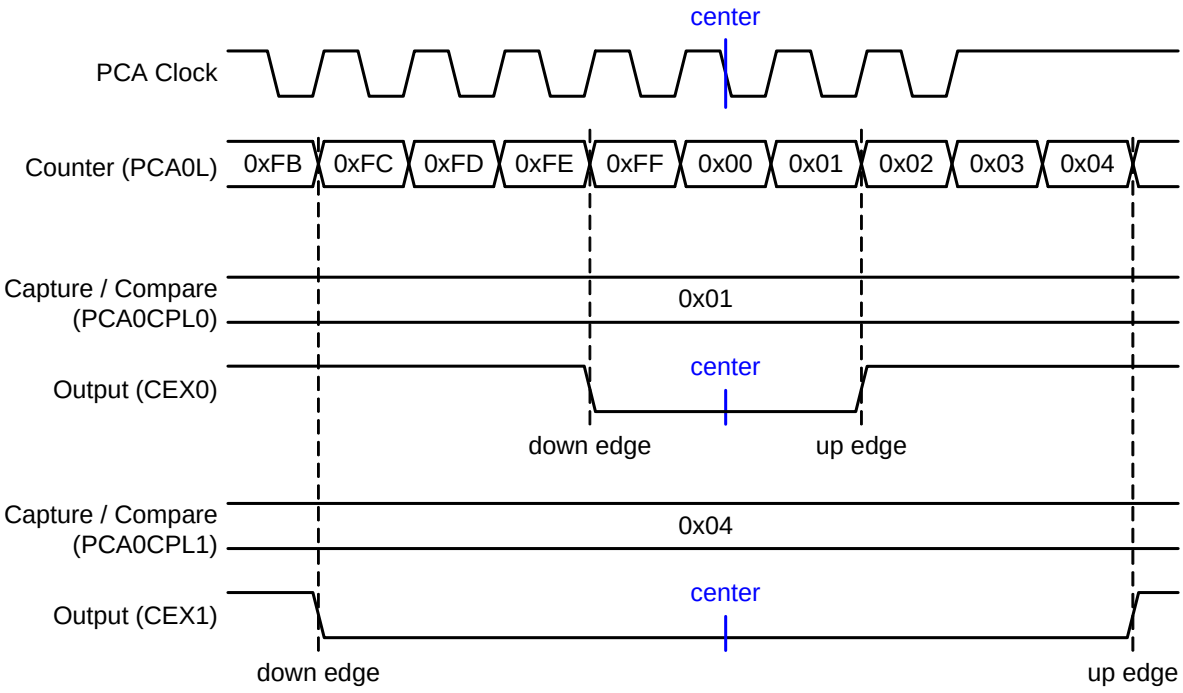

Figure 20.7. Center-Aligned PWM Timing

Register 20.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	Reserved			CPS			ECF
Type	RW	R			RW			RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD9								

Table 20.4. PCA0MD Register Bit Descriptions

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.
6:4	Reserved	Must write reset value.
3:1	CPS	**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). 110: Low frequency oscillator divided by 8. 111: 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.

Register 20.4. PCA0CLR: PCA Comparator Clear Control

Bit	7	6	5	4	3	2	1	0
Name	CPCPOL	Reserved				CPCE2	CPCE1	CPCE0
Type	RW	R				RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x9C								

Table 20.6. PCA0CLR Register Bit Descriptions

Bit	Name	Function
7	CPCPOL	**Comparator Clear Polarity.** Selects the polarity of the comparator result that will clear the PCA channel(s). 0: PCA channel(s) will be cleared when comparator result goes logic low. 1: PCA channel(s) will be cleared when comparator result goes logic high.
6:3	Reserved	Must write reset value.
2	CPCE2	**Comparator Clear Enable for CEX2.** Enables the comparator clear function on PCA channel 2.
1	CPCE1	**Comparator Clear Enable for CEX1.** Enables the comparator clear function on PCA channel 1.
0	CPCE0	**Comparator Clear Enable for CEX0.** Enables the comparator clear function on PCA channel 0.

Register 20.5. PCA0CPM0: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xDA								

Table 20.7. PCA0CPM0 Register Bit Descriptions

Bit	Name	Function
7	PWM16	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	**Comparator Function Enable.** This bit enables the comparator function.
5	CAPP	**Capture Positive Function Enable.** This bit enables the positive edge capture capability.
4	CAPN	**Capture Negative Function Enable.** This bit enables the negative edge capture capability.
3	MAT	**Match Function Enable.** This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF0 bit in the PCA0MD register to be set to logic 1.
2	TOG	**Toggle Function Enable.** This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX0 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX0 pin. 8 to 11-bit PWM is used if PWM16 is cleared; 16-bit mode is used if PWM16 is set to logic 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCF0) interrupt. 0: Disable CCF0 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF0 is set.

Register 21.2. XBR1: Port I/O Crossbar 1

Bit	7	6	5	4	3	2	1	0
Name	Reserved		T2E	T1E	T0E	ECIE	PCA0ME	
Type	R	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xE2								

Table 21.5. XBR1 Register Bit Descriptions

Bit	Name	Function
7:6	Reserved	Must write reset value.
5	T2E	**T2 Enable.** 0: T2 unavailable at Port pin. 1: T2 routed to Port pin.
4	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
3	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
2	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
1:0	PCA0ME	**PCA Module I/O Enable Bits.** 00: All PCA I/O unavailable at Port pins. 01: CEX0 routed to Port pin. 10: CEX0, CEX1 routed to Port pins. 11: CEX0, CEX1, CEX2 routed to Port pins.

Register 21.3. XBR2: Port I/O Crossbar 2

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved					
Type	RW	RW	R					
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xE3								

Table 21.6. XBR2 Register Bit Descriptions

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5:0	Reserved	Must write reset value.

25.1.2. 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.

26. Universal Asynchronous Receiver/Transmitter (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 “26.1. Enhanced Baud Rate Generation” on page 289). 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 (TI is set in SCON0), or a data byte has been received (RI 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).

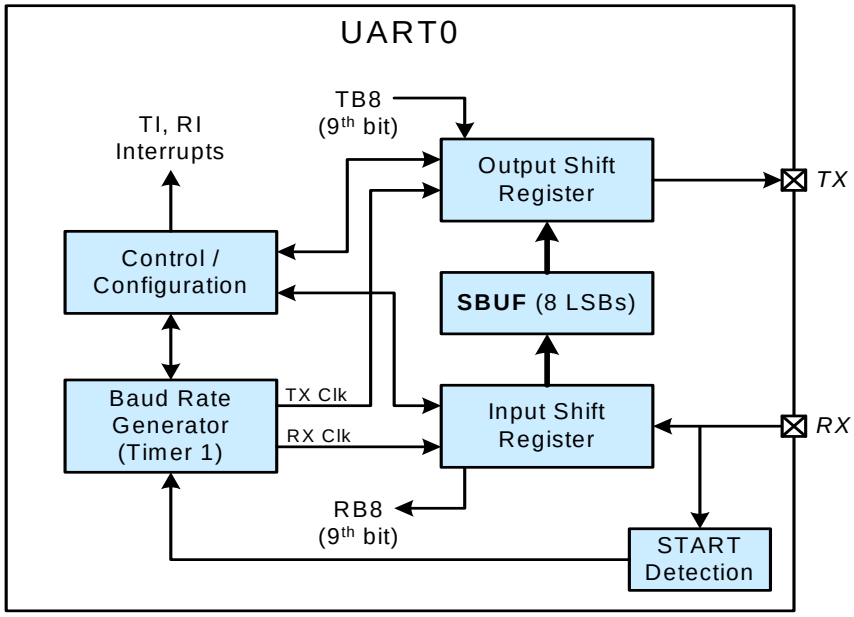

Figure 26.1. UART0 Block Diagram

26.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 26.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.