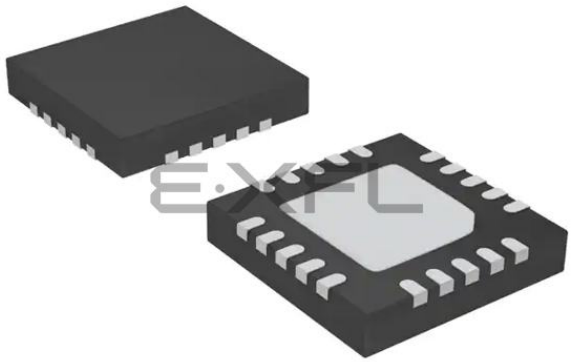

Welcome to E-XFL.COM

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	15
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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f855-c-gm

1. Electrical Specifications

1.1. Electrical Characteristics

All electrical parameters in all tables are specified under the conditions listed in Table 1.1, unless stated otherwise.

Table 1.1. Recommended Operating Conditions

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Operating Supply Voltage on VDD	V_{DD}		2.2	—	3.6	V
System Clock Frequency	f_{SYSCLK}		0	—	25	MHz
Operating Ambient Temperature	T_A	Commercial Grade Devices (-GM, -GS, -GU)	–40	—	85	°C
		Industrial Grade Devices (-IM, -IS, -IU)	–40	—	125	°C
Note: All voltages with respect to GND						

Table 1.2. Power Consumption

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Digital Core Supply Current (–Gx Devices, -40°C to +85°C)						
Normal Mode—Full speed with code executing from flash	I_{DD}	F_{SYSCLK} = 24.5 MHz[2]	—	4.45	4.85	mA
		F_{SYSCLK} = 1.53 MHz[2]	—	915	1150	µA
		F_{SYSCLK} = 80 kHz[3], T_A = 25 °C	—	250	290	µA
		F_{SYSCLK} = 80 kHz[3]	—	250	380	µA
Idle Mode—Core halted with peripherals running	I_{DD}	F_{SYSCLK} = 24.5 MHz[2]	—	2.05	2.3	mA
		F_{SYSCLK} = 1.53 MHz[2]	—	550	700	µA
		F_{SYSCLK} = 80 kHz[3], T_A = 25 °C	—	125	130	µA
		F_{SYSCLK} = 80 kHz[3]	—	125	200	µA
Stop Mode—Core halted and all clocks stopped, Supply monitor off.	I_{DD}	Internal LDO ON, T_A = 25 °C	—	105	120	µA
		Internal LDO ON	—	105	170	µA
		Internal LDO OFF	—	0.2	—	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 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			

9. Special Function Register Memory Map

This section details the special function register memory map for the C8051F85x/86x devices.

Table 9.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	P0MAT	P0MASK	VDM0CN
F0	B	P0MDIN	P1MDIN	EIP1	-	-	PRTDRV	PCA0PWM
E8	ADC0CN0	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	P1MAT	P1MASK	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF	-	EIE1	-
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	CRC0IN	CRC0DAT	ADC0PWR
D0	PSW	REF0CN	CRC0AUTO	CRC0CNT	P0SKIP	P1SKIP	SMB0ADM	SMB0ADR
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	CRC0CN	CRC0FLIP
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	OSCICL
B8	IP	ADC0TK	-	ADC0MX	ADC0CF	ADC0L	ADC0H	CPT1CN
B0	-	OSCLCN	ADC0CN1	ADC0AC	-	DEVICEID	REVID	FLKEY
A8	IE	CLKSEL	CPT1MX	CPT1MD	SMB0TC	DERIVID	-	-
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	-
98	SCON0	SBUF0	-	CPT0CN	PCA0CLR	CPT0MD	PCA0CENT	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	PCA0POL	WDTCN
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	-	-	-	PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

Table 9.2. Special Function Registers

Register	Address	Register Description	Page
ACC	0xE0	Accumulator	122
ADC0AC	0xB3	ADC0 Accumulator Configuration	102
ADC0CF	0xBC	ADC0 Configuration	101
ADC0CN0	0xE8	ADC0 Control 0	99
ADC0CN1	0xB2	ADC0 Control 1	100
ADC0GTH	0xC4	ADC0 Greater-Than High Byte	107
ADC0GTL	0xC3	ADC0 Greater-Than Low Byte	108
ADC0H	0xBE	ADC0 Data Word High Byte	105

12. Interrupts

The C8051F85x/86x includes an extended interrupt system supporting multiple interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE and EIE1). However, interrupts must first be globally enabled by setting the EA bit in the IE register to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

12.1. MCU Interrupt Sources and Vectors

The C8051F85x/86x MCUs support interrupt sources for each peripheral on the device. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 12.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

12.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 12.1.

12.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock

14.1. ADC0 Analog Multiplexer

ADC0 on C8051F85x/86x has an analog multiplexer capable of selecting any pin on ports P0 and P1 (up to 16 total), the on-chip temperature sensor, the internal regulated supply, the VDD supply, or GND. ADC0 input channels are selected using the ADC0MX register.

Table 14.1. ADC0 Input Multiplexer Channels

ADC0MX setting	Signal Name	QSOP24 Pin Name	QFN20 Pin Name	SOIC16 Pin Name
00000	ADC0.0	P0.0	P0.0	P0.0
00001	ADC0.1	P0.1	P0.1	P0.1
00010	ADC0.2	P0.2	P0.2	P0.2
00011	ADC0.3	P0.3	P0.3	P0.3
00100	ADC0.4	P0.4	P0.4	P0.4
00101	ADC0.5	P0.5	P0.5	P0.5
00110	ADC0.6	P0.6	P0.6	P0.6
00111	ADC0.7	P0.7	P0.7	P0.7
01000	ADC0.8	P1.0	P1.0	P1.0
01001	ADC0.9	P1.1	P1.1	P1.1
01010	ADC0.10	P1.2	P1.2	P1.2
01011	ADC0.11	P1.3	P1.3	P1.3
01100	ADC0.12	P1.4	P1.4	Reserved
01101	ADC0.13	P1.5	P1.5	Reserved
01110	ADC0.14	P1.6	P1.6	Reserved
01111	ADC0.15	P1.7	Reserved	Reserved
10000	Temp Sensor	Internal Temperature Sensor		
10001	LDO	Internal 1.8 V LDO Output		
10010	VDD	VDD Supply Pin		
10011	GND	GND Supply Pin		
10100-11111	None	No connection		

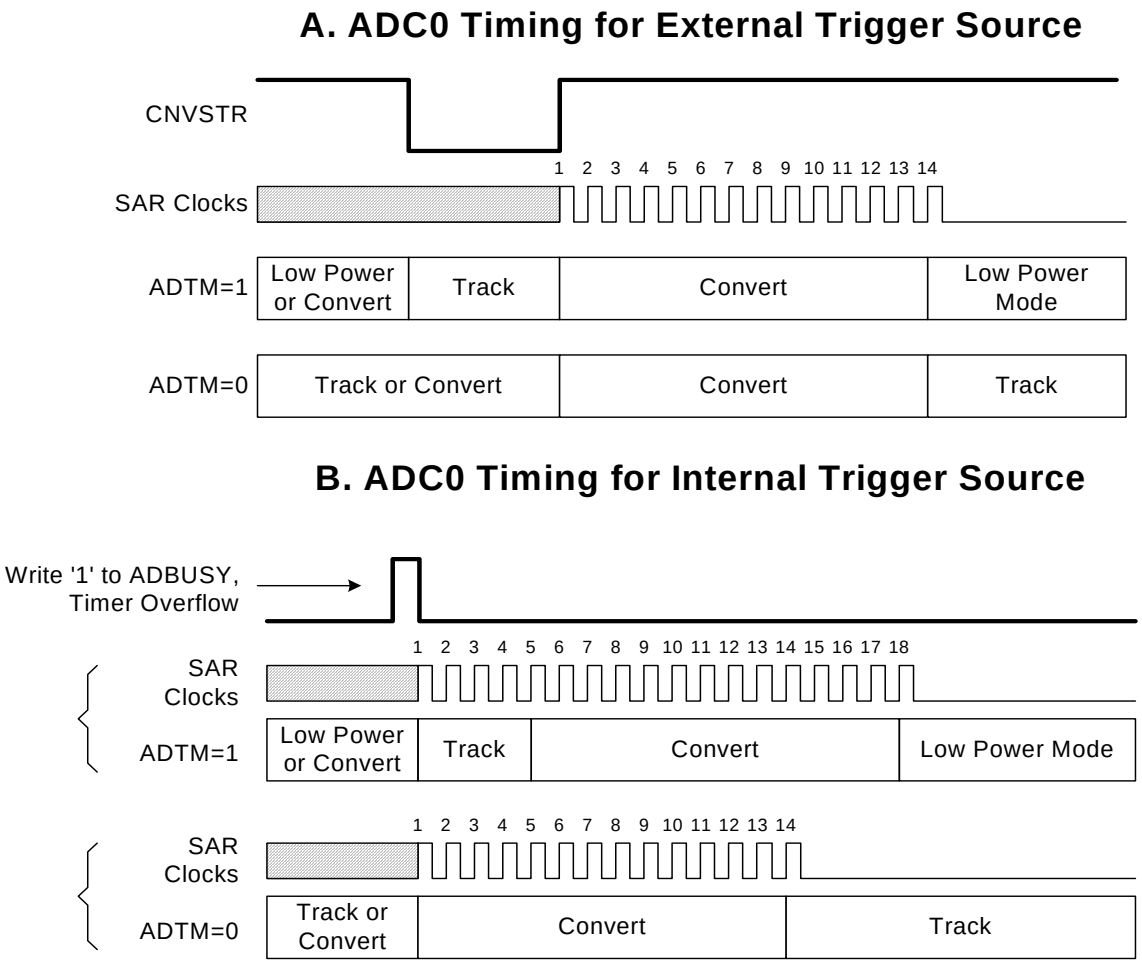

Figure 14.2. 10-Bit ADC Track and Conversion Example Timing (ADBMEN = 0)

14.2.3. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a low power state, accumulates 1, 4, 8, 16, 32, or 64 samples using the internal low-power high-frequency oscillator, then re-enters a low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (e.g. 80 kHz).

Burst Mode is enabled by setting ADBMEN to logic 1. When in Burst Mode, ADEN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If ADEN is set to logic 0, ADC0 is powered down after each burst. If ADEN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the ADPWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 14.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (ADINT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

Table 15.1. CIP-51 Instruction Set Summary (Continued)

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

20. Programmable Counter Array (PCA0)

The Programmable Counter Array (PCA0) provides three channels of enhanced timer and PWM functionality while requiring less CPU intervention than standard counter/timers. The PCA consists of a dedicated 16-bit counter/timer and three 16-bit capture/compare modules. The counter/timer is driven by a programmable timebase that can select between seven sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, low frequency oscillator 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 11-Bit PWM, or 16-Bit PWM. Additionally, all PWM modes support both center and edge-aligned operation. The external oscillator and LFO oscillator clock options allow the PCA to be clocked by an external oscillator or the LFO while the internal oscillator drives the system clock. 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 I/O signals have programmable polarity and Comparator 0 can optionally be used to perform a cycle-by-cycle kill operation on the PCA outputs. A PCA block diagram is shown in Figure 20.1

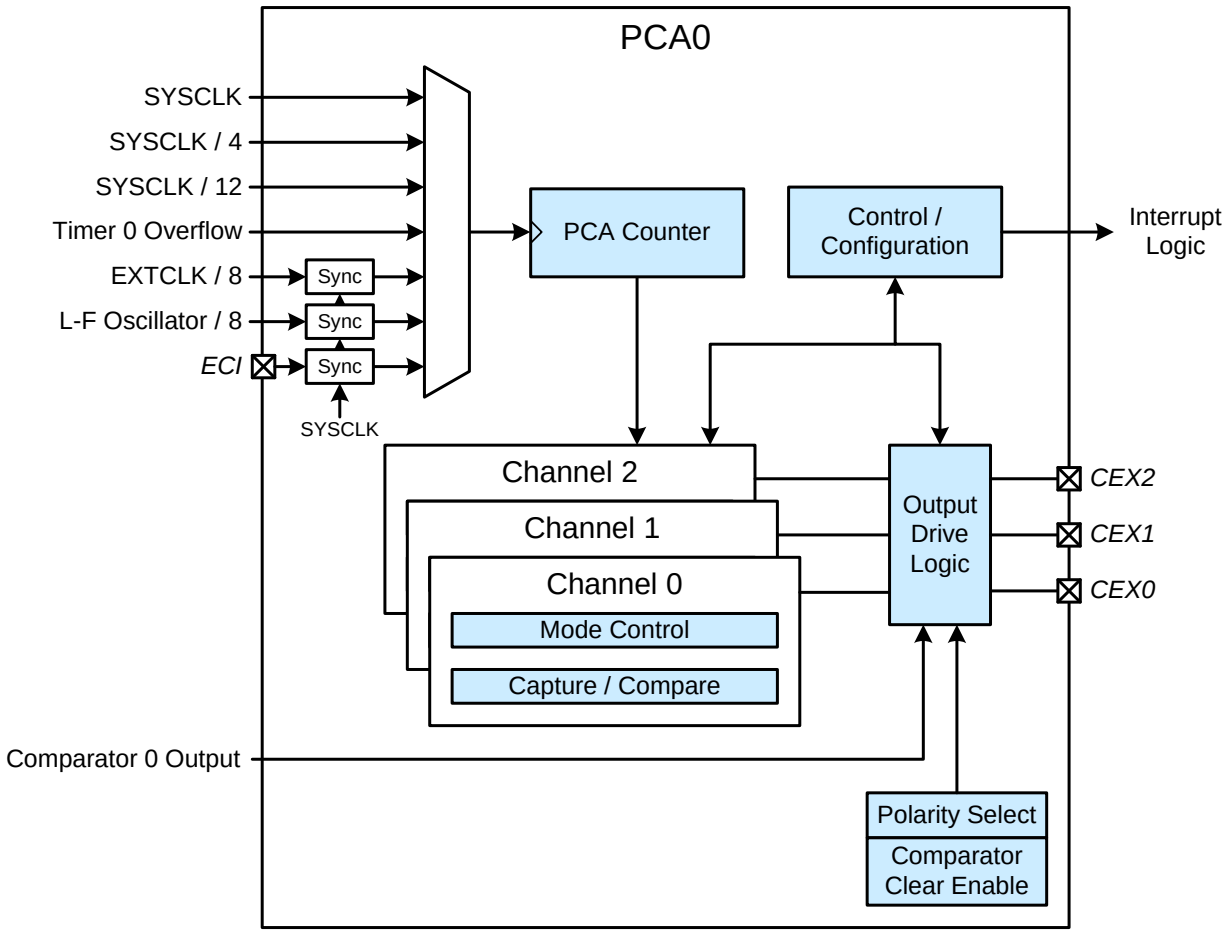

Figure 20.1. PCA0 Block Diagram

20.3.4. 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.

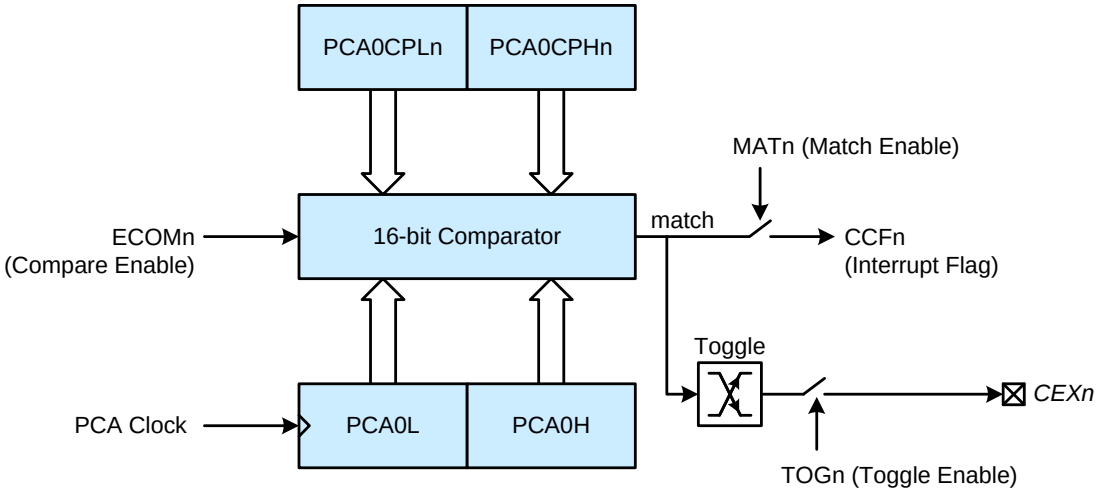

Figure 20.4. PCA High-Speed Output Mode Diagram

disable the comparison, and prevent the match edge from occuring. Note that although the PCA0CPn compare register determines the duty cycle, it is not always appropriate for firmware to update this register directly. See the sections on 8 to 11-bit and 16-bit PWM mode for additional details on adjusting duty cycle in the various modes.

$$\text{Duty Cycle} = \frac{(2^N - \text{PCA0CPn})}{2^N}$$

Equation 20.2. N-bit Edge-Aligned PWM Duty Cycle With CEXnPOL = 0 (N = PWM resolution)

$$\text{Duty Cycle} = \frac{\text{PCA0CPn}}{2^N}$$

Equation 20.3. N-bit Edge-Aligned PWM Duty Cycle With CEXnPOL = 0 (N = PWM resolution)

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.

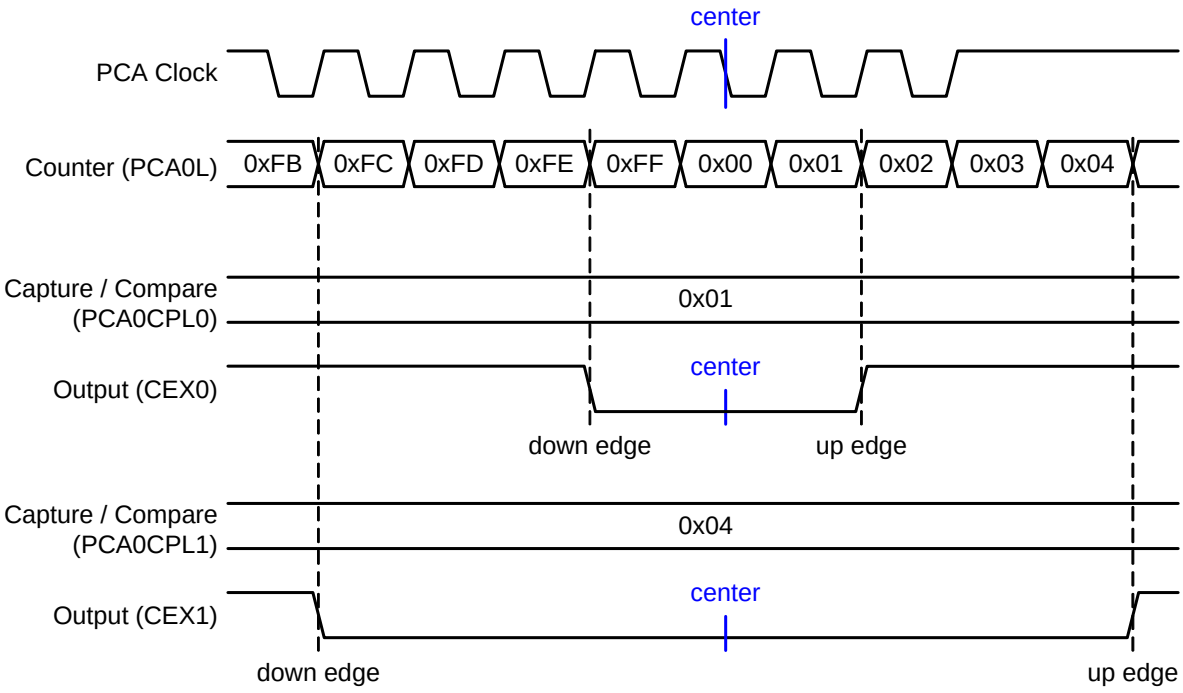

Figure 20.7. Center-Aligned PWM Timing

23. Serial Peripheral Interface (SPI0)

The serial peripheral interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

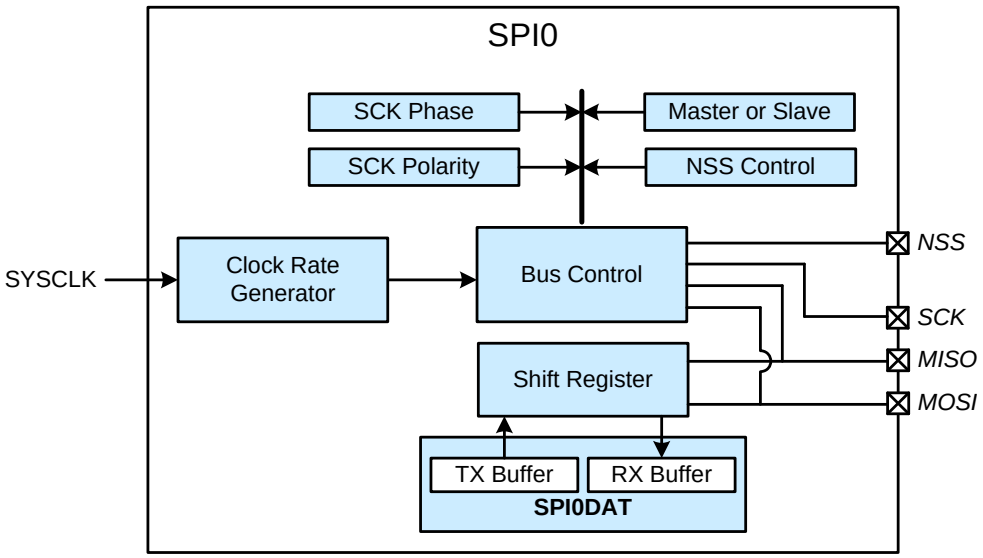

Figure 23.1. SPI0 Block Diagram

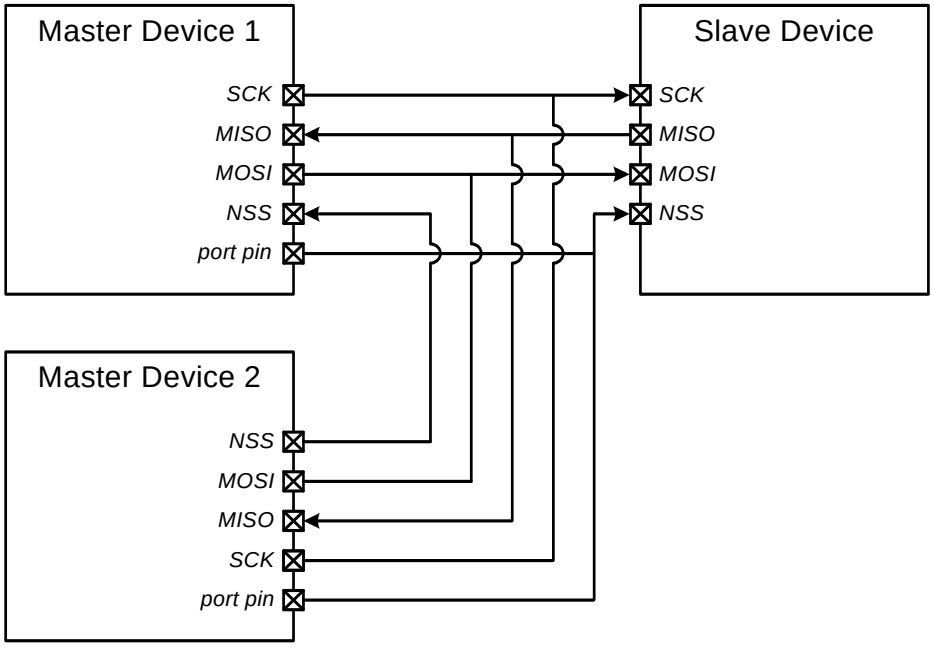

Figure 23.2. Multiple-Master Mode Connection Diagram

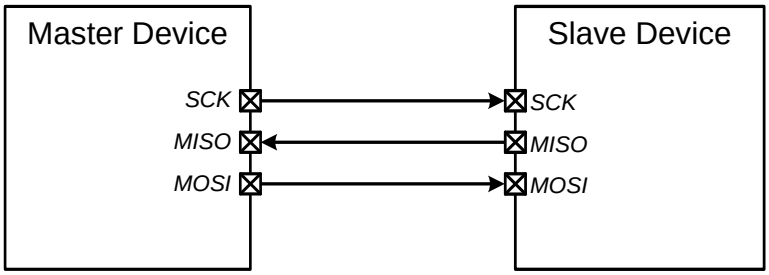

Figure 23.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

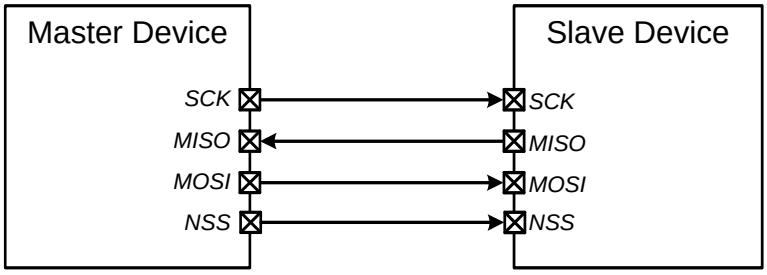

Figure 23.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

Table 24.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

24.4.6. 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.

Register 24.2. SMB0TC: SMBus0 Timing and Pin Control

Bit	7	6	5	4	3	2	1	0
Name	SWAP	Reserved					SDD	
Type	RW	R					RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xAC								

Table 24.8. SMB0TC Register Bit Descriptions

Bit	Name	Function
7	SWAP	**SMBus0 Swap Pins.** This bit swaps the order of the SMBus0 pins on the crossbar. 0: SDA is mapped to the lower-numbered port pin, and SCL is mapped to the higher-numbered port pin. 1: SCL is mapped to the lower-numbered port pin, and SDA is mapped to the higher-numbered port pin.
6:2	Reserved	Must write reset value.
1:0	SDD	**SMBus0 Start Detection Window.** These bits increase the hold time requirement between SDA falling and SCL falling for START detection. 00: No additional hold time window (0-1 SYSCLK). 01: Increase hold time window to 2-3 SYSCLKs. 10: Increase hold time window to 4-5 SYSCLKs. 11: Increase hold time window to 8-9 SYSCLKs.

25.2.3. Capture Mode

Capture mode allows an external input (Timer 2) or the low-frequency oscillator clock (Timer 3) to be measured against the system clock or an external oscillator source. The timer can be clocked from the system clock, the system clock divided by 12, or the external oscillator divided by 8, depending on the TnML, and TnXCLK settings.

Setting TFnCEN to 1 enables Capture Mode. In this mode, TnSPLIT should be set to 0, as the full 16-bit timer is used. Upon a falling edge of the input capture signal, the contents of the timer register (TMRnH:TMRnL) are loaded into the reload registers (TMRnRLH:TMRnRLL) and the TFnH flag is set. By recording the difference between two successive timer capture values, the period of the captured signal can be determined with respect to the selected timer clock.

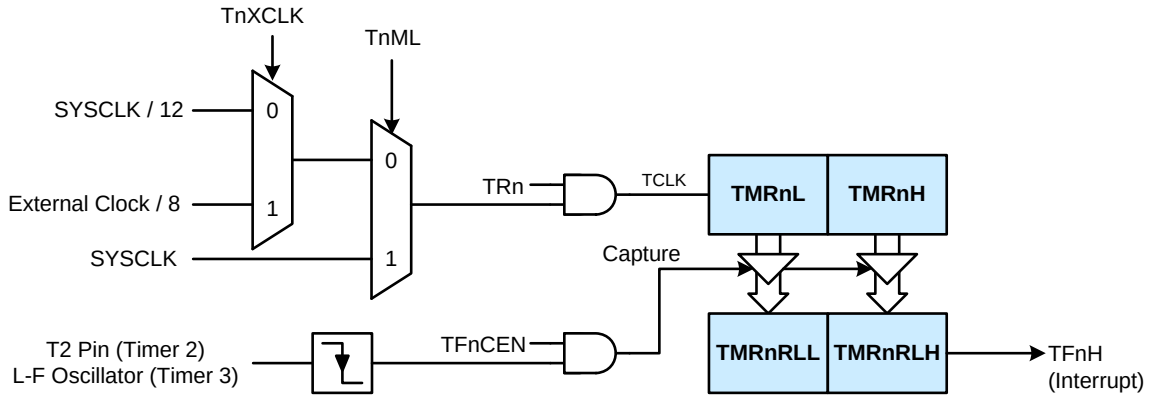

Figure 25.6. Capture Mode Block Diagram

Register 25.15. TMR3RLH: Timer 3 Reload High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x93								

Table 25.17. TMR3RLH Register Bit Descriptions

Bit	Name	Function
7:0	TMR3RLH	**Timer 3 Reload High Byte.** When operating in one of the auto-reload modes, TMR3RLH holds the reload value for the high byte of Timer 3 (TMR3H). When oeprating in capture mode, TMR3RLH is the captured value of TMR3H.

27. Watchdog Timer (WDT0)

The C8051F85x/86x family includes a programmable Watchdog Timer (WDT) running off the low-frequency oscillator. A WDT overflow will force the MCU into the reset state. To prevent the reset, the WDT must be restarted by application software before overflow. If the system experiences a software or hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset.

Following a reset the WDT is automatically enabled and running with the default maximum time interval. If desired the WDT can be disabled by system software or locked on to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

The WDT consists of an internal timer running from the low-frequency oscillator. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. When the WDT is active, the low-frequency oscillator is forced on. All watchdog features are controlled via the Watchdog Timer Control Register (WDTCN).

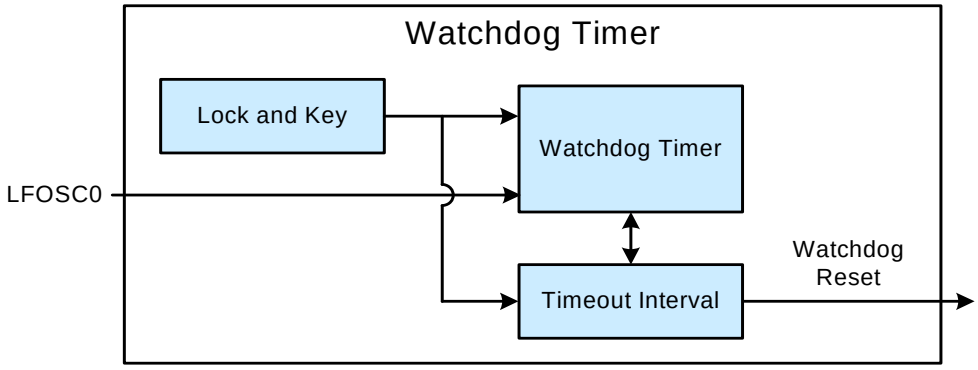

Figure 27.1. Watchdog Timer Block Diagram

29. C2 Interface

C8051F85x/86x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. Details on the C2 protocol can be found in the C2 Interface Specification.

29.1. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and flash programming may be performed. C2CK is shared with the $\overline{\text{RST}}$ pin, while the C2D signal is shared with a port I/O pin. 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 and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 29.1.

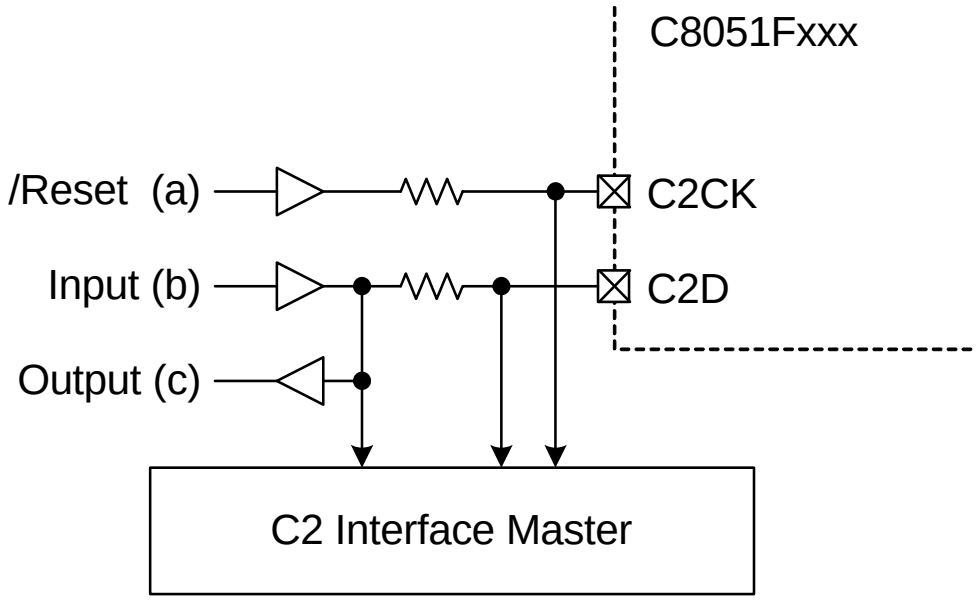

Figure 29.1. Typical C2 Pin Sharing

The configuration in Figure 29.1 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.

Register 29.2. C2DEVID: C2 Device ID

Bit	7	6	5	4	3	2	1	0
Name	C2DEVID							
Type	R							
Reset	0	0	1	1	0	0	0	0
C2 Address: 0x00								

Table 29.2. C2DEVID Register Bit Descriptions

Bit	Name	Function
7:0	C2DEVID	**Device ID.** This read-only register returns the 8-bit device ID: 0x30 (C8051F85x/86x).