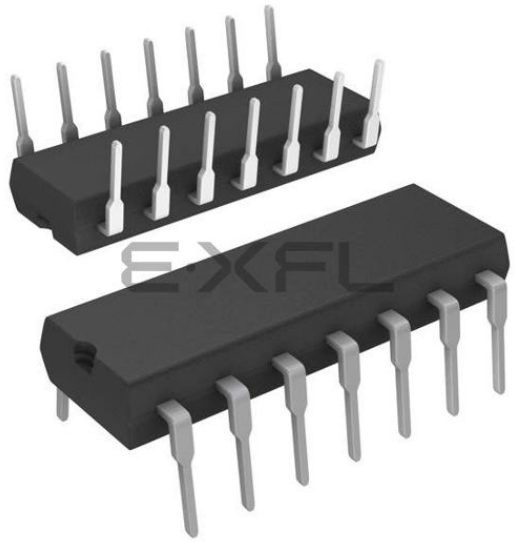

Welcome to E-XFL.COM

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, Temp Sensor, WDT
Number of I/O	8
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	14-DIP (0.300", 7.62mm)
Supplier Device Package	14-DIP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f300p

1.3. On-Chip Debug Circuitry

The C8051F300/1/2/3/4/5 devices include on-chip Silicon Labs 2-Wire (C2) debug circuitry that provides non-intrusive, full-speed, in-circuit debugging of the production part *installed in the end application.*

Silicon Labs' debugging system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADC and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F300DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F300/1/2/3/4/5 MCUs. The kit includes software with a developer's studio and debugger, an integrated 8051 assembler, and a C2 debug adapter. It also has a target application board with the associated MCU installed and large prototyping area, plus the necessary communication cables and wall-mount power supply. The Development Kit requires a computer with Windows® 98 SE or later. The Silicon Labs IDE interface is a vastly superior developing and debugging configuration, compared to standard MCU emulators that use onboard "ICE Chips" and require the MCU in the application board to be socketed. Silicon Labs' debug paradigm increases ease of use and preserves the performance of the precision analog peripherals.

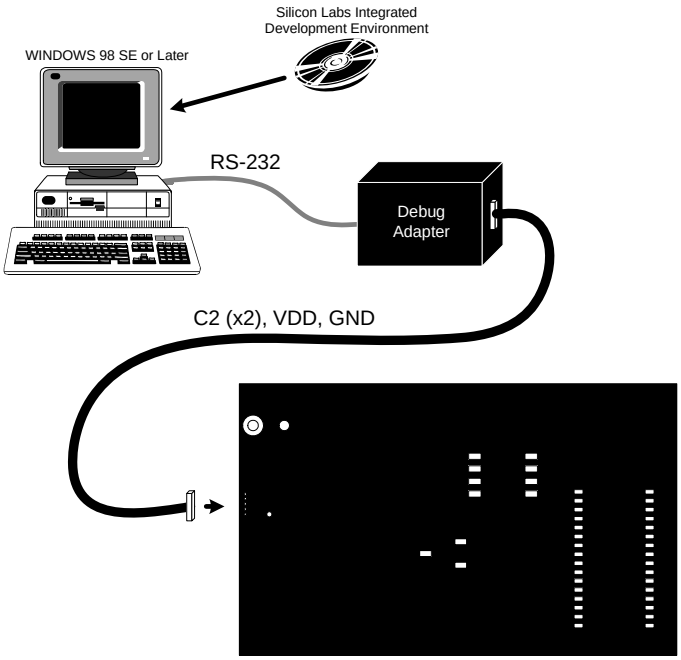

Figure 1.6. Development/In-System Debug Diagram

1.4. Programmable Digital I/O and Crossbar

C8051F300/1/2/3/4/5 devices include a byte-wide I/O Port that behaves like a typical 8051 Port with a few enhancements. Each Port pin may be configured as an analog input or a digital I/O pin. Pins selected as digital I/Os may additionally be configured for push-pull or open-drain output. The “weak pull-ups” that are fixed on typical 8051 devices may be globally disabled, providing power savings capabilities.

Table 3.1. Global Electrical Characteristics (Continued)

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
I_{DD} Supply Sensitivity (Note 3)	F = 25 MHz	—	47	—	%/V
	F = 1 MHz	—	59	—	%/V
I_{DD} Frequency Sensitivity (Note 3, Note 5)	V_{DD} = 3.0 V, F <= 1 MHz, T = 25 °C	—	0.27	—	mA/MHz
	V_{DD} = 3.0 V, F > 1 MHz, T = 25 °C	—	0.10	—	mA/MHz
	V_{DD} = 3.6 V, F <= 1 MHz, T = 25 °C	—	0.35	—	mA/MHz
	V_{DD} = 3.6 V, F > 1 MHz, T = 25 °C	—	0.12	—	mA/MHz
Digital Supply Current (Stop Mode, shutdown)	Oscillator not running, V_{DD} Monitor Disabled	—	< 0.1	—	µA

Notes:

1. Given in Table 9.2 on page 86.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization data; Not production tested.
4. Normal IDD can be estimated for frequencies <= 15 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} for >15 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number.
   For example: V_{DD} = 3.0 V; F = 20 MHz, I_{DD} = 6.6 mA – (25 MHz – 20 MHz) x 0.16 mA/MHz = 5.8 mA.
5. Idle IDD can be estimated for frequencies <= 1 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate Idle I_{DD} for >1 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number.
   For example: V_{DD} = 3.0 V; F = 5 MHz, Idle I_{DD} = 3.3 mA – (25 MHz – 5 MHz) x 0.10 mA/MHz = 1.3 mA.

5. ADC0 (8-Bit ADC, C8051F300/2)

The ADC0 subsystem for the C8051F300/2 consists of two analog multiplexers (referred to collectively as AMUX0) with 11 total input selections, a differential programmable gain amplifier (PGA), and a 500 ksps, 8-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 5.1). The AMUX0, PGA, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. ADC0 operates in both Single-ended and Differential modes, and may be configured to measure any Port pin, the Temperature Sensor output, or V_{DD} with respect to any Port pin or GND. 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.

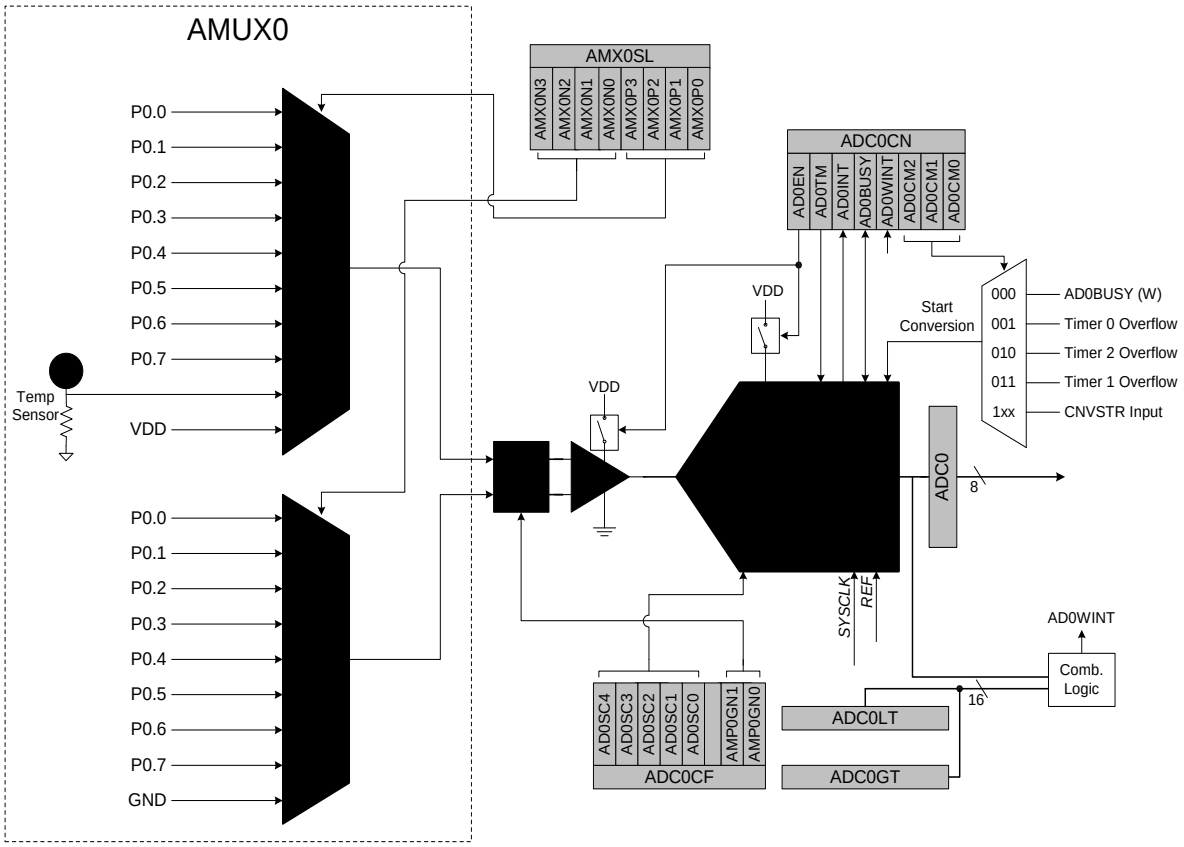

Figure 5.1. ADC0 Functional Block Diagram

5.1. Analog Multiplexer and PGA

The analog multiplexers (AMUX0) select the positive and negative inputs to the PGA, allowing any Port pin to be measured relative to any other Port pin or GND. Additionally, the on-chip temperature sensor or the positive power supply (V_{DD}) may be selected as the positive PGA input. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; all other times, ADC0 operates in Differential Mode.** The ADC0 input channels are selected in the AMX0SL register as described in SFR Definition 5.1.

The conversion code format differs in Single-ended versus Differential modes, as shown below. When in Single-ended Mode (negative input is selected GND), conversion codes are represented as 8-bit unsigned integers. Inputs are measured from ‘0’ to VREF x 255/256. Example codes are shown below.

Input Voltage	ADC0 Output (Conversion Code)
VREF x 255/256	0xFF
VREF x 128/256	0x80
VREF x 64/256	0x40
0	0x00

When in Differential Mode (negative input is not selected as GND), conversion codes are represented as 8-bit signed 2s complement numbers. Inputs are measured from –VREF to VREF x 127/128. Example codes are shown below.

Input Voltage	ADC0 Output (Conversion Code)
VREF x 127/128	0x7F
VREF x 64/128	0x40
0	0x00
–VREF x 64/128	0xC0
–VREF	0x80

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to ‘0’ the corresponding bit in register P0MDIN. To force the Crossbar to skip a Port pin, set to ‘1’ the corresponding bit in register XBR0. See **Section “12. Port Input/Output” on page 103** for more Port I/O configuration details.

The PGA amplifies the AMUX0 output signal as defined by the AMP0GN1-0 bits in the ADC0 Configuration register (SFR Definition 5.2). The PGA is software-programmable for gains of 0.5, 1, 2, or 4. The gain defaults to 0.5 on reset.

5.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the positive PGA input when the temperature sensor is selected by bits AMX0P2-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings.

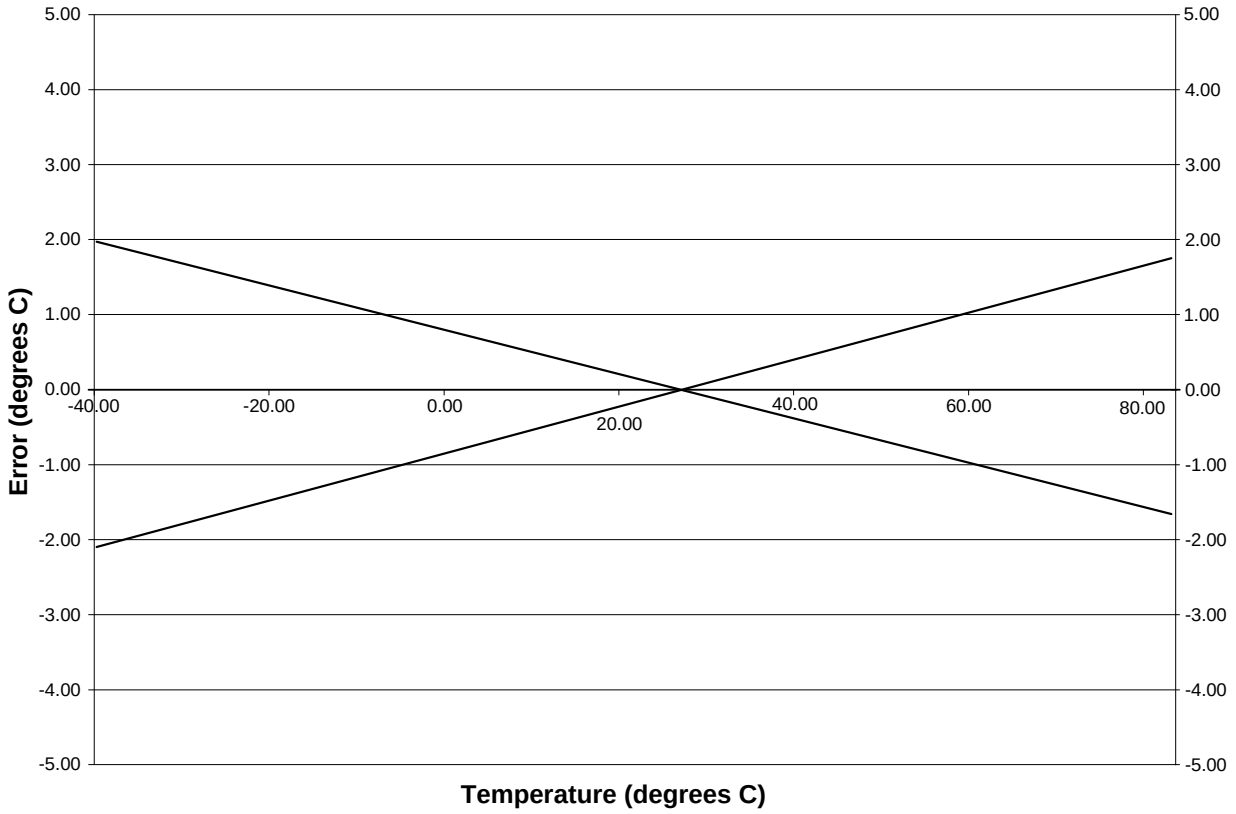

Figure 5.3. Temperature Sensor Error with 1-Point Calibration (VREF = 2.40 V)

5.3.2. Tracking Modes

According to Table 5.1 on page 47, each ADC0 conversion must be preceded by a minimum tracking time for the converted result to be accurate. The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked except when a conversion is in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 5.4). Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, due to the settling time requirements described in **Section "5.3.3. Settling Time Requirements" on page 41**.

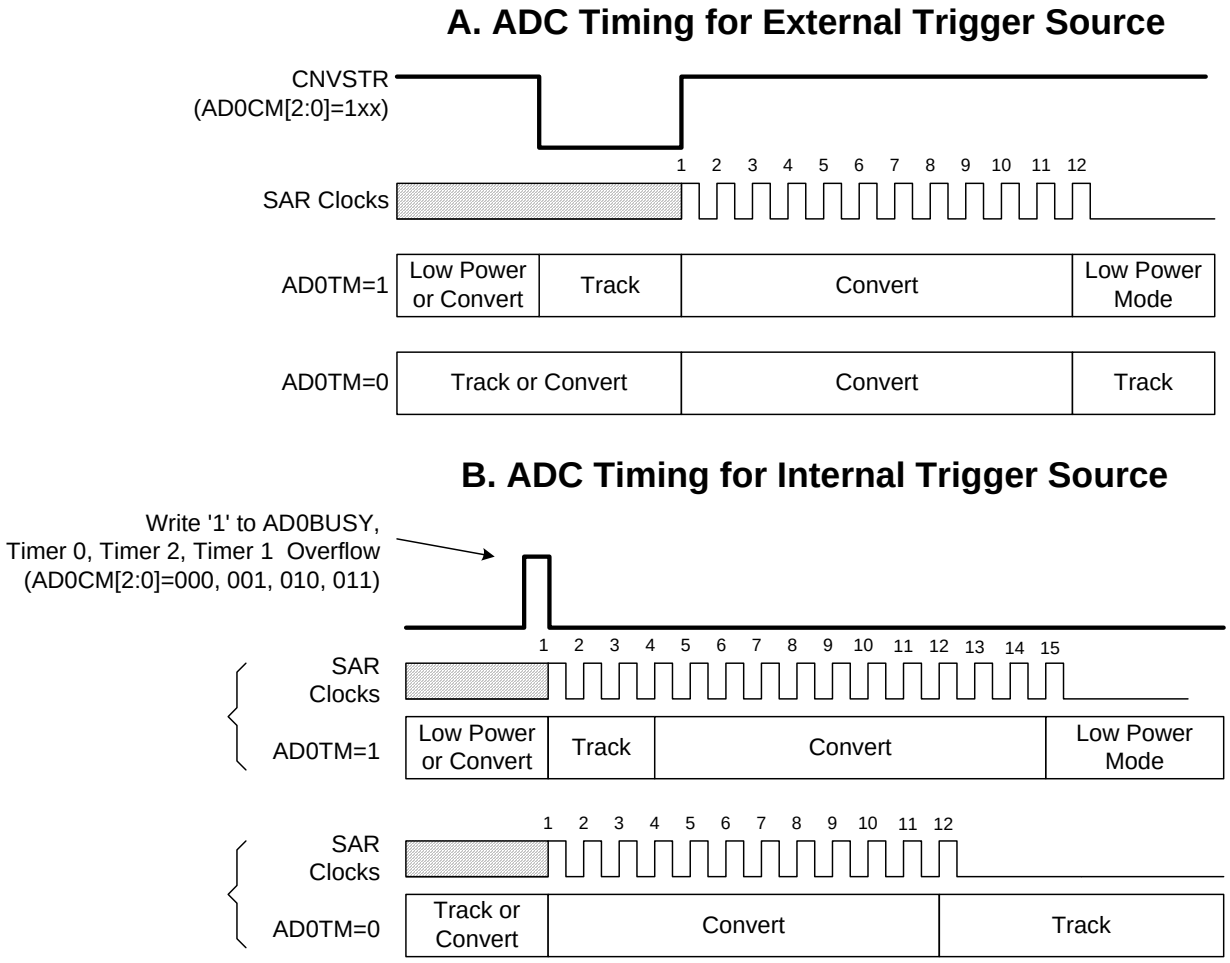

Figure 5.4. 8-Bit ADC Track and Conversion Example Timing

The output of Comparator0 can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator0 output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator0 output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 100 nA. See **Section "12.1. Priority Crossbar Decoder" on page 104** for details on configuring the Comparator0 output via the digital Crossbar. Comparator0 inputs can be externally driven from –0.25 to (V_{DD}) + 0.25 V without damage or upset. The complete electrical specifications for Comparator0 are given in Table 7.1.

The Comparator0 response time may be configured in software via the CP0MD1-0 bits in register CPT0MD (see SFR Definition 7.3). Selecting a longer response time reduces the amount of power consumed by Comparator0. See Table 7.1 for complete timing and power consumption specifications.

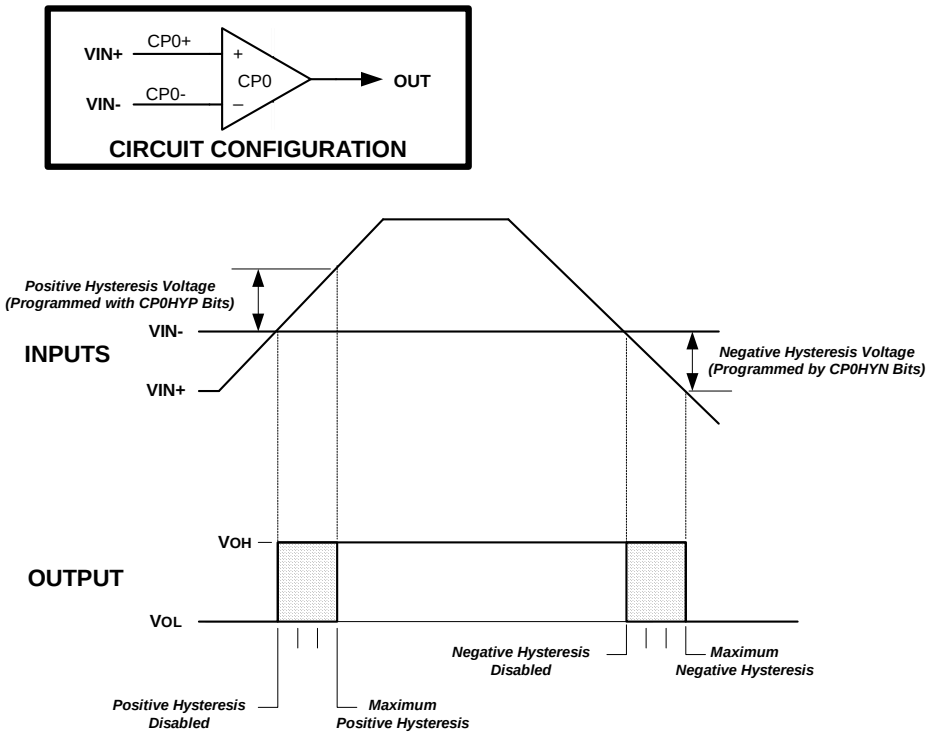

Figure 7.2. Comparator Hysteresis Plot

The hysteresis of Comparator0 is software-programmable via its Comparator0 Control register (CPT0CN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator0 hysteresis is programmed using Bits3–0 in the Comparator0 Control Register CPT0CN (shown in SFR Definition 7.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 7.2, 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.

SFR Definition 7.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	CMX0N1	CMX0N0	—	—	CMX0P1	CMX0P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bits6–4: CMX0N1–CMX0N0: Comparator0 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator0 negative input.

CMX0N1	CMX0N0	Negative Input
0	0	P0.1
0	1	P0.3
1	0	P0.5
1	1	P0.7

Bits3–2: UNUSED. Read = 00b, Write = don't care.
Bits1–0: CMX0P1–CMX0P0: Comparator0 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator0 positive input.

CMX0P1	CMX0P0	Positive Input
0	0	P0.0
0	1	P0.2
1	0	P0.4
1	1	P0.6

SFR Definition 7.3. CPT0MD: Comparator0 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7–2: UNUSED. Read = 000000b, Write = don't care.
Bits1–0: CP0MD1–CP0MD0: Comparator0 Mode Select.
These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Response Time (TYP)
0	0	0	Fastest Response Time
1	0	1	—
2	1	0	—
3	1	1	Lowest Power Consumption

Performance
The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

Programming and Debugging Support
In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2). Note that the re-programmable Flash can also be read and changed a single byte at a time by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources. C2 details can be found in **Section “17. C2 Interface” on page 173**.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, macro assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

8.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

8.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 8.1 is the

Table 8.3. Special Function Registers* (Continued)

Register	Address	Description	Page No.
TH1	0x8D	Timer/Counter 1 High	150
TL0	0x8A	Timer/Counter 0 Low	150
TL1	0x8B	Timer/Counter 1 Low	150
TMOD	0x89	Timer/Counter Mode	148
TMR2RLH	0xCB	Timer/Counter 2 Reload High	154
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	154
TMR2H	0xCD	Timer/Counter 2 High	154
TMR2L	0xCC	Timer/Counter 2 Low	154
XBR0	0xE1	Port I/O Crossbar Control 0	107
XBR1	0xE2	Port I/O Crossbar Control 1	107
XBR2	0xE3	Port I/O Crossbar Control 2	108
0x97, 0xAE, 0xAF, 0xB4, 0xB6, 0xBF, 0xCE, 0xD2, 0xD3, 0xD4, 0xD5, 0xD6, 0xD7, 0xDD, 0xDE, 0xDF, 0xF5		Reserved	
***Note:** SFRs are listed in alphabetical order. All undefined SFR locations are reserved			

8.2.7. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic l. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

SFR Definition 8.1. DPL: Data Pointer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x82

Bits7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory.

SFR Definition 8.8. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	PT2	PS0	PT1	PX1	PT0	PX0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB8

Bits7–6: UNUSED. Read = 11b, Write = don't care.

Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupts set to low priority level.
1: Timer 2 interrupts set to high priority level.

Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupts set to low priority level.
1: UART0 interrupts set to high priority level.

Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupts set to low priority level.
1: Timer 1 interrupts set to high priority level.

Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.

Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupts set to low priority level.
1: Timer 0 interrupts set to high priority level.

Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

SFR Definition 8.9. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	ECP0R	ECP0F	EPCA0	EADC0C	EWADC0	ESMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE6

Bits7–6: UNUSED. Read = 00b. Write = don't care.

Bit5: ECP0R: Enable Comparator0 (CP0) Rising Edge Interrupt.
This bit sets the masking of the CP0 Rising Edge interrupt.
0: Disable CP0 Rising Edge interrupt.
1: Enable interrupt requests generated by the CP0RIF flag.

Bit4: ECP0F: Enable Comparator0 (CP0) Falling Edge Interrupt.
This bit sets the masking of the CP0 Falling Edge interrupt.
0: Disable CP0 Falling Edge interrupt.
1: Enable interrupt requests generated by the CP0FIF flag.

Bit3: EPCA0: Enable Programmable Counter Array (PCA0) Interrupt.
This bit sets the masking of the PCA0 interrupts.
0: Disable all PCA0 interrupts.
1: Enable interrupt requests generated by PCA0.

Bit2: EADC0C: Enable ADC0 Conversion Complete Interrupt.
This bit sets the masking of the ADC0 Conversion Complete interrupt.
0: Disable ADC0 Conversion Complete interrupt.
1: Enable interrupt requests generated by the AD0INT flag.

Bit1: EWADC0: Enable Window Comparison ADC0 Interrupt.
This bit sets the masking of ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison interrupt.
1: Enable interrupt requests generated by ADC0 Window Compare flag.

Bit0: ESMB0: Enable SMBus Interrupt.
This bit sets the masking of the SMBus interrupt.
0: Disable all SMBus interrupts.
1: Enable interrupt requests generated by the SI flag.

NOTES:

13.5.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. After each byte is received, ACKRQ is set to ‘1’ and an interrupt is generated. Software must write the ACK bit (SMB0CN.1) to define the outgoing acknowledge value (Note: writing a ‘1’ to the ACK bit generates an ACK; writing a ‘0’ generates a NACK). Software should write a ‘0’ to the ACK bit after the last byte is received, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. Note that the interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 13.6 shows a typical Master Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

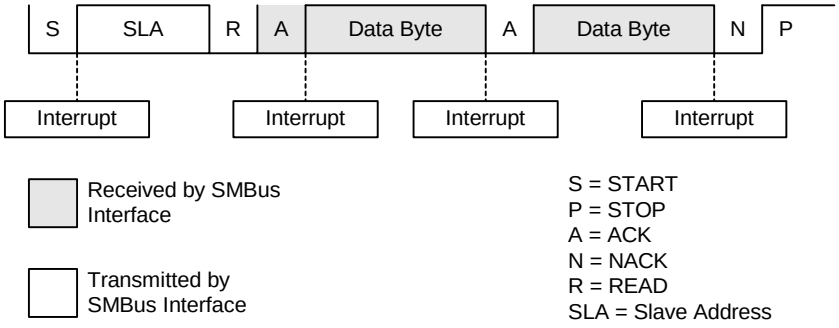

Figure 13.6. Typical Master Receiver Sequence

Table 14.3. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX[2]	1	0xD0
	115200	0.00%	192	SYSCLK	XX[2]	1	0xA0
	57600	0.00%	384	SYSCLK	XX[2]	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in **Section 15.1**.
2. X = Don't care.

SFR Definition 15.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 8.11).

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 8.11).

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

15.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

15.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 15.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

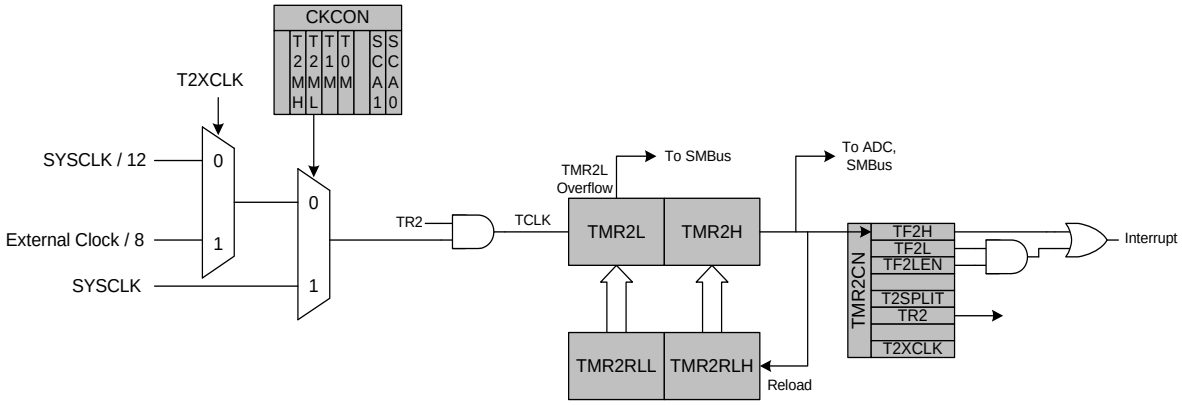

Figure 15.4. Timer 2 16-Bit Mode Block Diagram

SFR Definition 15.8. TMR2CN: Timer 2 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2H	TF2L	TF2LEN	—	T2SPLIT	TR2	—	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xC8

Bit7: 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. TF2H is not automatically cleared by hardware and must be cleared by software.

Bit6: TF2L: Timer 2 Low Byte Overflow Flag
Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF2LEN is set and Timer 2 interrupts are enabled. TF2L will set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.

Bit5: TF2LEN: Timer 2 Low Byte Interrupt Enable.
This bit enables/disables Timer 2 Low Byte interrupts. If TF2LEN is set and Timer 2 interrupts are enabled, an interrupt will be generated when the low byte of Timer 2 overflows. This bit should be cleared when operating Timer 2 in 16-bit mode.
0: Timer 2 Low Byte interrupts disabled.
1: Timer 2 Low Byte interrupts enabled.

Bit4: UNUSED. Read = 0b. Write = don't care.

Bit3: 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.

Bit2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in this mode.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit1: UNUSED. Read = 0b. Write = don't care.

Bit0: 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 external clock selection is the system clock divided by 12.
1: Timer 2 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

SFR Definition 16.4. PCA0L: PCA Counter/Timer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF9

Bits 7–0: PCA0L: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

SFR Definition 16.5. PCA0H: PCA Counter/Timer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xFA

Bits 7–0: PCA0H: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer.

C2 Register Definition 17.3. REVID: C2 Revision ID

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

This read-only register returns the 8-bit revision ID: 0x00 (Revision A)

C2 Register Definition 17.4. FPCTL: C2 Flash Programming Control

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0 FPCTL: Flash Programming Control Register
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 17.5. FPDAT: C2 Flash Programming Data

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0: FPDAT: C2 Flash Programming Data Register
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

Code	Command
0x06	Flash Block Read
0x07	Flash Block Write
0x08	Flash Page Erase
0x03	Device Erase