
**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | SMBus (2-Wire/I²C), UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 8 |
| Program Memory Size | 4KB (4K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 10-VFDFN Exposed Pad |
| Supplier Device Package | 11-QFN (3x3) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f304 |

## Table of Contents

## Table 5.1. ADC0 Electrical Characteristics

$V_{DD}$ = 3.0 V, VREF = 2.40 V (REFSL = 0), PGA Gain = 1, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 8 | | bits |
| Integral Nonlinearity | | — | ±0.5 | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | ±0.5 | ±1 | LSB |
| Offset Error | | –5.0 | 0.5 | 5.0 | LSB |
| Full Scale Error | Differential mode | –5.0 | –1 | 5.0 | LSB |
| **Dynamic Performance (10 kHz Sine-wave Differential Input, 1 dB below Full Scale, 500 ksps)** | | | | | |
| Signal-to-Noise Plus Distortion | | 45 | 48 | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –56 | — | dB |
| Spurious-Free Dynamic Range | | — | 58 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Conversion Clock | | — | — | 6 | MHz |
| Conversion Time in SAR Clocks | | 11 | — | — | clocks |
| Track/Hold Acquisition Time | | 300 | — | — | ns |
| Throughput Rate | | — | — | 500 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | | 0 | — | VREF | V |
| Input Capacitance | | — | 5 | — | pF |
| **Temperature Sensor** | | — | — | — | |
| Linearity[1,2,3] | | — | ±0.5 | — | °C |
| Gain[1,2,3] | | — | 3350 ±110 | — | µV / °C |
| Offset[1,2,3] | (Temp = 0 °C) | — | 897±31 | — | mV |
| **Power Specifications** | | | | | |
| Power Supply Current ($V_{DD}$ supplied to ADC0) | Operating Mode, 500 ksps | — | 400 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |

**Notes:**
1. Represents one standard deviation from the mean.
2. Measured with PGA Gain = 2.
3. Includes ADC offset, gain, and linearity variations.

## NOTES:

**Table 8.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| JNZ rel | Jump if A does not equal zero | 2 | 2/3 |
| CJNE A, direct, rel | Compare direct byte to A and jump if not equal | 3 | 3/4 |
| CJNE A, #data, rel | Compare immediate to A and jump if not equal | 3 | 3/4 |
| CJNE Rn, #data, rel | Compare immediate to Register and jump if not equal | 3 | 3/4 |
| CJNE @Ri, #data, rel | Compare immediate to indirect and jump if not equal | 3 | 4/5 |
| DJNZ Rn, rel | Decrement Register and jump if not zero | 2 | 2/3 |
| DJNZ direct, rel | Decrement direct byte and jump if not zero | 3 | 3/4 |
| NOP | No operation | 1 | 1 |

**Notes on Registers, Operands and Addressing Modes:**

**Rn** - Register R0-R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through R0 or R1.

**rel** - 8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

**direct** - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).

**#data** - 8-bit constant

**#data16** - 16-bit constant

**bit** - Direct-accessed bit in Data RAM or SFR

**addr11** - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction.

**addr16** - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8K-byte program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.
All mnemonics copyrighted © Intel Corporation 1980.

## SFR Definition 8.4. PSW: Program Status Word

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CY | AC | F0 | RS1 | RS0 | OV | F1 | PARITY | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xD0 |

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: RS1-RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

| RS1 | RS0 | Register Bank | Address |
|---|---|---|---|
| 0 | 0 | 0 | 0x00–0x07 |
| 0 | 1 | 1 | 0x08–0x0F |
| 1 | 0 | 2 | 0x10–0x17 |
| 1 | 1 | 3 | 0x18–0x1F |

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

**Table 8.4. Interrupt Summary**

| Interrupt Source | Interrupt Vector | Priority Order | Pending Flag | Bit addressable? | Cleared by HW? | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|
| Reset | 0x0000 | Top | None | N/A | N/A | Always Enabled | Always Highest |
| External Interrupt 0 (/INT0) | 0x0003 | 0 | IE0 (TCON.1) | Y | Y | EX0 (IE.0) | PX0 (IP.0) |
| Timer 0 Overflow | 0x000B | 1 | TF0 (TCON.5) | Y | Y | ET0 (IE.1) | PT0 (IP.1) |
| External Interrupt 1 (/INT1) | 0x0013 | 2 | IE1 (TCON.3) | Y | Y | EX1 (IE.2) | PX1 (IP.2) |
| Timer 1 Overflow | 0x001B | 3 | TF1 (TCON.7) | Y | Y | ET1 (IE.3) | PT1 (IP.3) |
| UART0 | 0x0023 | 4 | RI0 (SCON0.0)<br>TI0 (SCON0.1) | Y | N | ES0 (IE.4) | PS0 (IP.4) |
| Timer 2 Overflow | 0x002B | 5 | TF2H (TMR2CN.7)<br>TF2L (TMR2CN.6) | Y | N | ET2 (IE.5) | PT2 (IP.5) |
| SMBus Interface | 0x0033 | 6 | SI (SMB0CN.0) | Y | N | ESMB0 (EIE1.0) | PSMB0 (EIP1.0) |
| ADC0 Window Compare | 0x003B | 7 | AD0WINT (ADC0CN.3) | Y | N | EWADC0 (EIE1.1) | PWADC0 (EIP1.1) |
| ADC0 Conversion Complete | 0x0043 | 8 | AD0INT (ADC0CN.5) | Y | N | EADC0C (EIE1.2) | PADC0C (EIP1.2) |
| Programmable Counter Array | 0x004B | 9 | CF (PCA0CN.7)<br>CCFn (PCA0CN.n) | Y | N | EPCA0 (EIE1.3) | PPCA0 (EIP1.3) |
| Comparator0 Falling Edge | 0x0053 | 10 | CP0FIF (CPT0CN.4) | N | N | ECP0F (EIE1.4) | PCP0F (EIP1.4) |
| Comparator0 Rising Edge | 0x005B | 11 | CP0RIF (CPT0CN.5) | N | N | ECP0R (EIE1.5) | PCP0R (EIP1.5) |

### 8.4.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout of 100 µsec.

**SFR Definition 8.12. PCON: Power Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| GF5 | GF4 | GF3 | GF2 | GF1 | GF0 | STOP | IDLE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x87 |

Bits7–2: GF5–GF0: General Purpose Flags 5-0.
These are general purpose flags for use under software control.

Bit1: STOP: Stop Mode Select.
Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0.
1: CPU goes into Stop mode (turns off internal oscillator).

Bit0: IDLE: Idle Mode Select.
Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.
1: CPU goes into Idle mode (shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active).

## 12. Port Input/Output

Digital and analog resources are available through a byte-wide digital I/O Port, Port0. Each of the Port pins can be defined as general-purpose I/O (GPIO), analog input, or assigned to one of the internal digital resources as shown in Figure 12.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 12.3 and Figure 12.4). The registers XBR0, XBR1, and XBR2, defined in SFR Definition 12.1, SFR Definition 12.2, and SFR Definition 12.3 are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 12.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port0 Output Mode register (P0MDOUT). Complete Electrical Specifications for Port I/O are given in Table 12.1 on page 110.

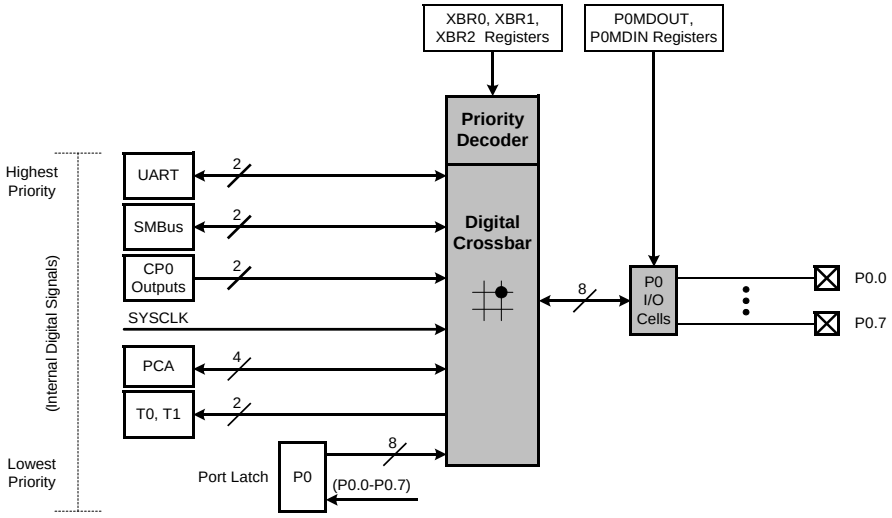


**Figure 12.1. Port I/O Functional Block Diagram**

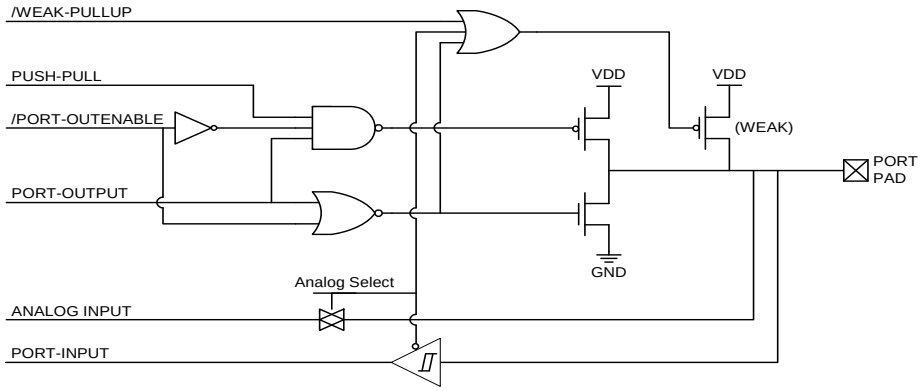


**Figure 12.2. Port I/O Cell Block Diagram**

The direction bit (R/W) occupies the least significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 13.3 illustrates a typical SMBus transaction.

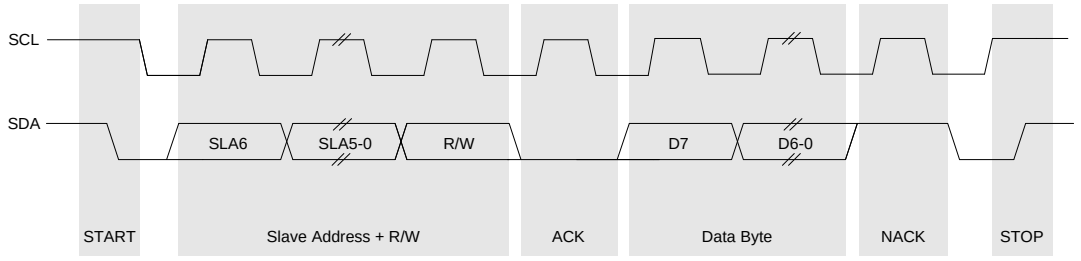


**Figure 13.3. SMBus Transaction**

### 13.3.1. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see **Section “13.3.4. SCL High (SMBus Free) Timeout” on page 114**). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

## SFR Definition 13.1. SMB0CF: SMBus Clock/Configuration

| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| ENSMB | INH | BUSY | EXTHOLD | SMBTOE | SMBFTE | SMBCS1 | SMBCS0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xC1 |

Bit7: ENSMB: SMBus Enable.
This bit enables/disables the SMBus interface. When enabled, the interface constantly monitors the SDA and SCL pins.
0: SMBus interface disabled.
1: SMBus interface enabled.

Bit6: INH: SMBus Slave Inhibit.
When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
0: SMBus Slave Mode enabled.
1: SMBus Slave Mode inhibited.

Bit5: BUSY: SMBus Busy Indicator.
This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free timeout is sensed.

Bit4: EXTHOLD: SMBus Setup and Hold Time Extension Enable.
This bit controls the SDA setup and hold times according to Table 13.2.
0: SDA Extended Setup and Hold Times disabled.
1: SDA Extended Setup and Hold Times enabled.

Bit3: SMBTOE: SMBus SCL Timeout Detection Enable.
This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 2 to reload while SCL is high and allows Timer 2 to count when SCL goes low. If Timer 2 is configured in split mode (T2SPLIT is set), only the high byte of Timer 2 is held in reload while SCL is high. Timer 2 should be programmed to generate interrupts at 25 ms, and the Timer 2 interrupt service routine should reset SMBus communication.

Bit2: SMBFTE: SMBus Free Timeout Detection Enable.
When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.

Bits1–0: SMBCS1-SMBCS0: SMBus Clock Source Selection.
These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 13.1.

| SMBCS1 | SMBCS0 | SMBus Clock Source |
|---|---|---|
| 0 | 0 | Timer 0 Overflow |
| 0 | 1 | Timer 1 Overflow |
| 1 | 0 | Timer 2 High Byte Overflow |
| 1 | 1 | Timer 2 Low Byte Overflow |

**Table 13.4. SMBus Status Decoding (Continued)**

| Mode | Values Read | | | | Current SMbus State | Typical Response Options | Values Written | | |
|---|---|---|---|---|---|---|---|---|---|
| | Status Vector | ACKRQ | ARBLOST | ACK | | | STA | STO | ACK |
| SLAVE RECEIVER | 0010 | 1 | 0 | X | A slave address was received; ACK requested. | Acknowledge received address (received slave address match, R/W bit = READ). | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received address. | 0 | 0 | 0 |
| | | | | | | Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see **Section 13.5.4** for procedure. | 0 | 0 | 1 |
| | | 1 | 1 | X | Lost arbitration as master; slave address received; ACK requested. | Acknowledge received address (received slave address match, R/W bit = READ). | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received address. | 0 | 0 | 0 |
| | | | | | | Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see **Section 13.5.4** for procedure. | 0 | 0 | 1 |
| | | | | | | Reschedule failed transfer; do not acknowledge received address | 1 | 0 | 0 |
| | 0010 | 0 | 1 | X | Lost arbitration while attempting a repeated START. | Abort failed transfer. | 0 | 0 | X |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X |
| | 0001 | 1 | 1 | X | Lost arbitration while attempting a STOP. | No action required (transfer complete/aborted). | 0 | 0 | 0 |
| | | 0 | 0 | X | A STOP was detected while addressed as a Slave Transmitter or Slave Receiver. | Clear STO. | 0 | 0 | X |
| | | 0 | 1 | X | Lost arbitration due to a detected STOP. | Abort transfer. | 0 | 0 | X |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X |
| | 0000 | 1 | 0 | X | A slave byte was received; ACK requested. | Acknowledge received byte; Read SMB0DAT. | 0 | 0 | 1 |
| | | | | | | Do not acknowledge received byte. | 0 | 0 | 0 |
| | | 1 | 1 | X | Lost arbitration while transmitting a data byte as master. | Abort failed transfer. | 0 | 0 | 0 |
| | | | | | | Reschedule failed transfer. | 1 | 0 | 0 |

## NOTES:

### 14.2.2. 9-Bit UART

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

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to '1'. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to '1'. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to '1'.

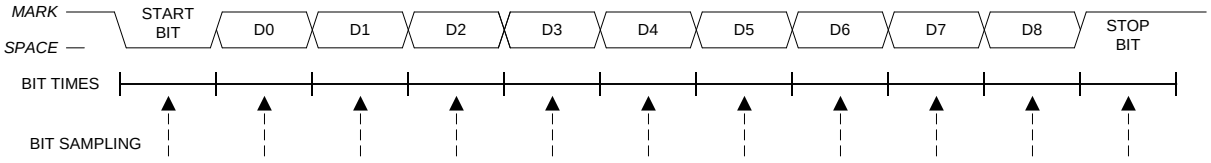


**Figure 14.5. 9-Bit UART Timing Diagram**

**Table 14.6. Timer Settings for Standard Baud Rates Using an External 3.6864 MHZ Oscillator**

| | Frequency: 3.6864 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% | 16 | SYSCLK | XX[2] | 1 | 0xF8 |
| | 115200 | 0.00% | 32 | SYSCLK | XX[2] | 1 | 0xF0 |
| | 57600 | 0.00% | 64 | SYSCLK | XX[2] | 1 | 0xE0 |
| | 28800 | 0.00% | 128 | SYSCLK | XX[2] | 1 | 0xC0 |
| | 14400 | 0.00% | 256 | SYSCLK | XX[2] | 1 | 0x80 |
| | 9600 | 0.00% | 384 | SYSCLK | XX[2] | 1 | 0x40 |
| | 2400 | 0.00% | 1536 | SYSCLK / 12 | 00 | 0 | 0xC0 |
| | 1200 | 0.00% | 3072 | SYSCLK / 12 | 00 | 0 | 0x80 |
| SYSCLK from Internal Osc. | 230400 | 0.00% | 16 | EXTCLK / 8 | 11 | 0 | 0xFF |
| | 115200 | 0.00% | 32 | EXTCLK / 8 | 11 | 0 | 0xFE |
| | 57600 | 0.00% | 64 | EXTCLK / 8 | 11 | 0 | 0xFC |
| | 28800 | 0.00% | 128 | EXTCLK / 8 | 11 | 0 | 0xF8 |
| | 14400 | 0.00% | 256 | EXTCLK / 8 | 11 | 0 | 0xF0 |
| | 9600 | 0.00% | 384 | EXTCLK / 8 | 11 | 0 | 0xE8 |

**Notes:**

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

### SFR Definition 15.4. TL0: Timer 0 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: |
| | | | | | | | | 0x8A |

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0

### SFR Definition 15.5. TL1: Timer 1 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: |
| | | | | | | | | 0x8B |

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

### SFR Definition 15.6. TH0: Timer 0 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: |
| | | | | | | | | 0x8C |

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

### SFR Definition 15.7. TH1: Timer 1 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: |
| | | | | | | | | 0x8D |

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

### 15.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 15.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

| T2MH | T2XCLK | TMR2H Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK / 12 |
| 0 | 1 | External Clock / 8 |
| 1 | X | SYSCLK |

| T2ML | T2XCLK | TMR2L Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK / 12 |
| 0 | 1 | External Clock / 8 |
| 1 | X | SYSCLK |

Note: External clock divided by 8 is synchronized with the system clock, and the external clock must be less than or equal to the system clock to operate in this mode.

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

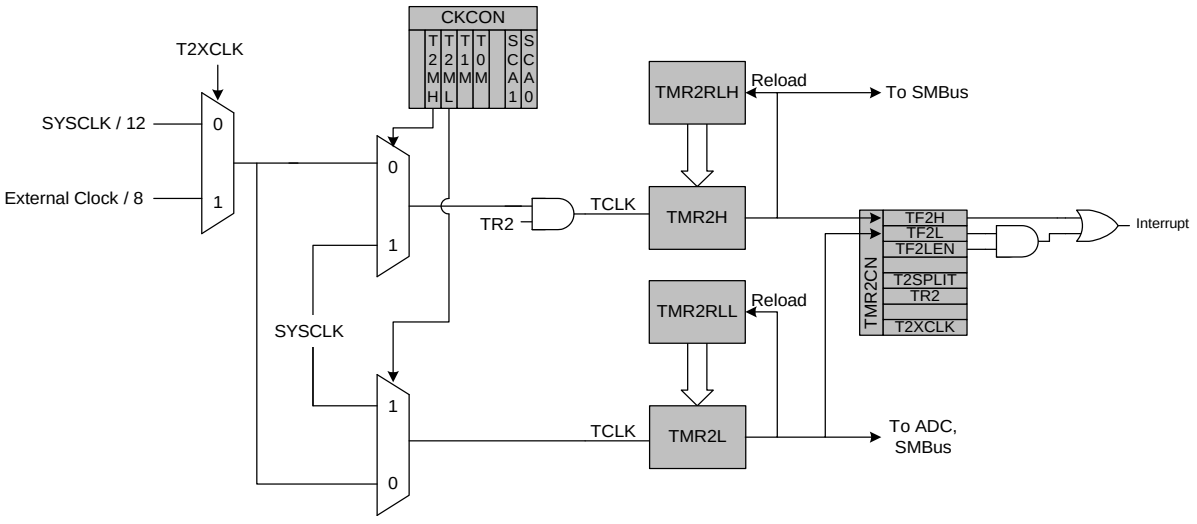


**Figure 15.5. Timer 2 8-Bit Mode Block Diagram**

## 16.1. 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 16.1. **Note that in 'External oscillator source divided by 8' mode, the external oscillator source is synchronized with the system clock, and must have a frequency less than or equal to the system clock.**

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 (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

**Table 16.1. 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* |

***Note:** External oscillator source divided by 8 is synchronized with the system clock.

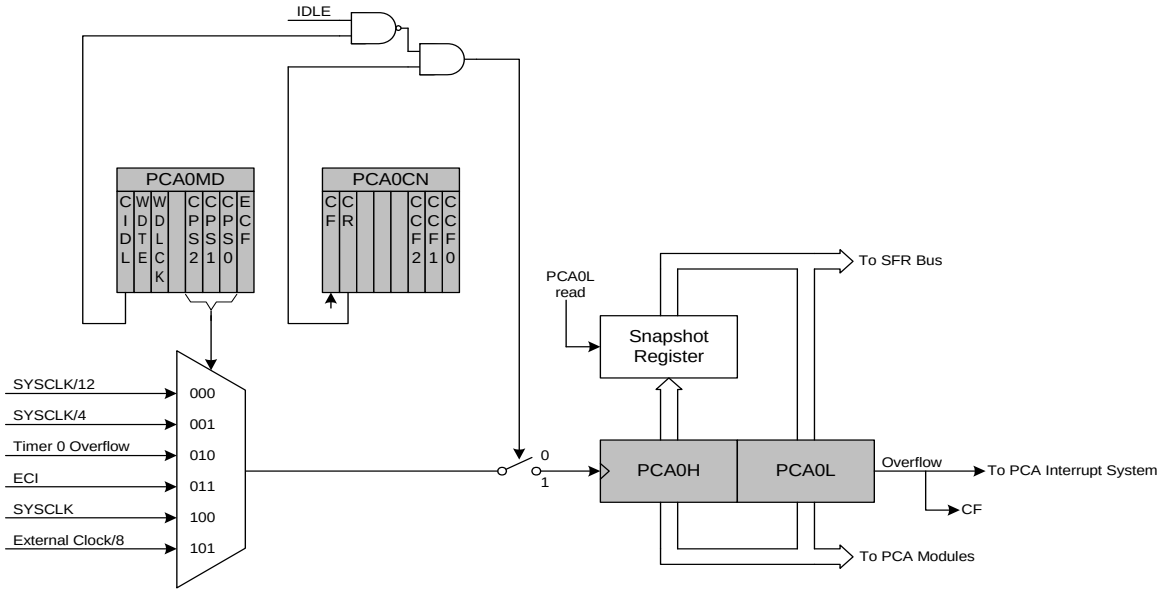


**Figure 16.2. PCA Counter/Timer Block Diagram**

## 16.4. Register Descriptions for PCA

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

### SFR Definition 16.1. PCA0CN: PCA Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CF | CR | — | — | — | CCF2 | CCF1 | CCF0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0xD8 |

Bit7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: CR: PCA Counter/Timer Run Control.
This bit enables/disables the PCA Counter/Timer.
0: PCA Counter/Timer disabled.
1: PCA Counter/Timer enabled.

Bits5–3: UNUSED. Read = 000b, Write = don't care.

Bit2: CCF2: PCA Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: CCF1: PCA Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: CCF0: PCA Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

# 17. C2 Interface

C8051F300/1/2/3/4/5 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 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.

## 17.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash 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 17.1. C2ADD: C2 Address

Reset Value: 00000000

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

| Address | Description |
|---|---|
| 0x00 | Selects the Device ID register for Data Read instructions |
| 0x01 | Selects the Revision ID register for Data Read instructions |
| 0x02 | Selects the C2 Flash Programming Control register for Data Read/Write instructions |
| 0xB4 | Selects the C2 Flash Programming Data register for Data Read/Write instructions |
| 0x80 | Selects the Port0 register for Data Read/Write instructions |
| 0xF1 | Selects the Port0 Input Mode register for Data Read/Write instructions |
| 0xA4 | Selects the Port0 Output Mode register for Data Read/Write instructions |

### C2 Register Definition 17.2. DEVICEID: C2 Device ID

Reset Value: 00000100

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

This read-only register returns the 8-bit device ID: 0x04 (C8051F300/1/2/3/4/5).

### 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 |