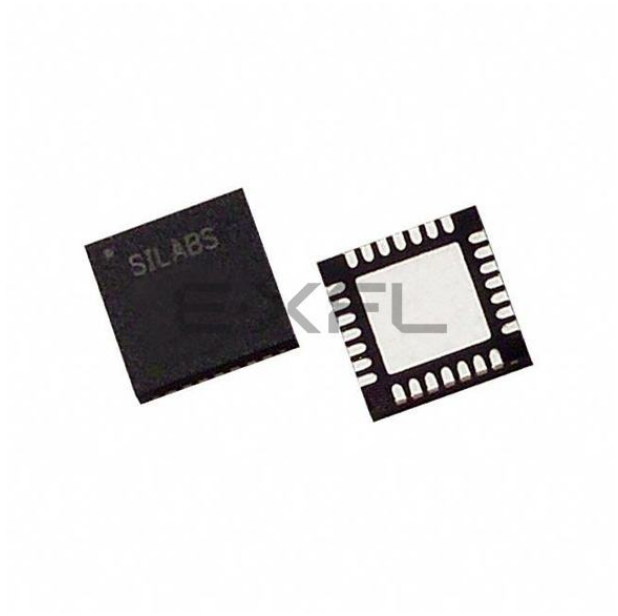


**Welcome to E-XFL.COM**

## 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 | Not For New Designs |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 25 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 17x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-VFQFN Exposed Pad |
| Supplier Device Package | 28-QFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f311-gm |

## List of Registers

## 2. Absolute Maximum Ratings

**Table 2.1. Absolute Maximum Ratings***

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Ambient temperature under bias | | –55 | — | 125 | °C |
| Storage Temperature | | –65 | — | 150 | °C |
| Voltage on any Port I/O Pin or $\overline{\text{RST}}$ with respect to GND | | –0.3 | — | 5.8 | V |
| Voltage on $V_{DD}$ with respect to GND | | –0.3 | — | 4.2 | V |
| Maximum Total current through $V_{DD}$ and GND | | — | — | 500 | mA |
| Maximum output current sunk by $\overline{\text{RST}}$ or any Port pin | | — | — | 100 | mA |

***Note:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**Table 4.1. Pin Definitions for the C8051F31x (Continued)**

| Name | Pin Numbers | | | Type | Description |
|---|---|---|---|---|---|
| | 'F310/2/4 | 'F311/3/5 | 'F316/7 | | |
| P1.5 | 21 | 17 | 13 | D I/O or A In | Port 1.5. See **Section 13** for a complete description. |
| P1.6 | 20 | 16 | | D I/O or A In | Port 1.6. See **Section 13** for a complete description. |
| P1.7 | 19 | 15 | | D I/O or A In | Port 1.7. See **Section 13** for a complete description. |
| P2.0 | 18 | 14 | 12 | D I/O or A In | Port 2.0. See **Section 13** for a complete description. |
| P2.1 | 17 | 13 | 11 | D I/O or A In | Port 2.1. See **Section 13** for a complete description. |
| P2.2 | 16 | 12 | 10 | D I/O or A In | Port 2.2. See **Section 13** for a complete description. |
| P2.3 | 15 | 11 | 9 | D I/O or A In | Port 2.3. See **Section 13** for a complete description. |
| P2.4 | 14 | 10 | 8 | D I/O or A In | Port 2.4. See **Section 13** for a complete description. |
| P2.5 | 13 | 9 | 7 | D I/O or A In | Port 2.5. See **Section 13** for a complete description. |
| P2.6 | 12 | 8 | | D I/O or A In | Port 2.6. See **Section 13** for a complete description. |
| P2.7 | 11 | 7 | | D I/O or A In | Port 2.7. See **Section 13** for a complete description. |
| P3.1 | 7 | | | D I/O or A In | Port 3.1. See **Section 13** for a complete description. |
| P3.2 | 8 | | | D I/O or A In | Port 3.2. See **Section 13** for a complete description. |
| P3.3 | 9 | | | D I/O or A In | Port 3.3. See **Section 13** for a complete description. |
| P3.4 | 10 | | | D I/O or A In | Port 3.4. See **Section 13** for a complete description. |

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, gain and/or offset calibration is recommended. Typically a 1-point calibration includes the following steps:

Step 1. Control/measure the ambient temperature (this temperature must be known).
Step 2. Power the device, and delay for a few seconds to allow for self-heating.
Step 3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
Step 4. Calculate the offset and/or gain characteristics, and store these values in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Note that parameters which affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

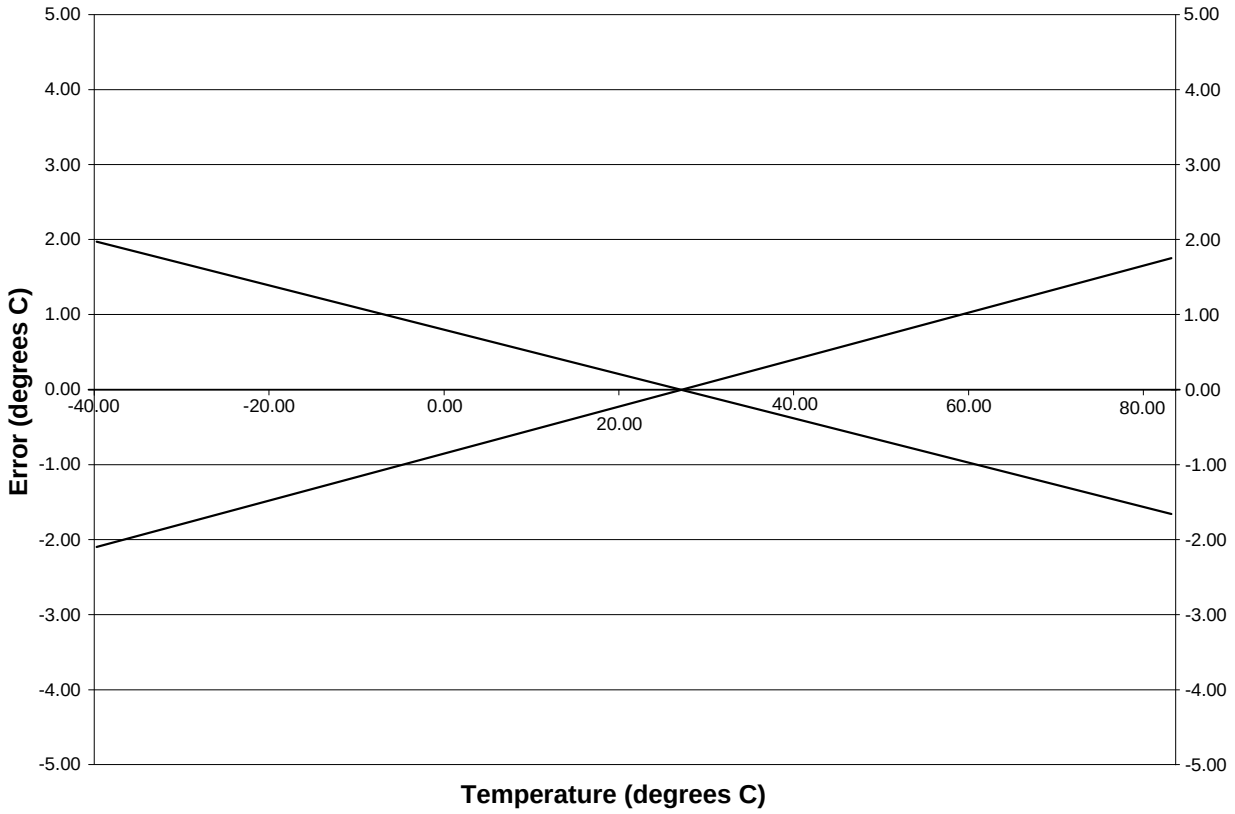


**Figure 5.3. Temperature Sensor Error with 1-Point Calibration**

## SFR Definition 5.1. AMX0P: AMUX0 Positive Channel Select

| R | R | R | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | AMX0P4 | AMX0P3 | AMX0P2 | AMX0P1 | AMX0P0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | | 0xBB |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–0: AMX0P4–0: AMUX0 Positive Input Selection

| AMX0P4–0 | ADC0 Positive Input |
|---|---|
| 00000 | P1.0 |
| 00001 | P1.1 |
| 00010 | P1.2 |
| 00011 | P1.3 |
| 00100 | P1.4 |
| 00101 | P1.5 |
| 00110 | P1.6(1) |
| 00111 | P1.7(1) |
| 01000 | P2.0 |
| 01001 | P2.1 |
| 01010 | P2.2 |
| 01011 | P2.3 |
| 01100 | P2.4 |
| 01101 | P2.5 |
| 01110 | P2.6(1) |
| 01111 | P2.7(1) |
| 10000 | P3.0 |
| 10001(2) | P3.1(2) |
| 10010(2) | P3.2(2) |
| 10011(2) | P3.3(2) |
| 10100(2) | P3.4(2) |
| 10101–11101 | RESERVED |
| 11110 | Temp Sensor |
| 11111 | $V_{DD}$ |

**Notes:**
1. Only applies to C8051F310/1/2/3/4/5; selection RESERVED on C8051F316/7 devices.
2. Only applies to C8051F310/2; selection RESERVED on C8051F311/3/6/7 devices.

## Table 5.1. ADC0 Electrical Characteristics

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

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 10 | | bits |
| Integral Nonlinearity | | — | ±0.5 | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | ±0.5 | ±1 | LSB |
| Offset Error | | –12 | 1 | +12 | LSB |
| Full Scale Error | Differential mode | –15 | –5 | +5 | LSB |
| Offset Temperature Coefficient | | — | 3.6 | — | ppm/°C |
| **Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)** | | | | | |
| Signal-to-Noise Plus Distortion | | 53 | 55.5 | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –67 | — | dB |
| Spurious-Free Dynamic Range | | — | 78 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Conversion Clock | | — | — | 3 | MHz |
| Conversion Time in SAR Clocks | | 10 | — | — | clocks |
| Track/Hold Acquisition Time | | 300 | — | — | ns |
| Throughput Rate | | — | — | 200 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | | 0 | — | VREF | V |
| Input Capacitance | | — | 5 | — | pF |
| **Temperature Sensor** | | — | — | — | |
| Linearity* | | — | ±0.5 | — | °C |
| Gain* | | — | 3350 ± 10 | — | µV / °C |
| Offset* | (Temp = 0 °C) | — | 897 ± 31 | — | mV |
| **Power Specifications** | | | | | |
| Power Supply Current ($V_{DD}$ supplied to ADC0) | Operating Mode, 200 ksps | — | 400 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |
| ***Note:** Represents one standard deviation from the mean. Includes ADC offset, gain, and linearity variations. | | | | | |

# 7. Comparators

C8051F31x devices include two on-chip programmable voltage comparators: Comparator0 is shown in Figure 7.1; Comparator1 is shown in Figure 7.2. The two comparators operate identically with the following exceptions: (1) Their input selections differ as shown in Figure 7.1 and Figure 7.2; (2) Comparator0 can be used as a reset source.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see **Section “13.2. Port I/O Initialization” on page 133**). Comparator0 may also be used as a reset source (see **Section “9.5. Comparator0 Reset” on page 108**).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 7.5). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 negative input.

**Important Note About Comparator Inputs:** The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section “13.3. General Purpose Port I/O” on page 135**).

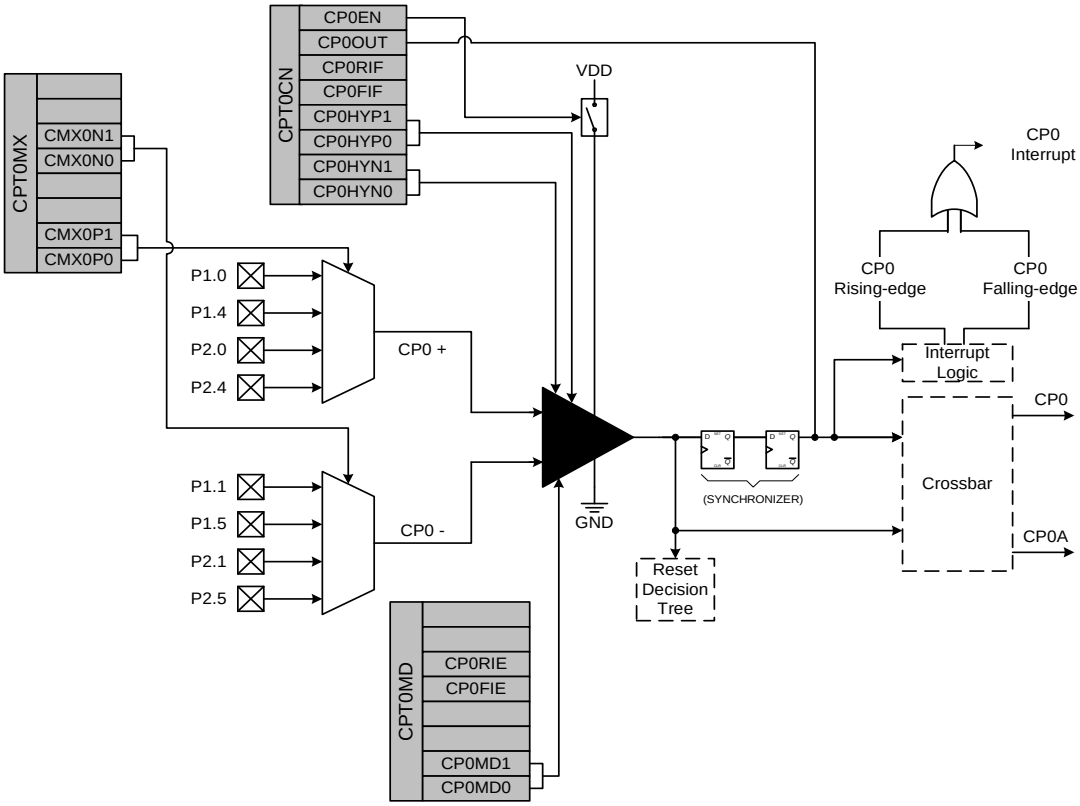


**Figure 7.1. Comparator0 Functional Block Diagram**

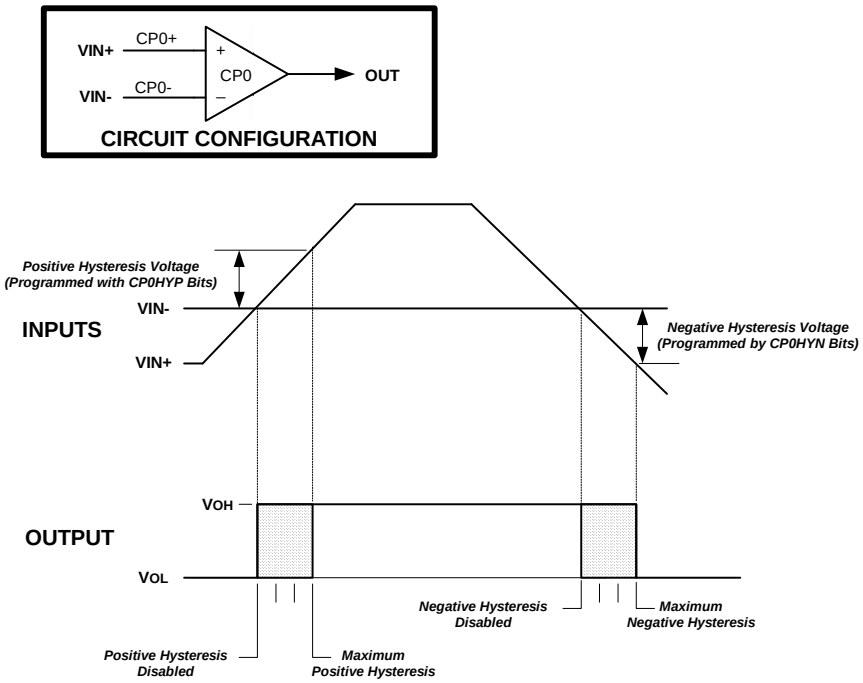


**Figure 7.3. Comparator Hysteresis Plot**

The Comparator hysteresis is software-programmable via its Comparator Control register CPTnCN (for n = 0 or 1). 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 Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in SFR Definition 7.1 and SFR Definition 7.4). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Table 7.1, 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 CPnHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section “8.3. Interrupt Handler” on page 93**). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge interrupt, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge interrupt. Once set, these bits remain set until cleared by software. The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.

The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered-on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed. This Power Up Time is specified in Table 7.1 on page 78.

**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). 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 “20. C2 Interface” on page 223**.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including an editor, evaluation compiler, 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

CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

### 8.1.2. MOVX Instruction and Program Memory

The MOVX instruction is typically used to access external data memory (Note: the C8051F31x does not support external data or program memory). In the CIP-51, the MOVX write instruction is used to accesses external RAM and the on-chip program memory space implemented as re-programmable Flash memory. The Flash access feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section "10. Flash Memory" on page 111** for further details.

**Table 8.1. CIP-51 Instruction Set Summary**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| **Arithmetic Operations** | | | |
| ADD A, Rn | Add register to A | 1 | 1 |
| ADD A, direct | Add direct byte to A | 2 | 2 |
| ADD A, @Ri | Add indirect RAM to A | 1 | 2 |
| ADD A, #data | Add immediate to A | 2 | 2 |
| ADDC A, Rn | Add register to A with carry | 1 | 1 |
| ADDC A, direct | Add direct byte to A with carry | 2 | 2 |
| ADDC A, @Ri | Add indirect RAM to A with carry | 1 | 2 |
| ADDC A, #data | Add immediate to A with carry | 2 | 2 |
| SUBB A, Rn | Subtract register from A with borrow | 1 | 1 |
| SUBB A, direct | Subtract direct byte from A with borrow | 2 | 2 |
| SUBB A, @Ri | Subtract indirect RAM from A with borrow | 1 | 2 |
| SUBB A, #data | Subtract immediate from A with borrow | 2 | 2 |
| INC A | Increment A | 1 | 1 |
| INC Rn | Increment register | 1 | 1 |
| INC direct | Increment direct byte | 2 | 2 |
| INC @Ri | Increment indirect RAM | 1 | 2 |
| DEC A | Decrement A | 1 | 1 |
| DEC Rn | Decrement register | 1 | 1 |
| DEC direct | Decrement direct byte | 2 | 2 |
| DEC @Ri | Decrement indirect RAM | 1 | 2 |
| INC DPTR | Increment Data Pointer | 1 | 1 |
| MUL AB | Multiply A and B | 1 | 4 |
| DIV AB | Divide A by B | 1 | 8 |
| DA A | Decimal adjust A | 1 | 1 |
| **Logical Operations** | | | |
| ANL A, Rn | AND Register to A | 1 | 1 |
| ANL A, direct | AND direct byte to A | 2 | 2 |
| ANL A, @Ri | AND indirect RAM to A | 1 | 2 |
| ANL A, #data | AND immediate to A | 2 | 2 |
| ANL direct, A | AND A to direct byte | 2 | 2 |
| ANL direct, #data | AND immediate to direct byte | 3 | 3 |
| ORL A, Rn | OR Register to A | 1 | 1 |

### SFR Definition 8.2. DPH: Data Pointer 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: 0x83 |

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

### SFR Definition 8.3. SP: Stack Pointer

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0x81 |

Bits7–0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

## 12.4. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 12.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 12.4. For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is first enabled, the oscillator amplitude detection circuit requires a settling time to achieve proper bias. Introducing a delay of 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Force the XTAL1 and XTAL2 pins low by writing 0s to the port latch.
Step 2. Configure XTAL1 and XTAL2 as analog inputs.
Step 3. Enable the external oscillator.
Step 4. Wait at least 1 ms.
Step 5. Poll for XTLVLD => '1'.
Step 6. Switch the system clock to the external oscillator.

**Note:** Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

**Note:** The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 12.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 12.2.

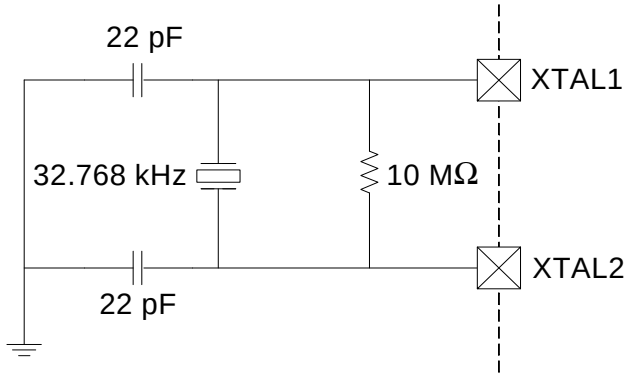


**Figure 12.2. 32.768 kHz External Crystal Example**

**Important Note on External Crystals:** Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

## SFR Definition 13.13. P2MDOUT: Port2 Output Mode

| 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: |
| | | | | | | | | 0xA6 |

Bits7–0: Output Configuration Bits for P2.7–P2.0 (respectively): ignored if corresponding bit in register P2MDIN is logic 0.
0: Corresponding P2.n Output is open-drain.
1: Corresponding P2.n Output is push-pull.

**Note:** Only P2.0–P2.5 are associated with Port pins on the C8051F316/7 devices.

## SFR Definition 13.14. P2SKIP: Port2 Skip

| 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: |
| | | | | | | | | 0xD6 |

Bits7–0: P2SKIP[7:0]: Port2 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P2.n pin is not skipped by the Crossbar.
1: Corresponding P2.n pin is skipped by the Crossbar.

**Note:** Only P2.0–P2.3 are associated with the Crossbar.

**Table 15.3. 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)* | T1M* | Timer 1 Reload Value (hex) |
| SYSCLK from External Osc. | 230400 | 0.00% | 96 | SYSCLK | XX | 1 | 0xD0 |
| | 115200 | 0.00% | 192 | SYSCLK | XX | 1 | 0xA0 |
| | 57600 | 0.00% | 384 | SYSCLK | XX | 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 |

X = Don't care

***Note:** SCA1–SCA0 and T1M bit definitions can be found in **Section 17.1**.

**Table 15.4. Timer Settings for Standard Baud Rates Using an External 18.432 MHz Oscillator**

| | Frequency: 18.432 MHz | | | | | | |
|---|---|---|---|---|---|---|---|
| | Target Baud Rate (bps) | Baud Rate % Error | Oscillator Divide Factor | Timer Clock Source | SCA1-SCA0 (pre-scale select)* | T1M* | Timer 1 Reload Value (hex) |
| SYSCLK from External Osc. | 230400 | 0.00% | 80 | SYSCLK | XX | 1 | 0xD8 |
| | 115200 | 0.00% | 160 | SYSCLK | XX | 1 | 0xB0 |
| | 57600 | 0.00% | 320 | SYSCLK | XX | 1 | 0x60 |
| | 28800 | 0.00% | 640 | SYSCLK / 4 | 01 | 0 | 0xB0 |
| | 14400 | 0.00% | 1280 | SYSCLK / 4 | 01 | 0 | 0x60 |
| | 9600 | 0.00% | 1920 | SYSCLK / 12 | 00 | 0 | 0xB0 |
| | 2400 | 0.00% | 7680 | SYSCLK / 48 | 10 | 0 | 0xB0 |
| | 1200 | 0.00% | 15360 | SYSCLK / 48 | 10 | 0 | 0x60 |
| SYSCLK from Internal Osc. | 230400 | 0.00% | 80 | EXTCLK / 8 | 11 | 0 | 0xFB |
| | 115200 | 0.00% | 160 | EXTCLK / 8 | 11 | 0 | 0xF6 |
| | 57600 | 0.00% | 320 | EXTCLK / 8 | 11 | 0 | 0xEC |
| | 28800 | 0.00% | 640 | EXTCLK / 8 | 11 | 0 | 0xD8 |
| | 14400 | 0.00% | 1280 | EXTCLK / 8 | 11 | 0 | 0xB0 |
| | 9600 | 0.00% | 1920 | EXTCLK / 8 | 11 | 0 | 0x88 |

X = Don't care

***Note:** SCA1–SCA0 and T1M bit definitions can be found in **Section 17.1**.

## 16.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 16.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 16.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 16.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

## SFR Definition 17.2. 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 |

## 18.1. 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 18.1. **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 (IE.7) and the EPCA0 bit in EIE1 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 18.1. 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. | | | |

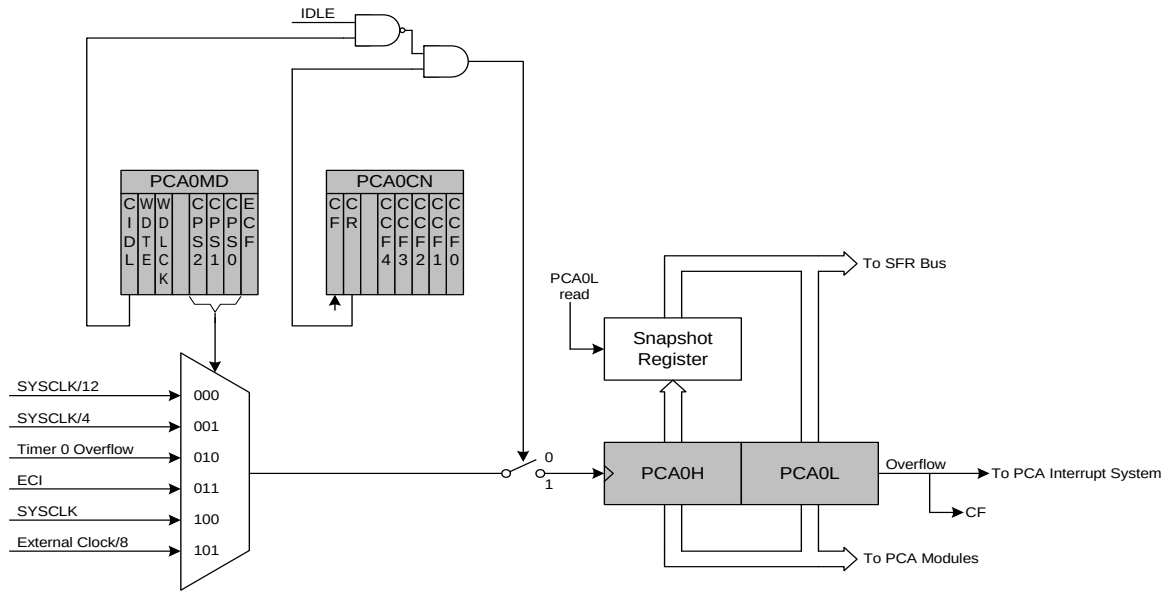


**Figure 18.2. PCA Counter/Timer Block Diagram**

### 18.2.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 18.1, where $F_{PCA}$ is the frequency of the clock selected by the CPS2-0 bits in the PCA mode register, PCA0MD.

**Equation 18.1. Square Wave Frequency Output**

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register.

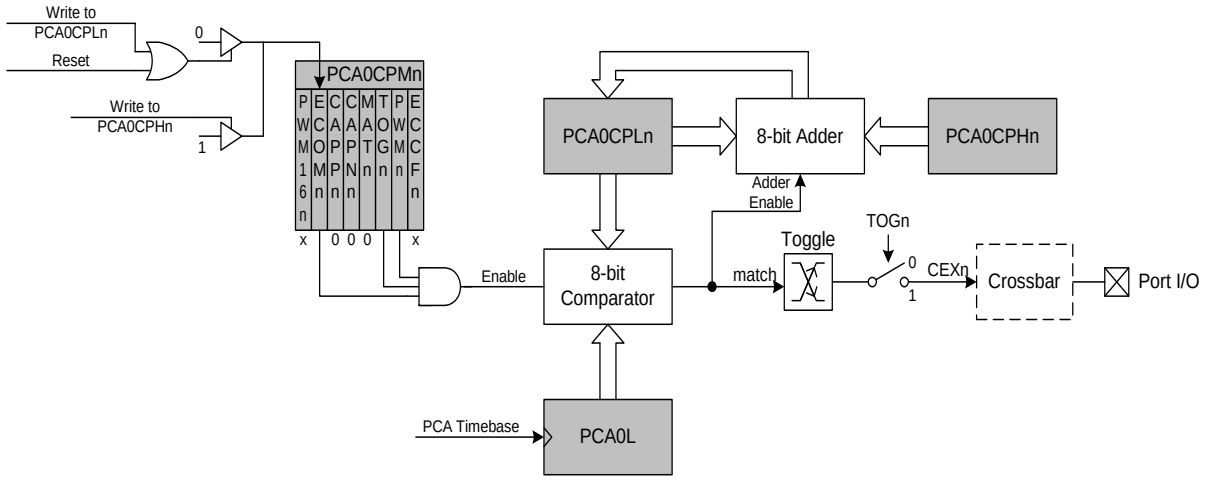


**Figure 18.7. PCA Frequency Output Mode**

## 20.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming functions may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (/RST) and C2D (P3.0) 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 20.1.

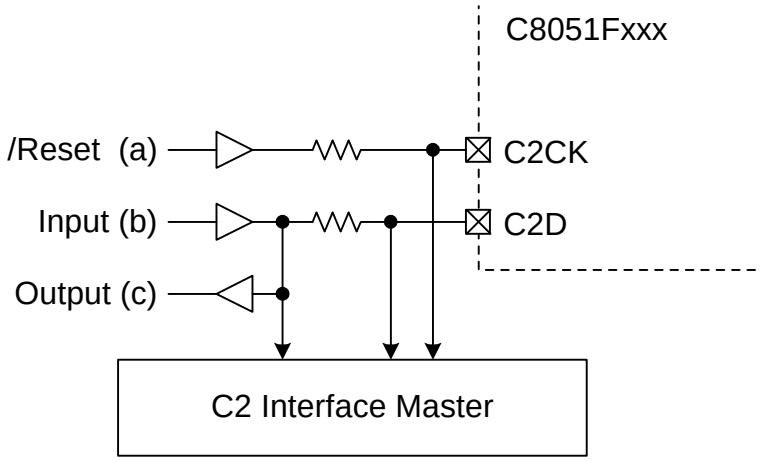


**Figure 20.1. Typical C2 Pin Sharing**

The configuration in Figure 20.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The /RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

**IoT Portfolio**
*www.silabs.com/IoT*

**SW/HW**
*www.silabs.com/simplicity*


**Quality**
*www.silabs.com/quality*

**Support and Community**
*community.silabs.com*



SILICON LABS®

**Silicon Laboratories Inc.**
**400 West Cesar Chavez**
**Austin, TX 78701**
**USA**

**http://www.silabs.com**