**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 | Last Time Buy |
| 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 | 2KB (2K 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 | 14-SOIC (0.154", 3.90mm Width) |
| Supplier Device Package | 14-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f305-gsr |

## 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.

## 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.

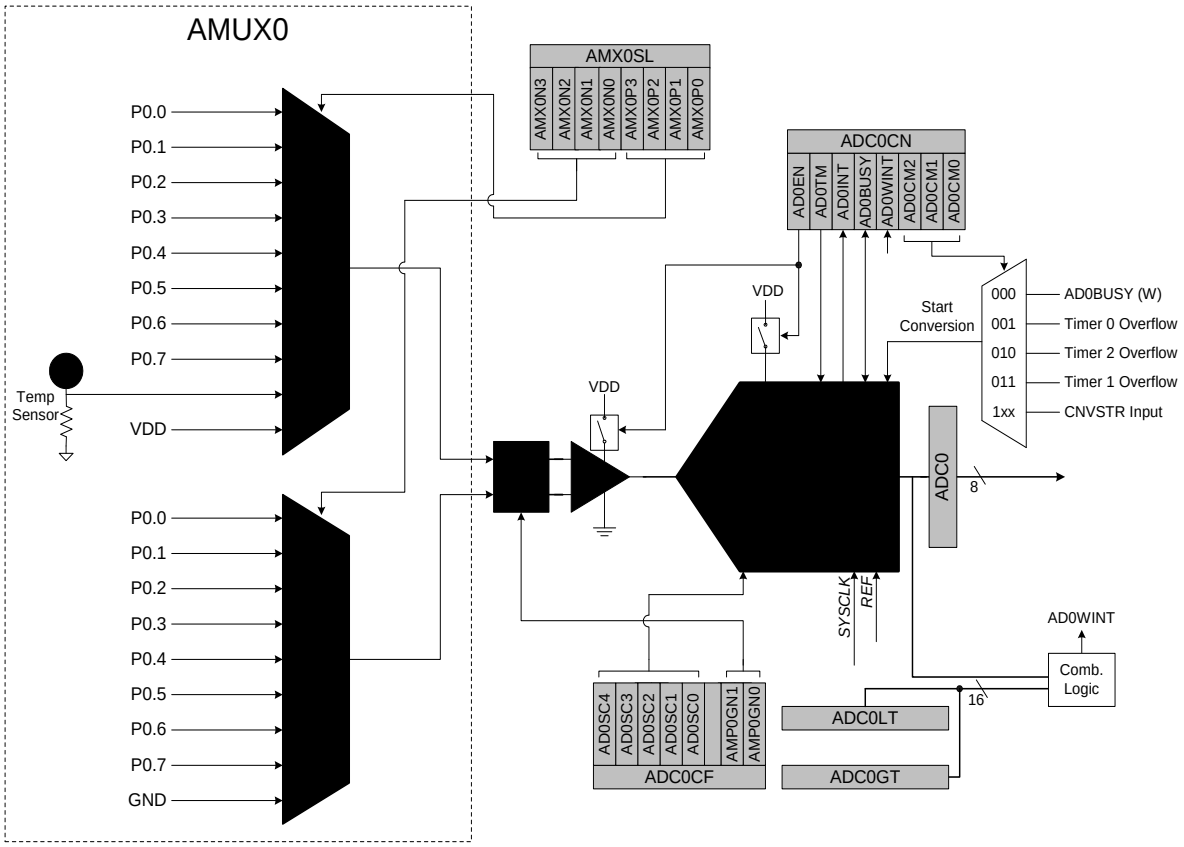


**Figure 5.1. ADC0 Functional Block Diagram**

## SFR Definition 5.2. ADC0CF: ADC0 Configuration (C8051F300/2)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | — | AMP0GN1 | AMP0GN0 | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: 0xBC |

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4-0. SAR Conversion clock requirements are given in Table 5.1.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bit2: UNUSED. Read = 0b; Write = don't care.
Bits1–0: AMP0GN1–0: ADC0 Internal Amplifier Gain (PGA).
00: Gain = 0.5
01: Gain = 1
10: Gain = 2
11: Gain = 4

## SFR Definition 5.3. ADC0: ADC0 Data Word (C8051F300/2)

| 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: 0xBE |

Bits7–0: ADC0 Data Word.
ADC0 holds the output data byte from the last ADC0 conversion. When in Single-ended mode, ADC0 holds an 8-bit unsigned integer. When in Differential mode, ADC0 holds a 2's complement signed 8-bit integer.

## 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.

# 7. Comparator0

C8051F300/1/2/3/4/5 devices include an on-chip programmable voltage comparator, which is shown in Figure 7.1. Comparator0 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), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows Comparator0 to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator0 output may be configured as open drain or push-pull (see **Section “12.2. Port I/O Initialization” on page 106**). Comparator0 may also be used as a reset source (see **Section “9.5. Comparator0 Reset” on page 85**).

The inputs for Comparator0 are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX-0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 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 “12.3. General Purpose Port I/O” on page 108**).

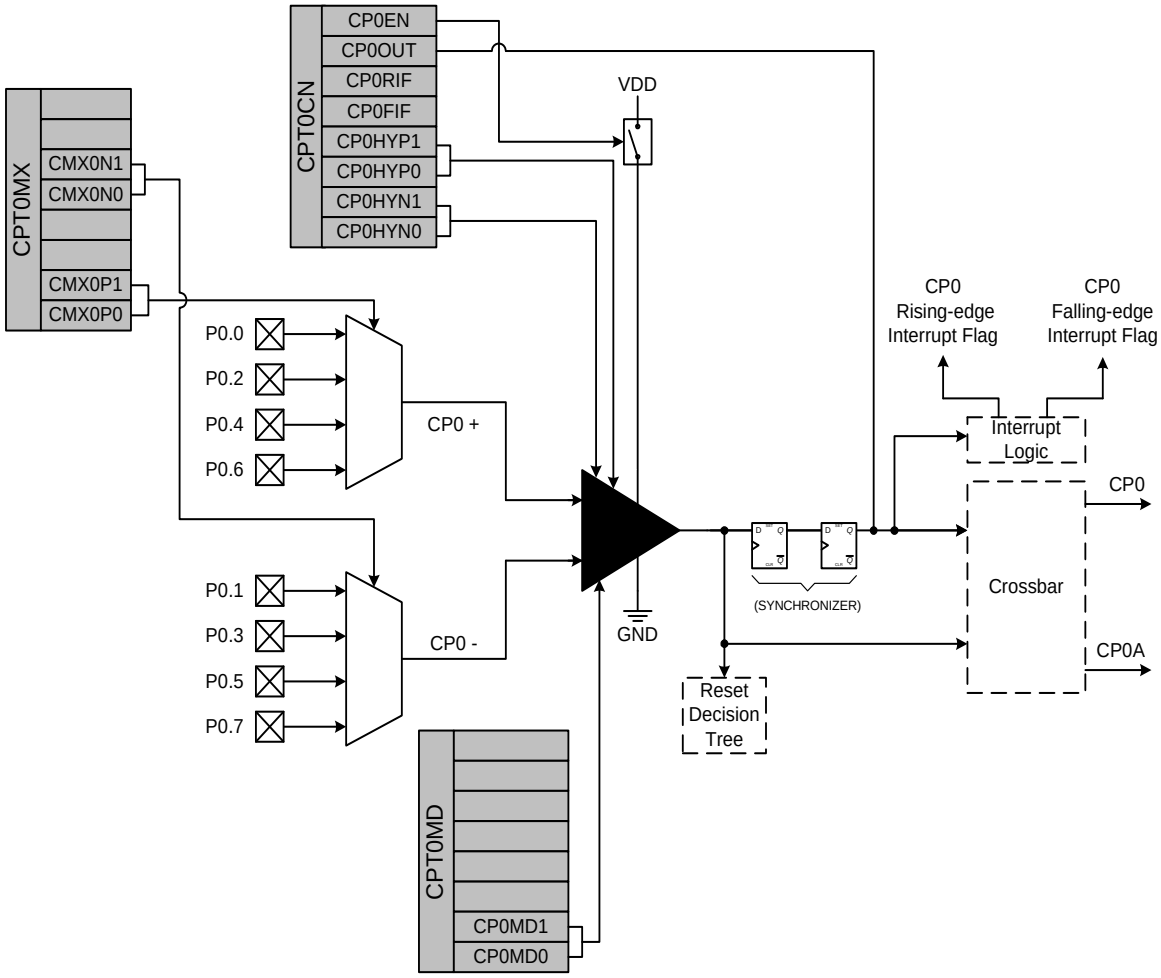


**Figure 7.1. Comparator0 Functional Block Diagram**

**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

**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 C8051F300/1/2/3/4/5 does not support external data or program memory). In the CIP-51, the MOVX instruction accesses the on-chip program memory space implemented as re-programmable Flash memory. This 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 89** 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 |

**Table 8.2. Special Function Register (SFR) Memory Map**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F8 | CPT0CN | PCA0L | PCA0H | PCA0CPL0 | PCA0CPH0 | | | |
| F0 | B | P0MDIN | | | | | EIP1 | |
| E8 | ADC0CN | PCA0CPL1 | PCA0CPH1 | PCA0CPL2 | PCA0CPH2 | | | RSTSRC |
| E0 | ACC | XBR0 | XBR1 | XBR2 | IT01CF | | EIE1 | |
| D8 | PCA0CN | PCA0MD | PCA0CPM0 | PCA0CPM1 | PCA0CPM2 | | | |
| D0 | PSW | REF0CN | | | | | | |
| C8 | TMR2CN | | TMR2RLL | TMR2RLH | TMR2L | TMR2H | | |
| C0 | SMB0CN | SMB0CF | SMB0DAT | | ADC0GT | | ADC0LT | |
| B8 | IP | | | AMX0SL | ADC0CF | | ADC0 | |
| B0 | | OSCXCN | OSCICN | OSCICL | | | FLSCL | FLKEY |
| A8 | IE | | | | | | | |
| A0 | | | | | P0MDOUT | | | |
| 98 | SCON0 | SBUF0 | | | | CPT0MD | | CPT0MX |
| 90 | | | | | | | | |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | PSCTL |
| 80 | P0 | SP | DPL | DPH | | | | PCON |
| | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

(bit addressable)

**Table 8.3. Special Function Registers***

| Register | Address | Description | Page No. |
|---|---|---|---|
| ACC | 0xE0 | Accumulator | 71 |
| ADC0CF | 0xBC | ADC0 Configuration | 43 |
| ADC0CN | 0xE8 | ADC0 Control | 44 |
| ADC0GT | 0xC4 | ADC0 Greater-Than Compare Word | 46 |
| ADC0LT | 0xC6 | ADC0 Less-Than Compare Word | 46 |
| ADC0 | 0xBE | ADC0 Data Word | 43 |
| AMX0SL | 0xBB | ADC0 Multiplexer Channel Select | 42 |
| B | 0xF0 | B Register | 71 |
| CKCON | 0x8E | Clock Control | 149 |
| CPT0CN | 0xF8 | Comparator0 Control | 53 |
| CPT0MD | 0x9D | Comparator0 Mode Selection | 54 |
| CPT0MX | 0x9F | Comparator0 MUX Selection | 54 |
| DPH | 0x83 | Data Pointer High | 69 |
| DPL | 0x82 | Data Pointer Low | 68 |
| EIE1 | 0xE6 | Extended Interrupt Enable 1 | 77 |
| EIP1 | 0xF6 | External Interrupt Priority 1 | 78 |
| FLKEY | 0xB7 | Flash Lock and Key | 93 |

***Note:** SFRs are listed in alphabetical order. All undefined SFR locations are reserved

**Table 8.3. Special Function Registers* (Continued)**

| Register | Address | Description | Page No. |
|---|---|---|---|
| FLSCL | 0xB6 | Flash Scale | 93 |
| IE | 0xA8 | Interrupt Enable | 75 |
| IP | 0xB8 | Interrupt Priority | 76 |
| IT01CF | 0xE4 | INT0/INT1 Configuration Register | 79 |
| OSCICL | 0xB3 | Internal Oscillator Calibration | 98 |
| OSCICN | 0xB2 | Internal Oscillator Control | 98 |
| OSCXCN | 0xB1 | External Oscillator Control | 100 |
| P0 | 0x80 | Port 0 Latch | 109 |
| P0MDIN | 0xF1 | Port 0 Input Mode Configuration | 109 |
| P0MDOUT | 0xA4 | Port 0 Output Mode Configuration | 110 |
| PCA0CN | 0xD8 | PCA Control | 167 |
| PCA0MD | 0xD9 | PCA Mode | 168 |
| PCA0CPH0 | 0xFC | PCA Capture 0 High | 171 |
| PCA0CPH1 | 0xEA | PCA Capture 1 High | 171 |
| PCA0CPH2 | 0xEC | PCA Capture 2 High | 171 |
| PCA0CPL0 | 0xFB | PCA Capture 0 Low | 171 |
| PCA0CPL1 | 0xE9 | PCA Capture 1 Low | 171 |
| PCA0CPL2 | 0xEB | PCA Capture 2 Low | 171 |
| PCA0CPM0 | 0xDA | PCA Module 0 Mode Register | 169 |
| PCA0CPM1 | 0xDB | PCA Module 1 Mode Register | 169 |
| PCA0CPM2 | 0xDC | PCA Module 2 Mode Register | 169 |
| PCA0H | 0xFA | PCA Counter High | 170 |
| PCA0L | 0xF9 | PCA Counter Low | 170 |
| PCON | 0x87 | Power Control | 81 |
| PSCTL | 0x8F | Program Store R/W Control | 92 |
| PSW | 0xD0 | Program Status Word | 70 |
| REF0CN | 0xD1 | Voltage Reference Control | 49 |
| RSTSRC | 0xEF | Reset Source Configuration/Status | 87 |
| SBUF0 | 0x99 | UART 0 Data Buffer | 137 |
| SCON0 | 0x98 | UART 0 Control | 136 |
| SMB0CF | 0xC1 | SMBus Configuration | 118 |
| SMB0CN | 0xC0 | SMBus Control | 120 |
| SMB0DAT | 0xC2 | SMBus Data | 122 |
| SP | 0x81 | Stack Pointer | 69 |
| TMR2CN | 0xC8 | Timer/Counter 2 Control | 154 |
| TCON | 0x88 | Timer/Counter Control | 147 |
| TH0 | 0x8C | Timer/Counter 0 High | 150 |

***Note:** SFRs are listed in alphabetical order. All undefined SFR locations are reserved

## 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.

## SFR Definition 8.11. IT01CF: INT0/INT1 Configuration

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| IN1PL | IN1SL2 | IN1SL1 | IN1SL0 | IN0PL | IN0SL2 | IN0SL1 | IN0SL0 | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | | 0xE4 |

**Note:** Refer to SFR Definition 15.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: /INT1 Polarity
0: /INT1 input is active low.
1: /INT1 input is active high.

Bits6–4: IN1SL2–0: /INT1 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register XBR0).

| IN1SL2–0 | /INT1 Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

Bit3: IN0PL: /INT0 Polarity
0: /INT0 interrupt is active low.
1: /INT0 interrupt is active high.

Bits2–0: INT0SL2–0: /INT0 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar. /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register XBR0).

| IN0SL2–0 | /INT0 Port Pin |
|---|---|
| 000 | P0.0 |
| 001 | P0.1 |
| 010 | P0.2 |
| 011 | P0.3 |
| 100 | P0.4 |
| 101 | P0.5 |
| 110 | P0.6 |
| 111 | P0.7 |

## NOTES:

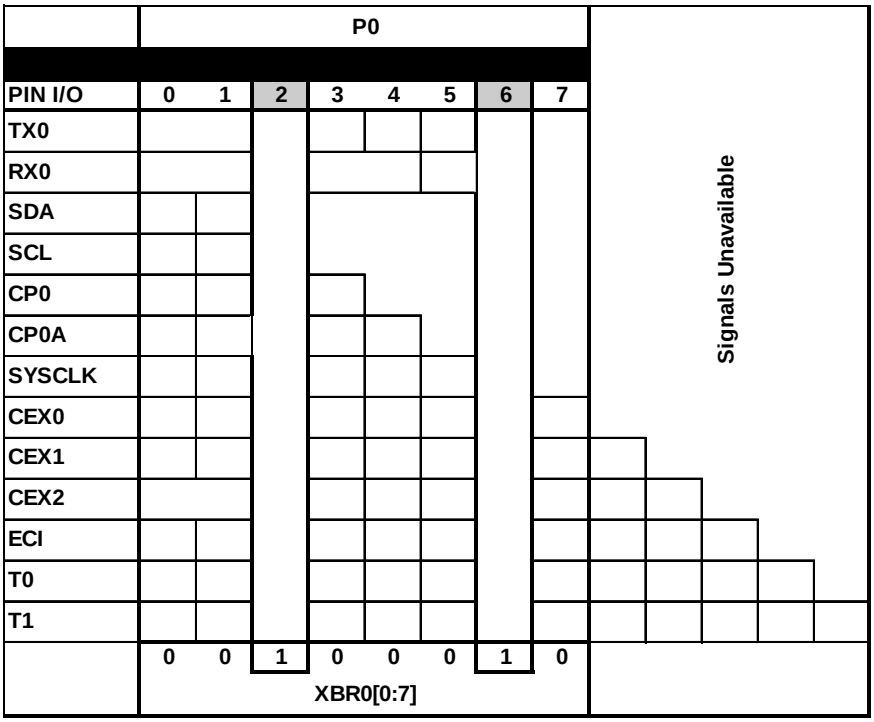


**Figure 12.4. Crossbar Priority Decoder with XBR0 = 0x44**

Registers XBR1 and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL). Either or both of the UART signals may be selected by the Crossbar. UART0 pin assignments are fixed for bootloading purposes: when UART TX0 is selected, it is always assigned to P0.4; when UART RX0 is selected, it is always assigned to P0.5. Standard Port I/Os appear contiguously after the prioritized functions have been assigned. For example, if assigned functions that take the first 3 Port I/O (P0.[2:0]), 5 Port I/O are left for analog or GPIO use.

## SFR Definition 12.1. XBR0: Port I/O Crossbar Register 0

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| — | XSKP6 | XSKP5 | XSKP4 | XSKP3 | XSKP2 | XSKP1 | XSKP0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | | 0xE1 |

Bit7: UNUSED. Read = 0b; Write = don't care.

Bits6–0: XSKP[6:0]: 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 P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

## SFR Definition 12.2. XBR1: Port I/O Crossbar Register 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PCA0ME | | CP0AOEN | CP0OEN | SYSCKE | SMB0OEN | URX0EN | UTX0EN | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | | 0xE2 |

Bits7–6: PCA0ME: PCA Module I/0 Enable Bits
00: All PCA I/O unavailable at Port pins.
01: CEX0 routed to Port pin.
10: CEX0, CEX1 routed to Port pins.
11: CEX0, CEX1, CEX2 routed to Port pins.

Bit5: CP0AOEN: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.

Bit4: CP0OEN: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit3: SYSCKE: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK output routed to Port pin.

Bit2: SMB0OEN: SMBus I/O Enable
0: SMBus I/O unavailable at Port pins.
1: SDA, SCL routed to Port pins.

Bit1: URX0EN: UART RX Enable
0: UART RX0 unavailable at Port pin.
1: UART RX0 routed to Port pin P0.5.

Bit0: UTX0EN: UART TX Output Enable
0: UART TX0 unavailable at Port pin.
1: UART TX0 routed to Port pin P0.4.

### 13.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 13.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER and TXMODE indicate the master/slave state and transmit/receive modes, respectively.

The STA bit indicates that a START has been detected or generated since the last SMBus interrupt. When set to '1', the STA bit will cause the SMBus to enter Master mode and generate a START when the bus becomes free. STA is not cleared by hardware after the START is generated; it must be cleared by software.

As a master, writing the STO bit will cause the hardware to generate a STOP condition and end the current transfer after the next ACK cycle. STO is cleared by hardware after the STOP condition is generated. As a slave, STO indicates that a STOP condition has been detected since the last SMBus interrupt. STO is also used in slave mode to manage the transition from slave receiver to slave transmitter; see **Section 13.5.4** for details on this procedure.

If STO and STA are both set to '1' (while in Master Mode), a STOP followed by a START will be generated.

As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 13.3 for more details.
**Important Note About the SI Bit:** The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

Table 13.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 13.4 for SMBus status decoding using the SMB0CN register.

## 13.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. In the table below, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. Note that the shown response options are only the typical responses; application-specific procedures are allowed as long as they conform with the SMBus specification. Highlighted responses are allowed but do not conform to the SMBus specification.

**Table 13.4. SMBus Status Decoding**

| Mode | Values Read | | | | Current SMbus State | Typical Response Options | Values Written | | |
|---|---|---|---|---|---|---|---|---|---|
| | Status Vector | ACKRQ | ARBLOST | ACK | | | STA | STO | ACK |
| Master Transmitter | 1110 | 0 | 0 | X | A master START was generated. | Load slave address + R/W into SMB0DAT. | 0 | 0 | X |
| | 1100 | 0 | 0 | 0 | A master data or address byte was transmitted; NACK received. | Set STA to restart transfer. | 1 | 0 | X |
| | | | | | | Abort transfer. | 0 | 1 | X |
| | | 0 | 0 | 1 | A master data or address byte was transmitted; ACK received. | Load next data byte into SMB0DAT | 0 | 0 | X |
| | | | | | | End transfer with STOP | 0 | 1 | X |
| | | | | | | End transfer with STOP and start another transfer. | 1 | 1 | X |
| | | | | | | Send repeated START | 1 | 0 | X |
| | | | | | | Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). | 0 | 0 | X |

## NOTES:

## 14.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 14.2), which is not user accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

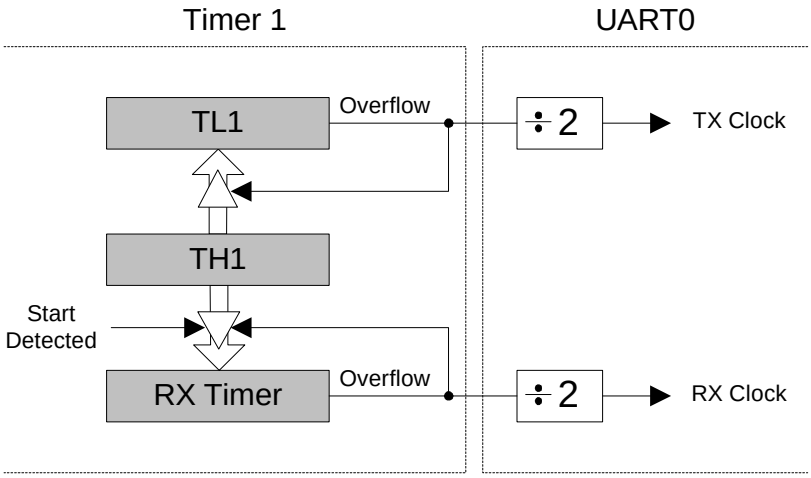


**Figure 14.2. UART0 Baud Rate Logic**

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see **Section "15.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload" on page 145**). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of five sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, or the external oscillator clock / 8. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 14.1.

$$UartBaudRate = \frac{T1_{CLK}}{(256 - T1H)} \times \frac{1}{2}$$

**Equation 14.1. UART0 Baud Rate**

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in **Section "15.2. Timer 2" on page 151**. A quick reference for typical baud rates and system clock frequencies is given in Tables 14.1 through 14.6. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1 (see **Section "15.1. Timer 0 and Timer 1" on page 143** for more details).

## SFR Definition 14.1. SCON0: Serial Port 0 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| S0MODE | — | MCE0 | REN0 | TB80 | RB80 | TI0 | RI0 | 01000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |
| | | | | | | | (bit addressable) | 0x98 |

Bit7: S0MODE: Serial Port 0 Operation Mode.
This bit selects the UART0 Operation Mode.
0: Mode 0: 8-bit UART with Variable Baud Rate
1: Mode 1: 9-bit UART with Variable Baud Rate

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: MCE0: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
Mode 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level 1.
Mode 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software

Bit0: RI0: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

### SFR Definition 15.9. TMR2RLL: Timer 2 Reload Register 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: |
| | | | | | | | | 0xCA |

Bits 7–0: TMR2RLL: Timer 2 Reload Register Low Byte.
TMR2RLL holds the low byte of the reload value for Timer 2.

### SFR Definition 15.10. TMR2RLH: Timer 2 Reload Register 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: |
| | | | | | | | | 0xCB |

Bits 7–0: TMR2RLH: Timer 2 Reload Register High Byte.
The TMR2RLH holds the high byte of the reload value for Timer 2.

### SFR Definition 15.11. TMR2L: Timer 2 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: |
| | | | | | | | | 0xCC |

Bits 7–0: TMR2L: Timer 2 Low Byte.
In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

### SFR Definition 15.12. TMR2H Timer 2 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: |
| | | | | | | | | 0xCD |

Bits 7–0: TMR2H: Timer 2 High Byte.
In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.