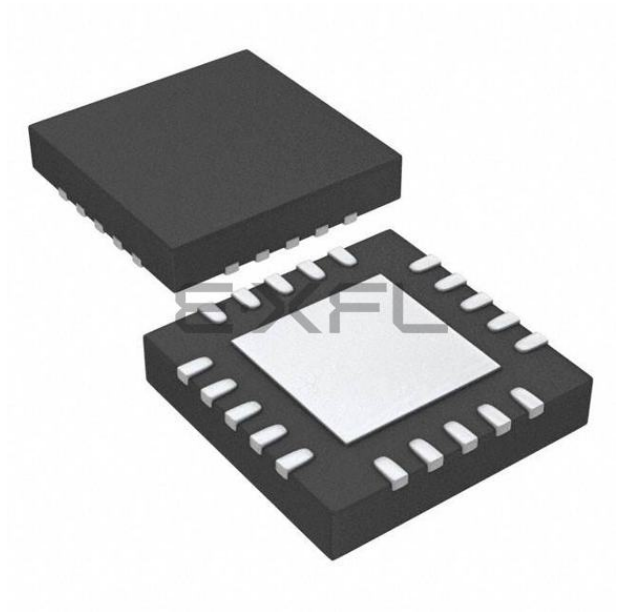

**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 50MHz |
| Connectivity | SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 17 |
| Program Memory Size | 4KB (4K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 16x10b; D/A 2x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-VFQFN Exposed Pad |
| Supplier Device Package | 20-QFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f398-a-gm |

31.3.1. 16-bit Timer with Auto-Reload ... 259
31.3.2. 8-bit Timers with Auto-Reload ... 260
31.3.3. Low-Frequency Oscillator (LFO) Capture Mode ... 261
31.4. Timer 4 ... 265
31.4.1. 16-bit Timer with Auto-Reload ... 265
31.4.2. 8-bit Timers with Auto-Reload ... 266
31.5. Timer 5 ... 270
31.5.1. 16-bit Timer with Auto-Reload ... 270
31.5.2. 8-bit Timers with Auto-Reload ... 271
**32. Programmable Counter Array ... 275**
32.1. PCA Counter/Timer ... 276
32.2. PCA0 Interrupt Sources ... 277
32.3. Capture/Compare Modules ... 278
32.3.1. Edge-triggered Capture Mode ... 279
32.3.2. Software Timer (Compare) Mode ... 280
32.3.3. High-Speed Output Mode ... 281
32.3.4. Frequency Output Mode ... 282
32.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes ... 282
32.3.5.1. 8-bit Pulse Width Modulator Mode ... 283
32.3.5.2. 9/10/11-bit Pulse Width Modulator Mode ... 284
32.3.6. 16-Bit Pulse Width Modulator Mode ... 285
32.4. Watchdog Timer Mode ... 286
32.4.1. Watchdog Timer Operation ... 286
32.4.2. Watchdog Timer Usage ... 287
32.5. Comparator Clear Function ... 288
32.6. Register Descriptions for PCA0 ... 290
**33. C2 Interface ... 297**
33.1. C2 Interface Registers ... 297
33.2. C2 Pin Sharing ... 300
**Document Change List ... 301**
**Contact Information ... 302**

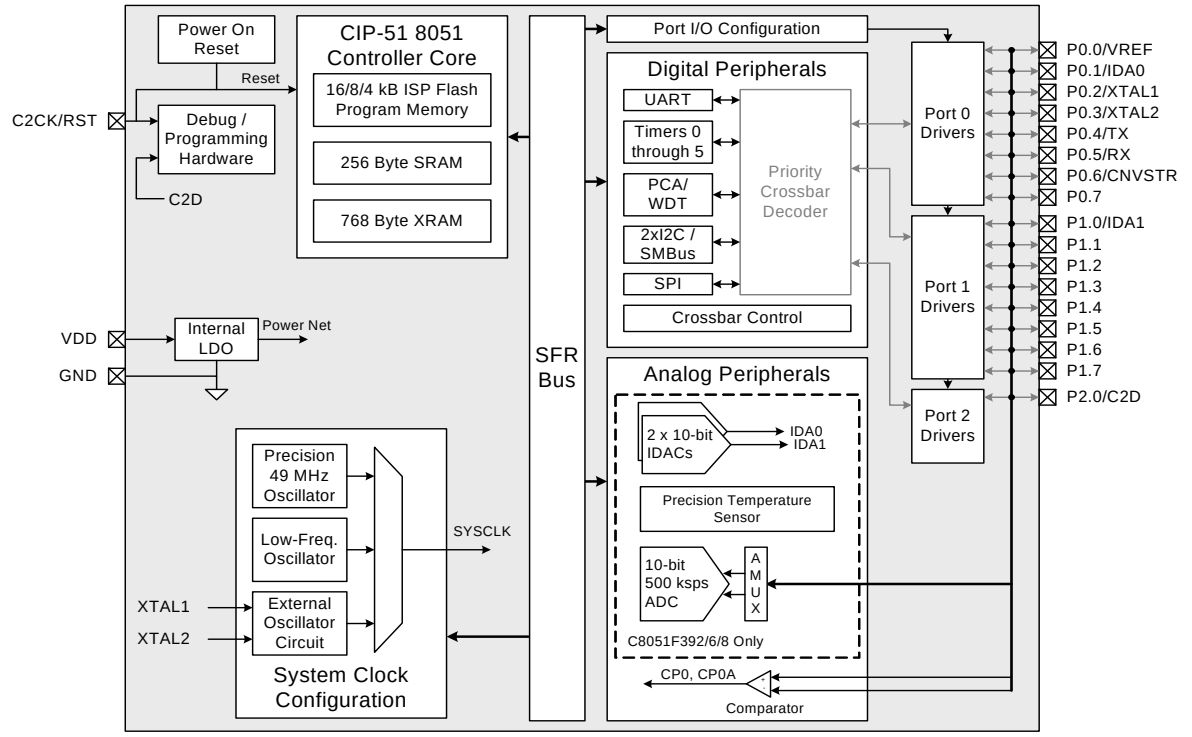


**Figure 1.1. C8051F392/3/6/7/8/9 Block Diagram**

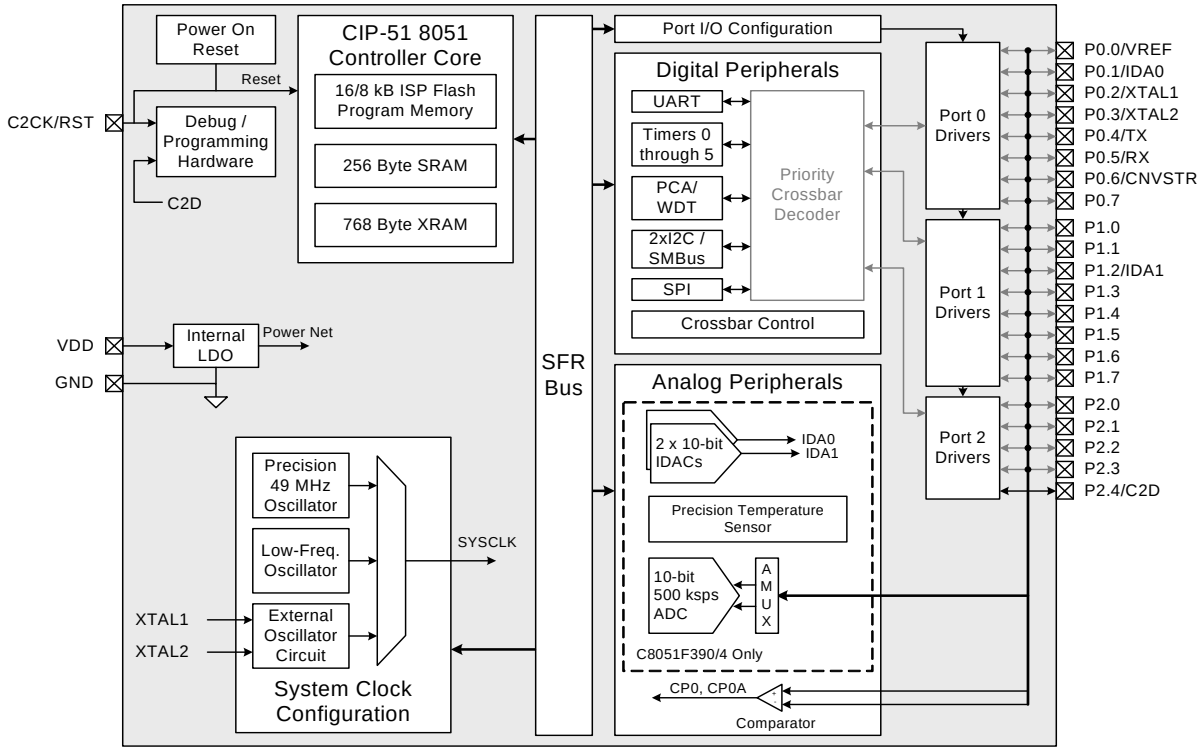


**Figure 1.2. C8051F390/1/4/5 Block Diagram**

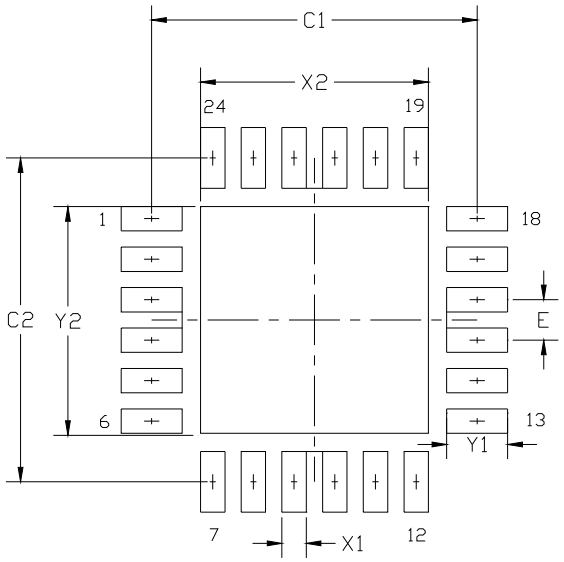


**Figure 6.2. QFN-24 Recommended PCB Land Pattern**

**Table 6.2. QFN-24 PCB Land Pattern Dimensions**

| Dimension | Min | Max |
|---|---|---|
| C1 | 3.90 | 4.00 |
| C2 | 3.90 | 4.00 |
| E | 0.50 BSC | |
| X1 | 0.20 | 0.30 |

| Dimension | Min | Max |
|---|---|---|
| X2 | 2.70 | 2.80 |
| Y1 | 0.65 | 0.75 |
| Y2 | 2.70 | 2.80 |

**Notes:**

**General**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

**Solder Mask Design**

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

**Stencil Design**

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
7. A 2 x 2 array of 1.10 mm x 1.10 mm openings on a 1.30 mm pitch should be used for the center pad.

**Card Assembly**

8. A No-Clean, Type-3 solder paste is recommended.
9. The recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

**Table 7.3. Port I/O DC Electrical Characteristics**

$V_{DD}$ = 1.8 to 3.6 V, –40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), unless otherwise specified.

| Parameters | Test Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| **Standard Port I/O** | | | | | |
| Output High Voltage | $I_{OH}$ = –3 mA, Port I/O push-pull | $V_{DD}$ – 0.7 | — | — | V |
| | $I_{OH}$ = –10 µA, Port I/O push-pull | $V_{DD}$ – 0.1 | — | — | V |
| | $I_{OH}$ = –10 mA, Port I/O push-pull | — | $V_{DD}$ – 0.8 | — | V |
| Output Low Voltage | $I_{OL}$ = 8.5 mA | — | — | 0.6 | V |
| | $I_{OL}$ = 10 µA | — | — | 0.1 | V |
| | $I_{OL}$ = 10 mA, 1.8 V ≤ $V_{DD}$ < 2.7 V | — | 0.8 | — | V |
| | $I_{OL}$ = 25 mA, 2.7 V ≤ $V_{DD}$ ≤ 3.6 V | — | 1.0 | — | V |
| Input High Voltage | 1.8 V ≤ $V_{DD}$ < 2.7 V | $V_{DD}$ – 0.4 | — | — | V |
| | 2.7 V ≤ $V_{DD}$ ≤ 3.6 V | $V_{DD}$ – 0.5 | — | — | V |
| Input Low Voltage | 1.8 V ≤ $V_{DD}$ < 2.7 V | — | — | 0.5 | V |
| | 2.7 V ≤ $V_{DD}$ ≤ 3.6 V | — | — | 0.6 | V |
| Input Leakage Current | Weak Pullup Off | — | — | ±1 | µA |
| | Weak Pullup On, $V_{IN}$ = 0 V | — | 20 | 100 | µA |
| **EESDA and EESCL (C8051F37x Only)*** | | | | | |
| Output Low Voltage (EESDA) | $I_{OL}$ = 0.15 mA, $V_{DD}$ = 1.8 V | — | — | 0.2 | V |
| Output Low Voltage (EESDA) | $I_{OL}$ = 2.1 mA, $V_{DD}$ = 3 V | — | — | 0.4 | V |
| Output Leakage Current (EESDA) | EEPUE = 0, $V_{DD}$ = 3.6 V, 0 V ≤ $V_{OUT}$ ≤ $V_{DD}$ | — | — | 2 | µA |
| Input High Voltage | | $V_{DD}$ x 0.7 | — | — | V |
| Input Low Voltage | | — | — | $V_{DD}$ x 0.3 | V |
| Input Leakage Current | EEPUE = 0, Standby, $V_{DD}$ = 3.6 V, 0 V ≤ $V_{IN}$ ≤ $V_{DD}$ | — | — | ±3 | µA |
| **Note:** Applicable when interfacing to the C8051F37x EEPROM. Otherwise, standard port I/O characteristics apply. | | | | | |

### 9.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked, except when a conversion is in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of three SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR. See Figure 9.2 for track and convert timing details. Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX settings are frequently changed, due to the settling time requirements described in Section “9.2.3. Settling Time Requirements” on page 54.

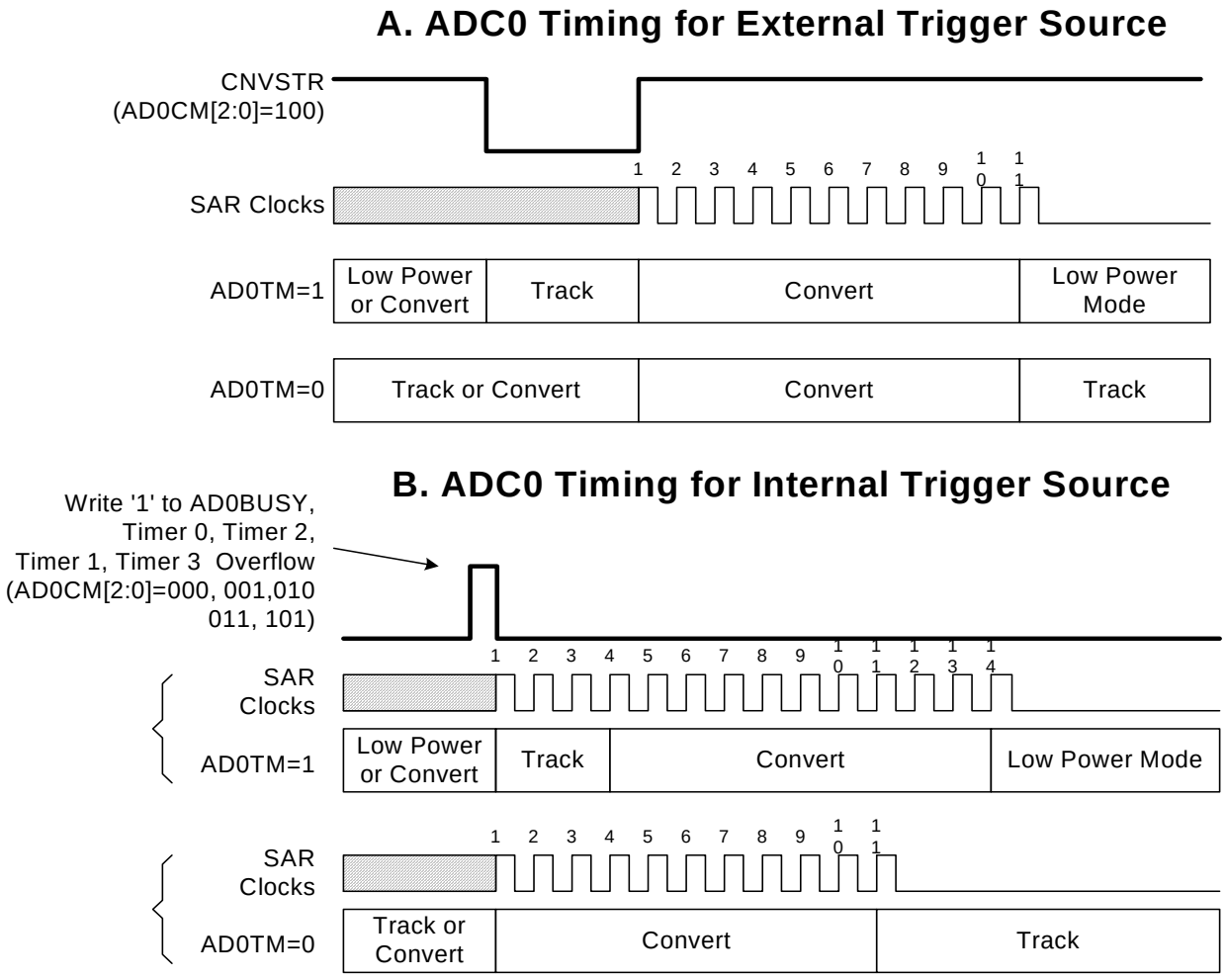


**Figure 9.2. 10-Bit ADC Track and Conversion Example Timing**

# 11. 10-Bit Current Mode DACs (IDA0, IDA1, C8051F390/2/4/6/8 and C8051F370/4 Only)

The C8051F390/2/4/6/8 and C8051F370/4 devices include two 10-bit current-mode Digital-to-Analog Converters (IDACs). The maximum current output of the IDACs can be adjusted for three different current settings; 0.5 mA, 1 mA, and 2 mA. The IDACs are enabled or disabled with the IDAnEN bit in the Control Register for that IDAC (see SFR Definition 11.1 and SFR Definition 11.4). When IDAnEN is set to 0, the IDAC output behaves as a normal GPIO pin. When IDAnEN is set to 1, the digital output drivers and weak pullup for the IDAC pin are automatically disabled, and the pin is connected to the IDAC output. An internal bandgap bias generator is used to generate a reference current for the IDAC whenever it is enabled. When using an IDAC, the crossbar skip functionality should be enabled on the IDAC output pin, to force the Crossbar to skip the output pin.

## 11.1. IDAC Output Scheduling

The IDACs feature a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. Three update modes are provided, allowing IDAC output updates on a write to IDAnH, on a Timer overflow, or on an external pin edge.

### 11.1.1. Update Output On-Demand

In its default mode (IDAnCN.[6:4] = 111) the IDAC output is updated “on-demand” on a write to the high-byte of the IDAC data register (IDAnH). It is important to note that writes to IDAnL are held in this mode, and have no effect on the IDAC output until a write to IDAnH takes place. If writing a full 10-bit word to the IDAC data registers, the 10-bit data word is written to the low byte (IDAnL) and high byte (IDAnH) data registers. Data is latched into the IDAC after a write to the IDAnH register, **so the write sequence should be IDAnL followed by IDAnH** if the full 10-bit resolution is required. The IDAC can be used in 8-bit mode by initializing IDAnL to the desired value (typically 0x00), and writing data to only IDAnH (see Section 11.3 for information on the format of the 10-bit IDAC data word within the 16-bit SFR space).

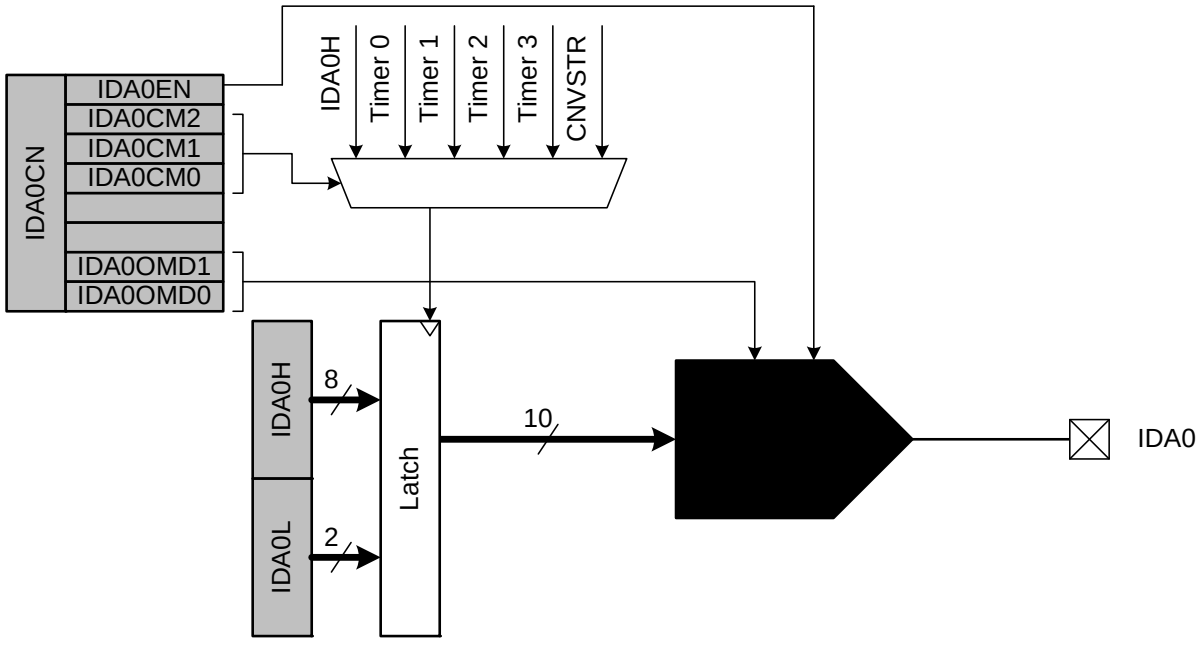


**Figure 11.1. IDA0 Functional Block Diagram**

## SFR Definition 14.3. CPT0MX: Comparator0 MUX Selection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CMX0N[3:0] | | | | CMX0P[3:0] | | | |
| Type | R/W | | | | R/W | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Address = 0x9F; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:4 | CMX0N[3:0] | **Comparator0 Negative Input MUX Selection.**<br>0000: P0.1<br>0001: P0.3<br>0010: P0.5<br>0011: P0.7<br>0100: P1.1<br>0101: P1.3<br>0110: P1.5<br>0111: P1.7<br>1000: P2.1 (C8051F390/1/4/5 and C8051F37x Only)<br>1001: P2.3 (C8051F390/1/4/5 and C8051F37x Only)<br>1010-1111: None |
| 3:0 | CMX0P[3:0] | **Comparator0 Positive Input MUX Selection.**<br>0000: P0.0<br>0001: P0.2<br>0010: P0.4<br>0011: P0.6<br>0100: P1.0<br>0101: P1.2<br>0110: P1.4<br>0111: P1.6<br>1000: P2.0 (C8051F390/1/4/5 and C8051F37x Only)<br>1001: P2.2 (C8051F390/1/4/5 and C8051F37x Only)<br>1010-1111: None |

| Notes on Registers, Operands and Addressing Modes: |
|---|
| **Rn** - Register R0–R7 of the currently selected register bank.<br><br>**@Ri** - Data RAM location addressed indirectly through R0 or R1.<br><br>**rel** - 8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.<br><br>**direct** - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).<br><br>**#data** - 8-bit constant<br><br>**#data16** - 16-bit constant<br><br>**bit** - Direct-accessed bit in Data RAM or SFR<br><br>**addr11** - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.<br><br>**addr16** - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.<br><br>There is one unused opcode (0xA5) that performs the same function as NOP.<br>All mnemonics copyrighted © Intel Corporation 1980. |

## 15.2. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should always be written to the value indicated in the SFR description. Future product versions may use these bits to implement new features in which case the reset value of the bit will be the indicated value, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

## SFR Definition 15.1. DPL: Data Pointer Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | DPL[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0x82; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | DPL[7:0] | **Data Pointer Low.**<br>The DPL register is the low byte of the 16-bit DPTR. |

## SFR Definition 15.2. DPH: Data Pointer High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | DPH[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0x83; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | DPH[7:0] | **Data Pointer High.**<br>The DPH register is the high byte of the 16-bit DPTR. |

**Table 19.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page |
|---|---|---|---|---|
| **PCA0CN** | 0xD8 | All Pages | PCA Control | 290 |
| **PCA0CPH0** | 0xFC | All Pages | PCA Capture 0 High | 296 |
| **PCA0CPH1** | 0xEA | All Pages | PCA Capture 1 High | 296 |
| **PCA0CPH2** | 0xEC | All Pages | PCA Capture 2 High | 296 |
| **PCA0CPL0** | 0xFB | All Pages | PCA Capture 0 Low | 296 |
| **PCA0CPL1** | 0xE9 | All Pages | PCA Capture 1 Low | 296 |
| **PCA0CPL2** | 0xEB | All Pages | PCA Capture 2 Low | 296 |
| **PCA0CPM0** | 0xDA | All Pages | PCA Module 0 Mode Register | 294 |
| **PCA0CPM1** | 0xDB | All Pages | PCA Module 1 Mode Register | 294 |
| **PCA0CPM2** | 0xDC | All Pages | PCA Module 2 Mode Register | 294 |
| **PCA0H** | 0xFA | All Pages | PCA Counter High | 295 |
| **PCA0L** | 0xF9 | All Pages | PCA Counter Low | 295 |
| **PCA0MD** | 0xD9 | All Pages | PCA Mode | 291 |
| **PCA0PWM** | 0xF7 | All Pages | PCA PWM Configuration | 292 |
| **PCON** | 0x87 | All Pages | Power Control | 163 |
| **PFE0CN** | 0xB5 | All Pages | Prefetch Engine Control | 92 |
| **PSCTL** | 0x8F | All Pages | Program Store R/W Control | 137 |
| **PSW** | 0xD0 | All Pages | Program Status Word | 91 |
| **REF0CN** | 0xD1 | All Pages | Voltage Reference Control | 74 |
| **REG0CN** | 0xC9 | All Pages | Voltage Regulator Control | 75 |
| **REVISION** | 0xAC | 0 | Device Revision | 98 |
| **RSTSRC** | 0xEF | All Pages | Reset Source Configuration/Status | 160 |
| **SBUF0** | 0x99 | All Pages | UART0 Data Buffer | 226 |
| **SCON0** | 0x98 | All Pages | UART0 Control | 225 |
| **SFRPAGE** | 0xBF | All Pages | SFR Page | 108 |
| **SFRPGCN** | 0xBF | All Pages | SFR Page Control | 109 |
| **SFRSTACK** | 0xBF | F | SFR Page Stack | 110 |
| **SMB0ADM** | 0xE7 | 0 | SMBus0 Slave Address Mask | 206 |
| **SMB0ADR** | 0xD7 | 0 | SMBus0 Slave Address | 205 |
| **SMB0CF** | 0xC1 | 0 | SMBus0 Configuration | 198 |
| **SMB0CN** | 0xC0 | 0 | SMBus0 Control | 202 |
| **SMB0DAT** | 0xC2 | 0 | SMBus0 Data | 209 |
| **SMB1ADM** | 0xE7 | F | SMBus1 Slave Address Mask | 208 |
| **SMB1ADR** | 0xD7 | F | SMBus1 Slave Address | 207 |

**Table 19.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page |
|---|---|---|---|---|
| **SMB1CF** | 0xC1 | F | SMBus1 Configuration | 199 |
| **SMB1CN** | 0xC0 | F | SMBus1 Control | 203 |
| **SMB1DAT** | 0xC2 | F | SMBus1 Data | 210 |
| **SMBTC** | 0xC7 | All Pages | SMBus Timing Control | 200 |
| **SN0** | 0xAB | F | Serial Number Byte 0 | 100 |
| **SN1** | 0xAC | F | Serial Number Byte 1 | 99 |
| **SN2** | 0xAD | F | Serial Number Byte 2 | 99 |
| **SN3** | 0xAE | F | Serial Number Byte 3 | 98 |
| **SP** | 0x81 | All Pages | Stack Pointer | 89 |
| **SPI0CFG** | 0xA1 | All Pages | SPI Configuration | 235 |
| **SPI0CKR** | 0xA2 | All Pages | SPI Clock Rate Control | 237 |
| **SPI0CN** | 0xF8 | All Pages | SPI Control | 236 |
| **SPI0DAT** | 0xA3 | All Pages | SPI Data | 238 |
| **TCON** | 0x88 | All Pages | Timer/Counter Control | 249 |
| **TH0** | 0x8C | All Pages | Timer/Counter 0 High | 252 |
| **TH1** | 0x8D | All Pages | Timer/Counter 1 High | 252 |
| **TL0** | 0x8A | All Pages | Timer/Counter 0 Low | 251 |
| **TL1** | 0x8B | All Pages | Timer/Counter 1 Low | 251 |
| **TMOD** | 0x89 | All Pages | Timer/Counter Mode | 250 |
| **TMR2CN** | 0xC8 | 0 | Timer/Counter 2 Control | 256 |
| **TMR2H** | 0xCD | 0 | Timer/Counter 2 High | 258 |
| **TMR2L** | 0xCC | 0 | Timer/Counter 2 Low | 257 |
| **TMR2RLH** | 0xCB | 0 | Timer/Counter 2 Reload High | 257 |
| **TMR2RLL** | 0xCA | 0 | Timer/Counter 2 Reload Low | 257 |
| **TMR3CN** | 0x91 | 0 | Timer/Counter 3 Control | 262 |
| **TMR3H** | 0x95 | 0 | Timer/Counter 3 High | 264 |
| **TMR3L** | 0x94 | 0 | Timer/Counter 3 Low | 263 |
| **TMR3RLH** | 0x93 | 0 | Timer/Counter 3 Reload High | 263 |
| **TMR3RLL** | 0x92 | 0 | Timer/Counter 3 Reload Low | 263 |
| **TMR4CN** | 0x91 | F | Timer/Counter 4 Control | 267 |
| **TMR4H** | 0x95 | F | Timer/Counter 4 High | 269 |
| **TMR4L** | 0x94 | F | Timer/Counter 4 Low | 268 |
| **TMR4RLH** | 0x93 | F | Timer/Counter 4 Reload High | 268 |
| **TMR4RLL** | 0x92 | F | Timer/Counter 4 Reload Low | 268 |

## 20.1. MCU Interrupt Sources and Vectors

The C8051F39x/37x MCUs support 18 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 20.2. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

### 20.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of four priority levels. This differs from the traditional two priority levels on the 8051 core. However, the implementation of the extra levels is backwards-compatible with legacy 8051 code.

An interrupt service routine can be preempted by any interrupt of higher priority. Interrupts at the highest priority level cannot be preempted. Each interrupt has two associated priority bits which are used to configure the priority level. For backwards compatibility, the bits are spread across two different registers. The LSBs of the priority setting are stored in the IP, EIP1 and EIP2 registers, while the MSBs are store in the IPH, EIP1H and EIP2H registers. Priority levels according to the MSB and LSB are decoded in Table 20.1. The lowest priority setting is the default for all interrupts. If two or more interrupts are recognized simultaneously, the interrupt with the highest priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 20.2. If legacy 8051 operation is desired, the bits of the “High” priority registers (IPH, EIP1H and EIP2H) should all be configured to 0 (this is the reset value of these registers).

| Priority MSB (from IPH, EIP1H or EIP2H) | Priority LSB (from IP, EIP1 or EIP2) | Priority Level |
|---|---|---|
| 0 | 0 | Priority 0 (lowest priority, default) |
| 0 | 1 | Priority 1 |
| 1 | 0 | Priority 2 |
| 1 | 1 | Priority 3 (highest priority) |

**Table 20.1. Configurable Interrupt Priority Decoding**

### 20.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction. If more than one interrupt is pending when the CPU exits an ISR, the CPU will service the next highest priority interrupt that is pending.

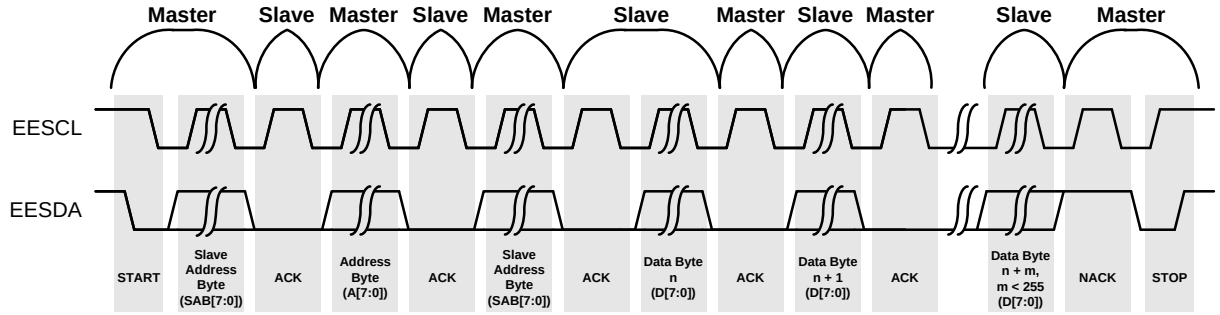


**Figure 22.7. Selective Address Read (Multiple Bytes)**

## 27.4. Port I/O Initialization

Port I/O initialization consists of the following steps:

1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
4. Assign Port pins to desired peripherals.
5. Enable the Crossbar (XBARE = ‘1’).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a ‘1’ indicates a digital input, and a ‘0’ indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 27.8 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is ‘0’, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a ‘0’ to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to ‘1’ enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. Port output drivers are disabled while the Crossbar is disabled.

low. With the associated timer enabled and configured to overflow after 25 ms (and SMBnTOE set), the timer interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

#### 28.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBnFTE bit in SMBnCF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. A clock source is required for free timeout detection, even in a slave-only implementation.

## 28.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information
- Optional hardware recognition of slave address and automatic acknowledgement of address/data

SMBus interrupts are generated for each data byte or slave address that is transferred. When hardware acknowledgment is disabled, the point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e., sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e., receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. If hardware acknowledgment is enabled, these interrupts are always generated after the ACK cycle. See Section 28.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMBnCN (SMBus Control register) to find the cause of the SMBus interrupt. The SMBnCN register is described in Section 28.4.4; Table 28.5 provides a quick SMBnCN decoding reference.

#### 28.4.1. SMBus Configuration Register

The SMBus Configuration register (SMBnCF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

## SFR Definition 28.1. SMB0CF: SMBus Clock/Configuration

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ENSMB0 | INH0 | BUSY0 | EXTHOLD0 | SMB0TOE | SMB0FTE | SMB0CS[1:0] | |
| Type | R/W | R/W | R | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xC1; SFR Page = 0

| Bit | Name | Function |
|---|---|---|
| 7 | ENSMB0 | **SMBus0 Enable.**<br>This bit enables the SMBus0 interface when set to 1. When enabled, the interface constantly monitors the SDA0 and SCL0 pins. |
| 6 | INH0 | **SMBus0 Slave Inhibit.**<br>When this bit is set to logic 1, the SMBus0 does not generate an interrupt when slave events occur. This effectively removes the SMBus0 slave from the bus. Master Mode interrupts are not affected. |
| 5 | BUSY0 | **SMBus0 Busy Indicator.**<br>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. |
| 4 | EXTHOLD0 | **SMBus0 Setup and Hold Time Extension Enable.**<br>This bit controls the SDA0 setup and hold times according to Table 28.2.<br>0: SDA0 Extended Setup and Hold Times disabled.<br>1: SDA0 Extended Setup and Hold Times enabled. |
| 3 | SMB0TOE | **SMBus0 SCL Timeout Detection Enable.**<br>This bit enables SCL low timeout detection. If set to logic 1, the SMBus0 forces Timer 3 to reload while SCL0 is high and allows Timer 3 to count when SCL0 goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL0 is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus0 communication. |
| 2 | SMB0FTE | **SMBus0 Free Timeout Detection Enable.**<br>When this bit is set to logic 1, the bus will be considered free if SCL0 and SDA0 remain high for more than 10 SMBus clock source periods. |
| 1:0 | SMB0CS[1:0] | **SMBus0 Clock Source Selection.**<br>These two bits select the SMBus0 clock source, which is used to generate the SMBus0 bit rate. The selected device should be configured according to Equation 28.1.<br>00: Timer 0 Overflow<br>01: Timer 1 Overflow<br>10: Timer 2 High Byte Overflow<br>11: Timer 2 Low Byte Overflow |

## SFR Definition 28.8. SMB1ADR: SMBus1 Slave Address

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SLV1[6:0] | | | | | | | GC1 |
| Type | R/W | | | | | | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xD7; SFR Page = F

| Bit | Name | Function |
|---|---|---|
| 7:1 | SLV1[6:0] | **SMBus1 Hardware Slave Address.**<br>Defines the SMBus1 Slave Address(es) for automatic hardware acknowledgment. Only address bits which have a 1 in the corresponding bit position in SLVM1[6:0] are checked against the incoming address. This allows multiple addresses to be recognized. |
| 0 | GC1 | **General Call Address Enable.**<br>When hardware address recognition is enabled (EHACK1 = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware.<br>0: General Call Address is ignored.<br>1: General Call Address is recognized. |

# 29. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “29.1. Enhanced Baud Rate Generation” on page 221). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

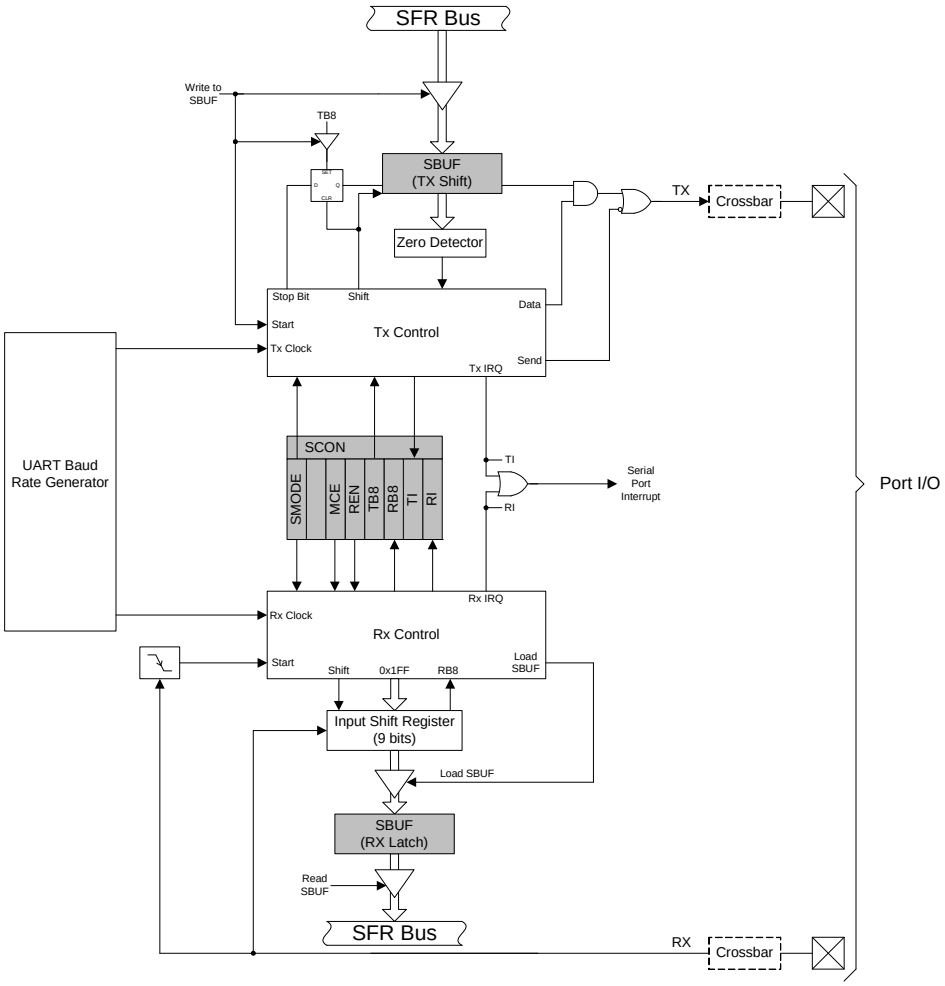


**Figure 29.1. UART0 Block Diagram**

## SFR Definition 31.13. TMR2H Timer 2 High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR2H[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xCD; SFR Page = 0

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR2H[7:0] | **Timer 2 Low Byte.**<br>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. |

## SFR Definition 31.28. TMR5H Timer 5 High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR5H[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xCD; SFR Page = F

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR5H[7:0] | **Timer 5 High Byte.**<br>In 16-bit mode, the TMR5H register contains the high byte of the 16-bit Timer 5. In 8-bit mode, TMR5H contains the 8-bit high byte timer value. |