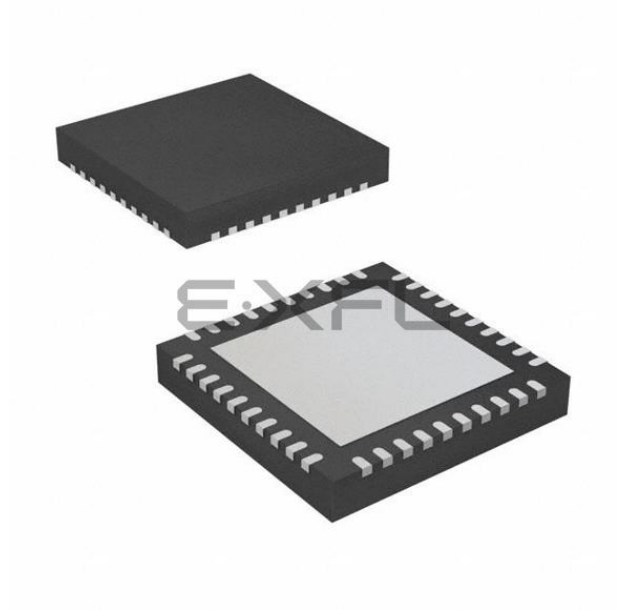

Welcome to E-XFL.COM

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | DMA, LCD, POR, PWM, WDT |
| Number of I/O | 34 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.8V |
| Data Converters | A/D 16x10b/12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 40-VFQFN Exposed Pad |
| Supplier Device Package | 40-QFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f969-a-gmr |

**15. Encoder/Decoder ..... 207**
15.1. Manchester Encoding ..... 208
15.2. Manchester Decoding ..... 209
15.3. Three-out-of-Six Encoding ..... 210
15.4. Three-out-of-Six Decoding ..... 211
15.5. Encoding/Decoding with SFR Access ..... 212
15.6. Decoder Error Interrupt ..... 212
15.7. Using the ENC0 module with the DMA ..... 213
**16. Special Function Registers ..... 216**
16.1. SFR Paging ..... 216
16.2. Interrupts and SFR Paging ..... 216
**17. Interrupt Handler ..... 232**
17.1. Enabling Interrupt Sources ..... 232
17.2. MCU Interrupt Sources and Vectors ..... 232
17.3. Interrupt Priorities ..... 233
17.4. Interrupt Latency ..... 233
17.5. Interrupt Register Descriptions ..... 235
17.6. External Interrupts INT0 and INT1 ..... 242
**18. Flash Memory ..... 244**
18.1. Programming the Flash Memory ..... 244
18.1.1. Flash Lock and Key Functions ..... 244
18.1.2. Flash Erase Procedure ..... 244
18.1.3. Flash Write Procedure ..... 245
18.1.4. Flash Write Optimization ..... 246
18.2. Non-volatile Data Storage ..... 247
18.3. Security Options ..... 247
18.4. Determining the Device Part Number at Run Time ..... 249
18.5. Flash Write and Erase Guidelines ..... 250
18.5.1. VDD Maintenance and the VDD Monitor ..... 250
18.5.2. PSWE Maintenance ..... 251
18.5.3. System Clock ..... 251
18.6. Minimizing Flash Read Current ..... 252
**19. Power Management ..... 257**
19.1. Normal Mode ..... 258
19.2. Idle Mode ..... 258
19.3. Stop Mode ..... 259
19.4. Low Power Idle Mode ..... 259
19.5. Suspend Mode ..... 263
19.6. Sleep Mode ..... 263
19.7. Configuring Wakeup Sources ..... 264
19.8. Determining the Event that Caused the Last Wakeup ..... 264
19.9. Power Management Specifications ..... 268
**20. On-Chip DC-DC Buck Converter (DC0) ..... 269**
20.1. Startup Behavior ..... 270
20.4. Optimizing Board Layout ..... 271

**Table 3.1. Pin Definitions for the C8051F96x (Continued)**

| Name | Pin Numbers | | | Type | Description |
|---|---|---|---|---|---|
| | DQFN76 | TQFP80 | QFN40 | | |
| P6.2 | B12 | 35 | | D I/O or A In | Port 6.2. See Port I/O Section for a complete description. |
| LCD26 | | | | A O | LCD Segment Pin 26 |
| P6.3 | B11 | 33 | | D I/O or A In | Port 6.3. See Port I/O Section for a complete description. |
| LCD27 | | | | A O | LCD Segment Pin 27 |
| P6.4 | B10 | 29 | | D I/O or A In | Port 6.4. See Port I/O Section for a complete description. |
| LCD28 | | | | A O | LCD Segment Pin 28 |
| P6.5 | B9 | 27 | | D I/O or A In | Port 6.5. See Port I/O Section for a complete description. |
| LCD29 | | | | A O | LCD Segment Pin 29 |
| P6.6 | B8 | 25 | | D I/O or A In | Port 6.6. See Port I/O Section for a complete description. |
| LCD30 | | | | A O | LCD Segment Pin 30 |
| P6.7 | D2 | 18 | | D I/O or A In | Port 6.7. See Port I/O Section for a complete description. |
| LCD31 | | | | A O | LCD Segment Pin 31 |

### Table 4.12. ADC0 Electrical Characteristics (Continued)

$V_{BAT}$ = 1.8 to 3.8 V, VREF = 1.65 V (REFSL[1:0] = 11), –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Analog Inputs** | | | | | |
| ADC Input Voltage Range | Single Ended (AIN+ – GND) | 0 | — | $V_{REF}$ | V |
| Absolute Pin Voltage with respect to GND | Single Ended | 0 | — | $V_{BAT}$ | V |
| Sampling Capacitance | 1x Gain<br>0.5x Gain | — | 16<br>13 | — | pF |
| Input Multiplexer Impedance | | — | 5 | — | kΩ |
| **Power Specifications** | | | | | |
| Power Supply Current ($V_{BAT}$ supplied to ADC0) | Normal Power Mode:<br>Conversion Mode (300 ksps)<br>Tracking Mode (0 ksps)<br>Low Power Mode:<br>Conversion Mode (150 ksps)<br>Tracking Mode (0 ksps) | <br>—<br>—<br><br>—<br>— | <br>650<br>740<br><br>370<br>400 | <br>—<br>—<br><br>—<br>— | µA |
| Power Supply Rejection | Internal High Speed VREF<br>External VREF | —<br>— | 67<br>74 | —<br>— | dB |

1. INL and DNL specifications for 12-bit mode do not include the first or last four ADC codes.
2. The maximum code in 12-bit mode is 0xFFFC. The Full Scale Error is referenced from the maximum code.
3. Performance in 8-bit mode is similar to 10-bit mode.

### Table 4.13. Temperature Sensor Electrical Characteristics

$V_{BAT}$ = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Linearity | | — | ±1 | — | °C |
| Slope | | — | 3.40 | — | mV/°C |
| Slope Error* | | — | 40 | — | µV/°C |
| Offset | Temp = 25 °C | — | 1025 | — | mV |
| Offset Error* | Temp = 25 °C | — | 18 | — | mV |
| Temperature Sensor Turn-On Time | | — | 1.7 | — | µs |
| Supply Current | | — | 35 | — | µA |

***Note:** Represents one standard deviation from the mean.

The 16-bit C8051F96x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 16-bit C8051F96x CRC algorithm:

**Table 12.1. Example 16-bit CRC Outputs**

| Input | Output |
|---|---|
| 0x63 | 0xBD35 |
| 0x8C | 0xB1F4 |
| 0x7D | 0x4ECA |
| 0xAA, 0xBB, 0xCC | 0x6CF6 |
| 0x00, 0x00, 0xAA, 0xBB, 0xCC | 0xB166 |

## 13.2. Endianness

The CRC1 module is optimized to process big endian data. Data written to the CRC1IN SFR should be in the normal bit order with the most significant bit stored in bit 7 and the least significant bit stored in bit 0. The input data is shifted left into the CRC engine. The CRC1 module will process one byte at a time and update the results for each byte. When used with the DMA, the first byte to be written should be stored in the lowest address.

Some communications systems may transmit data least significant bit first and may require calculation of a CRC in the transmission bit order. In this case, the bits must be flipped, using the CRC0FLIP SFR, before writing to the CRC1IN SFR. The final 16-bit result may be flipped using the flip bit in the CRC1CN SFR. Note that the polynomial is always written in big endian bit order.

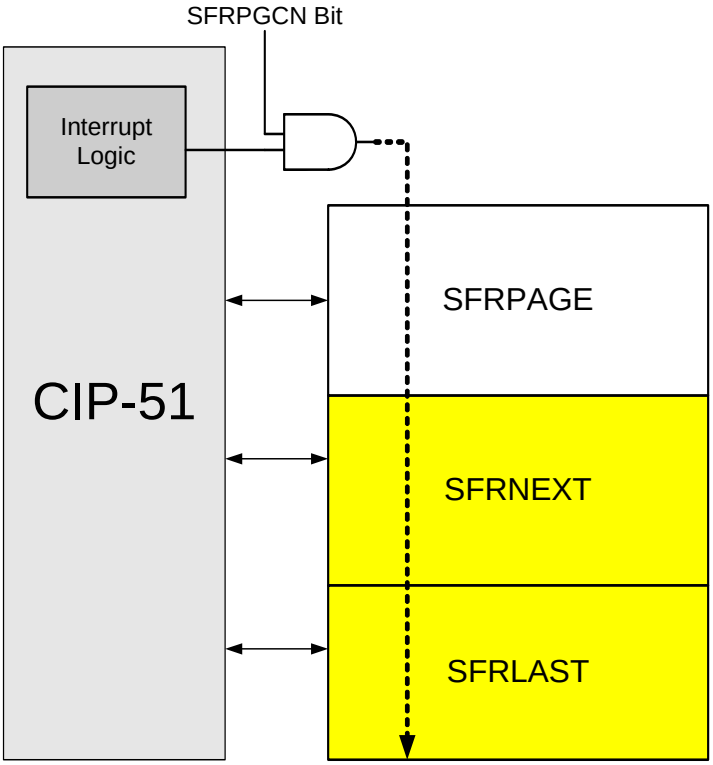


**Figure 16.1. SFR Page Stack**

Automatic hardware preserving and restoring of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to “enabled” upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) are provided in Table 16.3 in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Certain SFRs are accessible from ALL SFR pages, and are denoted by the “(ALL PAGES)” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “(ALL PAGES)” designation, indicating these SFRs are accessible from all SFR pages regardless of the SFRPAGE register value.

**Table 16.3. Special Function Registers**

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

| Register | Address | SFR Page | Description | Page |
|---|---|---|---|---|
| ADC0AC | 0xBA | 0x0 | ADC0 Accumulator Configuration | 88 |
| ADC0CF | 0xBC | 0x0 | ADC0 Configuration | 87 |
| ADC0CN | 0xE8 | All pages | ADC0 Control | 86 |
| ADC0GTH | 0xC4 | 0x0 | ADC0 Greater-Than Compare High | 92 |
| ADC0GTL | 0xC3 | 0x0 | ADC0 Greater-Than Compare Low | 92 |
| ADC0H | 0xBE | 0x0 | ADC0 High | 91 |
| ADC0L | 0xBD | 0x0 | ADC0 Low | 91 |
| ADC0LTH | 0xC6 | 0x0 | ADC0 Less-Than Compare Word High | 93 |
| ADC0LTL | 0xC5 | 0x0 | ADC0 Less-Than Compare Word Low | 93 |
| ADC0MX | 0xBB | 0x0 | ADC0 MUX | 96 |
| ADC0PWR | 0xBA | 0xF | ADC0 Burst Mode Power-Up Time | 89 |
| ADC0TK | 0xBB | 0xF | ADC0 Tracking Control | 90 |
| AES0BCFG | 0xE9 | 0x2 | AES0 Block Configuration | 202 |
| AES0BIN | 0xEB | 0x2 | AES0 Block Input | 204 |
| AES0DCFG | 0xEA | 0x2 | AES0 Data Configuration | 203 |
| AES0KIN | 0xED | 0x2 | AES0 Key Input | 205 |
| AES0XIN | 0xEC | 0x2 | AES0 XOR Input | 205 |
| AES0YOUT | 0xF5 | 0x2 | AES Y Out | 206 |
| CKCON | 0x8E | 0x0 | Clock Control | 445 |
| CLKMODE | 0xFD | 0xF | Clock Mode | 262 |
| CLKSEL | 0xA9 | 0x0 and 0xF | Clock Select | 291 |
| CPT0CN | 0x9B | 0x0 | Comparator0 Control | 108 |
| CPT0MD | 0x9D | 0x0 | Comparator0 Mode Selection | 109 |
| CPT0MX | 0x9F | 0x0 | Comparator0 Mux Selection | 113 |
| CPT1CN | 0x9A | 0x0 | Comparator1 Control | 110 |
| CPT1MD | 0x9C | 0x0 | Comparator1 Mode Selection | 111 |
| CPT1MX | 0x9E | 0x0 | Comparator1 Mux Selection | 114 |
| CRC0AUTO | 0x96 | 0xF | CRC0 Automatic Control | 166 |
| CRC0CNT | 0x97 | 0xF | CRC0 Automatic Flash Sector Count | 166 |
| CRC0CN | 0x92 | 0xF | CRC0 Control | 164 |
| CRC0DAT | 0x91 | 0xF | CRC0 Data | 165 |
| CRC0FLIP | 0x94 | 0xF | CRC0 Flip | 167 |
| CRC0IN | 0x93 | 0xF | CRC0 Input | 165 |

## SFR Definition 19.2. PCLKEN: Peripheral Clock Enable

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

SFR Page = 0xF; SFR Address = 0xFE

| Bit | Name | Function |
|---|---|---|
| 7:4 | Unused | Read = 0b; Write = don't care. |
| 3 | PCLKEN3 | **Clock Enable Controls for Peripherals in Low Power Idle Mode.**<br>0: Disable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode.<br>1: Enable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode. |
| 2 | PCLKEN2 | **Clock Enable Controls for Peripherals in Low Power Idle Mode.**<br>0: Disable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode.<br>1: Enable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode. |
| 1 | PCLKEN1 | **Clock Enable Controls for Peripherals in Low Power Idle Mode.**<br>0: Disableclocks to ADC0 and PCA0 in Low Power Idle Mode.<br>1: Enable clocks to ADC0 and PCA0 in Low Power Idle Mode. |
| 0 | PCLKEN0 | **Clock Enable Controls for Peripherals in Low Power Idle Mode.**<br>0: Disable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode.<br>1: Enable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode. |

## 23.4. Special Function Registers for Selecting and Configuring the System Clock

The clocking sources on C8051F96x devices are enabled and configured using the OSCICN, OSCICL, OSCXCN and the SmaRTClock internal registers. See Section “24. SmaRTClock (Real Time Clock)” on page 295 for SmaRTClock register descriptions. The system clock source for the MCU can be selected using the CLKSEL register. To minimize active mode current, the oneshot timer which sets Flash read time should by bypassed when the system clock is greater than 10 MHz. See the FLSCL register description for details.

The clock selected as the system clock can be divided by 1, 2, 4, 8, 16, 32, 64, or 128. When switching between two clock divide values, the transition may take up to 128 cycles of the undivided clock source. The CLKRDY flag can be polled to determine when the new clock divide value has been applied. The clock divider must be set to "divide by 1" when entering Suspend or Sleep Mode.

The system clock source may also be switched on-the-fly. The switchover takes effect after one clock period of the slower oscillator.

### SFR Definition 23.1. CLKSEL: Clock Select

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

SFR Page = All Pages; SFR Address = 0xA9

| Bit | Name | Function |
|---|---|---|
| 7 | CLKRDY | **System Clock Divider Clock Ready Flag.**<br>0: The selected clock divide setting has not been applied to the system clock.<br>1: The selected clock divide setting has been applied to the system clock. |
| 6:4 | CLKDIV[2:0] | **System Clock Divider Bits.**<br>Selects the clock division to be applied to the undivided system clock source.<br>000: System clock is divided by 1.<br>001: System clock is divided by 2.<br>010: System clock is divided by 4.<br>011: System clock is divided by 8.<br>100: System clock is divided by 16.<br>101: System clock is divided by 32.<br>110: System clock is divided by 64.<br>111: System clock is divided by 128. |
| 3 | Unused | Read = 0b. Must Write 0b. |
| 2:0 | CLKSEL[2:0] | **System Clock Select.**<br>Selects the oscillator to be used as the undivided system clock source.<br>000: Precision Internal Oscillator.<br>001: External Oscillator.<br>010: Low Power Oscillator divided by 8.<br>011: SmaRTClock Oscillator.<br>100: Low Power Oscillator.<br>All other values reserved. |

## Internal Register Definition 24.7. RTC0CF: SmaRTClock Configuration

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | ALRM2 | ALRM1 | ALRM0 | AUTORST | RTC2EN | RTC1EN | RTC0EN |
| Type | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SmaRTClock Address = 0x07

| Bit | Name | Function |
|---|---|---|
| 7 | Reserved | Read = 0b; Must write 0b. |
| 6 | ALRM2 | **Event Flag for Alarm 2.**<br>This bit must be cleared by software. Writing a '1' to this bit has no effect.<br>0: An Alarm 2 event has not occured since the last time the flag was cleared.<br>1: An Alarm 2 event has occured. |
| 5 | ALRM1 | **Event Flag for Alarm 1.**<br>This bit must be cleared by software. Writing a '1' to this bit has no effect.<br>0: An Alarm 1 event has not occured since the last time the flag was cleared.<br>1: An Alarm 1 event has occured. |
| 4 | ALRM0 | **Event Flag for Alarm 0.**<br>This bit must be cleared by software. Writing a '1' to this bit has no effect.<br>0: An Alarm 0 event has not occured since the last time the flag was cleared.<br>1: An Alarm 0 event has occured. |
| 3 | AUTORST | **Auto Reset Enable.**<br>Enables the Auto Reset function to clear the counter when an Alarm 0 event occurs.<br>0: Auto Reset is disabled<br>1: Auto Reset is enabled. |
| 2 | RTC2EN | **Alarm 2 Enable.**<br>0: Alarm 2 is disabled.<br>1: Alarm 2 is enabled. |
| 1 | RTC1EN | **Alarm 1 Enable.**<br>0: Alarm 1 is disabled.<br>1: Alarm 1 is enabled. |
| 0 | RTC0EN | **Alarm 0 Enable.**<br>0: Alarm 0 is disabled.<br>1: Alarm 0 is enabled. |

## SFR Definition 25.20. PC0INT0: PC0 Interrupt 0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CMP1F | CMP1EN | CMP0F | CMP0EN | OVRF | OVREN | DIRCHGF | DIRCHGEN |
| Type | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xFB; SFR Page = 0x2

| Bit | Name | Function |
|---|---|---|
| 7 | CMP1F | **Comparator 1 Flag**<br>0: Counter 1 did not match comparator 1 value.<br>1: Counter 1 matched comparator 1 value. |
| 6 | CMP1EN | **Comparator 1 Interrupt/Wake-up Source Enable**<br>0:CMP1F not enabled as interrupt or wake-up source.<br>1:CMP1F enabled as interrupt or wake-up source. |
| 5 | CMP0F | **Comparator 0 Flag**<br>0: Counter 0 did not match comparator 0 value.<br>1: Counter 0 matched comparator 0 value. |
| 4 | CMP0EN | **Comparator 0 Interrupt/Wake-up Source Enable**<br>0:CMP0F not enabled as interrupt or wake-up source.<br>1:CMP0F enabled as interrupt or wake-up source. |
| 3 | OVRF | **Counter Overflow Flag**<br>1:Neither of the counters has overflowed.<br>1:One of the counters has overflowed. |
| 2 | OVREN | **Counter Overflow Interrupt/Wake-up Source Enable**<br>0:OVRF not enabled as interrupt or wake-up source.<br>1:OVRF enabled as interrupt or wake-up source. |
| 1 | DIRCHGF | **Direction Change Flag**<br>Direction changed for quadrature mode only.<br>0:No change in direction detected.<br>1:Direction Change detected. |
| 0 | DIRCHGEN | **Direction Change Interrupt/Wake-up Source Enable**<br>0:DIRCHGF not enabled as interrupt or wake-up source.<br>1:DIRCHGF enabled as interrupt or wake-up source. |

### 27.1.3. Interfacing Port I/O to High Voltage Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage up to VBAT + 2.0 V. An external pull-up resistor to the higher supply voltage is typically required for most systems.

### 27.1.4. Increasing Port I/O Drive Strength

Port I/O output drivers support a high and low drive strength; the default is low drive strength. The drive strength of a Port I/O can be configured using the PnDRV registers. See Section “4. Electrical Characteristics” on page 56 for the difference in output drive strength between the two modes.

## 27.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P2.6 can be assigned to various analog, digital, and external interrupt functions. The Port pins assigned to analog functions should be configured for analog I/O and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

### 27.2.1. Assigning Port I/O Pins to Analog Functions

Table 27.1 shows all available analog functions that need Port I/O assignments. **Port pins selected for these analog functions should have their digital drivers disabled (PnMDOUT.n = 0 and Port Latch = 1) and their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 27.1 shows the potential mapping of Port I/O to each analog function.

**Table 27.1. Port I/O Assignment for Analog Functions**

| Analog Function | Potentially Assignable Port Pins | SFR(s) used for Assignment |
|---|---|---|
| ADC Input | P0.0–P0.7, P1.4–P2.3 | ADC0MX, PnSKIP |
| Comparator0 Input | P0.0–P0.7, P1.4–P2.3 | CPT0MX, PnSKIP |
| Comparator1 Input | P0.0–P0.7, P1.4–P2.3 | CPT1MX, PnSKIP |
| LCD Pins (LCD0) | P2.4–P6.7 | PnMDIN, PnSKIP |
| Pulse Counter (PC0) | P1.0, P1.1 | P1MDIN, PnSKIP |
| Voltage Reference (VREF0) | P0.0 | REF0CN, PnSKIP |
| Analog Ground Reference (AGND) | P0.1 | REF0CN, PnSKIP |
| Current Reference (IREF0) | P0.7 | IREF0CN, PnSKIP |
| External Oscillator Input (XTAL1) | P0.2 | OSCXCN, PnSKIP |
| External Oscillator Output (XTAL2) | P0.3 | OSCXCN, PnSKIP |
| SmaRTClock Input (XTAL3) | P1.2 | P1MDIN, PnSKIP |
| SmaRTClock Output (XTAL4) | P1.3 | P1MDIN, PnSKIP |

## SFR Definition 27.6. P1MASK: Port1 Mask Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P1MASK[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page= 0x0; SFR Address = 0xBF

| Bit | Name | Function |
|---|---|---|
| 7:0 | P1MASK[7:0] | **Port 1 Mask Value.**<br>Selects P1 pins to be compared to the corresponding bits in P1MAT.<br>0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event.<br>1: P1.n pin logic value is compared to P1MAT.n. |
| **Note:** | | |

## SFR Definition 27.7. P1MAT: Port1 Match Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P1MAT[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Page = 0x0; SFR Address = 0xCF

| Bit | Name | Function |
|---|---|---|
| 7:0 | P1MAT[7:0] | **Port 1 Match Value.**<br>Match comparison value used on Port 1 for bits in P1MASK which are set to 1.<br>0: P1.n pin logic value is compared with logic LOW.<br>1: P1.n pin logic value is compared with logic HIGH. |
| **Note:** | | |

## SFR Definition 27.20. P2MDIN: Port2 Input Mode

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | P2MDIN[6:0] | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Page = 0x0; SFR Address = 0xF3

| Bit | Name | Function |
|---|---|---|
| 7 | Reserved | Read = 1b; Must Write 1b. |
| 6:0 | P2MDIN[3:0] | **Analog Configuration Bits for P2.6–P2.0 (respectively).**<br>Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled.<br>0: Corresponding P2.n pin is configured for analog mode.<br>1: Corresponding P2.n pin is not configured for analog mode. |

## SFR Definition 27.21. P2MDOUT: Port2 Output Mode

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P2MDOUT[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0xA6

| Bit | Name | Function |
|---|---|---|
| 7:0 | P2MDOUT[7:0] | **Output Configuration Bits for P2.7–P2.0 (respectively).**<br>These bits control the digital driver even when the corresponding bit in register P2MDIN is logic 0.<br>0: Corresponding P2.n Output is open-drain.<br>1: Corresponding P2.n Output is push-pull. |

## SFR Definition 28.2. SMB0CN: SMBus Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MASTER | TXMODE | STA | STO | ACKRQ | ARBLOST | ACK | SI |
| Type | R | R | R/W | R/W | R | R | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0xC0; Bit-Addressable

| Bit | Name | Description | Read | Write |
|---|---|---|---|---|
| 7 | MASTER | **SMBus Master/Slave Indicator.** This read-only bit indicates when the SMBus is operating as a master. | 0: SMBus operating in slave mode.<br>1: SMBus operating in master mode. | N/A |
| 6 | TXMODE | **SMBus Transmit Mode Indicator.** This read-only bit indicates when the SMBus is operating as a transmitter. | 0: SMBus in Receiver Mode.<br>1: SMBus in Transmitter Mode. | N/A |
| 5 | STA | **SMBus Start Flag.** | 0: No Start or repeated Start detected.<br>1: Start or repeated Start detected. | 0: No Start generated.<br>1: When Configured as a Master, initiates a START or repeated START. |
| 4 | STO | **SMBus Stop Flag.** | 0: No Stop condition detected.<br>1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode). | 0: No STOP condition is transmitted.<br>1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle.<br>Cleared by Hardware. |
| 3 | ACKRQ | **SMBus Acknowledge Request.** | 0: No Ack requested<br>1: ACK requested | N/A |
| 2 | ARBLOST | **SMBus Arbitration Lost Indicator.** | 0: No arbitration error.<br>1: Arbitration Lost | N/A |
| 1 | ACK | **SMBus Acknowledge.** | 0: NACK received.<br>1: ACK received. | 0: Send NACK<br>1: Send ACK |
| 0 | SI | **SMBus Interrupt Flag.**<br>This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled. | **0: No interrupt pending**<br>1: Interrupt Pending | 0: Clear interrupt, and initiate next state machine event.<br>1: Force interrupt. |

**Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)**

| Mode | Values Read | | | | Current SMbus State | Typical Response Options | Values to Write | | | Next Status Vector Expected |
|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |
| | 1100 | 0 | 0 | 0 | A master data or address byte was transmitted; NACK received. | Set STA to restart transfer. | 1 | 0 | X | 1110 |
| | | | | | | 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 | 1100 |
| | | | | | | End transfer with STOP. | 0 | 1 | X | - |
| | | | | | | End transfer with STOP and start another transfer. | 1 | 1 | X | - |
| | | | | | | Send repeated START. | 1 | 0 | X | 1110 |
| | | | | | | Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). | 0 | 0 | X | 1000 |
| Master Receiver | 1000 | 1 | 0 | X | A master data byte was received; ACK requested. | Acknowledge received byte; Read SMB0DAT. | 0 | 0 | 1 | 1000 |
| | | | | | | Send NACK to indicate last byte, and send STOP. | 0 | 1 | 0 | - |
| | | | | | | Send NACK to indicate last byte, and send STOP followed by START. | 1 | 1 | 0 | 1110 |
| | | | | | | Send ACK followed by repeated START. | 1 | 0 | 1 | 1110 |
| | | | | | | Send NACK to indicate last byte, and send repeated START. | 1 | 0 | 0 | 1110 |
| | | | | | | Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI). | 0 | 0 | 1 | 1100 |
| | | | | | | Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI). | 0 | 0 | 0 | 1100 |

**Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)**

| Mode | Values Read | | | | Current SMbus State | Typical Response Options | Values to Write | | | Next Status Vector Expected |
|---|---|---|---|---|---|---|---|---|---|---|
| | Status Vector | ACKRQ | ARBLOST | ACK | | | STA | STO | ACK | |
| Slave Transmitter | 0100 | 0 | 0 | 0 | A slave byte was transmitted; NACK received. | No action required (expecting STOP condition). | 0 | 0 | X | 0001 |
| | | 0 | 0 | 1 | A slave byte was transmitted; ACK received. | Load SMB0DAT with next data byte to transmit. | 0 | 0 | X | 0100 |
| | | 0 | 1 | X | A Slave byte was transmitted; error detected. | No action required (expecting Master to end transfer). | 0 | 0 | X | 0001 |
| | 0101 | 0 | X | X | An illegal STOP or bus error was detected while a Slave Transmission was in progress. | Clear STO. | 0 | 0 | X | - |
| Slave Receiver | 0010 | 1 | 0 | X | A slave address + R/W was received; ACK requested. | If Write, Acknowledge received address | 0 | 0 | 1 | 0000 |
| | | | | | | If Read, Load SMB0DAT with data byte; ACK received address | 0 | 0 | 1 | 0100 |
| | | | | | | NACK received address. | 0 | 0 | 0 | - |
| | | 1 | 1 | X | Lost arbitration as master; slave address + R/W received; ACK requested. | If Write, Acknowledge received address | 0 | 0 | 1 | 0000 |
| | | | | | | If Read, Load SMB0DAT with data byte; ACK received address | 0 | 0 | 1 | 0100 |
| | | | | | | NACK received address. | 0 | 0 | 0 | - |
| | | | | | | Reschedule failed transfer; NACK received address. | 1 | 0 | 0 | 1110 |
| | 0001 | 0 | 0 | X | A STOP was detected while addressed as a Slave Transmitter or Slave Receiver. | Clear STO. | 0 | 0 | X | - |
| | | 1 | 1 | X | Lost arbitration while attempting a STOP. | No action required (transfer complete/aborted). | 0 | 0 | 0 | - |
| | 0000 | 1 | 0 | X | A slave byte was received; ACK requested. | Acknowledge received byte; Read SMB0DAT. | 0 | 0 | 1 | 0000 |
| | | | | | | NACK received byte. | 0 | 0 | 0 | - |
| Bus Error Condition | 0010 | 0 | 1 | X | Lost arbitration while attempting a repeated START. | Abort failed transfer. | 0 | 0 | X | - |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X | 1110 |
| | 0001 | 0 | 1 | X | Lost arbitration due to a detected STOP. | Abort failed transfer. | 0 | 0 | X | - |
| | | | | | | Reschedule failed transfer. | 1 | 0 | X | 1110 |
| | 0000 | 1 | 1 | X | Lost arbitration while transmitting a data byte as master. | Abort failed transfer. | 0 | 0 | 0 | - |
| | | | | | | Reschedule failed transfer. | 1 | 0 | 0 | 1110 |

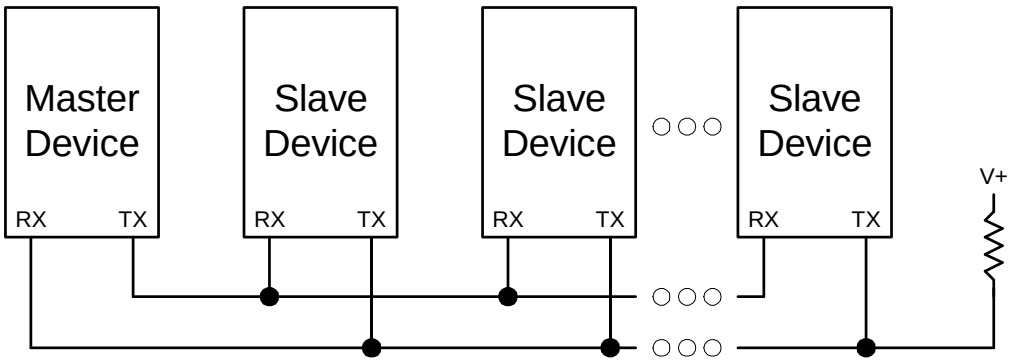


**Figure 29.6. UART Multi-Processor Mode Interconnect Diagram**

## SFR Definition 32.16. TMR3L: Timer 3 Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3L[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0x94

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3L[7:0] | **Timer 3 Low Byte.**<br>In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value. |

## SFR Definition 32.17. TMR3H Timer 3 High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3H[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0x95

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3H[7:0] | **Timer 3 High Byte.**<br>In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value. |

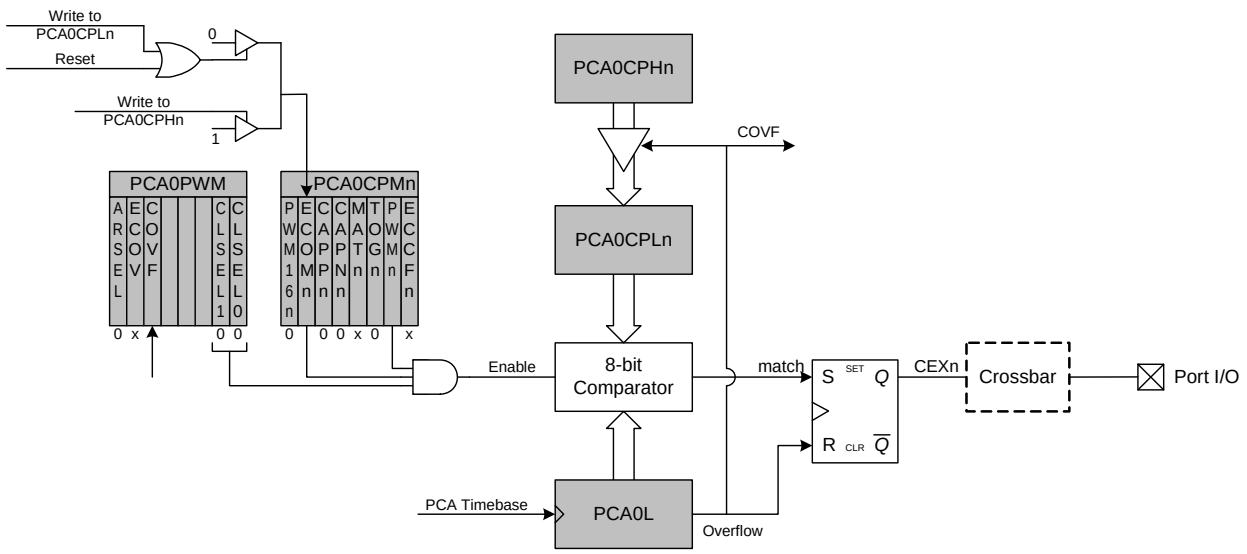


**Figure 33.8. PCA 8-Bit PWM Mode Diagram**

### 33.3.5.2. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an "Auto-Reload" Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module's capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 33.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module's auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 33.3, where N is the number of bits in the PWM cycle.

**Important Note About PCA0CPHn and PCA0CPLn Registers**: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(2^N - PCA0CPn)}{2^N}$$

**Equation 33.3. 9, 10, and 11-Bit PWM Duty Cycle**

A 0% duty cycle may be generated by clearing the ECOMn bit to 0.