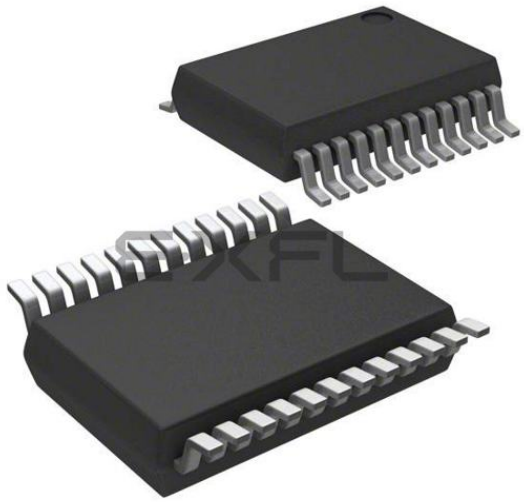


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT |
| Number of I/O | 16 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 768 x 8 |
| Voltage - Supply (Vcc/Vdd) | 0.9V ~ 3.6V |
| Data Converters | A/D 15x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 24-SSOP (0.154", 3.90mm Width) |
| Supplier Device Package | 24-QSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f901-gur |

## List of Tables

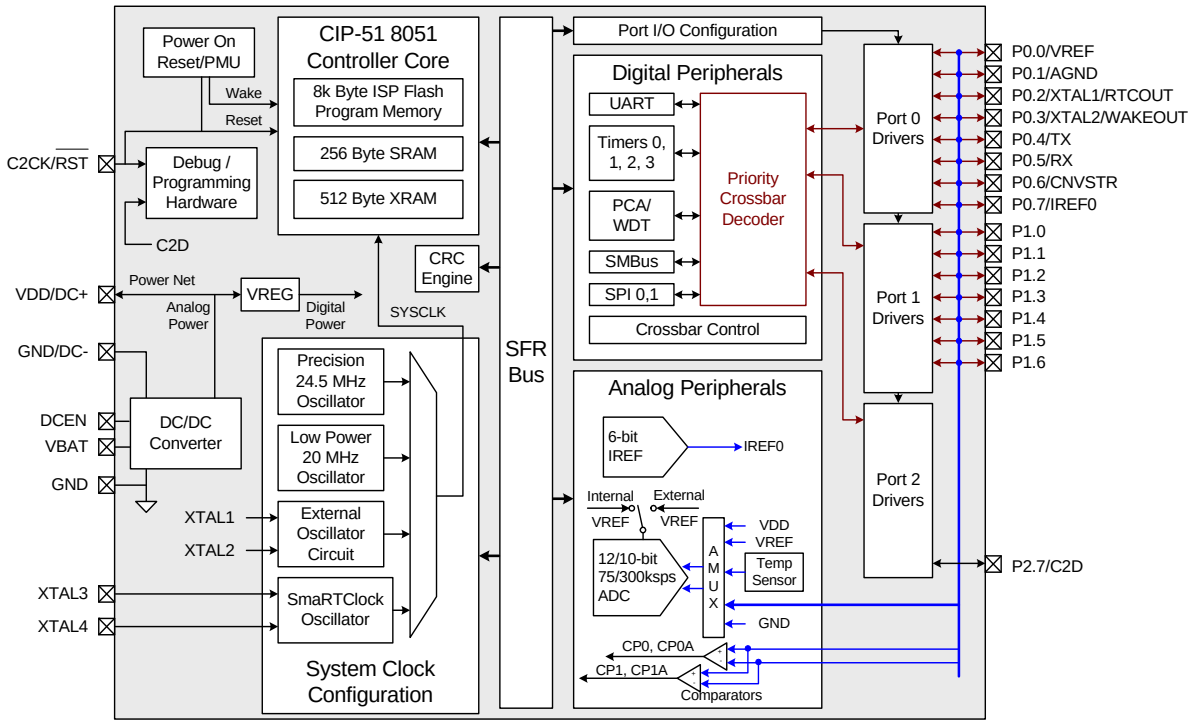



**Figure 1.3. C8051F902 Block Diagram**

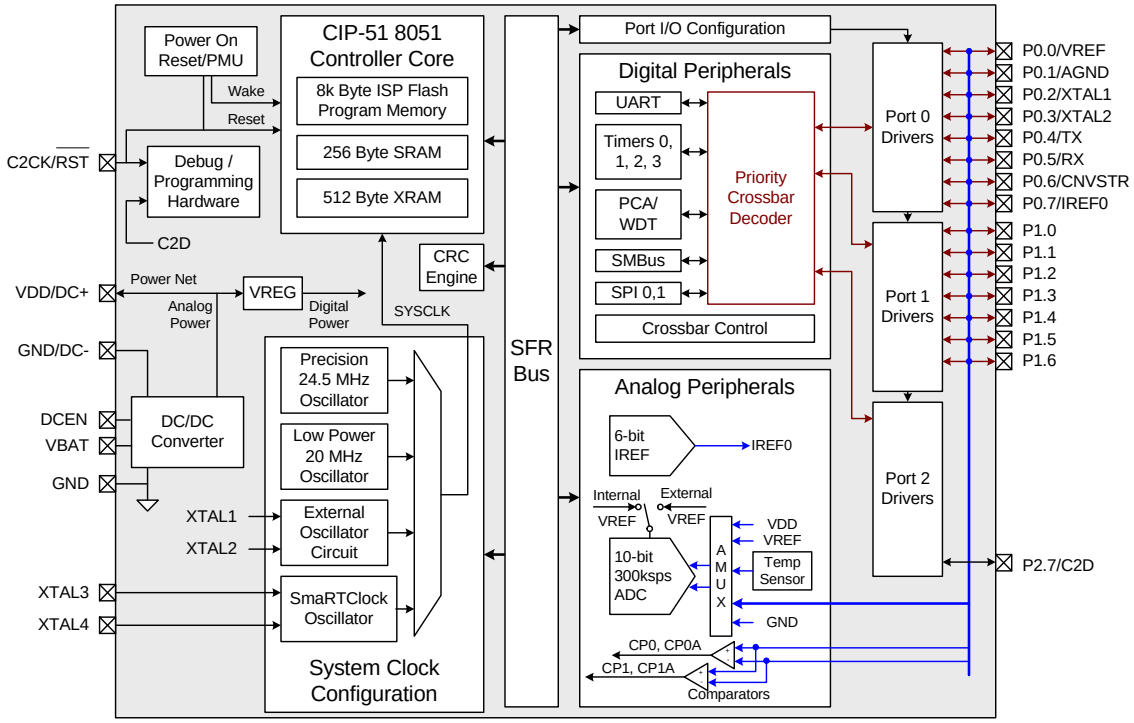



**Figure 1.4. C8051F901 Block Diagram**

### Table 4.12. Voltage Reference Electrical Characteristics

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

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Internal High Speed Reference (REFSL[1:0] = 11)** | | | | | |
| Output Voltage | –40 to +85 °C, $V_{DD}$ = 1.8–3.6 V | 1.60 | 1.65 | 1.70 | V |
| VREF Turn-on Time | | — | — | 1.5 | µs |
| Supply Current | Normal Power Mode<br>**Low Power Mode** | —<br>— | 260<br>**140** | —<br>— | µA |
| **Internal Precision Reference (REFSL[1:0] = 00, REFOE = 1)** | | | | | |
| Output Voltage | –40 to +85 °C, $V_{DD}$ = 1.8–3.6 V | 1.645 | 1.680 | 1.715 | V |
| VREF Short-Circuit Current | | — | 10 | — | mA |
| Load Regulation | Load = 0 to 200 µA to AGND | — | 400 | — | µV/µA |
| VREF Turn-on Time 1 | 4.7 µF tantalum, 0.1 µF ceramic bypass, settling to 0.5 LSB | — | 15 | — | ms |
| VREF Turn-on Time 2 | 0.1 µF ceramic bypass, settling to 0.5 LSB | — | 300 | — | µs |
| VREF Turn-on Time 3 | no bypass cap, settling to 0.5 LSB | — | 25 | — | µs |
| Supply Current | | — | 15 | — | µA |
| **External Reference (REFSL[1:0] = 00, REFOE = 0)** | | | | | |
| Input Voltage Range | | 0 | — | $V_{DD}$ | V |
| Input Current | Sample Rate = 300 ksps; VREF = 3.0 V | — | 5.25 | — | µA |

**Note:** **Blue** indicates a feature only available on ‘F912 and ‘F902 devices.

### Table 4.13. IREF0 Electrical Characteristics

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

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Static Performance** | | | | | |
| Resolution[1] | | | 6 | | bits |
| Output Compliance Range | Low Power Mode, Source<br>High Current Mode, Source<br>Low Power Mode, Sink<br>High Current Mode, Sink | 0<br>0<br>0.3<br>0.8 | —<br>—<br>—<br>— | $V_{DD}$ – 0.4<br>$V_{DD}$ – 0.8<br>$V_{DD}$<br>$V_{DD}$ | V |
| Integral Nonlinearity | | — | <±0.2 | ±1.0 | LSB |
| Differential Nonlinearity | | — | <±0.2 | ±1.0 | LSB |
| Offset Error | | — | <±0.1 | ±0.5 | LSB |
| Full Scale Error[2] | Low Power Mode, Source | — | — | ±5 | % |
| | High Current Mode, Source | — | — | ±6 | % |
| | Low Power Mode, Sink | — | — | ±8 | % |
| | High Current Mode, Sink | — | — | ±8 | % |
| Absolute Current Error | Low Power Mode<br>Sourcing 20 µA | — | <±1 | ±3 | % |
| **Dynamic Performance** | | | | | |
| Output Settling Time to 1/2 LSB | | — | 300 | — | ns |
| Startup Time | | — | 1 | — | µs |
| **Power Consumption** | | | | | |
| Net Power Supply Current ($V_{DD}$ supplied to IREF0 minus any output source current) | Low Power Mode, Source | | | | |
| | IREF0DAT = 000001 | — | 10 | — | µA |
| | IREF0DAT = 111111 | — | 10 | — | µA |
| | High Current Mode, Source | | | | |
| | IREF0DAT = 000001 | — | 10 | — | µA |
| | IREF0DAT = 111111 | — | 10 | — | µA |
| | Low Power Mode, Sink | | | | |
| | IREF0DAT = 000001 | — | 1 | — | µA |
| | IREF0DAT = 111111 | — | 11 | — | µA |
| | High Current Mode, Sink | | | | |
| | IREF0DAT = 000001 | — | 12 | — | µA |
| | IREF0DAT = 111111 | — | 81 | — | µA |

**Notes:**
1. Refer to “PWM Enhanced Mode” on page 86 for information on how to improve IREF0 resolution.
2. Full scale is 63 µA in Low Power Mode and 504 µA in High Power Mode.

### 5.2.3. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a low power state, accumulates 1, 4, 8, 16, 32, or 64 using an internal Burst Mode clock (approximately 20 MHz), then re-enters a low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

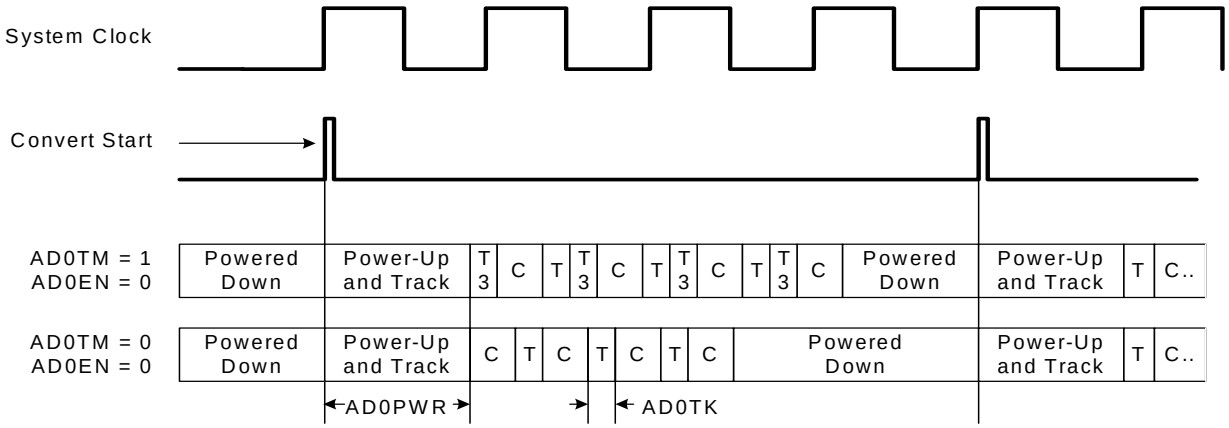
Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 5.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

In Burst Mode, tracking is determined by the settings in AD0PWR and AD0TK. The default settings for these registers will work in most applications without modification; however, settling time requirements may need adjustment in some applications. Refer to “5.2.4. Settling Time Requirements” on page 66 for more details.

**Notes:**

- Setting AD0TM to 1 will insert an additional 3 SAR clocks of tracking before each conversion, regardless of the settings of AD0PWR and AD0TK.
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.




T = Tracking set by AD0TK
T3 = Tracking set by AD0TM (3 SAR clocks)
C = Converting

**Figure 5.3. Burst Mode Tracking Example with Repeat Count Set to 4**

## 5.6. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

### SFR Definition 5.8. ADC0GTH: ADC0 Greater-Than High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AD0GT[15:8] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Page = 0x0; SFR Address = 0xC4

| Bit | Name | Function |
|---|---|---|
| 7:0 | AD0GT[15:8] | **ADC0 Greater-Than High Byte.**<br>Most Significant Byte of the 16-bit Greater-Than window compare register. |

### SFR Definition 5.9. ADC0GTL: ADC0 Greater-Than Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AD0GT[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Page = 0x0; SFR Address = 0xC3

| Bit | Name | Function |
|---|---|---|
| 7:0 | AD0GT[7:0] | **ADC0 Greater-Than Low Byte.**<br>Least Significant Byte of the 16-bit Greater-Than window compare register. |
| **Note:** In 8-bit mode, this register should be set to 0x00. | | |

# 6. Programmable Current Reference (IREF0)

C8051F91x-C8051F90x devices include an on-chip programmable current reference (source or sink) with two output current settings: Low Power Mode and High Current Mode. The maximum current output in Low Power Mode is 63 µA (1 µA steps) and the maximum current output in High Current Mode is 504 µA (8 µA steps).

The current source/sink is controlled though the IREF0CN special function register. It is enabled by setting the desired output current to a non-zero value. It is disabled by writing 0x00 to IREF0CN. The port I/O pin associated with ISRC0 should be configured as an analog input and skipped in the Crossbar. See Section "21. Port Input/Output" on page 205 for more details.

## SFR Definition 6.1. IREF0CN: Current Reference Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SINK | MODE | IREF0DAT | | | | | |
| Type | R/W | R/W | R/W | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0xB9

| Bit | Name | Function |
|---|---|---|
| 7 | SINK | **IREF0 Current Sink Enable.**<br>Selects if IREF0 is a current source or a current sink.<br>0: IREF0 is a current source.<br>1: IREF0 is a current sink. |
| 6 | MDSEL | **IREF0 Output Mode Select.**<br>Selects Low Power or High Current Mode.<br>0: Low Power Mode is selected (step size = 1 µA).<br>1: High Current Mode is selected (step size = 8 µA). |
| 5:0 | IREF0DAT[5:0] | **IREF0 Data Word.**<br>Specifies the number of steps required to achieve the desired output current.<br>Output current = direction x step size x IREF0DAT.<br>IREF0 is in a low power state when IREF0DAT is set to 0x00. |

## 6.1. PWM Enhanced Mode

On 'F912 and 'F902 devices, the precision of the current reference can be increased by fine tuning the IREF0 output using a PWM signal generated by the PCA. This mode allows the IREF0DAT bits to perform a course adjustment on the IREF0 output. Any available PCA channel can perform a fine adjustment on the IREF0 output. When enabled (PWMEN = 1), the CEX signal selected using the PWMSS bit field is internally routed to IREF0 to control the on time of a current source having the weight of 2 LSBs. With the two least significant bits of IREF0DAT set to 00b, applying a 100% duty cycle on the CEX signal will be equivalent to setting the two LSBs of IREF0DAT to 10b. PWM enhanced mode is enabled and setup using the IREF0CF register.

## 7.6. Comparator0 and Comparator1 Analog Multiplexers

Comparator0 and Comparator1 on C8051F91x-C8051F90x devices have analog input multiplexers to connect Port I/O pins and internal signals the comparator inputs; CP0+/CP0- are the positive and negative input multiplexers for Comparator0 and CP1+/CP1- are the positive and negative input multiplexers for Comparator1.

The comparator input multiplexers directly support capacitive touch switches. When the Capacitive Touch Sense Compare input is selected on the positive or negative multiplexer, any Port I/O pin connected to the other multiplexer can be directly connected to a capacitive touch switch with no additional external components. The Capacitive Touch Sense Compare provides the appropriate reference level for detecting when the capacitive touch switches have charged or discharged through the on-chip Rsense resistor. The Comparator outputs can be routed to Timer2 or Timer3 for capturing sense capacitor's charge and discharge time. See Section "25. Timers" on page 270 for details.

Any of the following may be selected as comparator inputs: Port I/O pins, Capacitive Touch Sense Compare, VDD/DC+ Supply Voltage, Regulated Digital Supply Voltage (Output of VREG0), the VBAT Supply voltage or ground. The Comparator's supply voltage divided by 2 is also available as an input; the resistors used to divide the voltage only draw current when this setting is selected. The Comparator input multiplexers are configured using the CPT0MX and CPT1MX registers described in SFR Definition 7.5 and SFR Definition 7.6.

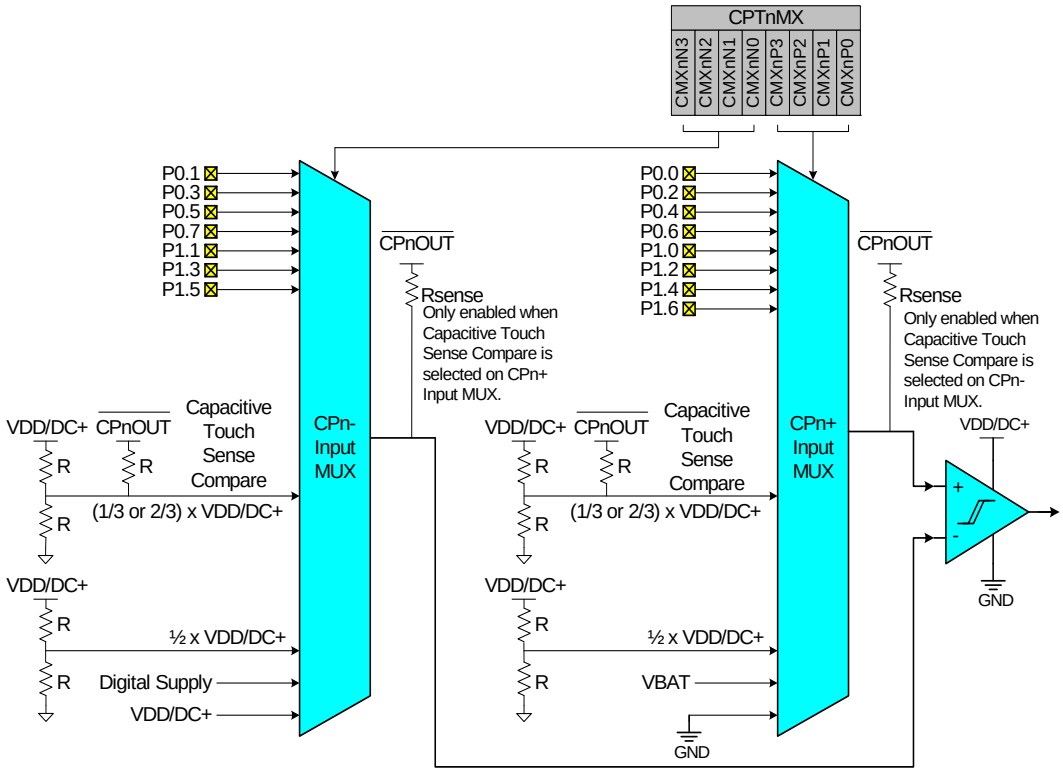



**Figure 7.4. CPn Multiplexer Block Diagram**

**Important Note About Comparator Input Configuration:** Port pins selected as comparator inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN and disable the digital driver (PnMDOUT = 0 and Port Latch = 1). To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section "21. Port Input/Output" on page 205 for more Port I/O configuration details.

## SFR Definition 14.2. PMU0MD: Power Management Unit Mode

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | RTCOE | WAKEOE | MONDIS | | | | | |
| 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 Page = 0xF; SFR Address = 0xB5

| Bit | Name | Function |
|---|---|---|
| 7 | RTCOE | **Buffered SmaRTClock Output Enable.**<br>Enables the buffered SmaRTClock oscillator output on P0.2. Only available on ‘F912 and ‘F902 devices.<br>0: Buffered SmaRTClock output is not enabled.<br>1: Buffered SmaRTClock output is enabled. |
| 6 | WAKEOE | **Wakeup Request Output Enable.**<br>Enables the Sleep Mode wake-up request signal on P0.3. Only available on ‘F912 and ‘F902 devices.<br>0: Wake-up request signal is not enabled.<br>1: Wake-up request signal is enabled. |
| 5 | MONDIS | **VBAT Supply Monitor Disable.**<br>Writing a 1 to this bit disables the VBAT supply monitor. Writing a 0 to this bit when the VBAT supply monitor is disabled will trigger a power-on reset. Only available on ‘F912 and ‘F902 devices. |
| 4:0 | Unused | **Unused.**<br>Read = 00000b. Write = Don’t Care. |

## SFR Definition 15.5. CRC0CNT: CRC0 Automatic Flash Sector Count

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | CRC0CNT[5:0] | | | | | |
| Type | R/W | | | | | | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0xF; SFR Address = 0x97

| Bit | Name | Function |
|---|---|---|
| 7:6 | Unused | **Unused.**<br>Read = 00b; Write = Don't Care. |
| 5:0 | CRC0CNT[5:0] | **Automatic CRC Calculation Flash Sector Count.**<br>These bits specify the number of Flash sectors to include in an automatic CRC calculation. The starting address of the last Flash sector included in the automatic CRC calculation is (CRC0ST+CRC0CNT) x Page Size.<br><br>**Note:** 'F91x and 'F90x devices have a page size of 512 bytes. |

### 20.3.2. Setting a SmaRTClock Alarm

The SmaRTClock alarm function compares the 32-bit value of SmaRTClock Timer to the value of the ALARMn registers. An alarm event is triggered if the SmaRTClock timer is **equal to** the ALARMn registers. If Auto Reset is enabled, the 32-bit timer will be cleared to zero one SmaRTClock cycle after the alarm event.

The SmaRTClock alarm event can be configured to reset the MCU, wake it up from a low power mode, or generate an interrupt. See Section “12. Interrupt Handler” on page 120, Section “14. Power Management” on page 143, and Section “18. Reset Sources” on page 171 for more information.

The following steps can be used to set up a SmaRTClock Alarm:

1. Disable SmaRTClock Alarm Events (RTC0AEN = 0).
2. Set the ALARMn registers to the desired value.
3. Enable SmaRTClock Alarm Events (RTC0AEN = 1).

**Notes:**

- The ALRM bit, which is used as the SmaRTClock Alarm Event flag, is cleared by disabling SmaRTClock Alarm Events (RTC0AEN = 0).
- If AutoReset is disabled, disabling (RTC0AEN = 0) then Re-enabling Alarm Events (RTC0AEN = 1) after a SmaRTClock Alarm without modifying ALARMn registers will automatically schedule the next alarm after 2^32 SmaRTClock cycles (approximately 36 hours using a 32.768 kHz crystal).
- The SmaRTClock Alarm Event flag will remain asserted for a maximum of one SmaRTClock cycle. See Section “14. Power Management” on page 143 for information on how to capture a SmaRTClock Alarm event using a flag which is not automatically cleared by hardware.

### 20.3.3. Software Considerations for using the SmaRTClock Timer and Alarm

The SmaRTClock timer and alarm have two operating modes to suit varying applications. The two modes are described below:

**Mode 1:**
The first mode uses the SmaRTClock timer as a perpetual timebase which is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. The alarm interval is software managed and is added to the ALRMn registers by software after each alarm. This allows the alarm match value to always stay ahead of the timer by one software managed interval. If software uses 32-bit unsigned addition to increment the alarm match value, then it does not need to handle overflows since both the timer and the alarm match value will overflow in the same manner.

This mode is ideal for applications which have a long alarm interval (e.g., 24 or 36 hours) and/or have a need for a perpetual timebase. An example of an application that needs a perpetual timebase is one whose wake-up interval is constantly changing. For these applications, software can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

**Mode 2:**
The second mode uses the SmaRTClock timer as a general purpose up counter which is auto reset to zero by hardware after each alarm. The alarm interval is managed by hardware and stored in the ALRMn registers. Software only needs to set the alarm interval once during device initialization. After each alarm, software should keep a count of the number of alarms that have occurred in order to keep track of time.

This mode is ideal for applications that require minimal software intervention and/or have a fixed alarm interval. This mode is the most power efficient since it requires less CPU time per alarm.

## 21.3. Priority Crossbar Decoder

The Priority Crossbar Decoder assigns a Port I/O pin to each software selected digital function using the fixed peripheral priority order shown in Figure 21.3. The registers XBR0, XBR1, and XBR2 defined in SFR Definition 21.1, SFR Definition 21.2, and SFR Definition 21.3 are used to select digital functions in the Crossbar. The Port pins available for assignment by the Crossbar include all Port pins (P0.0–P1.6) which have their corresponding bit in PnSKIP set to 0.

From Figure 21.3, the highest priority peripheral is UART0. If UART0 is selected in the Crossbar (using the XBRn registers), then P0.4 and P0.5 will be assigned to UART0. The next highest priority peripheral is SPI1. If SPI1 is selected in the Crossbar, then P1.0–P1.3 will be assigned to SPI1. The user should ensure that the pins to be assigned by the Crossbar have their PnSKIP bits set to 0.

For all remaining digital functions selected in the Crossbar, starting at the top of Figure 21.3 going down, the least-significant unskipped, unassigned Port pin(s) are assigned to that function. If a Port pin is already assigned (e.g. UART0 or SPI1 pins), or if its PnSKIP bit is set to 1, then the Crossbar will skip over the pin and find next available unskipped, unassigned Port pin. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

Figure 21.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP = 0x00); Figure 21.4 shows the Crossbar Decoder priority with the External Oscillator pins (XTAL1 and XTAL2) skipped (P0SKIP = 0x0C).

**Notes:**
- The Crossbar must be enabled (XBARE = 1) before any Port pin is used as a digital output. Port output drivers are disabled while the Crossbar is disabled.
- When SMBus is selected in the Crossbar, the pins associated with SDA and SCL will automatically be forced into open-drain output mode regardless of the PnMDOUT setting.
- SPI0 can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1-NSSMD0 bits in register SPI0CN. The NSS signal is only routed to a Port pin when 4-wire mode is selected. When SPI0 is selected in the Crossbar, the SPI0 mode (3-wire or 4-wire) will affect the pinout of all digital functions lower in priority than SPI0.
- For given XBRn, PnSKIP, and SPInCN register settings, one can determine the I/O pin-out of the device using Figure 21.3 and Figure 21.4.

## SFR Definition 21.13. P1: Port1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | P1[6:0] | | | | | | |
| Type | | R/W | | | | | | |
| Reset | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Page = All Pages; SFR Address = 0x90; Bit-Addressable

| Bit | Name | Description | Write | Read |
|---|---|---|---|---|
| 7 | Unused | **Unused.**<br>Read =0b; Write = Don't Care. | | |
| 6:0 | P1[6:0] | **Port 1 Data.**<br>Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O. | 0: Set output latch to logic LOW.<br>1: Set output latch to logic HIGH. | 0: P1.n Port pin is logic LOW.<br>1: P1.n Port pin is logic HIGH. |

## SFR Definition 21.14. P1SKIP: Port1 Skip

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | P1SKIP[6:0] | | | | | | |
| Type | | R/W | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0xD5

| Bit | Name | Function |
|---|---|---|
| 7 | Unused | **Unused.**<br>Read =0b; Write = Don't Care. |
| 6:0 | P1SKIP[6:0] | **Port 1 Crossbar Skip Enable Bits.**<br>These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar.<br>0: Corresponding P1.n pin is not skipped by the Crossbar.<br>1: Corresponding P1.n pin is skipped by the Crossbar. |

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 22.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

**Table 22.2. Minimum SDA Setup and Hold Times**

| EXTHOLD | Minimum SDA Setup Time | Minimum SDA Hold Time |
|---|---|---|
| 0 | $T_{low}$ – 4 system clocks<br>or<br>1 system clock + s/w delay* | 3 system clocks |
| 1 | 11 system clocks | 12 system clocks |
| ***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero. | | |

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “22.3.4. SCL Low Timeout” on page 228). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 22.4).

## SFR Definition 22.3. SMB0ADR: SMBus Slave Address

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

SFR Page = 0x0; SFR Address = 0xF4

| Bit | Name | Function |
|---|---|---|
| 7:1 | SLV[6:0] | **SMBus Hardware Slave Address.**<br>Defines the SMBus Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM[6:0] are checked against the incoming address. This allows multiple addresses to be recognized. |
| 0 | GC | **General Call Address Enable.**<br>When hardware address recognition is enabled (EHACK = 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. |

## SFR Definition 22.4. SMB0ADM: SMBus Slave Address Mask

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

SFR Page = 0x0; SFR Address = 0xF5

| Bit | Name | Function |
|---|---|---|
| 7:1 | SLVM[6:0] | **SMBus Slave Address Mask.**<br>Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM[6:0] enables comparisons with the corresponding bit in SLV[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address). |
| 0 | EHACK | **Hardware Acknowledge Enable.**<br>Enables hardware acknowledgement of slave address and received data bytes.<br>0: Firmware must manually acknowledge all incoming address and data bytes.<br>1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled. |

## SFR Definition 23.1. SCON0: Serial Port 0 Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | S0MODE | | MCE0 | REN0 | TB80 | RB80 | TI0 | RI0 |
| Type | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0x98; Bit-Addressable

| Bit | Name | Function |
|---|---|---|
| 7 | S0MODE | **Serial Port 0 Operation Mode.**<br>Selects the UART0 Operation Mode.<br>0: 8-bit UART with Variable Baud Rate.<br>1: 9-bit UART with Variable Baud Rate. |
| 6 | Unused | **Unused.**<br>Read = 1b. Write = Don't Care. |
| 5 | MCE0 | **Multiprocessor Communication Enable.**<br>**For Mode 0 (8-bit UART): Checks for valid stop bit.**<br>0: Logic level of stop bit is ignored.<br>1: RI0 will only be activated if stop bit is logic level 1.<br>**For Mode 1 (9-bit UART): Multiprocessor Communications Enable.**<br>0: Logic level of ninth bit is ignored.<br>1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1. |
| 4 | REN0 | **Receive Enable.**<br>0: UART0 reception disabled.<br>1: UART0 reception enabled. |
| 3 | TB80 | **Ninth Transmission Bit.**<br>The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0). |
| 2 | RB80 | **Ninth Receive Bit.**<br>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. |
| 1 | TI0 | **Transmit Interrupt Flag.**<br>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. |
| 0 | RI0 | **Receive Interrupt Flag.**<br>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. |

# 24. Enhanced Serial Peripheral Interface (SPI0 and SPI1)

The enhanced serial peripheral interfaces (SPI0 and SPI1) provide access to two identical, flexible, full-duplex synchronous serial busses. Both SPI0 and SPI1 will be referred to collectively as SPIn. SPIn can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPIn in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

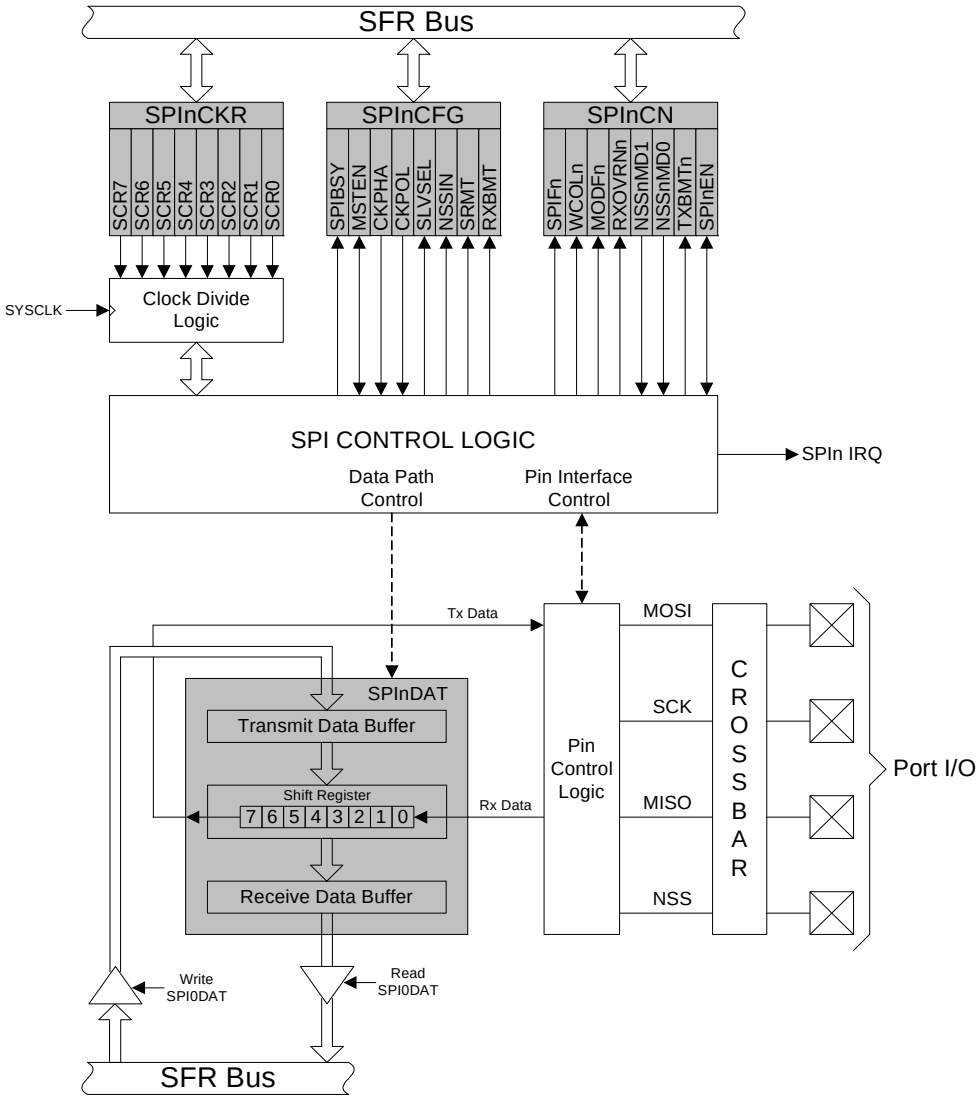



**Figure 24.1. SPI Block Diagram**

## SFR Definition 25.14. TMR3RLL: Timer 3 Reload Register Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3RLL[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0x92

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3RLL[7:0] | **Timer 3 Reload Register Low Byte.**<br>TMR3RLL holds the low byte of the reload value for Timer 3. |

## SFR Definition 25.15. TMR3RLH: Timer 3 Reload Register High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TMR3RLH[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Page = 0x0; SFR Address = 0x93

| Bit | Name | Function |
|---|---|---|
| 7:0 | TMR3RLH[7:0] | **Timer 3 Reload Register High Byte.**<br>TMR3RLH holds the high byte of the reload value for Timer 3. |

### 26.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 26.1.

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

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

**Equation 26.1. Square Wave Frequency Output**

Where $F_{PCA}$ is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

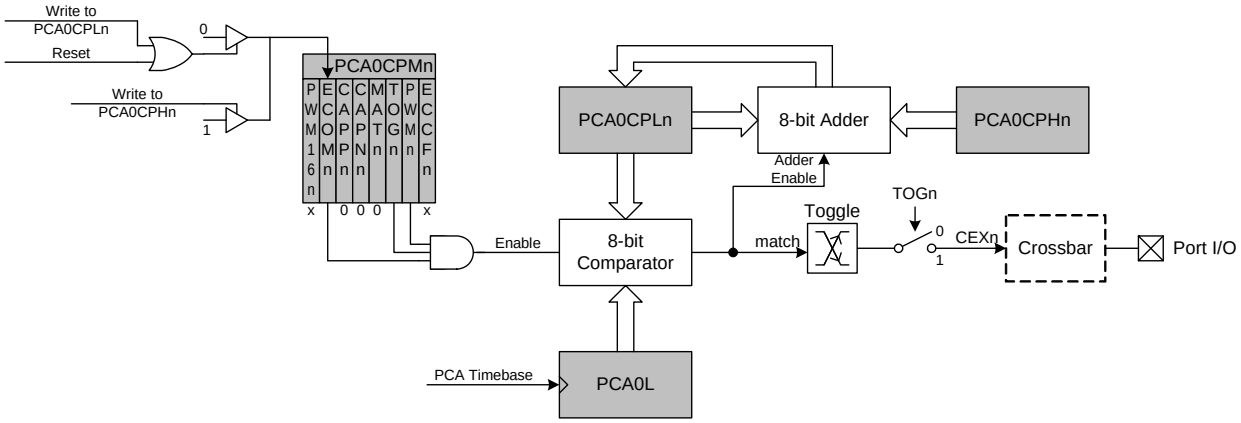



**Figure 26.7. PCA Frequency Output Mode**