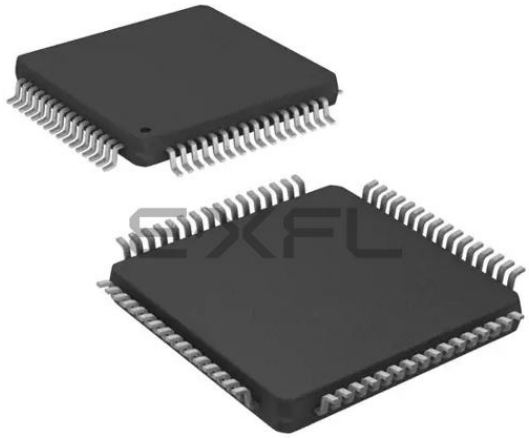


Welcome to E-XFL.COM

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	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f047

Table of Contents

1.4. Programmable Digital I/O and Crossbar

The standard 8051 Ports (0, 1, 2, and 3) are available on the MCUs. The C8051F040/2/4/6 have 4 additional 8-bit ports (4, 5, 6, and 7) for a total of 64 general-purpose I/O Ports. The Ports behave like the standard 8051 with a few enhancements.

Each port pin can be configured as either a push-pull or open-drain output. Also, the "weak pullups" which are normally fixed on an 8051 can be globally disabled, providing additional power saving capabilities for low-power applications.

Perhaps the most unique enhancement is the Digital Crossbar. This is essentially a large digital switching network that allows mapping of internal digital system resources to Port I/O pins on P0, P1, P2, and P3 (See Figure 1.9). Unlike microcontrollers with standard multiplexed digital I/O ports, all combinations of functions are supported with all package options offered.

The on-chip counter/timers, serial buses, HW interrupts, ADC Start of Conversion input, comparator outputs, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for the particular application.

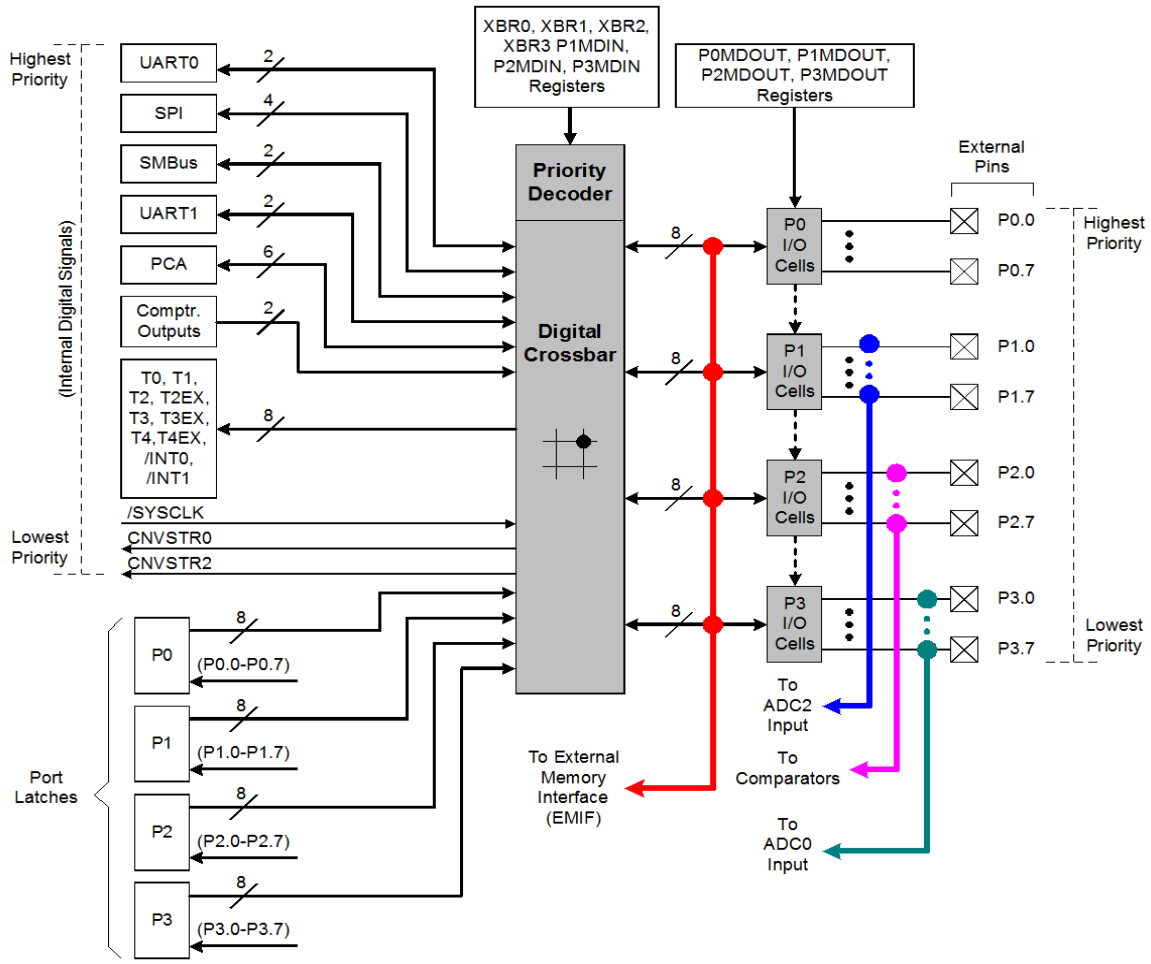


Figure 1.9. Digital Crossbar Diagram

5.1.1. Analog Input Configuration

The analog multiplexer routes signals from external analog input pins, Port 3 I/O pins (See **Section "17.1.5. Configuring Port 1, 2, and 3 Pins as Analog Inputs" on page 207**), a High Voltage Difference Amplifier, and an on-chip temperature sensor as shown in Figure 5.2.

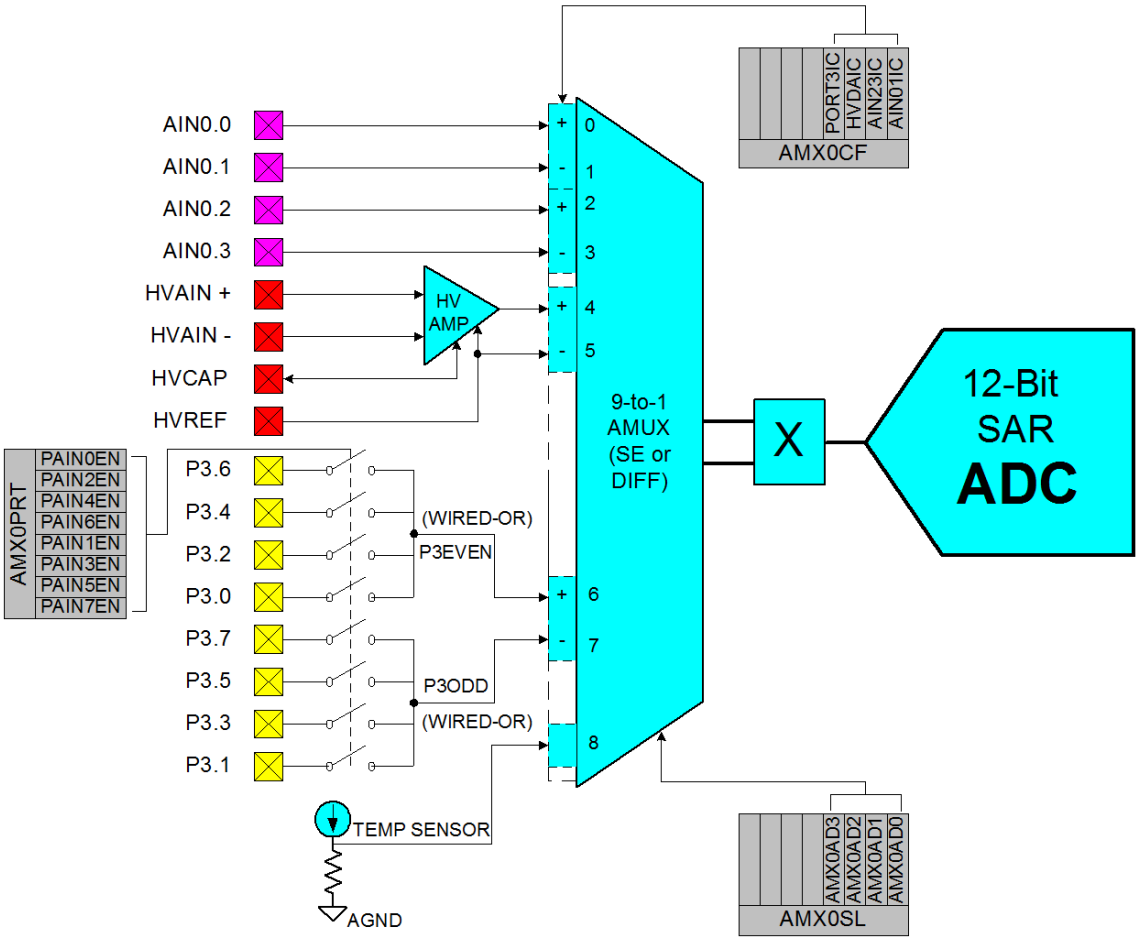


Figure 5.2. Analog Input Diagram

Analog signals may be input from four external analog input pins (AIN0.0 through AIN0.3) as differential or single-ended measurements. Additionally, Port 3 I/O Port Pins may be configured to input analog signals. Port 3 pins configured as analog inputs are selected using the Port Pin Selection register (AMX0PRT). Any number of Port 3 pins may be selected simultaneously as inputs to the AMUX. Even numbered Port 3 pins and odd numbered Port 3 pins are routed to separate AMUX inputs. (**Note:** Even port pins and odd port pins that are simultaneously selected will be shorted together as "wired-OR".) In this way, differential measurements may be made when using the Port 3 pins (voltage difference between selected even and odd Port 3 pins) as shown in Figure 5.2.

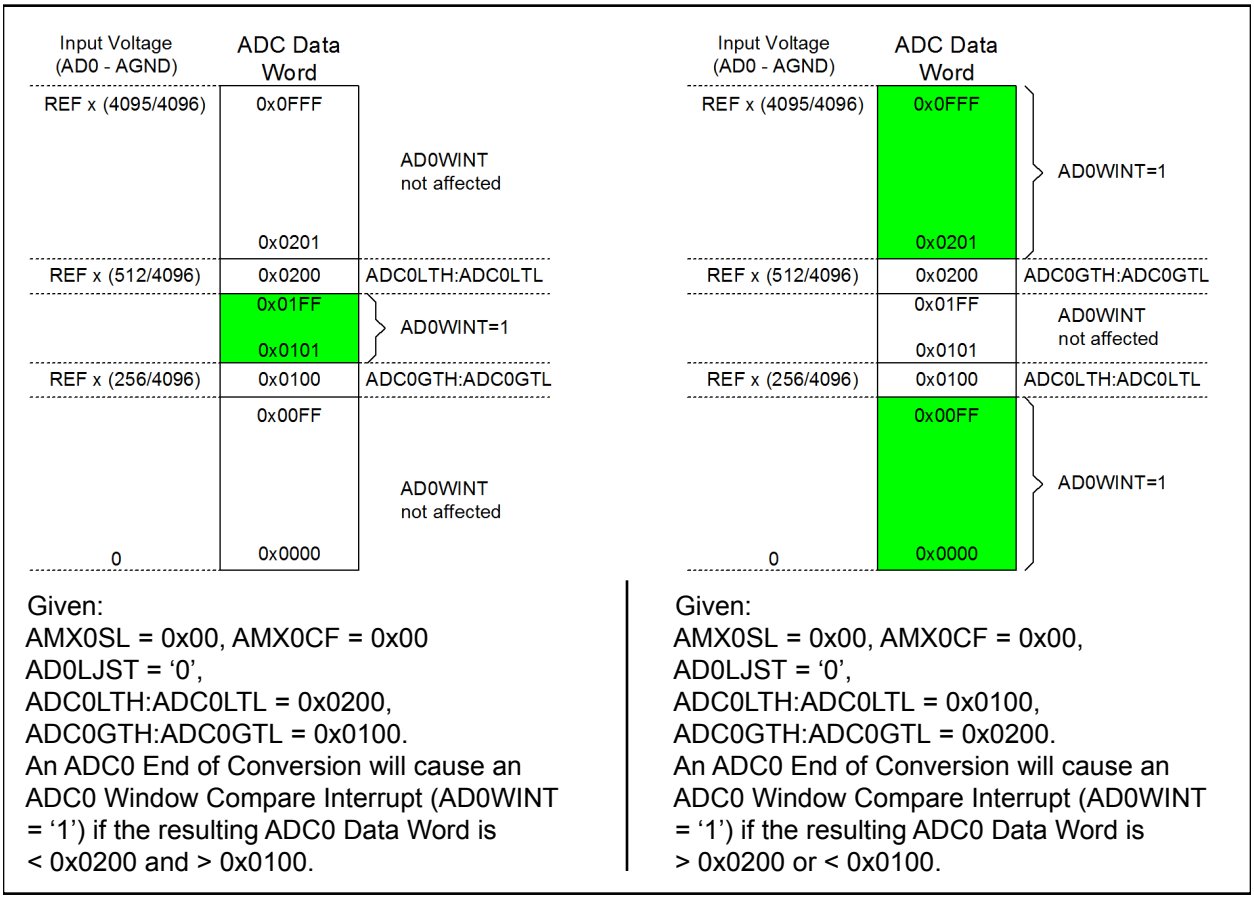
The High Voltage Difference Amplifier (HVDA) will accept analog input signals and reject up to 60 volts common-mode for differential measurement of up to the reference voltage to the ADC (0 to VREF volts). The output of the HVDA can be selected as an input to the ADC using the AMUX as any other channel is selected for input. (See **Section "5.2. High-Voltage Difference Amplifier" on page 52**).

SFR Definition 5.12. ADC0LTL: ADC0 Less-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC6
SFR Page: 0

Bits7-0: Low byte of ADC0 Less-Than Data Word.



Given:
AMX0SL = 0x00, AMX0CF = 0x00
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0200,
ADC0GTH:ADC0GTL = 0x0100.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x0200 and > 0x0100.

Given:
AMX0SL = 0x00, AMX0CF = 0x00,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0100,
ADC0GTH:ADC0GTL = 0x0200.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is > 0x0200 or < 0x0100.

Figure 5.8. 12-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data

6.3.3. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 6.5 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (*SA*) may be approximated by Equation 6.2. When measuring the Temperature Sensor output, R_{TOTAL} reduces to R_{MUX}. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the tracking requirements. See Table 6.2 for absolute minimum settling/tracking time requirements.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 6.2. ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the ADC0 MUX resistance and any external source resistance.
n is the ADC resolution in bits (10).

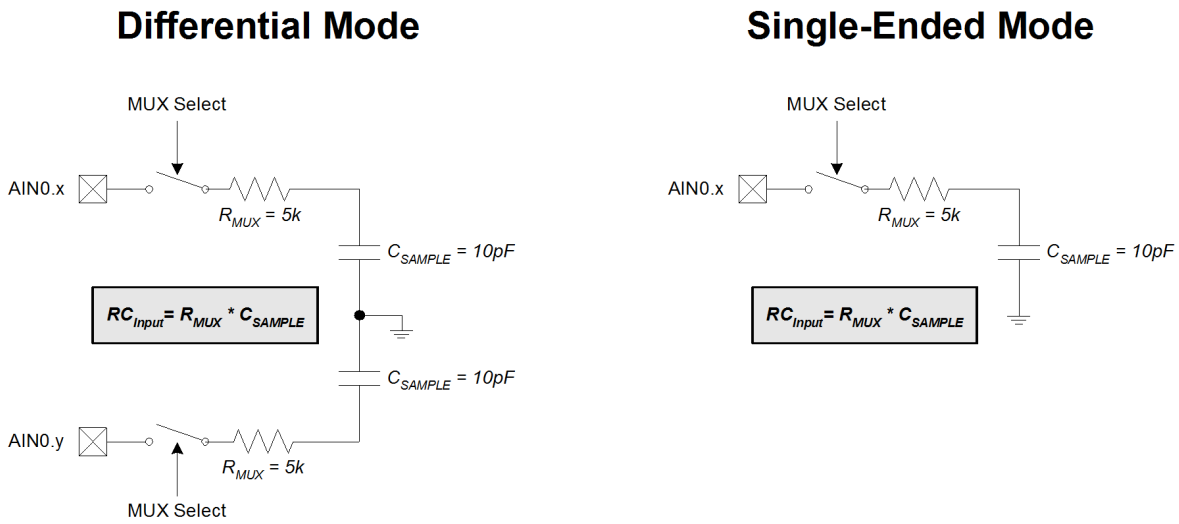


Figure 6.5. ADC0 Equivalent Input Circuits

SFR Definition 6.7. ADC0H: ADC0 Data Word MSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0xBF
SFR Page: 0

Bits7-0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7-2 are the sign extension of Bit 1. Bits 0 and 1 are the upper 2 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7-0 are the most-significant bits of the 10-bit ADC0 Data Word.

SFR Definition 6.8. ADC0L: ADC0 Data Word LSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0xBE
SFR Page: 0

Bits7-0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7-0 are the lower 8 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 6 and 7 are the lower 2 bits of the 10-bit ADC0 Data Word. Bits 5-0

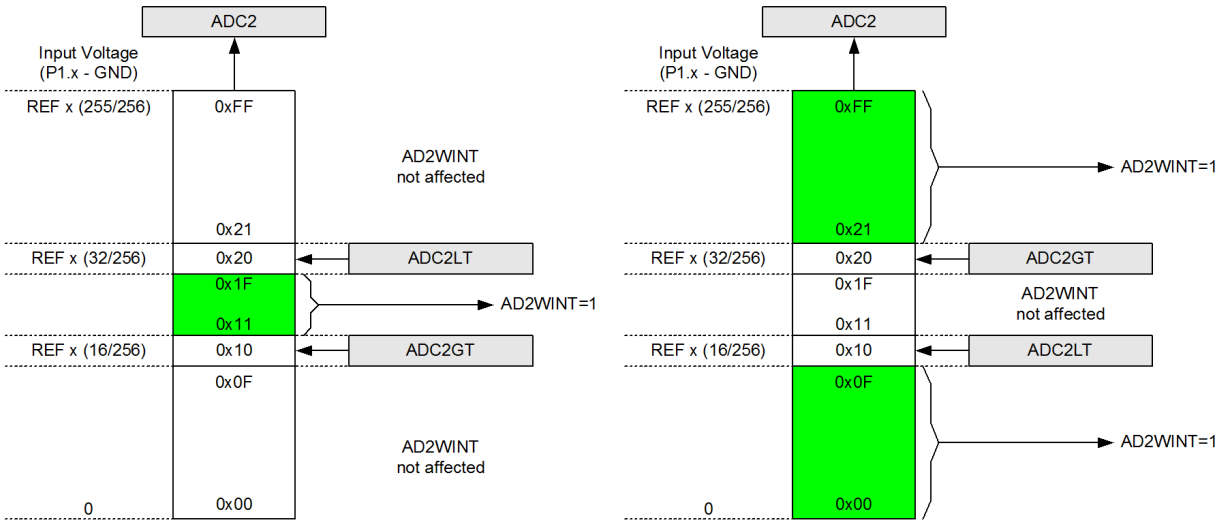


Figure 7.5. ADC Window Compare Examples, Single-Ended Mode

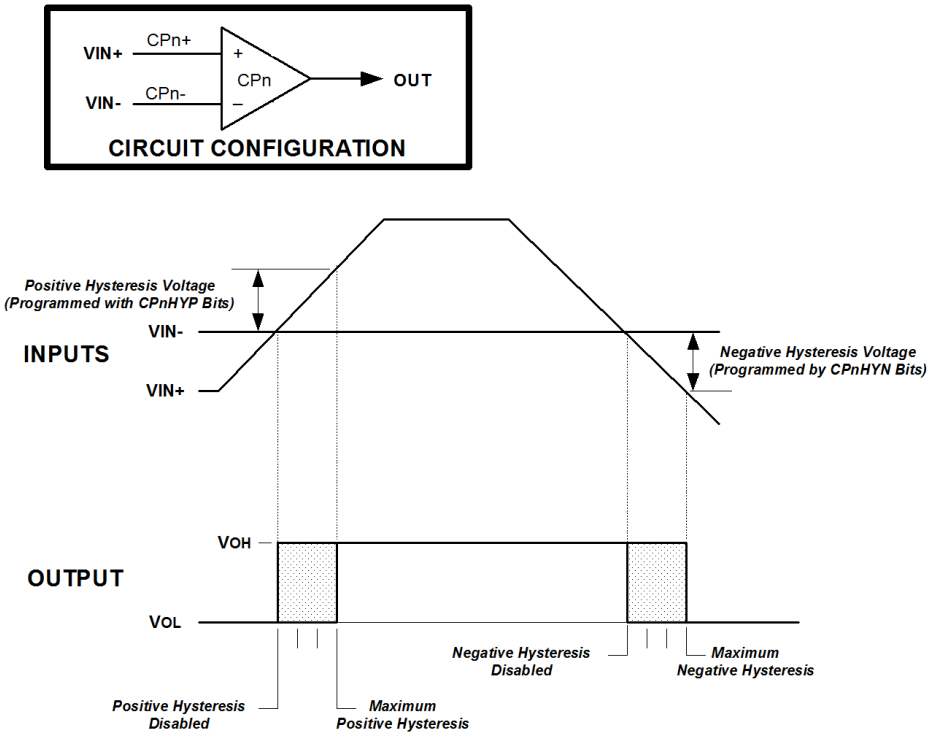


Figure 11.2. Comparator Hysteresis Plot

The hysteresis of the Comparator is software-programmable via its Comparator Control register (CPTnCN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in SFR Definition 11.1). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Table 11.1, settings of approximately 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CPnHYP bits.

Comparator interrupts can be generated on either rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section "12.3. Interrupt Handler" on page 153**). The rising and/or falling -edge interrupts are enabled using the comparator's Rising/Falling Edge Interrupt Enable Bits (CPnRIE and CPnFIE) in their respective Comparator Mode Selection Register (CPTnMD), shown in SFR Definition 11.2. These bits allow the user to control which edge (or both) will cause a comparator interrupt. However, the comparator interrupt must also be enabled in the Extended Interrupt Enable Register (EIE1). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge interrupt, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge interrupt. Once set, these bits remain set until cleared by software. The output state of a Comparator can be obtained at any time by reading the CPnOUT bit. A Comparator is enabled by setting its respective CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.Upon enabling a comparator, the output of the comparator is not immediately valid. Before using a comparator as an interrupt or reset source, software should wait for a minimum of the specified "Power-up time" as specified in Table 11.1, "Comparator Electrical Characteristics," on page 126.

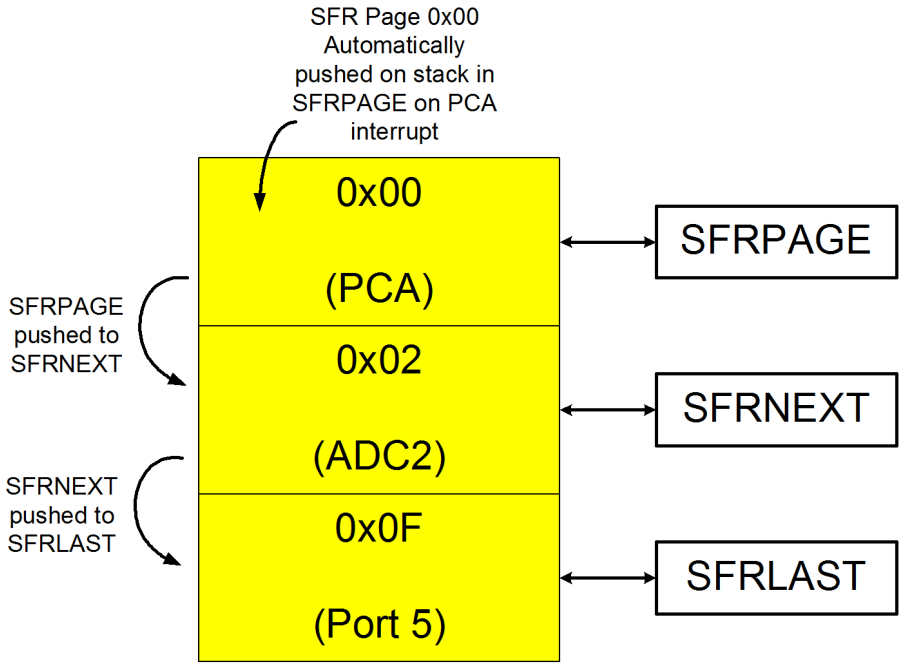


Figure 12.6. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR

On exit from the PCA interrupt service routine, the CIP-51 will return to the ADC2 Window Comparator ISR. On execution of the RETI instruction, SFR Page 0x00 used to access the PCA registers will be automatically popped off of the SFR Page Stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Software in the ADC2 ISR can continue to access SFR's as it did prior to the PCA interrupt. Likewise, the contents of SFRLAST are moved to the SFRNEXT register. Recall this was the SFR Page value 0x0F being used to access Port 5 before the ADC2 interrupt occurred. See Figure 12.7 below.

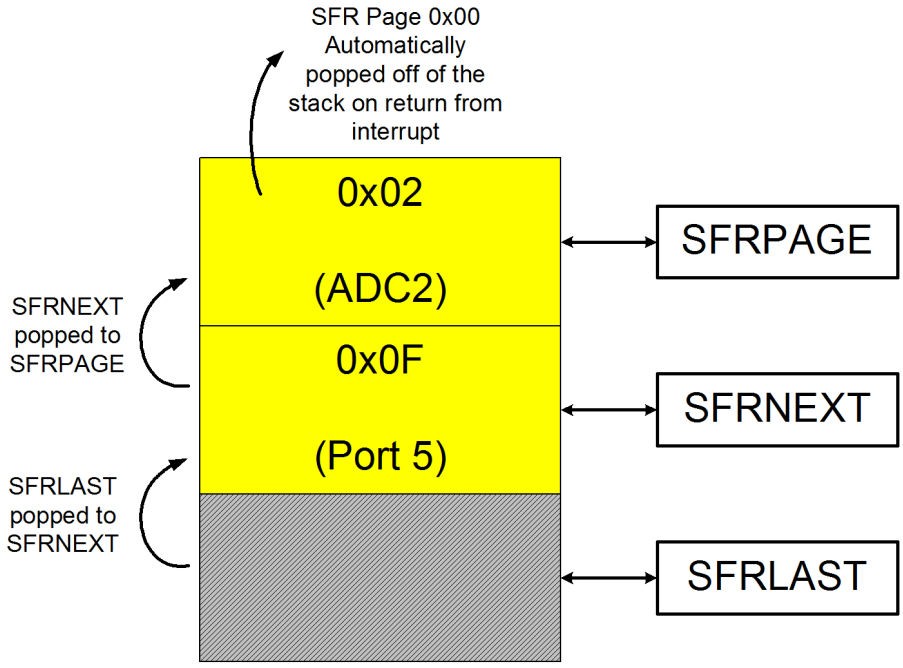


Figure 12.7. SFR Page Stack Upon Return From PCA Interrupt

On the execution of the RETI instruction in the ADC2 Window Comparator ISR, the value in SFRPAGE register is overwritten with the contents of SFRNEXT. The CIP-51 may now access the Port 5 SFR bits as it did prior to the interrupts occurring. See Figure 12.8 below.

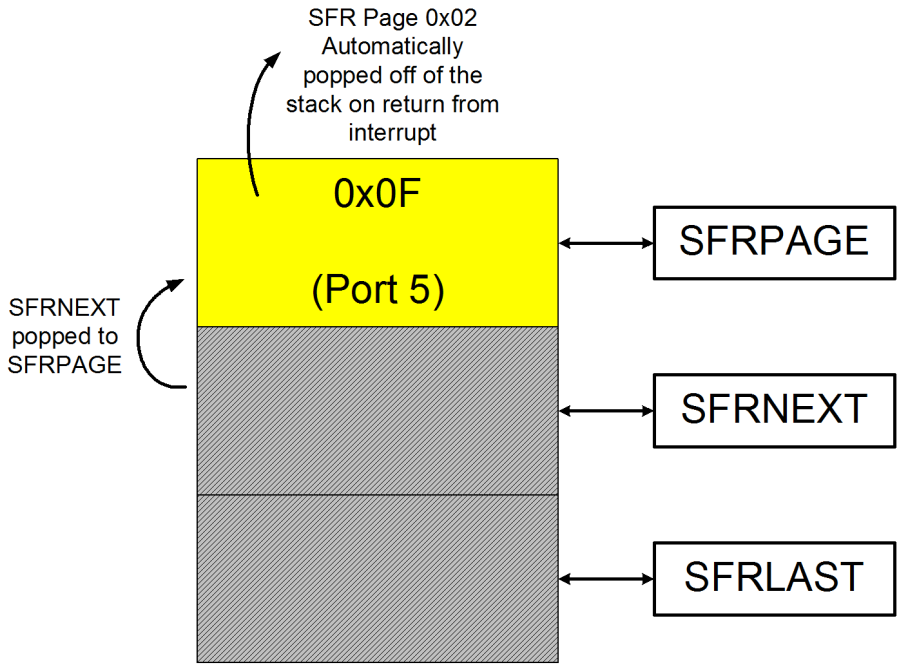


Figure 12.8. SFR Page Stack Upon Return From ADC2 Window Interrupt

Note that in the above example, all three bytes in the SFR Page Stack are accessible via the SFRPAGE, SFRNEXT, and SFRLAST special function registers. If the stack is altered while servicing an interrupt, it is possible to return to a different SFR Page upon interrupt exit than selected prior to the interrupt call. Direct access to the SFR Page stack can be useful to enable real-time operating systems to control and manage context switching between multiple tasks.

Push operations on the SFR Page Stack only occur on interrupt service, and pop operations only occur on interrupt exit (execution on the RETI instruction). The automatic switching of the SFRPAGE and operation of the SFR Page Stack as described above can be disabled in software by clearing the SFR Automatic Page Enable Bit (SFRPGEN) in the SFR Page Control Register (SFRPGCN). See SFR Definition 12.1.

16. External Data Memory Interface and On-Chip XRAM

The C8051F04x MCUs include 4 kB of on-chip RAM mapped into the external data memory space (XRAM), as well as an External Data Memory Interface which can be used to access off-chip memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 16.1). **Note**: the MOVX instruction can also be used for writing to the Flash memory. See **Section "15. Flash Memory" on page 179** for details. The MOVX instruction accesses XRAM by default. The EMIF can be configured to appear on the lower GPIO Ports (P0-P3) or the upper GPIO Ports (P4-P7).

16.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

16.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

16.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

Table 16.1. AC Parameters for External Memory Interface

Parameter	Description	Min	Max	Units
T_{SYSCLK}	System Clock Period	40	—	ns
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20	—	ns
T_{RDH}	Read Data Hold Time	0	—	ns

SFR Definition 17.10. P2: Port2 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addessable

SFR Address: 0xA0
SFR Page: All Pages

Bits7-0: P2.[7:0]: Port2 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P2MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P2.n pin is logic low.
1: P2.n pin is logic high.

Note: P2.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Multiplexed mode, or as Address[7:0] in Non-multiplexed mode). See **Section "16. External Data Memory Interface and On-Chip XRAM" on page 187** for more information about the External Memory Interface.

SFR Definition 17.11. P2MDIN: Port2 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xAE
SFR Page: F

Bits7-0: P1MDIN.[7:0]: Port 2 Input Mode Bits.
0: Port Pin is configured in Analog Input mode. The digital input path is disabled (a read from the Port bit will always return '0'). The weak pullup on the pin is disabled.
1: Port Pin is configured in Digital Input mode. A read from the Port bit will return the logic level at the Pin. The state of the weak pullup is determined by the WEAKPUD bit (XBR2.7, see SFR Definition 17.3).

Notes:
1. When P2.0 is configured to Analog Input mode, the crossbar does not skip over this pin, and the crossbar is allowed to allocate digital peripherals on this pin.
2. When P2.1 is configured to Analog Input mode, the crossbar does not skip over this pin, and the crossbar is allowed to allocate digital peripherals on this pin.

We will adjust the length of the 4 bit segments so that their sum is as close as possible to the desired bit time. Since each segment must be an integer multiple of the time quantum (t_q), the closest achievable bit time is 22 t_q (994.642 ns), yielding a bit rate of 1.00539 Mbit/sec. The Sync_Seg is a constant 1 t_q. The Prop_Seg must be greater than or equal to the propagation delay of 400 ns; we choose 9 t_q (406.899 ns).

The remaining time quanta (t_q) in the bit time are divided between Phase_Seg1 and Phase_Seg2 as shown in Figure 18.1. We select Phase_Seg1 = 6 t_q and Phase_Seg2 = 6 t_q.

$$\text{Phase\_Seg1} + \text{Phase\_Seg2} = \text{Bit Time} - (\text{Sync\_Seg} + \text{Prop\_Seg})$$

Note 1: If Phase_Seg1 + Phase_Seg2 is even, then Phase_Seg2 = Phase_Seg1.

Note 2: Phase_Seg2 should be at least 2 t_q.

Equation 18.1. Assigning the Phase Segments

The Synchronization Jump Width (SJW) timing parameter is defined by Figure 18.2. It is used for determining the value written to the Bit Timing Register and for determining the required oscillator tolerance. Since we are using a quartz crystal as the system clock source, an oscillator tolerance calculation is not needed.

$$\text{SJW} = \min (4, \text{Phase\_Seg1})$$

Equation 18.2. Synchronization Jump Width (SJW)

The value written to the Bit Timing Register can be calculated using Equation 18.3. The BRP Extension register is left at its reset value of 0x0000.

BRPE = BRP - 1 = BRP Extension Register = 0x0000

SJWp = SJW - 1 = min (4, 6) – 1 = 3

TSEG1 = (Prop_Seg + Phase_Seg1 - 1) = 9 + 6 - 1 = 14

TSEG2 = (Phase_Seg2 - 1) = 5

Bit Timing Register = (TSEG2 * 0x1000) + (TSEG1 * 0x0100) + (SJWp * 0x0040) + BRPE = 0x5EC0

Equation 18.3. Calculating the Bit Timing Register Value

The following steps are performed to initialize the CAN timing registers:

Step 1. Set the SFRPAGE register to CAN0_PAGE.
Step 2. Set the INIT the CCE bits to ‘1’ in the CAN Control Register accessible through the CAN0CN SFR.
Step 3. Set the CAN0ADR to 0x03 to point to the Bit Timing Register.

SFR Definition 18.3. CAN0CN: CAN Control

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
*	*	*	CANIF	*	*	*	*	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF8
SFR Page: 1

Bit 4: CANIF: CAN Interrupt Flag. Write = don't care.
0: CAN interrupt has not occurred.
1: CAN interrupt has occurred and is active.

CANIF is controlled by the CAN controller and is cleared by hardware once all interrupt conditions have been cleared in the CAN controller. See Section 3.4.1 in the *Bosch CAN User's Guide* (page 24) for more information concerning CAN controller interrupts.

***All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide with the exception of the CANIF bit.**

This register may be accessed directly in the CIP-51 SFR register space, or through the indirect, index method (See Section "**18.2.5. Using CAN0ADR, CAN0DATH, and CANDATL to Access CAN Registers**" on page **232**).

SFR Definition 18.4. CAN0TST: CAN Test

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Please see the Bosch CAN User's Guide for a complete definition of this register								
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xDB
SFR Page: 1

All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.
This register may be accessed directly in the CIP-51 SFR register space, or through the indirect, index method (See Section "**18.2.5. Using CAN0ADR, CAN0DATH, and CANDATL to Access CAN Registers**" on page **232**).

SFR Definition 19.4. SMB0ADR: SMBus0 Address

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SLV6	SLV5	SLV4	SLV3	SLV2	SLV1	SLV0	GC	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC3
SFR Page: 0

Bits7-1: SLV6-SLV0: SMBus0 Slave Address.
These bits are loaded with the 7-bit slave address to which SMBus0 will respond when operating as a slave transmitter or slave receiver. SLV6 is the most significant bit of the address and corresponds to the first bit of the address byte received.

Bit0: GC: General Call Address Enable.
This bit is used to enable general call address (0x00) recognition.
0: General call address is ignored.
1: General call address is recognized.

19.4.5. Status Register

The SMB0STA Status register holds an 8-bit status code indicating the current state of the SMBus0 interface. There are 28 possible SMBus0 states, each with a corresponding unique status code. The five most significant bits of the status code vary while the three least-significant bits of a valid status code are fixed at zero when SI = '1'. Therefore, all possible status codes are multiples of eight. This facilitates the use of status codes in software as an index used to branch to appropriate service routines (allowing 8 bytes of code to service the state or jump to a more extensive service routine).

For the purposes of user software, the contents of the SMB0STA register is only defined when the SI flag is logic 1. Software should never write to the SMB0STA register; doing so will yield indeterminate results. The 28 SMBus0 states, along with their corresponding status codes, are given in Table 19.1.

22.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB81 (SCON1.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB81 (SCON1.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: (1) RI1 must be logic 0, and (2) if MCE1 is logic 1, the 9th bit must be logic 1 (when MCE1 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF1, the ninth bit is stored in RB81, and the RI1 flag is set to '1'. If the above conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set to '1'. A UART1 interrupt will occur if enabled when either TI1 or RI1 is set to '1'.

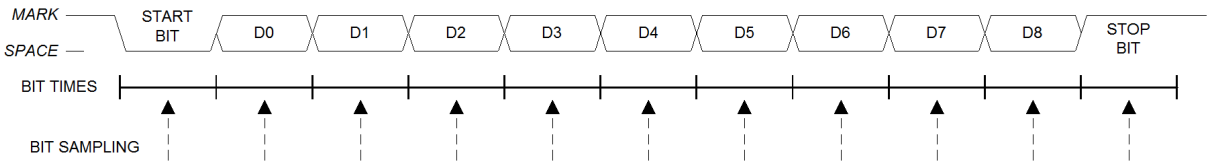


Figure 22.5. 9-Bit UART Timing Diagram

SFR Definition 22.1. SCON1: Serial Port 1 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S1MODE	-	MCE1	REN1	TB81	RB81	TI1	RI1	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98
SFR Page: 1

Bit7: S1MODE: Serial Port 1 Operation Mode.
This bit selects the UART1 Operation Mode.
0: Mode 0: 8-bit UART with Variable Baud Rate
1: Mode 1: 9-bit UART with Variable Baud Rate

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

Bit5: MCE1: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
Mode 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI1 will only be activated if stop bit is logic level 1.
Mode 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI1 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART receiver.
0: UART1 reception disabled.
1: UART1 reception enabled.

Bit3: TB81: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB81: Ninth Receive Bit.
RB81 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI1: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART1 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART1 interrupt is enabled, setting this bit causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI1: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART1 (set at the STOP bit sampling time). When the UART1 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

24. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. PCA0 consists of a dedicated 16-bit counter/timer and six 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See **Section "17.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 204**). The counter/timer is driven by a programmable timebase that can select between six inputs as its source: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI line. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each is described in **Section 24.2**). The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 24.1.

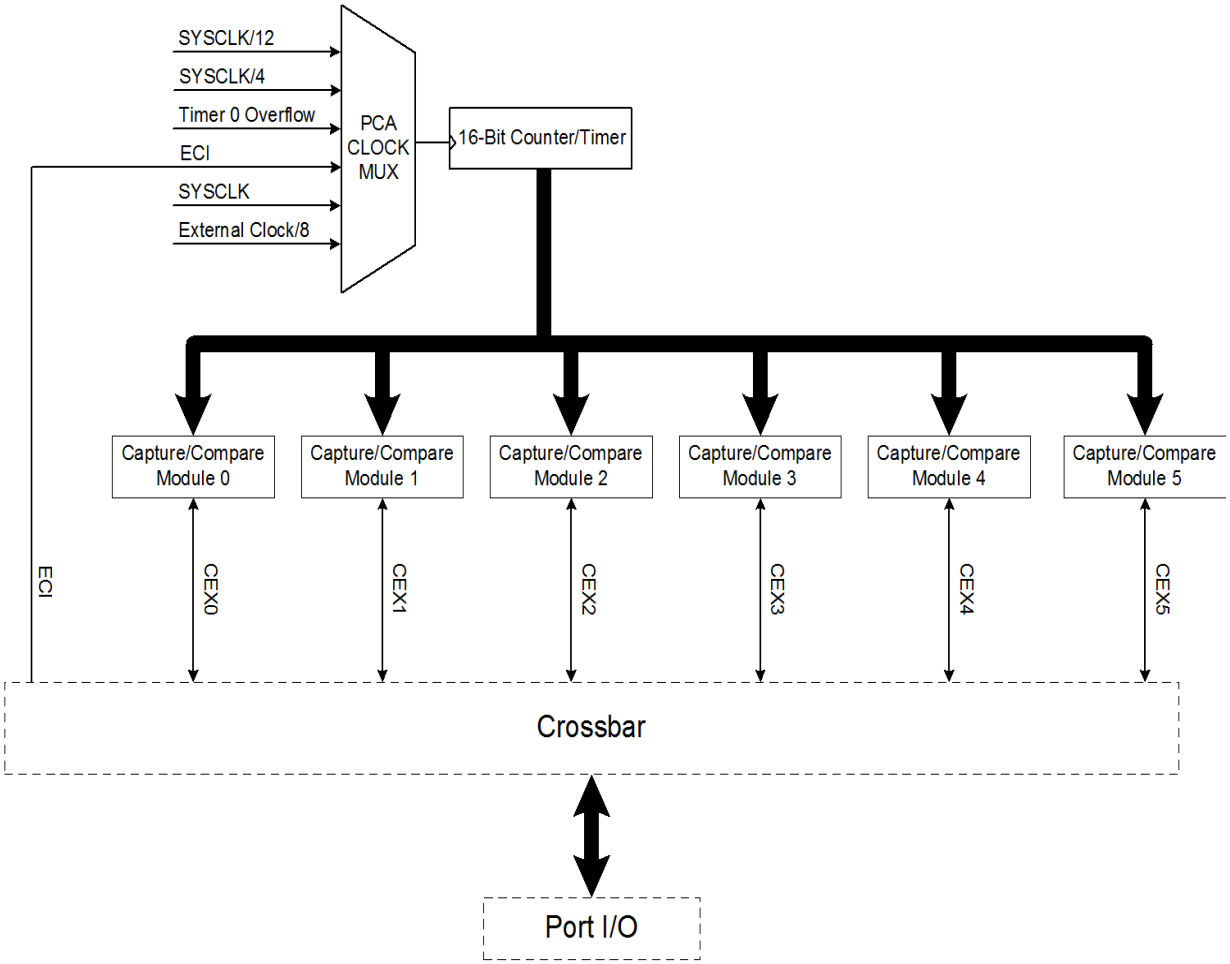


Figure 24.1. PCA Block Diagram