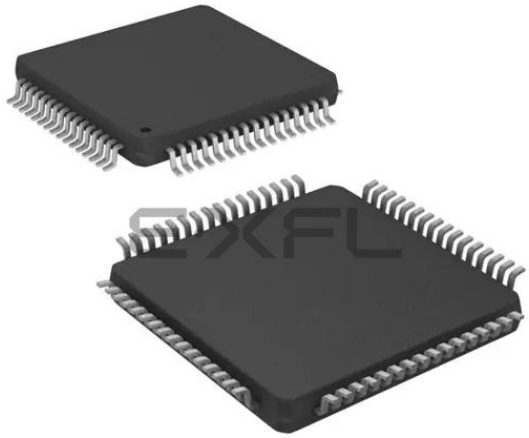


Welcome to E-XFL.COM

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, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	24
Program Memory Size	64KB (64K 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 2x16b, 8x10b; D/A 2x12b
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/c8051f063

1.4. Programmable Digital I/O and Crossbar

Three standard 8051 Ports (0, 1, and 2) are available on the MCUs. The C8051F060/2/4/6 have 4 additional 8-bit ports (3, 5, 6, and 7), and a 3-bit port (port 4) for a total of 59 general-purpose I/O Pins. 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 pull-ups" 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 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, 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.

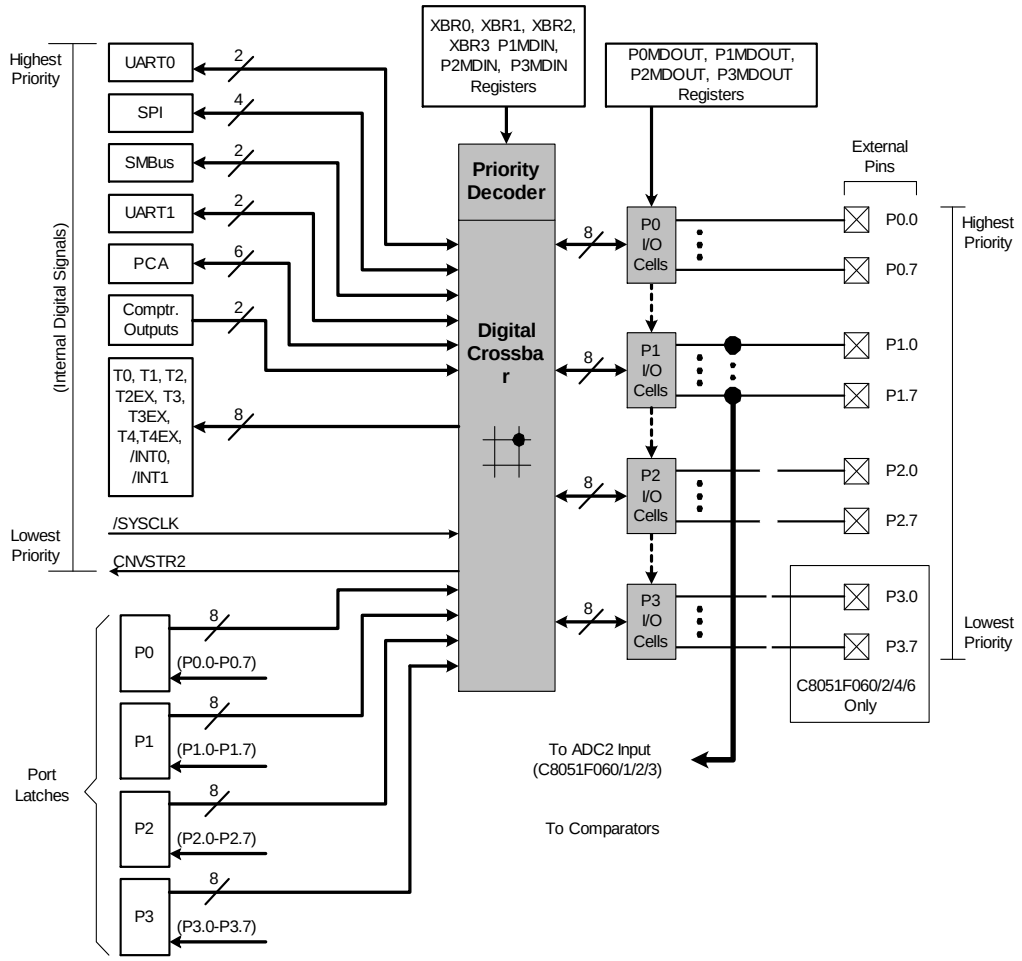


Figure 1.9. Digital Crossbar Diagram

1.8. 16-Bit Analog to Digital Converters

The C8051F060/1/2/3/4/5/6/7 devices have two on-chip 16-bit SAR ADCs (ADC0 and ADC1), which can be used independently in single-ended mode, or together in differential mode. ADC0 and ADC1 can directly access on-chip or external RAM, using the DMA interface. With a maximum throughput of 1 Msps, the ADCs offer 16 bit performance with two available linearity grades. ADC0 and ADC1 each have the capability to use dedicated, on-chip voltage reference circuitry or an external voltage reference source.

The ADCs are under full control of the CIP-51 microcontroller via the associated Special Function Registers. The system controller can also put the ADCs into shutdown mode to save power.

Conversions can be started in four ways; a software command, an overflow of Timer 2, an overflow of Timer 3, or an external signal input. This flexibility allows the start of conversion to be triggered by software events, external HW signals, or a periodic timer overflow signal. The two ADCs can operate independently, or be synchronized to perform conversions at the same time. Conversion completions are indicated by status bits, and can generate interrupts. The resulting 16-bit data words are latched into SFRs upon completion of a conversion. A DMA interface is also provided, which can gather conversions from the ADCs, and directly store them to on-chip or external RAM.

ADC0 also contains Window Compare registers, which can be configured to interrupt the controller when ADC0 data is within or outside of a specified range. ADC0 can monitor a key voltage continuously in background mode, and not interrupt the controller unless the converted data is within the specified window.

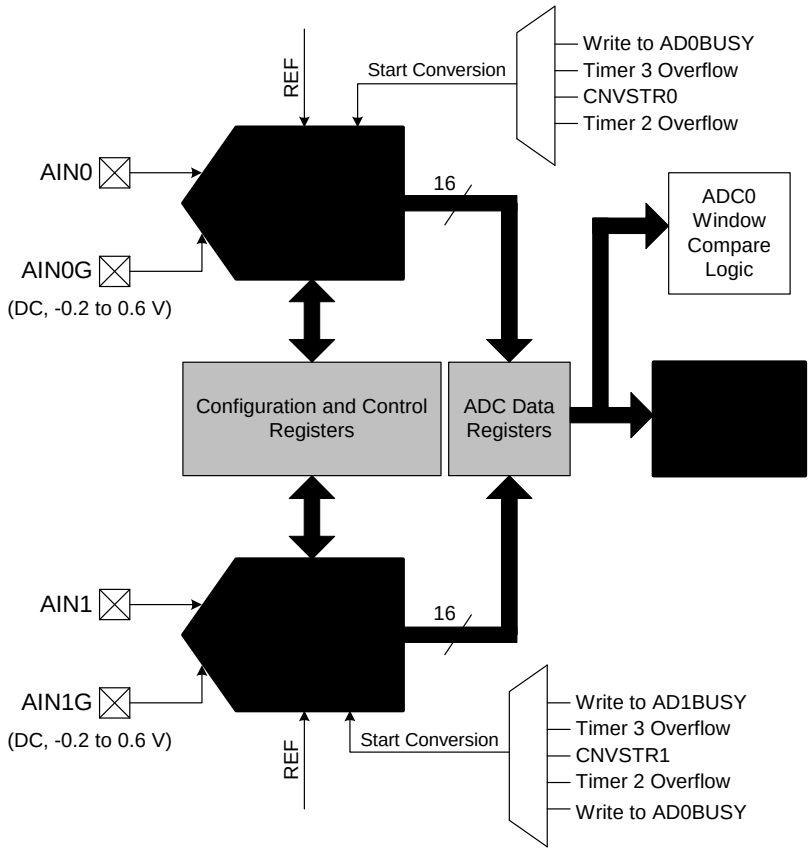


Figure 1.12. 16-Bit ADC Block Diagram

1.11. Analog Comparators

The C8051F060/1/2/3/4/5/6/7 MCUs include three analog comparators on-chip. The comparators have software programmable hysteresis and response time. Each comparator can generate an interrupt on its rising edge, falling edge, or both. The interrupts are capable of waking up the MCU from sleep mode, and Comparator 0 can be used as a reset source. The output state of the comparators can be polled in software or routed to Port I/O pins via the Crossbar. Outputs from the comparator can be routed through the crossbar. The comparators can be programmed to a low power shutdown mode when not in use.

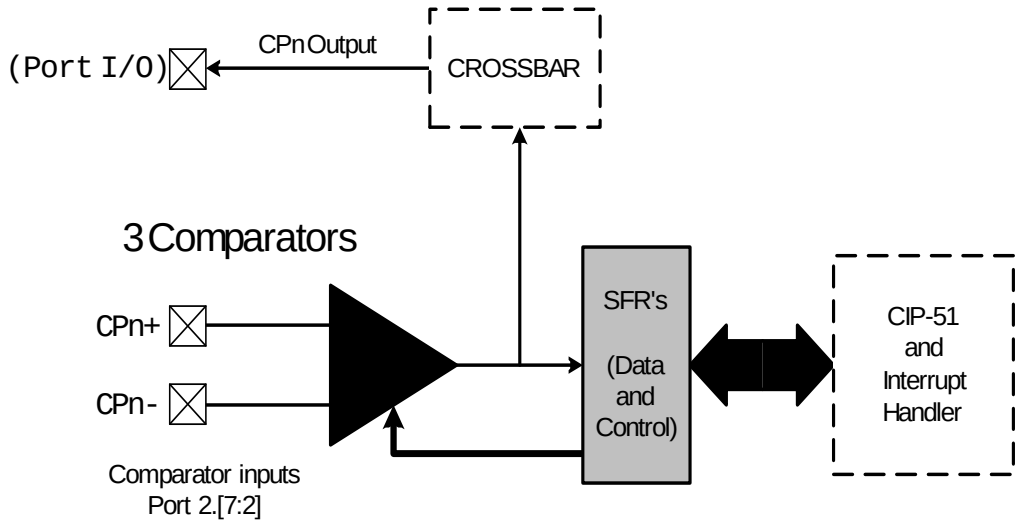
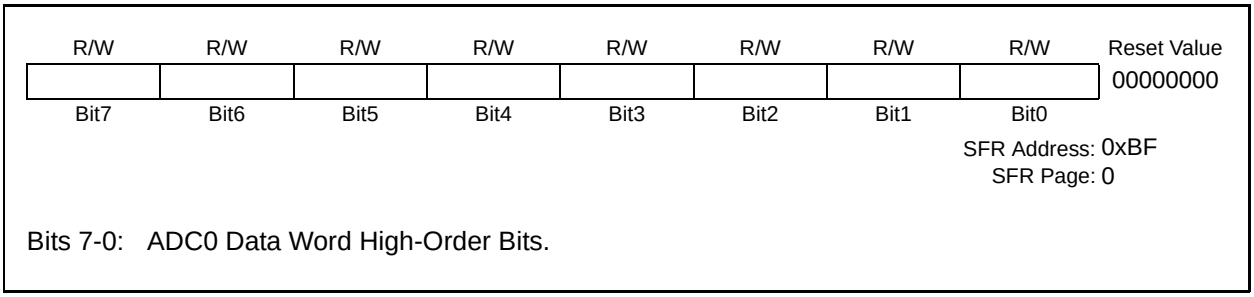


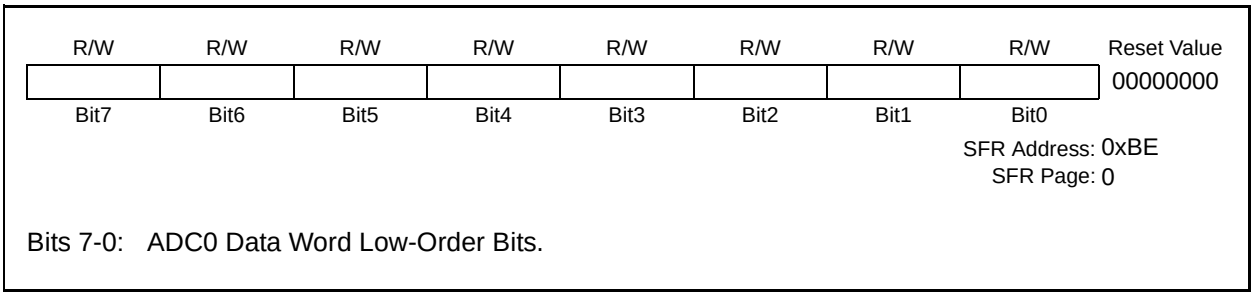
Figure 1.15. Comparator Block Diagram

Figure 5.13. ADC0H: ADC0 Data Word MSB Register



Bits 7-0: ADC0 Data Word High-Order Bits.

Figure 5.14. ADC0L: ADC0 Data Word LSB Register



Bits 7-0: ADC0 Data Word Low-Order Bits.

6.5. Instruction Execution in Mode 1

When the DMA interface begins an operation cycle, the DMA Instruction Status Register (DMA0ISW, Figure 6.9) is loaded with the address contained within the DMA Instruction Boundary Register (DMA0BND, Figure 6.8). The instruction is fetched from the Instruction Buffer, and the DMA Control Logic waits for data from the appropriate ADC(s). At the end of an instruction, the Repeat Counter (Registers DMA0CSH and DMA0CSL) is decremented, and the instruction will be repeated until the Repeat Counter reaches 0x0000. The Repeat Counter is then reset to the Repeat Counter Limit value (Registers DMA0CTH and DMA0CTL), and the DMA will increment DMA0ISW to the next instruction address. When the current DMA instruction is an End of Operation instruction, the Instruction Status Register is reset to the Instruction Boundary Register. If the Continuous Conversion bit (bit 7, CCNV) in the End of Operation instruction word is set to '1', the DMA will continue to execute instructions. When CCNV is set to '0', the DMA will stop executing instructions at this point. An example of Mode 1 operation is shown in Figure 6.3.

Figure 6.3. DMA Mode 1 Operation

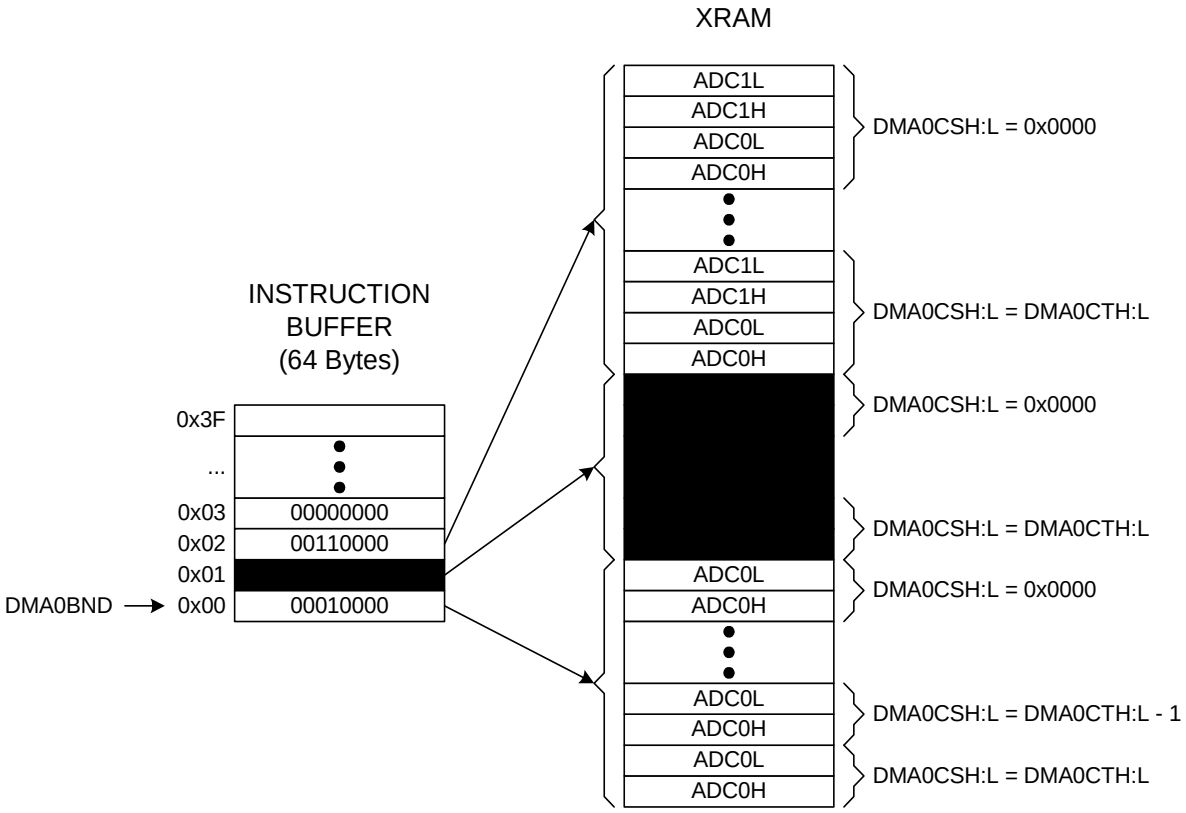


Figure 6.8. DMA0BND: DMA0 Instruction Boundary Register

SFR Page: 3
SFR Address: 0xFD

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	

Bits 7-6: Unused.
Bits 5-0: DMA0 instruction address to begin with when executing DMA instructions.

Figure 6.9. DMA0ISW: DMA0 Instruction Status Register

SFR Page: 3
SFR Address: 0xFE

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	

Bits 7-6: Unused.
Bits 5-0: Contains the address of the current DMA0 Instruction to be executed.

While in the ADC2 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a *high* priority interrupt, while the ADC2 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR Page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 2 for ADC2) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x0F for Port 5) is pushed down to the SFRLAST register, the "bottom" of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 13.6 below.

Figure 13.6. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR

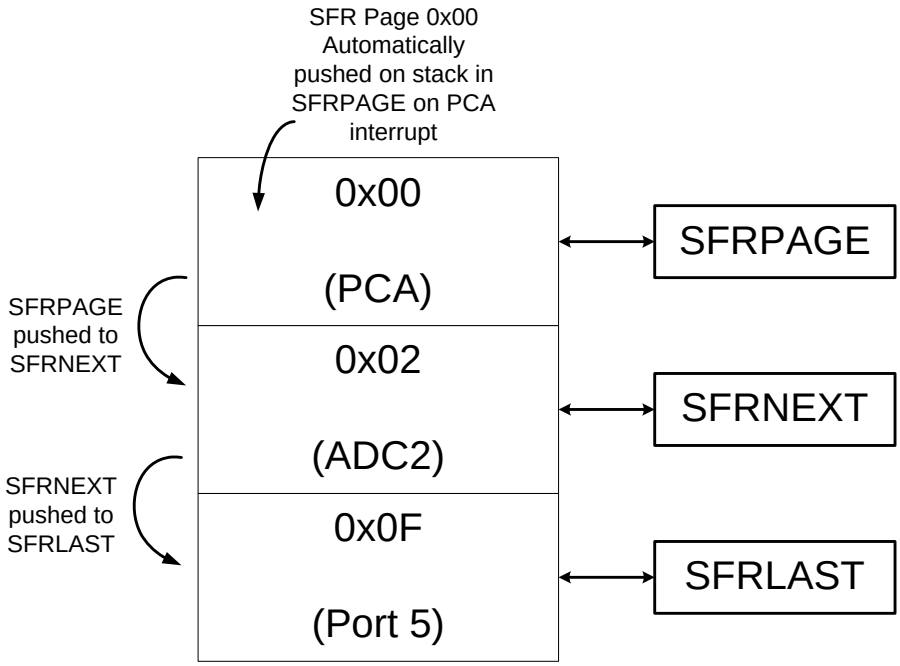


Figure 13.9. SFRPGCN: SFR Page Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	-	SFRPGEN	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x96
SFR Page: F

Bits7-1: Reserved.

Bit0: SFRPGEN: SFR Automatic Page Control Enable.
Upon interrupt, the C8051 Core will vector to the specified interrupt service routine and automatically switch the SFR page to the corresponding peripheral or function's SFR page. This bit is used to control this autopaging function.
0: SFR Automatic Paging disabled. C8051 core will not automatically change to the appropriate SFR page (i.e., the SFR page that contains the SFRs for the peripheral/function that was the source of the interrupt).
1: SFR Automatic Paging enabled. Upon interrupt, the C8051 will switch the SFR page to the page that contains the SFRs for the peripheral or function that is the source of the interrupt.

Figure 13.10. SFRPAGE: SFR Page Register

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: 0x84
SFR Page: All Pages

Bits7-0: SFR Page Bits: Byte Represents the SFR Page the C8051 MCU uses when reading or modifying SFRs.
Write: Sets the SFR Page.
Read: Byte is the SFR page the C8051 MCU is using.

When enabled in the SFR Page Control Register (SFRPGCN), the C8051 will automatically switch to the SFR Page that contains the SFRs of the corresponding peripheral/function that caused the interrupt, and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt).
SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and **not** by reading/writing to the SFRPAGE register)

Table 13.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
P2MDIN	0xAE	F	Port 2 Input Mode	page 217
P2MDOUT	0xA6	F	Port 2 Output Mode Configuration	page 217
P3	0xB0	All Pages	Port 3 Latch	page 218*1
P3MDOUT	0xA7	F	Port 3 Output Mode Configuration	page 218*1
P4	0xC8	F	Port 4 Latch	page 221*1
P4MDOUT	0x9C	F	Port 4 Output Mode Configuration	page 221*1
P5	0xD8	F	Port 5 Latch	page 222*1
P5MDOUT	0x9D	F	Port 5 Output Mode Configuration	page 222*1
P6	0xE8	F	Port 6 Latch	page 223*1
P6MDOUT	0x9E	F	Port 6 Output Mode Configuration	page 223*1
P7	0xF8	F	Port 7 Latch	page 224*1
P7MDOUT	0x9F	F	Port 7 Output Mode Configuration	page 224*1
PCA0CN	0xD8	0	PCA Control	page 312
PCA0CPH0	0xFC	0	PCA Capture 0 High	page 316
PCA0CPH1	0xFE	0	PCA Capture 1 High	page 316
PCA0CPH2	0xEA	0	PCA Capture 2 High	page 316
PCA0CPH3	0xEC	0	PCA Capture 3 High	page 316
PCA0CPH4	0xEE	0	PCA Capture 4 High	page 316
PCA0CPH5	0xE2	0	PCA Capture 5 High	page 316
PCA0CPL0	0xFB	0	PCA Capture 0 Low	page 316
PCA0CPL1	0xFD	0	PCA Capture 1 Low	page 316
PCA0CPL2	0xE9	0	PCA Capture 2 Low	page 316
PCA0CPL3	0xEB	0	PCA Capture 3 Low	page 316
PCA0CPL4	0xED	0	PCA Capture 4 Low	page 316
PCA0CPL5	0xE1	0	PCA Capture 5 Low	page 316
PCA0CPM0	0xDA	0	PCA Module 0 Mode Register	page 314
PCA0CPM1	0xDB	0	PCA Module 1 Mode Register	page 314
PCA0CPM2	0xDC	0	PCA Module 2 Mode Register	page 314
PCA0CPM3	0xDD	0	PCA Module 3 Mode Register	page 314
PCA0CPM4	0xDE	0	PCA Module 4 Mode Register	page 314
PCA0CPM5	0xDF	0	PCA Module 5 Mode Register	page 314
PCA0H	0xFA	0	PCA Counter High	page 315
PCA0L	0xF9	0	PCA Counter Low	page 315
PCA0MD	0xD9	0	PCA Mode	page 313
PCON	0x87	All Pages	Power Control	page 161
PSCTL	0x8F	0	Program Store R/W Control	page 185
PSW	0xD0	All Pages	Program Status Word	page 149
RCAP2H	0xCB	0	Timer/Counter 2 Capture/Reload High	page 301
RCAP2L	0xCA	0	Timer/Counter 2 Capture/Reload Low	page 301
RCAP3H	0xCB	1	Timer/Counter 3 Capture/Reload High	page 301
RCAP3L	0xCA	1	Timer/Counter 3 Capture/Reload Low	page 301
RCAP4H	0xCB	2	Timer/Counter 4 Capture/Reload High	page 301
RCAP4L	0xCA	2	Timer/Counter 4 Capture/Reload Low	page 301

Figure 13.20. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS0	PT1	PX1	PT0	PX0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xB8
SFR Page: All Pages

Bits7-6: UNUSED. Read = 11b, Write = don't care.
Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority level.
1: Timer 2 interrupt set to high priority level.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupt set to high priority level.
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupt set to high priority level.
Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.
Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.
Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

17.5.3. Split Mode with Bank Select

When EMI0CF.[3:2] are set to ‘10’, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the 4 kB boundary will access on-chip XRAM space.
- Effective addresses beyond the 4 kB boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in “Bank Select” mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

17.5.4. External Only

When EMI0CF[3:2] are set to ‘11’, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the 4 kB boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in “Split Mode without Bank Select” described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

The PnMDOUT registers control the output modes of the port pins regardless of whether the Crossbar has allocated the Port pin for a digital peripheral or not. The exceptions to this rule are: the Port pins connected to SDA, SCL, RX0 (if UART0 is in Mode 0), and RX1 (if UART1 is in Mode 0) are always configured as Open-Drain outputs, regardless of the settings of the associated bits in the PnMDOUT registers.

18.1.3. Configuring Port Pins as Digital Inputs

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic 1 to the associated bit in the Port Data register. For example, P3.7 is configured as a digital input by setting P3MDOUT.7 to a logic 0 and P3.7 to a logic 1.

If the Port pin has been assigned to a digital peripheral by the Crossbar and that pin functions as an input (for example RX0, the UART0 receive pin), then the output drivers on that pin are automatically disabled.

18.1.4. Weak Pull-ups

By default, each Port pin has an internal weak pull-up device enabled which provides a resistive connection (about 100 kΩ) between the pin and VDD. The weak pull-up devices can be globally disabled by writing a logic 1 to the Weak Pull-up Disable bit, (WEAKPUD, XBR2.7). The weak pull-up is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pull-up device. The weak pull-up device can also be explicitly disabled on a Port 1 pin by configuring the pin as an Analog Input, as described below.

18.1.5. Configuring Port 1 and 2 pins as Analog Inputs

The pins on Port 1 can serve as analog inputs to the ADC2 analog MUX (C8051F060/1/2/3 only) and the pins on Port 2 can serve as analog inputs to the Comparators (all devices). A Port pin is configured as an Analog Input by writing a logic 0 to the associated bit in the PnMDIN registers. All Port pins default to a Digital Input mode. Configuring a Port pin as an analog input:

1. Disables the digital input path from the pin. This prevents additional power supply current from being drawn when the voltage at the pin is near VDD / 2. A read of the Port Data bit will return a logic 0 regardless of the voltage at the Port pin.
2. Disables the weak pull-up device on the pin.
3. Causes the Crossbar to “skip over” the pin when allocating Port pins for digital peripherals.

Note that the output drivers on a pin configured as an Analog Input are not explicitly disabled. Therefore, the associated PnMDOUT bits of pins configured as Analog Inputs should explicitly be set to logic 0 (Open-Drain output mode), and the associated Port Data bits should be set to logic 1 (high-impedance). Also note that it is not required to configure a Port pin as an Analog Input in order to use it as an input to ADC2 or the Comparators, however, it is strongly recommended. See the analog peripheral’s corresponding section in this datasheet for further information.

Figure 19.1. CAN Controller Diagram

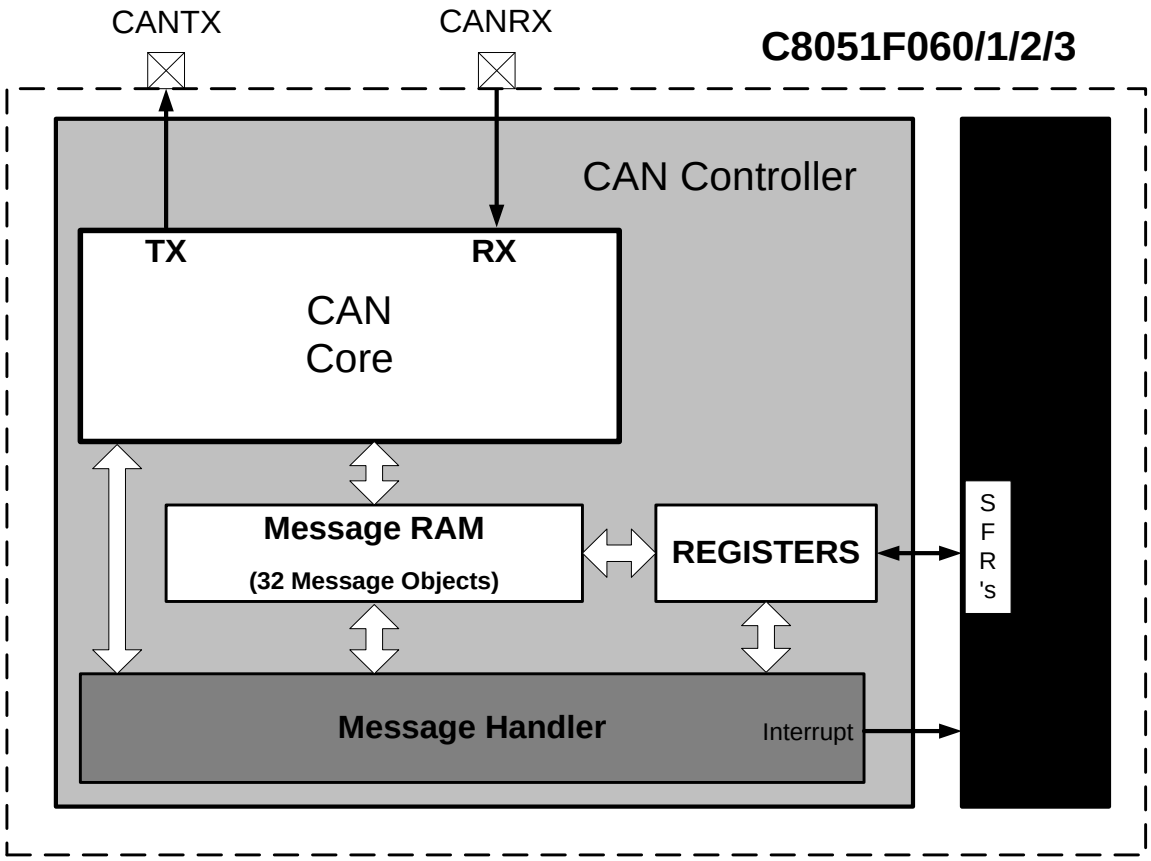
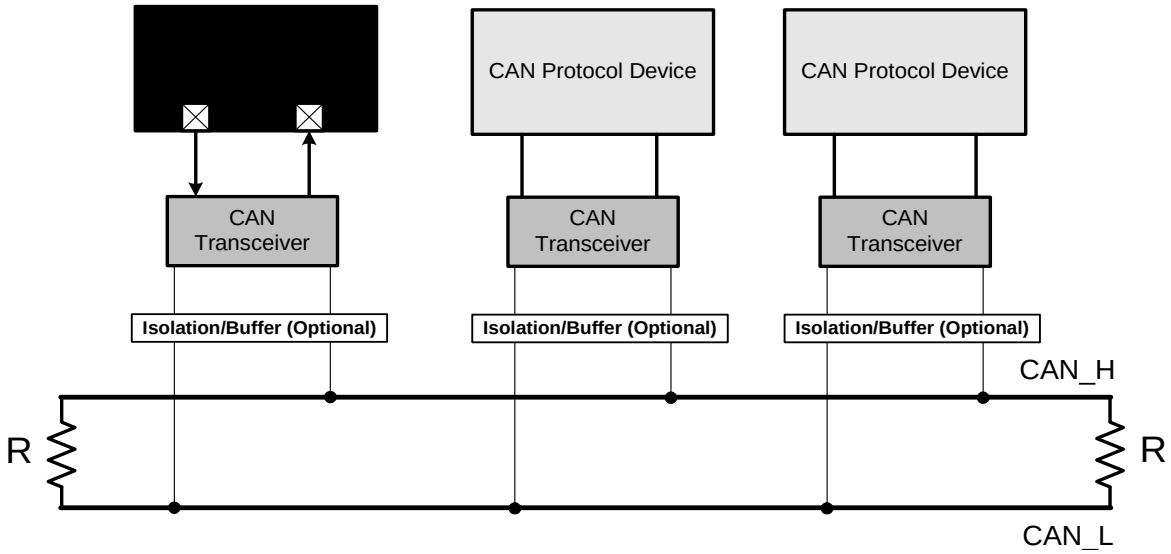


Figure 19.2. Typical CAN Bus Configuration



19.2. CAN Registers

CAN registers are classified as follows:

1. CAN Controller Protocol Registers: CAN control, interrupt, error control, bus status, test modes.
2. Message Object Interface Registers: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The C8051 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. Message Handler Registers: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).
4. C8051 MCU Special Function Registers (SFR): Five registers located in the C8051 MCU memory map that allow direct access to certain CAN Controller Protocol Registers, and Indexed indirect access to all CAN registers.

19.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes. The CAN controller protocol registers are accessible using C8051 MCU SFRs by an indexed method, and some can be accessed directly by addressing the SFRs in the C8051 SFR map for convenience.

The registers are: CAN Control Register (CAN0CN), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register. CAN0STA, CAN0CN, and CAN0TST can be accessed via C8051 MCU SFRs. All others are accessed indirectly using the CAN address indexed method via CAN0ADR, CAN0DATH, and CAN0DATL.

Please refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

19.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers. These registers are accessed via the C8051's CAN0ADR and CAN0DAT registers using the indirect indexed address method.

Please refer to the Bosch CAN User's Guide for information on the function and use of the Message Object Interface Registers.

19.2.3. Message Handler Registers

The Message Handler Registers are *read only* registers. Their flags can be read via the indexed access method with CAN0ADR, CAN0DATH, and CAN0DATL. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

21.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

Figure 21.8. SPI0CFG: SPI0 Configuration Register

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x9A
SFR Page: 0

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.†
1: Data centered on second edge of SCK period.†

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0.
NOTE: RXBMT = 1 when in Master Mode.

†In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 21.1 for timing parameters.

Figure 22.8. SCON0: UART0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SM00	SM10	SM20	REN0	TB80	RB80	TI0	RI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98
SFR Page: 0

Bits7-6: SM00-SM10: Serial Port Operation Mode:
Write:
When written, these bits select the Serial Port Operation Mode as follows:

SM00	SM10	Mode
0	0	Mode 0: Synchronous Mode
0	1	Mode 1: 8-Bit UART, Variable Baud Rate
1	0	Mode 2: 9-Bit UART, Fixed Baud Rate
1	1	Mode 3: 9-Bit UART, Variable Baud Rate

Reading these bits returns the current UART0 mode as defined above.

Bit5: SM20: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port Operation Mode.
Mode 0: No effect.
Mode 1: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level 1.
Mode 2 and 3: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1 and the received address matches the UART0 address or the broadcast address.

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

Bit3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in Modes 2 and 3. It is not used in Modes 0 and 1. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
The bit is assigned the logic level of the ninth bit received in Modes 2 and 3. In Mode 1, if SM20 is logic 0, RB80 is assigned the logic level of the received stop bit. RB8 is not used in Mode 0.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in Mode 0, or at the beginning of the stop bit in other modes). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI0: Receive Interrupt Flag.
Set by hardware when a byte of data has been received by UART0 (as selected by the SM20 bit). 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.

Figure 23.8. SBUF1: Serial (UART1) Port Data Buffer Register

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: 0x99
SFR Page: 1

Bits7-0: SBUF1[7:0]: Serial Data Buffer Bits 7-0 (MSB-LSB).
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF1, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF1 is what initiates the transmission. A read of SBUF1 returns the contents of the receive latch.

Table 23.5. Timer Settings for Standard Baud Rates Using an External Oscillator

	Frequency: 11.0592 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscilla-tor Divide Factor	Timer Clock Source	SCA1-SCA0 (pre-scale select)†	T1M†	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	48	SYSCLK	XX	1	0xE8
	115200	0.00%	96	SYSCLK	XX	1	0xD0
	57600	0.00%	192	SYSCLK	XX	1	0xA0
	28800	0.00%	384	SYSCLK	XX	1	0x40
	14400	0.00%	768	SYSCLK / 12	00	0	0xE0
	9600	0.00%	1152	SYSCLK / 12	00	0	0xD0
	2400	0.00%	4608	SYSCLK / 12	00	0	0x40
	1200	0.00%	9216	SYSCLK / 48	10	0	0xA0
SYSCLK from Internal Osc.	230400	0.00%	48	EXTCLK / 8	11	0	0xFD
	115200	0.00%	96	EXTCLK / 8	11	0	0xFA
	57600	0.00%	192	EXTCLK / 8	11	0	0xF4
	28800	0.00%	384	EXTCLK / 8	11	0	0xE8
	14400	0.00%	768	EXTCLK / 8	11	0	0xD0
	9600	0.00%	1152	EXTCLK / 8	11	0	0xB8

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

Table 23.6. Timer Settings for Standard Baud Rates Using an External Oscillator

	Frequency: 3.6864 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscilla-tor Divide Factor	Timer Clock Source	SCA1-SCA0 (pre-scale select)†	T1M†	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	16	SYSCLK	XX	1	0xF8
	115200	0.00%	32	SYSCLK	XX	1	0xF0
	57600	0.00%	64	SYSCLK	XX	1	0xE0
	28800	0.00%	128	SYSCLK	XX	1	0xC0
	14400	0.00%	256	SYSCLK	XX	1	0x80
	9600	0.00%	384	SYSCLK	XX	1	0x40
	2400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	1200	0.00%	3072	SYSCLK / 12	00	0	0x80
SYSCLK from Internal Osc.	230400	0.00%	16	EXTCLK / 8	11	0	0xFF
	115200	0.00%	32	EXTCLK / 8	11	0	0xFE
	57600	0.00%	64	EXTCLK / 8	11	0	0xFC
	28800	0.00%	128	EXTCLK / 8	11	0	0xF8
	14400	0.00%	256	EXTCLK / 8	11	0	0xF0
	9600	0.00%	384	EXTCLK / 8	11	0	0xE8

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

24.1.2. Mode 1: 16-bit Counter/Timer

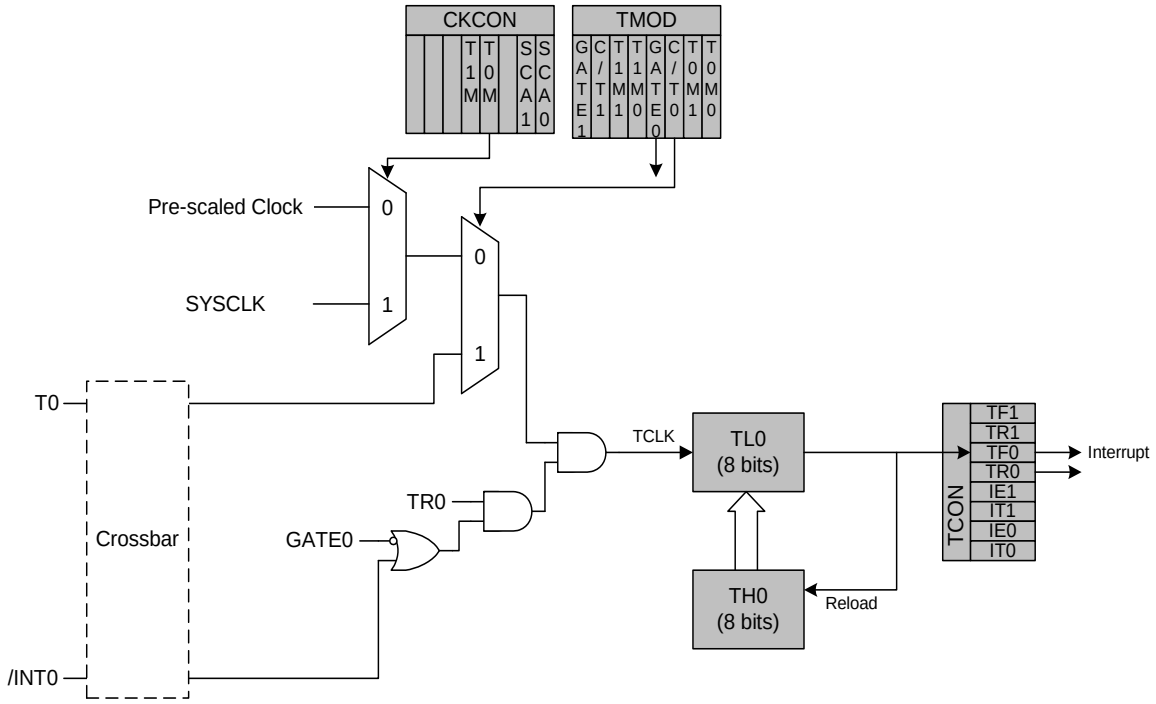
Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

24.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 or Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from 0xFF to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal /INT0 is low.

Figure 24.2. T0 Mode 2 Block Diagram



25.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate pulse width modulated (PWM) outputs on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA0 counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA0 counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be high. When the count value in PCA0L overflows, the CEXn output will be low (see Figure 25.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the counter/timer's high byte (PCA0H) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 25.2.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare 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’.

Equation 25.2. 8-Bit PWM Duty Cycle

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Figure 25.8. PCA 8-Bit PWM Mode Diagram