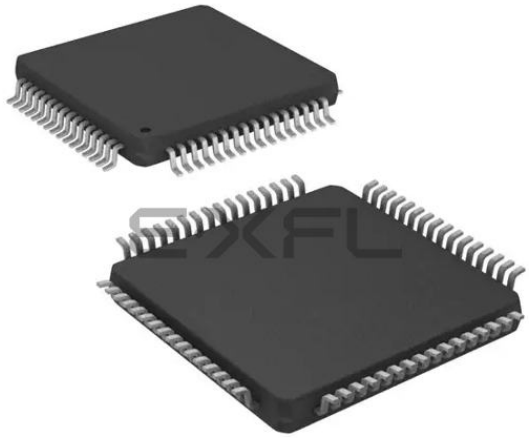

Welcome to E-XFL.COM

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	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 8x8b, 13x12b; D/A 2x10b, 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/c8051f041r

1. System Overview

The C8051F04x family of devices are fully integrated mixed-signal System-on-a-Chip MCUs with 64 digital I/O pins (C8051F040/2/4/6) or 32 digital I/O pins (C8051F041/3/5/7), and an integrated CAN 2.0B controller. Highlighted features are listed below; refer to Table 1.1 for specific product feature selection.

- High-Speed pipelined 8051-compatible CIP-51 microcontroller core (up to 25 MIPS)
- Controller Area Network (CAN 2.0B) Controller with 32 message objects, each with its own indentifier mask.
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 12-bit (C8051F040/1) or 10-bit (C8051F042/3/4/5/6/7) 100 ksps 8-channel ADC with PGA and analog multiplexer
- High Voltage Difference Amplifier input to the 12/10-bit ADC (60 V Peak-to-Peak) with programmable gain.
- True 8-bit 500 ksps 8-channel ADC with PGA and analog multiplexer (C8051F040/1/2/3)
- Two 12-bit DACs with programmable update scheduling (C8051F040/1/2/3)
- 64 kB (C8051F040/1/2/3/4/5) or 32 kB (C8051F046/7) of in-system programmable Flash memory
- 4352 (4096 + 256) bytes of on-chip RAM
- External Data Memory Interface with 64 kB address space
- SPI, SMBus/I^2C, and (2) UART serial interfaces implemented in hardware
- Five general purpose 16-bit Timers
- Programmable Counter/Timer Array with six capture/compare modules
- On-chip Watchdog Timer, V_{DD} Monitor, and Temperature Sensor

With on-chip V_{DD} monitor, Watchdog Timer, and clock oscillator, the C8051F04x family of devices are truly stand-alone System-on-a-Chip solutions. All analog and digital peripherals are enabled/disabled and configured by user firmware. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware.

On-board JTAG debug circuitry allows non-intrusive (uses no on-chip resources), full speed, in-circuit programming and debugging using the production MCU installed in the final application. This debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, Run, and Halt commands. All analog and digital peripherals are fully functional while debugging using JTAG.

Each MCU is specified for 2.7 V to 3.6 V operation over the industrial temperature range (–45 to +85 °C). The Port I/Os, /RST, and JTAG pins are tolerant for input signals up to 5 V. The C8051F040/2/4/6 are available in a 100-pin TQFP and the C8051F041/3/5/7 are available in a 64-pin TQFP.

Table 1.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory	RAM	External Memory Interface	SMBus/I^2C and SPI	CAN	UARTS	Timers (16-bit)	Programmable Counter Array	Digital Port I/O's	12-bit 100ksps ADC	10-bit 100ksps ADC	8-bit 500 ksps ADC Inputs	High Voltage Diff Amp	Voltage Reference	Temperature Sensor	DAC Resolution (bits)	DAC Outputs	Analog Comparators	Lead-free (RoHS Compliant)	Package
C8051F040	25	64 kB	4352	✓	✓	✓	2	5	✓	64	✓	-	8	✓	✓	✓	12	2	3	-	100TQFP
C8051F040-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	64	✓	-	8	✓	✓	✓	12	2	3	✓	100TQFP
C8051F041	25	64 kB	4352	✓	✓	✓	2	5	✓	32	✓	-	8	✓	✓	✓	12	2	3	-	64TQFP
C8051F041-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	32	✓	-	8	✓	✓	✓	12	2	3	✓	64TQFP
C8051F042	25	64 kB	4352	✓	✓	✓	2	5	✓	64	-	✓	8	✓	✓	✓	12	2	3	-	100TQFP
C8051F042-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	64	-	✓	8	✓	✓	✓	12	2	3	✓	100TQFP
C8051F043	25	64 kB	4352	✓	✓	✓	2	5	✓	32	-	✓	8	✓	✓	✓	12	2	3	-	64TQFP
C8051F043-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	32	-	✓	8	✓	✓	✓	12	2	3	✓	64TQFP
C8051F044	25	64 kB	4352	✓	✓	✓	2	5	✓	64	-	✓		✓	✓	✓			3	-	100TQFP
C8051F044-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	64	-	✓		✓	✓	✓			3	✓	100TQFP
C8051F045	25	64 kB	4352	✓	✓	✓	2	5	✓	32	-	✓		✓	✓	✓			3	-	64TQFP
C8051F045-GQ	25	64 kB	4352	✓	✓	✓	2	5	✓	32	-	✓		✓	✓	✓			3	✓	64TQFP
C8051F046	25	32 kB	4352	✓	✓	✓	2	5	✓	64	-	✓		✓	✓	✓			3	-	100TQFP
C8051F046-GQ	25	32 kB	4352	✓	✓	✓	2	5	✓	64	-	✓		✓	✓	✓			3	✓	100TQFP
C8051F047	25	32 kB	4352	✓	✓	✓	2	5	✓	32	-	✓		✓	✓	✓			3	-	64TQFP
C8051F047-GQ	25	32 kB	4352	✓	✓	✓	2	5	✓	32	-	✓		✓	✓	✓			3	✓	64TQFP

1.3. JTAG Debug and Boundary Scan

The C8051F04x family has on-chip JTAG boundary scan and debug circuitry that provides *non-intrusive, full speed, in-circuit debugging using the production part installed in the end application*, via the four-pin JTAG interface. The JTAG port is fully compliant to IEEE 1149.1, providing full boundary scan for test and manufacturing purposes.

Silicon Labs' debugging system supports inspection and modification of memory and registers, breakpoints, watchpoints, a stack monitor, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADC and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized with instruction execution.

The C8051F040DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F04x MCUs. The development kit includes two target boards and a cable to facilitate evaluating a simple CAN communication network. The kit also includes software with a developer's studio and debugger, a target application board with the associated MCU installed, and the required cables and wall-mount power supply. The Serial Adapter takes its power from the application board; it requires roughly 20 mA at 2.7-3.6 V. For applications where there is not sufficient power available from the target system, the provided power supply can be connected directly to the Serial Adapter.

Silicon Labs' debug environment is a vastly superior configuration for developing and debugging embedded applications compared to standard MCU emulators, which use on-board "ICE Chips" and target cables and require the MCU in the application board to be socketed. Silicon Labs' debug environment both increases ease of use and preserves the performance of the precision, on-chip analog peripherals.

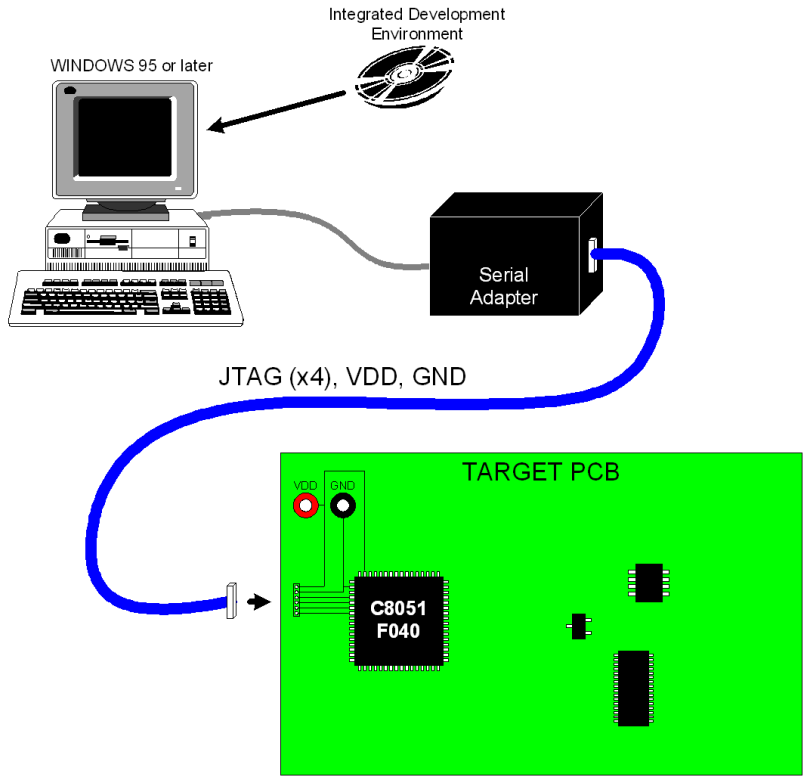

Figure 1.8. Development/In-System Debug Diagram

1.10. Comparators and DACs

Each C8051F040/1/2/3 MCU has two 12-bit DACs, and all C8051F04x devices have three comparators on chip. The MCU data and control interface to each comparator and DAC is via the Special Function Registers. The MCU can place any DAC or comparator in low power shutdown mode.

The comparators have software programmable hysteresis and response time. Each comparator can generate an interrupt on its rising edge, falling edge, or both; these interrupts are capable of waking up the MCU from sleep mode. The comparators' output state can also be polled in software. The comparator outputs can be programmed to appear on the Port I/O pins via the Crossbar.

The DACs are voltage output mode and include a flexible output scheduling mechanism. This scheduling mechanism allows DAC output updates to be forced by a software write or a Timer 2, 3, or 4 overflow. The DAC voltage reference is supplied via the dedicated VREFD input pin on C8051F040/2 devices or via the internal voltage reference on C8051F041/3 devices. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADC.

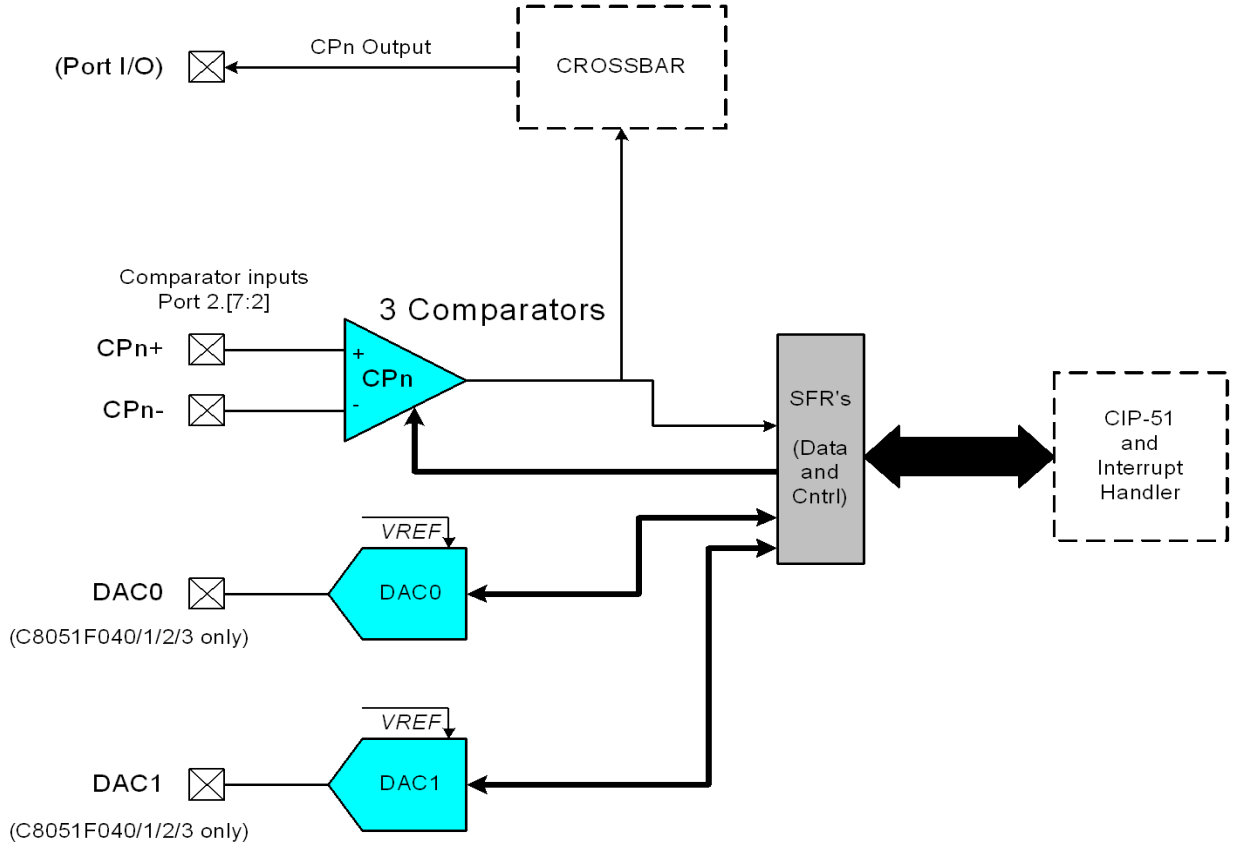

Figure 1.14. Comparator and DAC Diagram

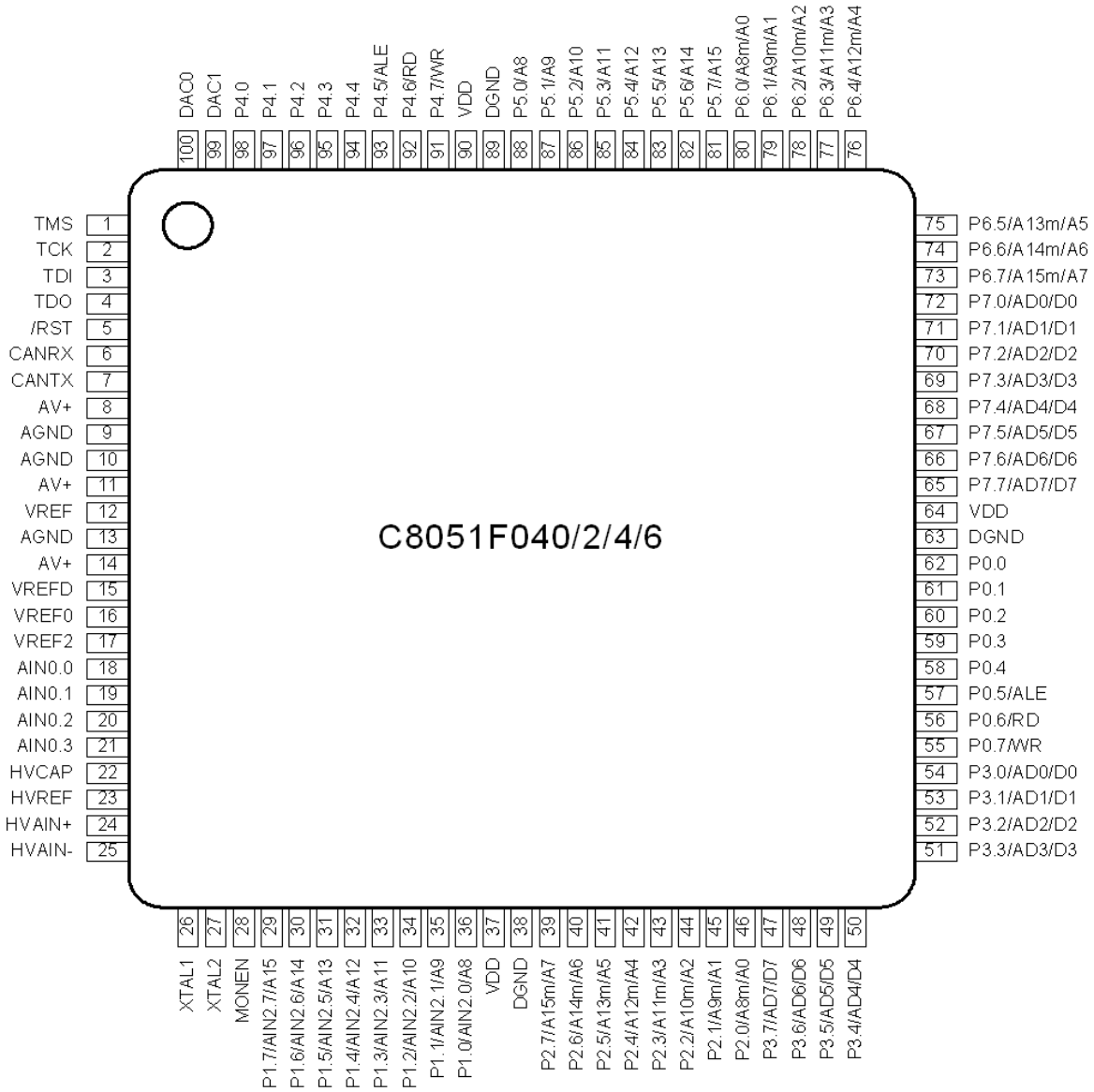

Figure 4.1. TQFP-100 Pinout Diagram

6.1.1. Analog Input Configuration

The analog multiplexer routes signals from external analog input pins, Port 3 I/O pins (programmed to be analog inputs), a High Voltage Difference Amplifier, and an on-chip temperature sensor as shown in Figure 6.2.

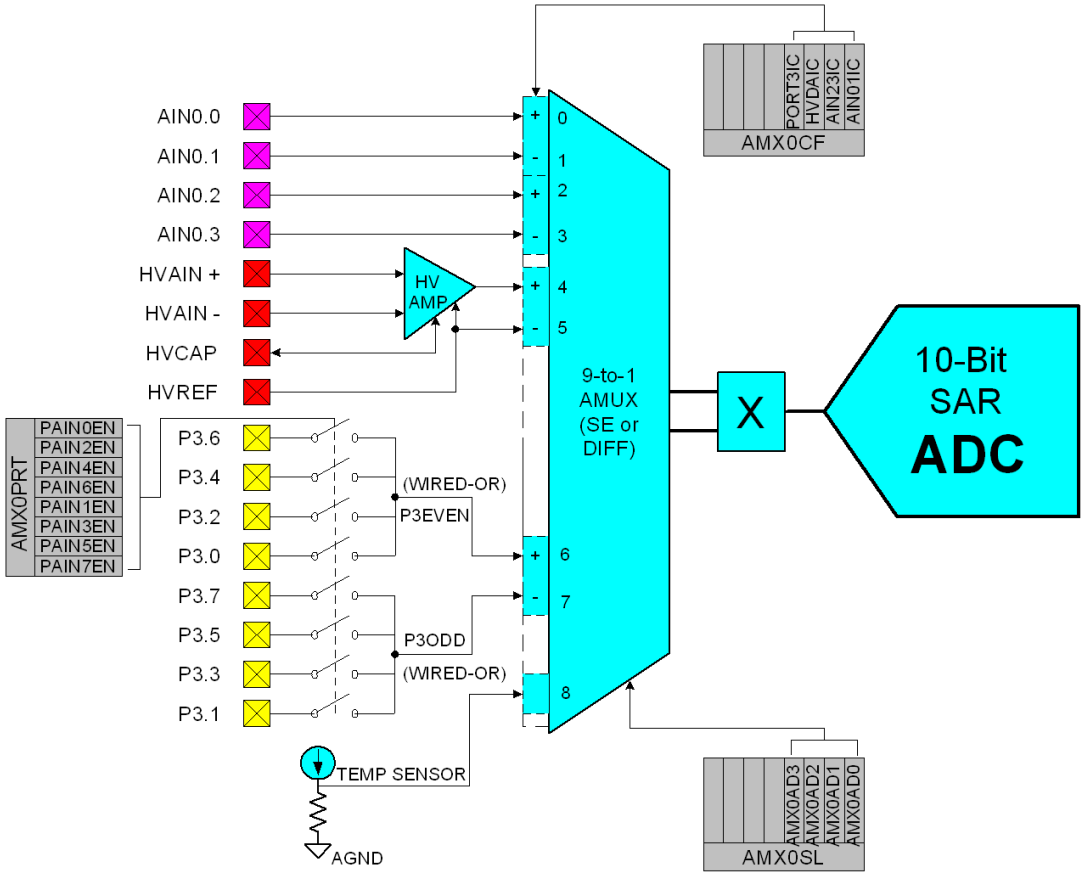

Figure 6.2. Analog Input Diagram

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 6.2.

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 measurement.

SFR Definition 6.3. AMX0PRT: Port 3 Pin Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PAIN7EN	PAIN6EN	PAIN5EN	PAIN4EN	PAIN3EN	PAIN2EN	PAIN1EN	PAIN0EN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBA
SFR Page: 0

Bit7: PAIN7EN: Pin 7 Analog Input Enable Bit
0: P3.7 is not selected as an analog input to the AMUX.
1: P3.7 is selected as an analog input to the AMUX.

Bit6: PAIN6EN: Pin 6 Analog Input Enable Bit
0: P3.6 is not selected as an analog input to the AMUX.
1: P3.6 is selected as an analog input to the AMUX.

Bit5: PAIN5EN: Pin 5 Analog Input Enable Bit
0: P3.5 is not selected as an analog input to the AMUX.
1: P3.5 is selected as an analog input to the AMUX.

Bit4: PAIN4EN: Pin 4 Analog Input Enable Bit
0: P3.4 is not selected as an analog input to the AMUX.
1: P3.4 is selected as an analog input to the AMUX.

Bit3: PAIN3EN: Pin 3 Analog Input Enable Bit
0: P3.3 is not selected as an analog input to the AMUX.
1: P3.3 is enabled as an analog input to the AMUX.

Bit2: PAIN2EN: Pin 2 Analog Input Enable Bit
0: P3.2 is not selected as an analog input to the AMUX.
1: P3.2 is enabled as an analog input to the AMUX.

Bit1: PAIN1EN: Pin 1 Analog Input Enable Bit
0: P3.1 is not selected as an analog input to the AMUX.
1: P3.1 is enabled as an analog input to the AMUX.

Bit0: PAIN0EN: Pin 0 Analog Input Enable Bit
0: P3.0 is not selected as an analog input to the AMUX.
1: P3.0 is enabled as an analog input to the AMUX.

NOTE: Any number of Port 3 pins may be selected simultaneously inputs to the AMUX. Odd numbered and even numbered pins that are selected simultaneously are shorted together as “wired-OR”.

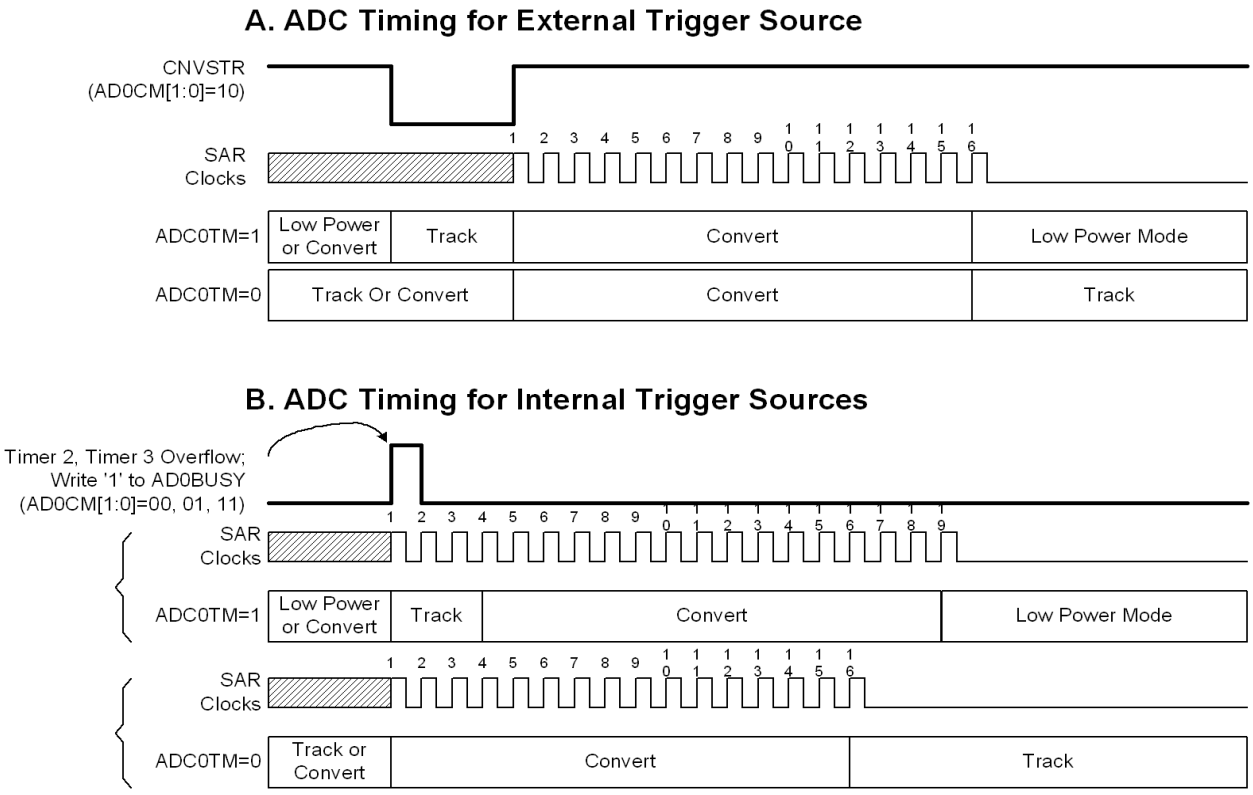

Figure 6.4. 10-Bit ADC Track and Conversion Example Timing

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

Table 10.1. Voltage Reference Electrical Characteristics

V_{DD} = 3.0 V, AV+ = 3.0 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal Reference (REFBE = 1)					
Output Voltage	25 °C ambient	2.36	2.43	2.48	V
VREF Short-Circuit Current		—	—	30	mA
VREF Temperature Coefficient		—	15	—	ppm/°C
Load Regulation	Load = 0 to 200 µA to AGND	—	0.5	—	ppm/µA
VREF Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass	—	2	—	ms
VREF Turn-on Time 2	0.1 µF ceramic bypass	—	20	—	µs
VREF Turn-on Time 3	no bypass cap	—	10	—	µs
Reference Buffer Power Supply Current		—	40	—	µA
Power Supply Rejection		—	140	—	ppm/V
External Reference (REFBE = 0)					
Input Voltage Range		1.00	—	(AV+) – 0.3	V
Input Current		—	0	1	µA

SFR Definition 11.2. CPTnMD: Comparator Mode Selection

R/W	R/W	R/W	R/W	R	R	R/W	R/W	Reset Value
-	-	CPnRIE	CPnFIE	-	-	CPnMD1	CPnMD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: CPT0MD: 0x89; CPT1MD: 0x89;CPT2MD: 0x89

SFR Page: CPT0MD:page 1;CPT1MD:page 2; CPT2MD:page 3

Bits7-6: UNUSED. Read = 00b, Write = don't care.

Bit 5: CPnRIE: Comparator Rising-Edge Interrupt Enable Bit.
0: Comparator rising-edge interrupt disabled.
1: Comparator rising-edge interrupt enabled.

Bit 4: CPnFIE: Comparator Falling-Edge Interrupt Enable Bit.
0: Comparator falling-edge interrupt disabled.
1: Comparator falling-edge interrupt enabled.

Bits3-2: UNUSED. Read = 00b, Write = don't care.

Bits1-0: CPnMD1-CPnMD0: Comparator Mode Select
These bits select the response time for the Comparator.

Mode	CPnMD1	CPnMD0	CPn Typical Response Time
0	0	0	Fastest Response Time
1	0	1	—
2	1	0	—
3	1	1	Lowest Power Consumption

and a RET pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the 32-bit shift register, and can notify the debug software even with the MCU running at speed.

12.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFR's). The SFR's provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFR's found in a typical 8051 implementation as well as implementing additional SFR's used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 12.2 lists the SFR's implemented in the CIP-51 System Controller.

The SFR registers are accessed whenever the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFR's with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFR's are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 12.3, for a detailed description of each register.

12.2.6.1. SFR Paging

The CIP-51 features *SFR paging,* allowing the device to map many SFR's into the 0x80 to 0xFF memory address space. The SFR memory space has 256 *pages*. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFR's. The C8051F04x family of devices utilizes five SFR pages: 0, 1, 2, 3, and F. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE (see SFR Definition 12.2). The procedure for reading and writing an SFR is as follows:

1. Select the appropriate SFR page number using the SFRPAGE register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

12.2.6.2. Interrupts and SFR Paging

When an interrupt occurs, the SFR Page Register will automatically switch to the SFR page containing the flag bit that caused the interrupt. The automatic SFR Page switch function conveniently removes the burden of switching SFR pages from the interrupt service routine. Upon execution of the RETI instruction, the SFR page is automatically restored to the SFR Page in use prior to the interrupt. This is accomplished via a three-byte *SFR Page Stack*. The top byte of the stack is SFRPAGE, the current SFR Page. The second byte of the SFR Page Stack is SFRNEXT. The third, or bottom byte of the SFR Page Stack is SFRLAST. On interrupt, the current SFRPAGE value is pushed to the SFRNEXT byte, and the value of SFRNEXT is pushed to SFRLAST. Hardware then loads SFRPAGE with the SFR Page containing the flag bit associated with the interrupt. On a return from interrupt, the SFR Page Stack is popped resulting in the value of SFRNEXT returning to the SFRPAGE register, thereby restoring the SFR page context without software intervention. The value in SFRLAST (0x00 if there is no SFR Page value in the bottom of the stack) of the stack is placed in SFRNEXT register. If desired, the values stored in SFRNEXT and SFRLAST may be modified during an interrupt, enabling the CPU to return to a different SFR Page upon execution of the RETI instruction (on interrupt exit). Modifying registers in the SFR Page Stack does not cause a push or pop of the stack. Only interrupt calls and returns will cause push/pop operations on the SFR Page Stack.

Table 13.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{RST}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 V to 3.6 V	—	—	0.6	V
$\overline{RST}$ Input High Voltage		0.7 x V_{DD}	—	—	V
$\overline{RST}$ Input Low Voltage		—	—	0.3 x V_{DD}	
$\overline{RST}$ Input Leakage Current	$\overline{RST}$ = 0.0 V	—	50	—	µA
V_{DD} for /RST Output Valid		1.0	—	—	V
AV+ for /RST Output Valid		1.0	—	—	V
V_{DD} POR Threshold (V_{RST})		2.40	2.55	2.70	V
Minimum /RST Low Time to Generate a System Reset		10	—	—	ns
Reset Time Delay	$\overline{RST}$ rising edge after V_{DD} crosses V_{RST} threshold	80	100	120	ms
Missing Clock Detector Timeout	Time from last system clock to reset initiation	100	220	500	µs

19.4. SMBus Special Function Registers

The SMBus0 serial interface is accessed and controlled through five SFRs: SMB0CN Control Register, SMB0CR Clock Rate Register, SMB0ADR Address Register, SMB0DAT Data Register and SMB0STA Status Register. The five special function registers related to the operation of the SMBus0 interface are described in the following sections.

19.4.1. Control Register

The SMBus0 Control register SMB0CN is used to configure and control the SMBus0 interface. All of the bits in the register can be read or written by software. Two of the control bits are also affected by the SMBus0 hardware. The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by the hardware when a valid serial interrupt condition occurs. It can only be cleared by software. The Stop flag (STO, SMB0CN.4) is set to logic 1 by software. It is cleared to logic 0 by hardware when a STOP condition is detected on the bus.

Setting the ENSMB flag to logic 1 enables the SMBus0 interface. Clearing the ENSMB flag to logic 0 disables the SMBus0 interface and removes it from the bus. Momentarily clearing the ENSMB flag and then resetting it to logic 1 will reset SMBus0 communication. However, ENSMB should not be used to temporarily remove a device from the bus since the bus state information will be lost. Instead, the Assert Acknowledge (AA) flag should be used to temporarily remove the device from the bus (see description of AA flag below).

Setting the Start flag (STA, SMB0CN.5) to logic 1 will put SMBus0 in a master mode. If the bus is free, SMBus0 will generate a START condition. If the bus is not free, SMBus0 waits for a STOP condition to free the bus and then generates a START condition after a 5 µs delay per the SMB0CR value (In accordance with the SMBus protocol, the SMBus0 interface also considers the bus free if the bus is idle for 50 µs and no STOP condition was recognized). If STA is set to logic 1 while SMBus0 is in master mode and one or more bytes have been transferred, a repeated START condition will be generated.

When the Stop flag (STO, SMB0CN.4) is set to logic 1 while the SMBus0 interface is in master mode, the interface generates a STOP condition. In a slave mode, the STO flag may be used to recover from an error condition. In this case, a STOP condition is not generated on the bus, but the SMBus hardware behaves as if a STOP condition has been received and enters the "not addressed" slave receiver mode. Note that this simulated STOP will not cause the bus to appear free to SMBus0. The bus will remain occupied until a STOP appears on the bus or a Bus Free Timeout occurs. Hardware automatically clears the STO flag to logic 0 when a STOP condition is detected on the bus.

The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by hardware when the SMBus0 interface enters any one of the 28 possible states except the Idle state. If interrupts are enabled for the SMBus0 interface, an interrupt request is generated when the SI flag is set. The SI flag must be cleared by software.

Important Note: If SI is set to logic 1 while the SCL line is low, the clock-low period of the serial clock will be stretched and the serial transfer is suspended until SI is cleared to logic 0. A high level on SCL is not affected by the setting of the SI flag.

The Assert Acknowledge flag (AA, SMB0CN.2) is used to set the level of the SDA line during the acknowledge clock cycle on the SCL line. Setting the AA flag to logic 1 will cause an ACK (low level on SDA) to be sent during the acknowledge cycle if the device has been addressed. Setting the AA flag to logic 0 will cause a NACK (high level on SDA) to be sent during acknowledge cycle. After the transmission of a byte in slave mode, the slave can be temporarily removed from the bus by clearing the AA flag. The slave's own address and general call address will be ignored. To resume operation on the bus, the AA flag must be reset to logic 1 to allow the slave's address to be recognized.

19.4.2. Clock Rate Register

SFR Definition 19.2. SMB0CR: SMBus0 Clock Rate

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: 0xCF
SFR Page: 0

Bits7-0: SMB0CR.[7:0]: SMBus0 Clock Rate Preset
The SMB0CR Clock Rate register controls the frequency of the serial clock SCL in master mode. The 8-bit word stored in the SMB0CR Register preloads a dedicated 8-bit timer. The timer counts up, and when it rolls over to 0x00, the SCL logic state toggles.

The SMB0CR setting should be bounded by the following equation, where *SMB0CR* is the unsigned 8-bit value in register SMB0CR, and *SYSCLK* is the system clock frequency in Hz:

$$SMB0CR < ((288 - 0.85 \times SYSCLK) / 1.124\text{E}6)$$

The resulting SCL signal high and low times are given by the following equations:

$$T_{LOW} = (256 - SMB0CR) / SYSCLK$$

$$T_{HIGH} \cong (258 - SMB0CR) / SYSCLK + 625ns$$

Using the same value of SMB0CR from above, the Bus Free Timeout period is given in the following equation:

$$T_{BFT} \cong 10 \times \frac{(256 - SMB0CR) + 1}{SYSCLK}$$

Table 19.1. SMB0STA Status Codes and States (Continued)

Mode	Status Code	SMBus State	Typical Action
Slave Receiver	0x60	Own slave address + W received. ACK transmitted.	Wait for data.
	0x68	Arbitration lost in sending SLA + R/W as master. Own address + W received. ACK transmitted.	Save current data for retry when bus is free. Wait for data.
	0x70	General call address received. ACK transmitted.	Wait for data.
	0x78	Arbitration lost in sending SLA + R/W as master. General call address received. ACK transmitted.	Save current data for retry when bus is free.
	0x80	Data byte received. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x88	Data byte received. NACK transmitted.	Set STO to reset SMBus.
	0x90	Data byte received after general call address. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x98	Data byte received after general call address. NACK transmitted.	Set STO to reset SMBus.
	0xA0	STOP or repeated START received.	No action necessary.
Slave Transmitter	0xA8	Own address + R received. ACK transmitted.	Load SMB0DAT with data to transmit.
	0xB0	Arbitration lost in transmitting SLA + R/W as master. Own address + R received. ACK transmitted.	Save current data for retry when bus is free. Load SMB0DAT with data to transmit.
	0xB8	Data byte transmitted. ACK received.	Load SMB0DAT with data to transmit.
	0xC0	Data byte transmitted. NACK received.	Wait for STOP.
	0xC8	Last data byte transmitted (AA=0). ACK received.	Set STO to reset SMBus.
Slave	0xD0	SCL Clock High Timer per SMB0CR timed out	Set STO to reset SMBus.
All	0x00	Bus Error (illegal START or STOP)	Set STO to reset SMBus.
	0xF8	Idle	State does not set SI.

21.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

Mode 1 provides standard asynchronous, full-duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if SM20 is logic 1, the stop bit must be logic 1.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 are set.

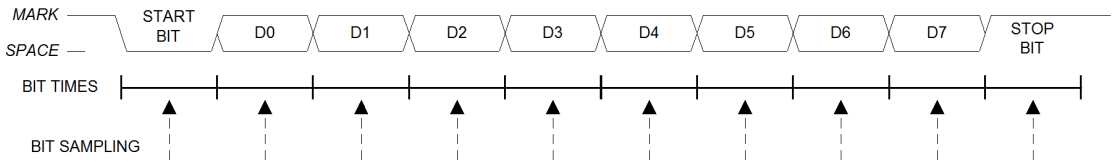

Figure 21.4. UART0 Mode 1 Timing Diagram

The baud rate generated in Mode 1 is a function of timer overflow, shown in Equation 21.1 and Equation 21.3. UART0 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 2, 3, or 4 operating in *Auto-reload Mode* to generate the baud rate (note that the TX and RX clocks are selected separately). On each timer overflow event (a rollover from all ones—0xFF for Timer 1, 0xFFFF for Timers 2, 3 and 4—to zero) a clock is sent to the baud rate logic.

Timers 1, 2, 3, and 4 are selected as the baud rate source with bits in the SSTA0 register (see SFR Definition 21.2). The transmit baud rate clock is selected using the S0TCLK1 and S0TCLK0 bits, and the receive baud rate clock is selected using the S0RCLK1 and S0RCLK0 bits.

The Mode 1 baud rate equations are shown below, where T1M is bit4 of register CKCON, TH1 is the 8-bit reload register for Timer 1, and [RCAPnH, RCAPnL] is the 16-bit reload register for Timer 2, 3, or 4.

When SMOD0 = 0:

$$\text{Mode1\_BaudRate} = 1/32 \times \text{Timer1\_OverflowRate}$$

When SMOD0 = 1:

$$\text{Mode1\_BaudRate} = 1/16 \times \text{Timer1\_OverflowRate}$$

Equation 21.1. Mode 1 Baud Rate using Timer 1

The Timer 1 overflow rate is determined by the Timer 1 clock source (T1CLK) and reload value (TH1). The frequency of T1CLK is selected as described in **Section "23.1. Timer 0 and Timer 1" on page 289**. The Timer 1 overflow rate is calculated as shown in Equation 21.2.

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

Mode 2 configures Timer 0 and 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.

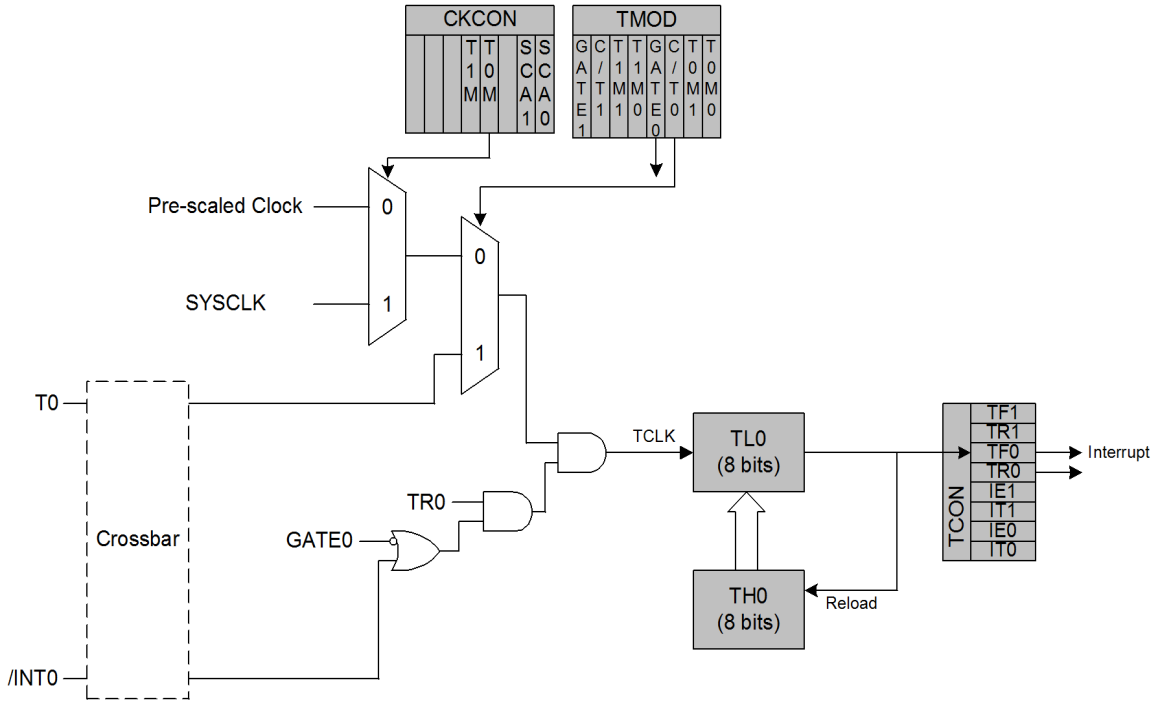

Figure 23.2. T0 Mode 2 Block Diagram

SFR Definition 23.13. TMRnH Timer n High 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: TMR2H: 0xCD; TMR3H: 0xCD; TMR4H: 0xCD

SFR Page: TMR2H: page 0; TMR3H: page 1; TMR4H: page 2

Bits 7-0: TMRnH: Timer n High Byte.
The TMRnH register contains the high byte of the 16-bit Timer n

SFR Definition 24.4. PCA0L: PCA0 Counter/Timer 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: 0xF9
SFR Page: 0

Bits 7-0: PCA0L: PCA0 Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA0 Counter/Timer.

SFR Definition 24.5. PCA0H: PCA0 Counter/Timer High 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: 0xFA
SFR Page: 0

Bits 7-0: PCA0H: PCA0 Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA0 Counter/Timer.