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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	8
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	10-VFDFN Exposed Pad
Supplier Device Package	11-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f300-gmr

1.1.3. Additional Features

The C8051F300/1/2/3/4/5 SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

The extended interrupt handler provides 12 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multitasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} monitor (forces reset when power supply voltage drops below 2.7 V), a Watchdog Timer, a Missing Clock Detector, a voltage level detection from Comparator0, a forced software reset, an external reset pin, and an illegal Flash read/write protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash protection may be disabled by the user in software. The WDT may be permanently enabled in software after a power-on reset during MCU initialization.

The internal oscillator is available as a factory calibrated 24.5 MHz ±2% (C8051F300/1 devices); an uncalibrated version is available on C8051F302/3/4/5 devices. On all C8051F300/1/2/3/4/5 devices, the internal oscillator period may be user programmed in ~0.5% increments. An external oscillator drive circuit is also included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock. If desired, the system clock source may be switched on-the-fly to the external oscillator circuit. An external oscillator can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the fast (up to 25 MHz) internal oscillator as needed.

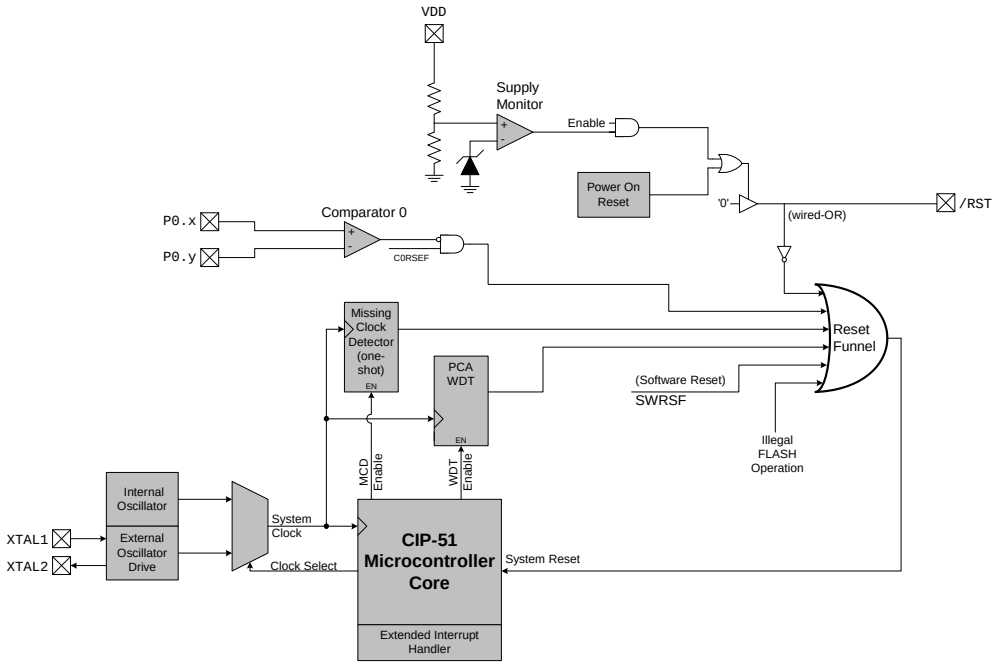

Figure 1.4. On-Chip Clock and Reset

5. ADC0 (8-Bit ADC, C8051F300/2)

The ADC0 subsystem for the C8051F300/2 consists of two analog multiplexers (referred to collectively as AMUX0) with 11 total input selections, a differential programmable gain amplifier (PGA), and a 500 ksps, 8-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 5.1). The AMUX0, PGA, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. ADC0 operates in both Single-ended and Differential modes, and may be configured to measure any Port pin, the Temperature Sensor output, or V_{DD} with respect to any Port pin or GND. The ADC0 subsystem is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

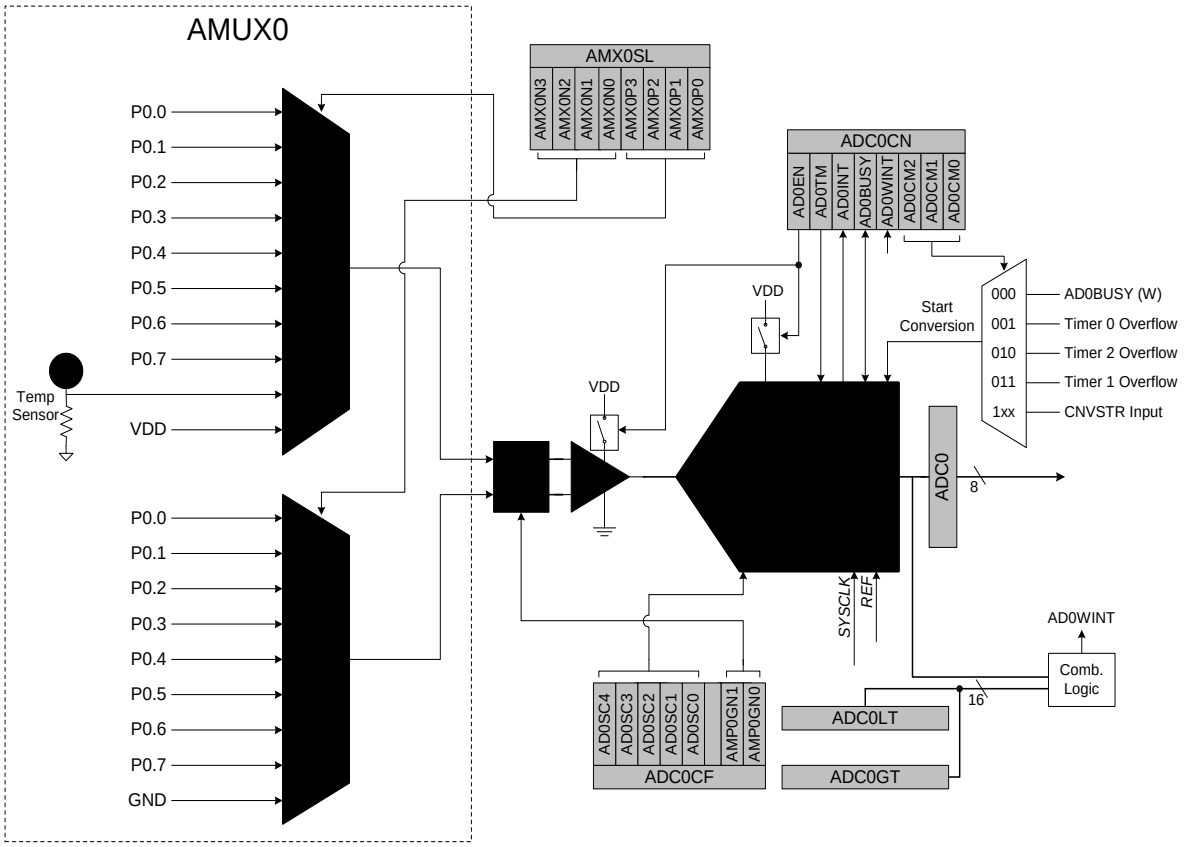

Figure 5.1. ADC0 Functional Block Diagram

5.1. Analog Multiplexer and PGA

The analog multiplexers (AMUX0) select the positive and negative inputs to the PGA, allowing any Port pin to be measured relative to any other Port pin or GND. Additionally, the on-chip temperature sensor or the positive power supply (V_{DD}) may be selected as the positive PGA input. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; all other times, ADC0 operates in Differential Mode.** The ADC0 input channels are selected in the AMX0SL register as described in SFR Definition 5.1.

The conversion code format differs in Single-ended versus Differential modes, as shown below. When in Single-ended Mode (negative input is selected GND), conversion codes are represented as 8-bit unsigned integers. Inputs are measured from ‘0’ to VREF x 255/256. Example codes are shown below.

Input Voltage	ADC0 Output (Conversion Code)
VREF x 255/256	0xFF
VREF x 128/256	0x80
VREF x 64/256	0x40
0	0x00

When in Differential Mode (negative input is not selected as GND), conversion codes are represented as 8-bit signed 2s complement numbers. Inputs are measured from –VREF to VREF x 127/128. Example codes are shown below.

Input Voltage	ADC0 Output (Conversion Code)
VREF x 127/128	0x7F
VREF x 64/128	0x40
0	0x00
–VREF x 64/128	0xC0
–VREF	0x80

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to ‘0’ the corresponding bit in register P0MDIN. To force the Crossbar to skip a Port pin, set to ‘1’ the corresponding bit in register XBR0. See **Section “12. Port Input/Output” on page 103** for more Port I/O configuration details.

The PGA amplifies the AMUX0 output signal as defined by the AMP0GN1-0 bits in the ADC0 Configuration register (SFR Definition 5.2). The PGA is software-programmable for gains of 0.5, 1, 2, or 4. The gain defaults to 0.5 on reset.

5.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the positive PGA input when the temperature sensor is selected by bits AMX0P2-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings.

SFR Definition 6.1. REF0CN: Reference Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	REFSL	TEMPE	BIASE	—	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD1

Bits7–3: UNUSED. Read = 00000b; Write = don't care.
Bit3: REFSL: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: VREF input pin used as voltage reference.
1: V_{DD} used as voltage reference.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit1: BIASE: Internal Analog Bias Generator Enable Bit. (Must be '1' if using ADC).
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit0: UNUSED. Read = 0b. Write = don't care.

Table 6.1. External Voltage Reference Circuit Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range		0	—	V_{DD}	V
Input Current	Sample Rate = 500 ksps; VREF = 3.0 V	—	12	—	µA

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3

Table 8.3. Special Function Registers* (Continued)

Register	Address	Description	Page No.
TH1	0x8D	Timer/Counter 1 High	150
TL0	0x8A	Timer/Counter 0 Low	150
TL1	0x8B	Timer/Counter 1 Low	150
TMOD	0x89	Timer/Counter Mode	148
TMR2RLH	0xCB	Timer/Counter 2 Reload High	154
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	154
TMR2H	0xCD	Timer/Counter 2 High	154
TMR2L	0xCC	Timer/Counter 2 Low	154
XBR0	0xE1	Port I/O Crossbar Control 0	107
XBR1	0xE2	Port I/O Crossbar Control 1	107
XBR2	0xE3	Port I/O Crossbar Control 2	108
0x97, 0xAE, 0xAF, 0xB4, 0xB6, 0xBF, 0xCE, 0xD2, 0xD3, 0xD4, 0xD5, 0xD6, 0xD7, 0xDD, 0xDE, 0xDF, 0xF5		Reserved	
***Note:** SFRs are listed in alphabetical order. All undefined SFR locations are reserved			

8.2.7. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic l. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

SFR Definition 8.1. DPL: Data Pointer 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: 0x82

Bits7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory.

8.4.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout of 100 µsec.

SFR Definition 8.12. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x87

Bits7–2: GF5–GF0: General Purpose Flags 5-0.
These are general purpose flags for use under software control.

Bit1: STOP: Stop Mode Select.
Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0.
1: CPU goes into Stop mode (turns off internal oscillator).

Bit0: IDLE: Idle Mode Select.
Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.
1: CPU goes into Idle mode (shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active).

NOTES:

9. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. Refer to **Section “11. Oscillators” on page 97** for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (**Section “16.3. Watchdog Timer Mode” on page 164** details the use of the Watchdog Timer). Once the system clock source is stable, program execution begins at location 0x0000.

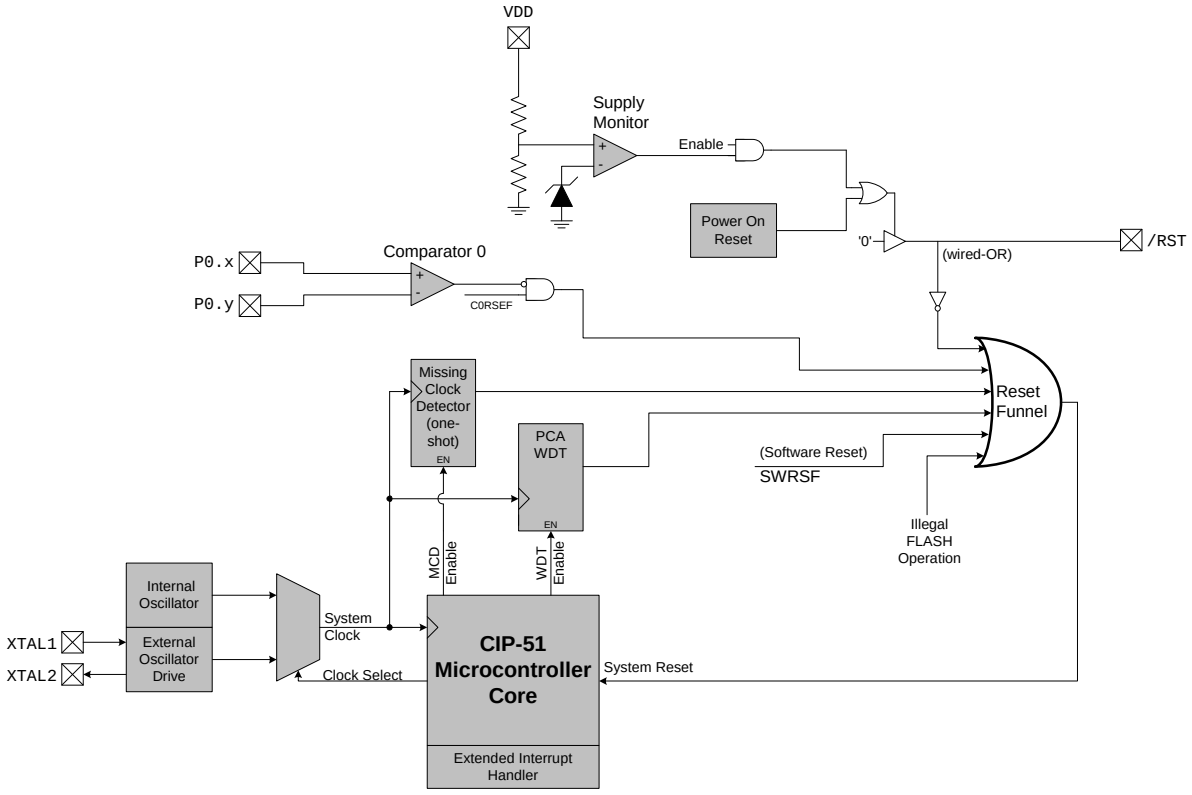

Figure 9.1. Reset Sources

13.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

SMBus interrupts are generated for each data byte or slave address that is transferred. When transmitting, this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data, this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See **Section “13.5. SMBus Transfer Modes” on page 123** for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in **Section “13.4.2. SMB0CN Control Register” on page 119**; Table 13.4 provides a quick SMB0CN decoding reference.

SMBus configuration options include:

- Timeout detection (SCL Low Timeout and/or Bus Free Timeout)
- SDA setup and hold time extensions
- Slave event enable/disable
- Clock source selection

These options are selected in the SMB0CF register, as described in **Section “13.4.1. SMBus Configuration Register” on page 116**.

Table 13.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	• A START is generated.	• A STOP is generated. • Arbitration is lost.
TXMODE	• START is generated. • The SMBus interface enters transmitter mode (after SMB0DAT is written before the start of an SMBus frame).	• A START is detected. • Arbitration is lost. • SMB0DAT is not written before the start of an SMBus frame.
STA	• A START followed by an address byte is received.	• Must be cleared by software.
STO	• A STOP is detected while addressed as a slave. • Arbitration is lost due to a detected STOP.	• A pending STOP is generated.
ACKRQ	• A byte has been received and an ACK response value is needed.	• After each ACK cycle.
ARBLOST	• A repeated START is detected as a MASTER when STA is low (unwanted repeated START). • SCL is sensed low while attempting to generate a STOP or repeated START condition. • SDA is sensed low while transmitting a ‘1’ (excluding ACK bits).	• Each time SI is cleared.
ACK	• The incoming ACK value is low (ACKNOWLEDGE).	• The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	• A START has been generated. • Lost arbitration. • A byte has been transmitted and an ACK/NACK received. • A byte has been received. • A START or repeated START followed by a slave address + R/W has been received. • A STOP has been received.	• Must be cleared by software.

13.4.3. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 13.3. SMB0DAT: SMBus Data

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: 0xC2

Bits7–0: SMB0DAT: SMBus Data.
The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic one. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

13.5.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. After each byte is received, ACKRQ is set to ‘1’ and an interrupt is generated. Software must write the ACK bit (SMB0CN.1) to define the outgoing acknowledge value (Note: writing a ‘1’ to the ACK bit generates an ACK; writing a ‘0’ generates a NACK). Software should write a ‘0’ to the ACK bit after the last byte is received, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. Note that the interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 13.6 shows a typical Master Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

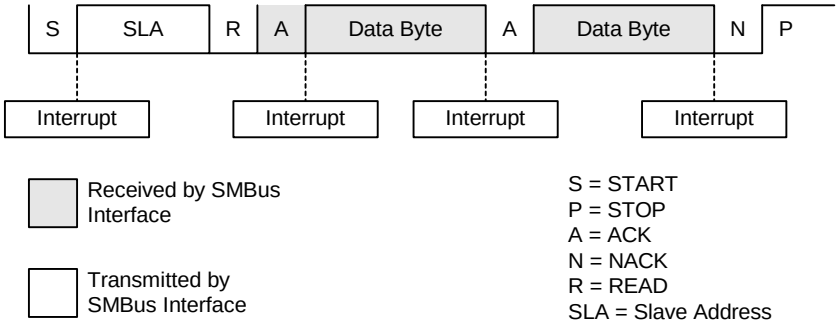

Figure 13.6. Typical Master Receiver Sequence

13.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. In the table below, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. Note that the shown response options are only the typical responses; application-specific procedures are allowed as long as they conform with the SMBus specification. Highlighted responses are allowed but do not conform to the SMBus specification.

Table 13.4. SMBus Status Decoding

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X
						Abort transfer.	0	1	X
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT	0	0	X
						End transfer with STOP	0	1	X
						End transfer with STOP and start another transfer.	1	1	X
						Send repeated START	1	0	X
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X

SFR Definition 14.1. SCON0: Serial Port 0 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S0MODE	—	MCE0	REN0	TB80	RB80	TI0	RI0	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0x98

Bit7: S0MODE: Serial Port 0 Operation Mode.
This bit selects the UART0 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: MCE0: 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: RI0 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: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART 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 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software

Bit0: RI0: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 14.2. SBUF0: Serial (UART0) Port Data Buffer

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

Bits7–0: SBUF0[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 SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 is what initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Table 14.4. Timer Settings for Standard Baud Rates Using an External 18.432 MHz Oscillator

	Frequency: 18.432 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	80	SYSCLK	XX[2]	1	0xD8
	115200	0.00%	160	SYSCLK	XX[2]	1	0xB0
	57600	0.00%	320	SYSCLK	XX[2]	1	0x60
	28800	0.00%	640	SYSCLK / 4	01	0	0xB0
	14400	0.00%	1280	SYSCLK / 4	01	0	0x60
	9600	0.00%	1920	SYSCLK / 12	00	0	0xB0
	2400	0.00%	7680	SYSCLK / 48	10	0	0xB0
	1200	0.00%	15360	SYSCLK / 48	10	0	0x60
SYSCLK from Internal Osc.	230400	0.00%	80	EXTCLK / 8	11	0	0xFB
	115200	0.00%	160	EXTCLK / 8	11	0	0xF6
	57600	0.00%	320	EXTCLK / 8	11	0	0xEC
	28800	0.00%	640	EXTCLK / 8	11	0	0xD8
	14400	0.00%	1280	EXTCLK / 8	11	0	0xB0
	9600	0.00%	1920	EXTCLK / 8	11	0	0x88

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in **Section 15.1**.
2. X = Don't care

15.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

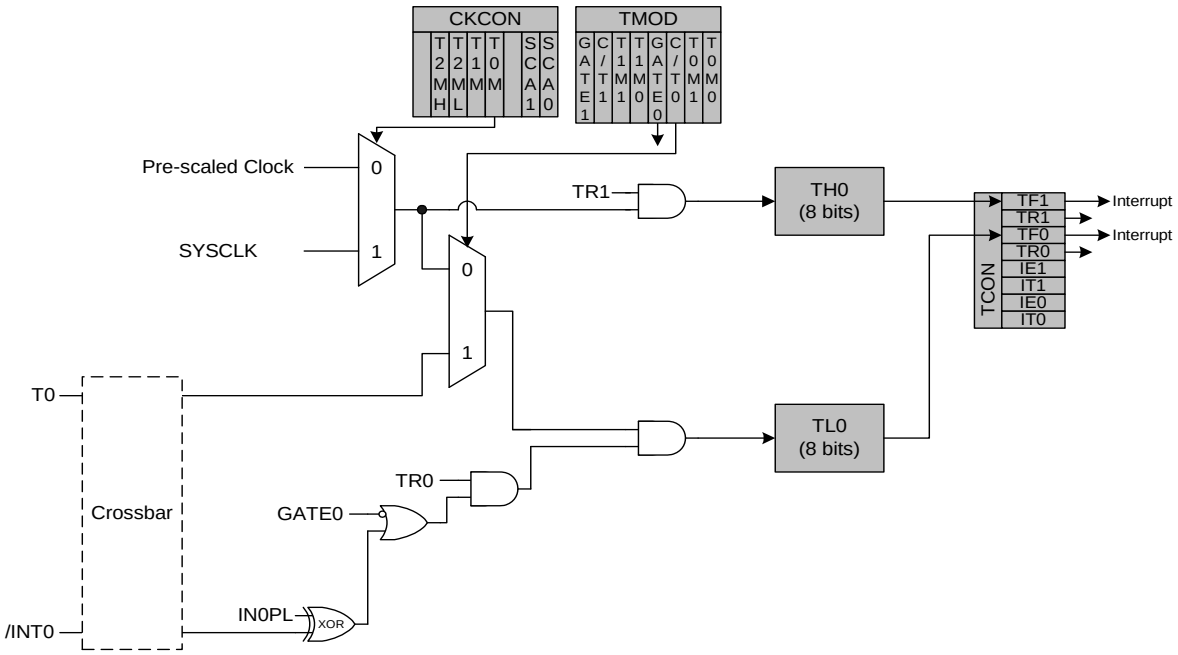

Figure 15.3. T0 Mode 3 Block Diagram

16.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-bit Pulse Width Modulator, or 16-bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 16.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1. See Figure 16.3 for details on the PCA interrupt configuration.

Table 16.2. PCA0CPM Register Settings for PCA Capture/Compare Modules

PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Operation Mode
X*	X*	1	0	0	0	0	X*	Capture triggered by positive edge on CEXn
X*	X*	0	1	0	0	0	X*	Capture triggered by negative edge on CEXn
X*	X*	1	1	0	0	0	X*	Capture triggered by transition on CEXn
X*	1	0	0	1	0	0	X*	Software Timer
X*	1	0	0	1	1	0	X*	High Speed Output
X*	1	0	0	X*	1	1	X*	Frequency Output
0	1	0	0	X*	0	1	X*	8-bit Pulse Width Modulator
1	1	0	0	X*	0	1	X*	16-bit Pulse Width Modulator

***Note:** X = Don't Care

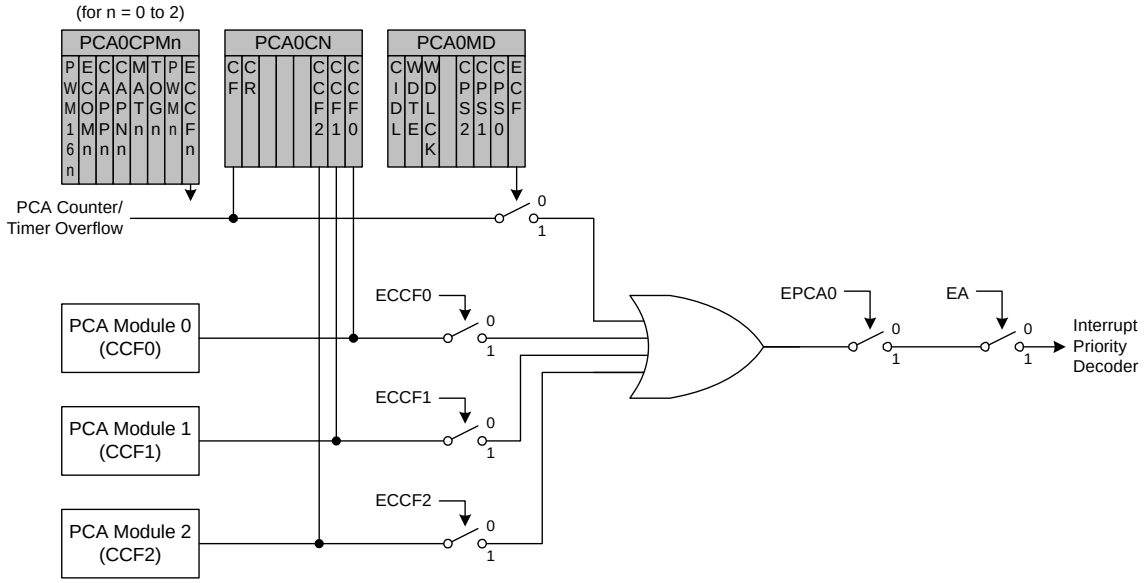

Figure 16.3. PCA Interrupt Block Diagram