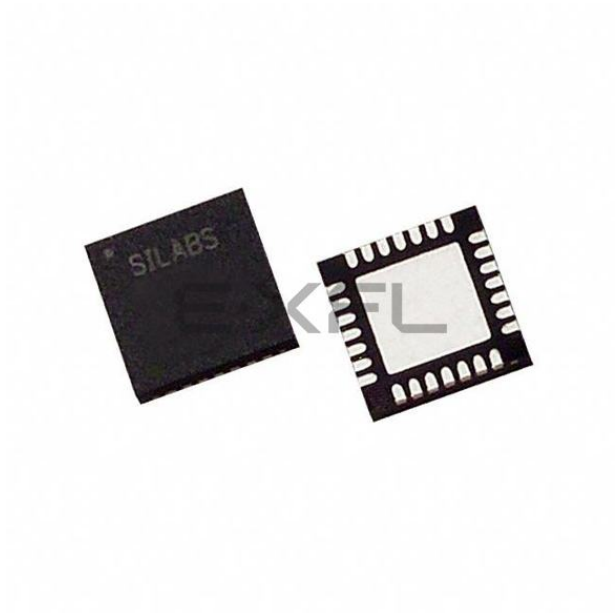

Welcome to E-XFL.COM

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	48 MIPS
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t326-gmr

6. Electrical Characteristics

6.1. Absolute Maximum Specifications

Table 6.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient temperature under bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on $\overline{RST}$ or any Port I/O Pin (except V_{PP} during programming) with respect to GND	$V_{DD} \geq 2.2$ V $V_{DD} < 2.2$ V	–0.3 –0.3	— —	5.8 V_{DD} + 3.6	V V
Voltage on V_{PP} with respect to GND during a programming operation	$V_{DD} \geq 2.4$ V	–0.3	—	7.0	V
Duration of High-voltage on V_{PP} pin (cumulative)	$V_{PP} > (V_{DD} + 3.6$ V)	—	—	10	s
Voltage on V_{DD} with respect to GND	Regulator1 in Normal Mode Regulator1 in Bypass Mode	–0.3 –0.3	— —	4.2 1.98	V V
Maximum Total current through V_{DD}, V_{IO}, REGIN, or GND		—	—	500	mA
Maximum output current sunk by $\overline{RST}$ or any Port pin		—	—	100	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

8. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 25), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 8.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 48 MIPS Peak Throughput with 48 MHz Clock
- 0 to 48 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

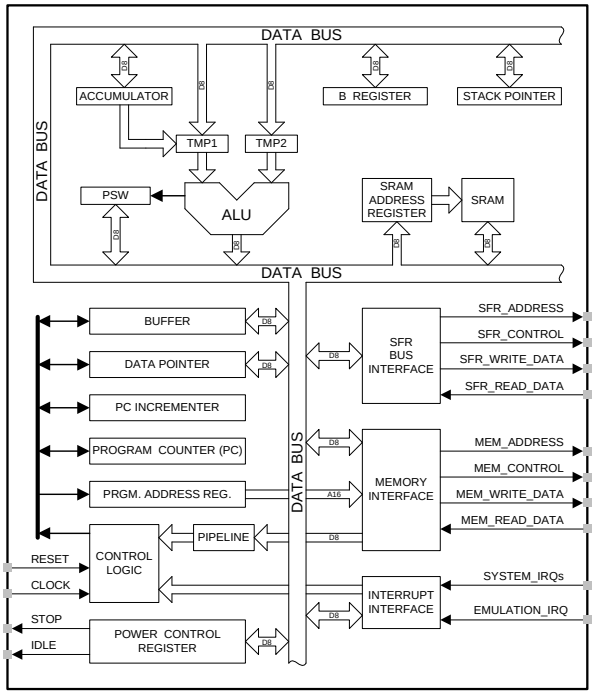

Figure 8.1. CIP-51 Block Diagram

SFR Definition 10.2. EMI0CF: External Memory Configuration

Bit	7	6	5	4	3	2	1	0
Name		USBFAE						
Type	R	R/W	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	1

SFR Address = 0x85

Bit	Name	Function
7	Unused	Unused. Read = 0b; Write = Don't Care
6	USBFAE	**USB FIFO Access Enable.** 0: USB FIFO RAM not available through MOVX instructions. 1: USB FIFO RAM available using MOVX instructions. The 256 bytes of USB RAM will be mapped in XRAM space at addresses 0x0400 to 0x04FF. **The USB clock must be active and greater than or equal to twice the SYSCLK (USBCLK > 2 x SYSCLK) to access this area with MOVX instructions.**
5:0	Unused	Unused. Read = 000011b; Write = Don't Care

SFR Definition 12.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	Reserved	Reserved	EPCA0	Reserved	Reserved	EUSB0	ESMB0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE6

Bit	Name	Function
7	ET3	**Enable Timer 3 Interrupt.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6:5	Reserved	Reserved. Must Write 00b.
4	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3:2	Reserved	Reserved. Must Write 00b.
1	EUSB0	**Enable USB (USB0) Interrupt.** This bit sets the masking of the USB0 interrupt. 0: Disable all USB0 interrupts. 1: Enable interrupt requests generated by USB0.
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

14. Power Management Modes

The C8051T622/3 and C8051T326/7 devices have three software programmable power management modes: Idle, Stop, and Suspend. Idle mode and stop mode are part of the standard 8051 architecture, while suspend mode is an enhanced power-saving mode implemented by the high-speed oscillator peripheral.

Idle mode halts the CPU while leaving the peripherals and clocks active. In stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Suspend mode is similar to stop mode in that the internal oscillator is halted, but the device can wake on events such as a Port Mismatch, Timer 3 overflow, or activity with the USB transceiver. Additionally, the CPU is not halted in suspend mode, so it can run on another oscillator, if desired. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode and suspend mode consume the least power because the majority of the device is shut down with no clocks active. SFR Definition 14.1 describes the Power Control Register (PCON) used to control the C8051T622/3 and C8051T326/7's Stop and Idle power management modes. Suspend mode is controlled by the SUSPEND bit in the OSCICN register (SFR Definition 16.3).

Although the C8051T622/3 and C8051T326/7 has Idle, Stop, and suspend modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off oscillators lowers power consumption considerably, at the expense of reduced functionality.

14.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the hardware to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

Note: If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in ‘C’:
PCON |= 0x01;                  // set IDLE bit
PCON = PCON;                   // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h                 ; set IDLE bit
MOV PCON, PCON                 ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This pro-

SFR Definition 16.3. OSCICN: Internal H-F Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY	SUSPEND				IFCN[1:0]	
Type	R/W	R	R/W	R	R	R	R/W	
Reset	1	1	0	0	0	0	0	0

SFR Address = 0xB2

Bit	Name	Function
7	IOSCEN	**Internal H-F Oscillator Enable Bit.** 0: Internal H-F Oscillator Disabled. 1: Internal H-F Oscillator Enabled.
6	IFRDY	**Internal H-F Oscillator Frequency Ready Flag.** 0: Internal H-F Oscillator is not running at programmed frequency. 1: Internal H-F Oscillator is running at programmed frequency.
5	SUSPEND	**Internal Oscillator Suspend Enable Bit.** Setting this bit to logic 1 places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occurs.
4:2	Unused	Unused. Read = 000b; Write = Don't Care
1:0	IFCN[1:0]	**Internal H-F Oscillator Frequency Divider Control Bits.** The Internal H-F Oscillator is divided by the IFCN bit setting after a divide-by-4 stage. 00: SYSCLK can be derived from Internal H-F Oscillator divided by 8 (1.5 MHz). 01: SYSCLK can be derived from Internal H-F Oscillator divided by 4 (3 MHz). 10: SYSCLK can be derived from Internal H-F Oscillator divided by 2 (6 MHz). 11: SYSCLK can be derived from Internal H-F Oscillator divided by 1 (12 MHz).

16.6. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. Figure 16.1 shows a block diagram of the four external oscillator options. The external oscillator is enabled and configured using the OSCXCN register (see SFR Definition 16.6).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2, respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pin used by the oscillator circuit; see Section “17.3. Priority Crossbar Decoder” on page 100 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “17.4. Port I/O Initialization” on page 104 for details on Port input mode selection.

The external oscillator output may be selected as the system clock or used to clock some of the digital peripherals (e.g. Timers, PCA, etc.). See the data sheet chapters for each digital peripheral for details. See Section “6. Electrical Characteristics” on page 28 for complete oscillator specifications.

16.6.1. External Crystal Mode

If a crystal or ceramic resonator is used as the external oscillator, the crystal/resonator and a 10 MΩ resistor must be wired across the XTAL1 and XTAL2 pins as shown in Figure 16.1, “Crystal Mode”. Appropriate loading capacitors should be added to XTAL1 and XTAL2, and both pins should be configured for analog I/O with the digital output drivers disabled.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are “in series” as seen by the crystal and “in parallel” with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The recommended load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

The equation for determining the load capacitance for two capacitors is

$$C_L = \frac{C_A \times C_B}{C_A + C_B} + C_S$$

Where:
C_A and C_B are the capacitors connected to the crystal leads.
C_S is the total stray capacitance of the PCB.
The stray capacitance for a typical layout where the crystal is as close as possible to the pins is 2-5 pF per pin.

If C_A and C_B are the same (C), then the equation becomes

$$C_L = \frac{C}{2} + C_S$$

For example, a tuning-fork crystal of 32 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 16.1, Option 1. With a stray capacitance of 3 pF per pin (6 pF total), the 13 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 16.2.

SFR Definition 17.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
4	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
3	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
2:0	PCA0ME[2:0]	**PCA Module I/O Enable Bits.** 000: All PCA I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100-111: Reserved.

Table 18.2. USB0 Controller Registers

USB Register Name	USB Register Address	Description	Page Number
Interrupt Registers			
IN1INT	0x02	Endpoint0 and Endpoints1-2 IN Interrupt Flags	131
OUT1INT	0x04	Endpoints1-2 OUT Interrupt Flags	132
CMINT	0x06	Common USB Interrupt Flags	133
IN1IE	0x07	Endpoint0 and Endpoints1-2 IN Interrupt Enables	134
OUT1IE	0x09	Endpoints1-2 OUT Interrupt Enables	135
CMIE	0x0B	Common USB Interrupt Enables	136
Common Registers			
FADDR	0x00	Function Address	127
POWER	0x01	Power Management	129
FRAMEL	0x0C	Frame Number Low Byte	130
FRAMEH	0x0D	Frame Number High Byte	130
INDEX	0x0E	Endpoint Index Selection	123
CLKREC	0x0F	Clock Recovery Control	124
EENABLE	0x1E	Endpoint Enable	141
FIFOn	0x20-0x22	Endpoints0-2 FIFOs	126
Indexed Registers			
E0CSR	0x11	Endpoint0 Control / Status	139
EINCSRL		Endpoint IN Control / Status Low Byte	143
EINCSRH	0x12	Endpoint IN Control / Status High Byte	144
EOUTCSRL	0x14	Endpoint OUT Control / Status Low Byte	146
EOUTCSRH	0x15	Endpoint OUT Control / Status High Byte	147
E0CNT	0x16	Number of Received Bytes in Endpoint0 FIFO	140
EOUTCNTL		Endpoint OUT Packet Count Low Byte	147
EOUTCNTH	0x17	Endpoint OUT Packet Count High Byte	148

Suspend Interrupt Service Routine (ISR) should perform application-specific configuration tasks such as disabling appropriate peripherals and/or configuring clock sources for low power modes. See Section “16.3. Programmable Internal High-Frequency (H-F) Oscillator” on page 89 for more details on internal oscillator configuration, including the Suspend mode feature of the internal oscillator.

USB0 exits Suspend mode when any of the following occur: (1) Resume signaling is detected or generated, (2) Reset signaling is detected, or (3) a device or USB reset occurs. If suspended, the internal oscillator will exit Suspend mode upon any of the above listed events.

Resume Signaling: USB0 will exit Suspend mode if Resume signaling is detected on the bus. A Resume interrupt will be generated upon detection if enabled (RESINTE = 1). Software may force a Remote Wakeup by writing 1 to the RESUME bit (POWER.2). When forcing a Remote Wakeup, software should write RESUME = 0 to end Resume signaling 10-15 ms after the Remote Wakeup is initiated (RESUME = 1).

ISO Update: When software writes 1 to the ISOUP bit (POWER.7), the ISO Update function is enabled. With ISO Update enabled, new packets written to an ISO IN endpoint will not be transmitted until a new Start-Of-Frame (SOF) is received. If the ISO IN endpoint receives an IN token before a SOF, USB0 will transmit a zero-length packet. When ISOUP = 1, ISO Update is enabled for all ISO endpoints.

USB Enable: USB0 is disabled following a Power-On-Reset (POR). USB0 is enabled by clearing the USBINH bit (POWER.4). Once written to 0, the USBINH can only be set to 1 by one of the following: (1) a Power-On-Reset (POR), or (2) an asynchronous USB0 reset generated by writing 1 to the USBRST bit (POWER.3).

Software should perform all USB0 configuration before enabling USB0. The configuration sequence should be performed as follows:

1. Select and enable the USB clock source.
2. Reset USB0 by writing USBRST= 1.
3. Configure and enable the USB Transceiver.
4. Perform any USB0 function configuration (interrupts, Suspend detect).
5. Enable USB0 by writing USBINH = 0.

USB Register Definition 18.18. E0CNT: USB0 Endpoint0 Data Count

Bit	7	6	5	4	3	2	1	0
Name		E0CNT[6:0]						
Type	R	R						
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x16

Bit	Name	Function
7	Unused	Unused. Read = 0b. Write = don't care.
6:0	E0CNT[6:0]	**Endpoint 0 Data Count.** This 7-bit number indicates the number of received data bytes in the Endpoint 0 FIFO. This number is only valid while bit OPRDY is a 1.

18.11. Configuring Endpoints1-2

Endpoints1-2 are configured and controlled through their own sets of the following control/status registers: IN registers EINCSRL and EINCSRH, and OUT registers EOUTCSRL and EOUTCSRH. Only one set of endpoint control/status registers is mapped into the USB register address space at a time, defined by the contents of the INDEX register (USB Register Definition 18.4).

Endpoints1-2 can be configured as IN, OUT, or both IN/OUT (Split Mode) as described in Section 18.5.1. The endpoint mode (Split/Normal) is selected via the SPLIT bit in register EINCSRH.

When SPLIT = 1, the corresponding endpoint FIFO is split, and both IN and OUT pipes are available.

When SPLIT = 0, the corresponding endpoint functions as either IN or OUT; the endpoint direction is selected by the DIRSEL bit in register EINCSRH.

Endpoints1-2 can be disabled individually by the corresponding bits in the ENABLE register. When an Endpoint is disabled, it will not respond to bus traffic or stall the bus. All Endpoints are enabled by default.

USB Register Definition 18.20. EINCSRL: USB0 IN Endpoint Control Low

Bit	7	6	5	4	3	2	1	0
Name		CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY
Type	R	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x11

Bit	Name	Description	Write	Read
7	Unused	Unused. Read = 0b. Write = don't care.		
6	CLRDT	**Clear Data Toggle Bit.**	Software should write 1 to this bit to reset the IN End-point data toggle to 0.	This bit always reads 0.
5	STSTL	**Sent Stall Bit.** Hardware sets this bit to 1 when a STALL handshake signal is transmitted. The FIFO is flushed, and the INPRDY bit cleared. This flag must be cleared by software.		
4	SDSTL	**Send Stall.** Software should write 1 to this bit to generate a STALL handshake in response to an IN token. Software should write 0 to this bit to terminate the STALL signal. This bit has no effect in ISO mode.		
3	FLUSH	**FIFO Flush Bit.** Writing a 1 to this bit flushes the next packet to be transmitted from the IN Endpoint FIFO. The FIFO pointer is reset and the INPRDY bit is cleared. If the FIFO contains multiple packets, software must write 1 to FLUSH for each packet. Hardware resets the FLUSH bit to 0 when the FIFO flush is complete.		
2	UNDRUN	**Data Underrun Bit.** The function of this bit depends on the IN Endpoint mode: ISO: Set when a zero-length packet is sent after an IN token is received while bit INPRDY = 0. Interrupt/Bulk: Set when a NAK is returned in response to an IN token. This bit must be cleared by software.		
1	FIFONE	**FIFO Not Empty.** 0: The IN Endpoint FIFO is empty. 1. The IN Endpoint FIFO contains one or more packets.		
0	INPRDY	**In Packet Ready.** Software should write 1 to this bit after loading a data packet into the IN Endpoint FIFO. Hardware clears INPRDY due to any of the following: 1) A data packet is transmitted. 2) Double buffering is enabled (DBIEN = 1) and there is an open FIFO packet slot. 3) If the endpoint is in Isochronous Mode (ISO = 1) and ISOUD = 1, INPRDY will read 0 until the next SOF is received. **Note:** An interrupt (if enabled) will be generated when hardware clears INPRDY as a result of a packet being transmitted.		

SFR Definition 19.4. SMB0ADM: SMBus Slave Address Mask

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

SFR Address = 0xCF

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

Table 19.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
		1	1	X	Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—

Table 19.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—

20. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “20.1. Enhanced Baud Rate Generation” on page 172). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

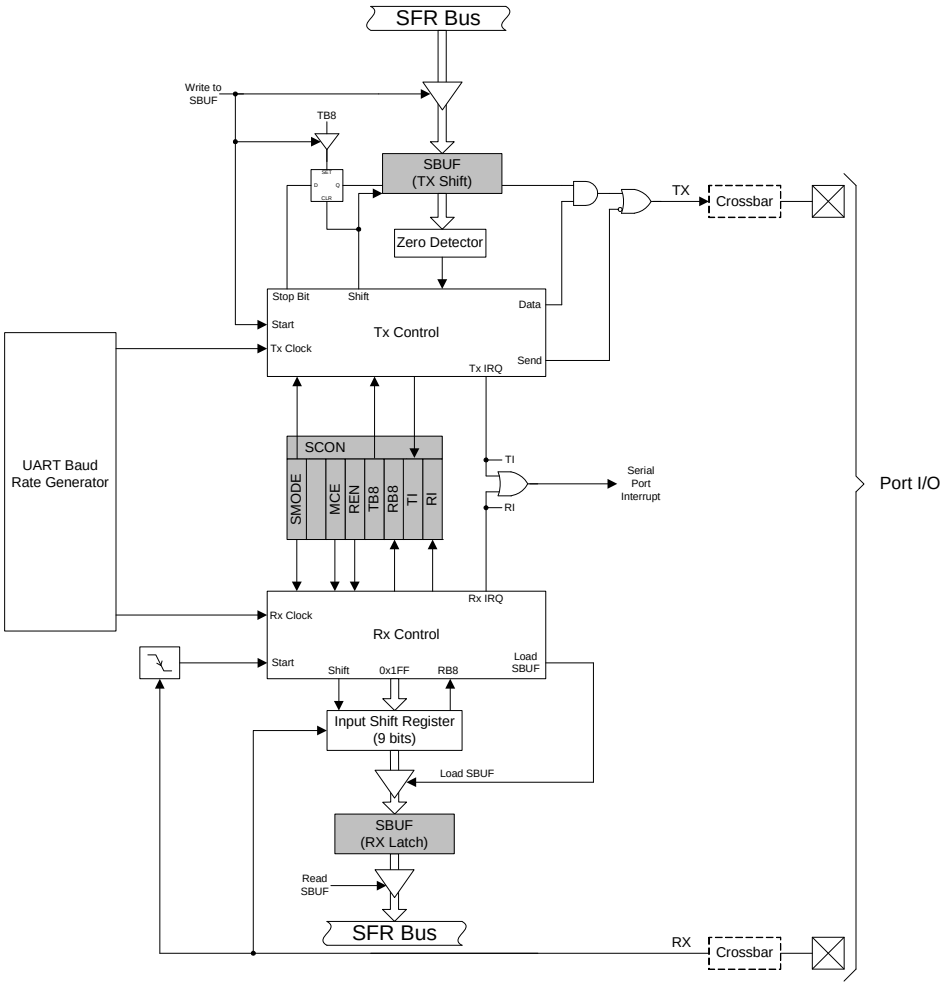

Figure 20.1. UART0 Block Diagram

SFR Definition 21.3. SBUF1: UART1 Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD3

Bit	Name	Description	Write	Read
7:0	SBUF1[7:0]	**Serial Data Buffer Bits.** This SFR is used to both send data from the UART and to read received data from the UART1 receive FIFO.	Writing a byte to SBUF1 initiates the transmission. When data is written to SBUF1, it first goes to the Transmit Holding Register, where it is held for serial transmission. When the transmit shift register is available, data is transferred into the shift register, and SBUF1 may be written again.	Reading SBUF1 retrieves data from the receive FIFO. When read, the oldest byte in the receive FIFO is returned, and removed from the FIFO. Up to three bytes may be held in the FIFO. If there are additional bytes available in the FIFO, the RI1 bit will remain at logic 1, even after being cleared by software.

SFR Definition 23.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{INT1}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{INT1}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 12.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{INT0}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{INT0}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 12.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

SFR Definition 24.7. PCA0CPLn: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: 0xFB (n = 0), 0xE9 (n = 1), 0xEB (n = 2),

Bit	Name	Function
7:0	PCA0CPn[7:0]	**PCA Capture Module Low Byte.** The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOMn bit to a 0.		

SFR Definition 24.8. PCA0CPHn: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: 0xFC (n = 0), 0xEA (n = 1), 0xEC (n = 2)

Bit	Name	Function
7:0	PCA0CPn[15:8]	**PCA Capture Module High Byte.** The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOMn bit to a 1.		