



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Last Time Buy
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	17
Program Memory Size	4KB (4K × 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	768 × 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t633-gm

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

List of Figures

1.	System Overview	
••	Figure 1.1. C8051T630/1/2/3/4/5 Block Diagram	16
3.	Pin Definitions	
•	Figure 3.1. QFN-20 Pinout Diagram (Top View)	20
4.	QFN-20 Package Specifications	
	Figure 4.1. QFN-20 Package Drawing	21
	Figure 4.2. QFN-20 Recommended PCB Land Pattern	22
5.	Electrical Characteristics	
	Figure 5.1. Normal Mode Digital Supply Current vs. Frequency (MPCE = 1)	32
	Figure 5.2. Idle Mode Digital Supply Current vs. Frequency (MPCE = 1)	32
6.	10-Bit ADC (ADC0, C8051T630/2/4 only)	
	Figure 6.1. ADC0 Functional Block Diagram	33
	Figure 6.2. 10-Bit ADC Track and Conversion Example Timing	35
	Figure 6.3. ADC0 Equivalent Input Circuits	36
	Figure 6.4. ADC Window Compare Example: Right-Justified Data	42
	Figure 6.5. ADC Window Compare Example: Left-Justified Data	42
	Figure 6.6. ADC0 Multiplexer Block Diagram	43
7 . '	Temperature Sensor (C8051T630/2/4 only)	
	Figure 7.1. Temperature Sensor Transfer Function	45
	Figure 7.2. Temperature Sensor Error with 1-Point Calibration at 0 Celsius	46
8.	10-Bit Current Mode DAC (IDA0, C8051T630/2/4 only)	
	Figure 8.1. IDA0 Functional Block Diagram	48
	Figure 8.2. IDA0 Data Word Mapping	49
9.	Voltage Reference Options	
	Figure 9.1. Voltage Reference Functional Block Diagram	53
11	. Comparator0	
	Figure 11.1. Comparator0 Functional Block Diagram	57
	Figure 11.2. Comparator Hysteresis Plot	58
	Figure 11.3. Comparator Input Multiplexer Block Diagram	61
12	. CIP-51 Microcontroller	~~
	Figure 12.1. CIP-51 Block Diagram	63
13	. Memory Organization	70
	Figure 13.1. Memory Map	72
40	Figure 13.2. Program Memory Map	13
10	. Reset Sources	05
	Figure 18.1. Reset Sources	95
10	Pigure 18.2. Power-On and VDD Monitor Reset Timing	90
19	Eigure 10.1 Occillator Options	01
20	Port Input/Output	01
20	Figure 20.1. Port I/O Functional Block Diagram	na
	Figure 20.2. Port I/O Cell Block Diagram	11
	Figure 20.3. Crossbar Priority Decoder with No Pins Skinned	14
	rigare 20.0. Crossbar i honry Decouci with No 1 ins Oripped	



SFR Definition 6.4. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0	
Nam	e AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT		AD0CM[2:0]		
Туре	R/W	R/W	R/W	R/W	R/W		R/W		
Rese	et 0	0	0	0	0	0	0	0	
SFR /	Address = 0xE	8; Bit-Addres	sable				I	II	
Bit	Name				Function				
7	AD0EN	ADC0 Enable Bit. 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.							
6	ADOTM	 ADC0 Track Mode Bit. 0: Normal Track Mode: When ADC0 is enabled, tracking is continuous unless a conversion is in progress. Conversion begins immediately on start-of-conversion event, as defined by AD0CM[2:0]. 1: Delayed Track Mode: When ADC0 is enabled, input is tracked when a conversion is not in progress. A start-of-conversion signal initiates three SAR clocks of additional tracking, and then begins the conversion. 							
5	AD0INT	ADC0 Conv 0: ADC0 has 1: ADC0 has	version Con s not comple s completed	plete Interr eted a data c a data conve	upt Flag. onversion sir ersion.	nce AD0INT	was last clea	ared.	
4	AD0BUSY	ADC0 Busy	Bit. Rea	d:		Write:			
			0: Al prog 1: Al ress	DC0 convers ress. DC0 convers	ion is not in ion is in prog	0: No Ef 1: Initiat g- AD0CM	fect. es ADC0 Co [2 : 0] = 000b	nversion if	
3	AD0WINT	ADC0 Wind	ow Compai	e Interrupt	Flag.				
		0: ADC0 Win cleared. 1: ADC0 Win	ndow Compa ndow Compa	arison Data r arison Data r	match has no match has oc	ot occurred s	since this flag	g was last	
2:0	AD0CM[2:0]	ADC0 Start	of Convers	ion Mode S	elect.				
		 ADC0 start of Conversion Mode Select. 000: ADC0 start-of-conversion source is write of 1 to AD0BUSY. 001: ADC0 start-of-conversion source is overflow of Timer 0. 010: ADC0 start-of-conversion source is overflow of Timer 2. 011: ADC0 start-of-conversion source is overflow of Timer 1. 100: ADC0 start-of-conversion source is rising edge of external CNVSTR. 101: ADC0 start-of-conversion source is overflow of Timer 3. 11x: Reserved. 							



SFR Definition 7.1. TOFFH: Temperature Offset Measurement High Byte

Bit	7	6	5	4	3	2	1	0			
Name TOFF[9:2]							L				
Туре	9	R/W									
Rese	t Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies			
SFR A	Address = 0x8	36									
Bit	Name				Function						
7:0	TOFF[9:2]	Temperatur	e Sensor O	ffset High C	rder Bits.						
		The temperature sensor offset registers represent the output of the ADC when mea- suring the temperature sensor at 0 °C, with the voltage reference set to the internal regulator. The temperature sensor offset information is left-justified. One LSB of this measurement is equivalent to one LSB of the ADC output under the measurement									

SFR Definition 7.2. TOFFL: Temperature Offset Measurement Low Byte

conditions.

Bit	7	6	5	4	3	2	1	0
Name	TOFF[1:0]							
Туре	R/W		R	R	R	R	R	R
Reset	Varies	Varies	0	0	0	0	0	0

SFR Address = 0x85

Bit	Name	Function
7:6	TOFF[1:0]	Temperature Sensor Offset Low Order Bits.
		The temperature sensor offset registers represent the output of the ADC when mea- suring the temperature sensor at 0 °C, with the voltage reference set to the internal regulator. The temperature sensor offset information is left-justified. One LSB of this measurement is equivalent to one LSB of the ADC output under the measurement conditions.
5:0	Unused	Unused. Read = 000000b; Write = Don't Care.



8.1.2. Update Output Based on Timer Overflow

Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the IDAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the IDAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the IDAC output. When the IDA0CM bits (IDA0CN.[6:4]) are set to 000, 001, 010 or 011, writes to both IDAC data registers (IDA0L and IDA0H) are held until an associated Timer overflow event (Timer 0, Timer 1, Timer 2 or Timer 3, respectively) occurs, at which time the IDA0H:IDA0L contents are copied to the IDAC input latches, allowing the IDAC output to change to the new value.

8.1.3. Update Output Based on CNVSTR Edge

The IDAC output can also be configured to update on a rising edge, falling edge, or both edges of the external CNVSTR signal. When the IDA0CM bits (IDA0CN.[6:4]) are set to 100, 101, or 110, writes to both IDAC data registers (IDA0L and IDA0H) are held until an edge occurs on the CNVSTR input pin. The particular setting of the IDA0CM bits determines whether IDAC outputs are updated on rising, falling, or both edges of CNVSTR. When a corresponding edge occurs, the IDA0H:IDA0L contents are copied to the IDAC input latches, allowing the IDAC output to change to the new value.

8.2. IDAC Output Mapping

The IDAC data registers (IDA0H and IDA0L) are left-justified, meaning that the eight MSBs of the IDAC output word are mapped to bits 7–0 of the IDA0H register, and the two LSBs of the IDAC output word are mapped to bits 7 and 6 of the IDA0L register. The data word mapping for the IDAC is shown in Figure 8.2.

	IDA0H										ID/	40L			
B9	B8	B7	B6	B5	B4	B3	B2	2 B1 B0							
Input Data Word		Output Current					Output Current				Ou	itput C	urren	:	
(IDA09–IDA00)		10	IDA0OMD[1:0] = 1x				IDA0OMD[1:0] = 01			1	IDA0OMD[1:0] = 00				
	0x000	C		0 mA				0 mA					0 m	A	
	0x00	1		1/1024 x 2 mA				1/1024 x 1 mA				1/1024 x 0.5 mA		١	
	0x200	C		512/1024 x 2 mA				512/1024 x 1 mA				512/1024 x 0.5 mA		A	
0x3FF		x3FF 1023/1024 x 2 mA				1023/	1024 >	<1 mA		1023	8/1024	x 0.5 n	nΑ		

Figure 8.2. IDA0 Data Word Mapping

The full-scale output current of the IDAC is selected using the IDA0OMD bits (IDA0CN[1:0]). By default, the IDAC is set to a full-scale output current of 2 mA. The IDA0OMD bits can also be configured to provide full-scale output currents of 1 mA or 0.5 mA, as shown in SFR Definition 8.1.



SFR Definition 10.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0		
Nam	e STOPC	F BYPASS						MPCE		
Туре	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Rese	t 0	0	0	0	0	0	0	0		
SFR Address = 0xC7										
Bit	Name				Function					
7	STOPCF	Stop Mode Co This bit config 0: Regulator is device. 1: Regulator is the device.	 Stop Mode Configuration. This bit configures the regulator's behavior when the device enters STOP mode. 0: Regulator is still active in STOP mode. Any enabled reset source will reset the device. 1: Regulator is shut down in STOP mode. Only the RST pin or power cycle can reset the device. 							
6	BYPASS	 Bypass Internal Regulator. This bit places the regulator in bypass mode, turning off the regulator, and allowing the core to run directly from the V_{DD} supply pin. 0: Normal Mode—Regulator is on. 1: Bypass Mode—Regulator is off, and the microcontroller core operates directly from the V_{DD} supply voltage. IMPORTANT: Bypass mode is for use with an external regulator as the supply voltage only. Never place the regulator in bypass mode when the V_{DD} supply voltage is greater than the specifications given in Table 5.1 on page 23. Doing so 								
5:1	Reserved	Reserved. Mu	st Write 000	00b						
0	MPCE	 Memory Power Controller Enable. This bit can help the system save power at slower system clock frequencies (about 2.0 MHz or less) by automatically shutting down the EPROM memory between clock when information is not being fetched from the EPROM memory. 0: Normal Mode—Memory power controller disabled (EPROM memory is always or 1: Low Power Mode—Memory power controller enabled (EPROM memory turns on as needed). Note: If an external clock source is used with the Memory Power Controller enabled, and th clock frequency changes from slow (<2.0 MHz) to fast (> 2.0 MHz), the EPROM power will turn on, and up to 20 clocks may be "skipped" to ensure that the EPROM power is stable before reading memory. 						s (about een clocks lways on). turns on/off d, and the OM power 1 power is		



C8051T630/1/2/3/4/5

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 11.2). Selecting a longer response time reduces the Comparator supply current.



Figure 11.2. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 11.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 11.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section "15.1. MCU Interrupt Sources and Vectors" on page 81). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CP0RIF flag is set to logic 1 upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE to a logic 1. The Comparator0 falling-edge interrupt mask is enabled by setting CP0FIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed.



C8051T630/1/2/3/4/5

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0–R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through R0 or R1.

rel - 8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00– 0x7F) or an SFR (0x80–0xFF).

#data - 8-bit constant

#data16 - 16-bit constant

bit - Direct-accessed bit in Data RAM or SFR

addr11 - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP. All mnemonics copyrighted © Intel Corporation 1980.



SFR Definition 15.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Туре	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 = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	 Enable All Interrupts. Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	 Enable Serial Peripheral Interface (SPI0) Interrupt. This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	 Enable Timer 2 Interrupt. This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	Enable UART0 Interrupt. This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	 Enable Timer 1 Interrupt. This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	Enable External Interrupt 1. This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the /INT1 input.
1	ETO	 Enable Timer 0 Interrupt. This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	 Enable External Interrupt 0. This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the INTO input.



SFR Definition 18.2. RSTSRC: Reset Source

Bit	7	6	5	4	3	2	1	0
Name		MEMERR	CORSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF
Туре	R	R	R/W	R/W	R	R/W	R/W	R
Reset	0	Varies						

SFR Address = 0xEF

Bit	Name	Description	Write	Read
7	Unused	Unused.	Don't care.	0
6	MEMERR	EPROM Error Reset Flag.	N/A	Set to 1 if EPROM read/write error caused the last reset.
5	CORSEF	Comparator0 Reset Enable and Flag.	Writing a 1 enables Comparator0 as a reset source (active-low).	Set to 1 if Comparator0 caused the last reset.
4	SWRSF	Software Reset Force and Flag.	Writing a 1 forces a sys- tem reset.	Set to 1 if last reset was caused by a write to SWRSF.
3	WDTRSF	Watchdog Timer Reset Flag.	N/A	Set to 1 if Watchdog Timer overflow caused the last reset.
2	MCDRSF	Missing Clock Detector Enable and Flag.	Writing a 1 enables the Missing Clock Detector. The MCD triggers a reset if a missing clock condition is detected.	Set to 1 if Missing Clock Detector timeout caused the last reset.
1	PORSF	Power-On/V _{DD} Monitor Reset Flag, and V _{DD} monitor Reset Enable.	Writing a 1 enables the V_{DD} monitor as a reset source. Writing 1 to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset.	Set to 1 anytime a power- on or V _{DD} monitor reset occurs. When set to 1 all other RSTSRC flags are inde- terminate.
0	PINRSF	HW Pin Reset Flag.	N/A	Set to 1 if RST pin caused the last reset.
Note:	Do not use	read-modify-write operations on this	s register	



19.2. Programmable Internal High-Frequency (H-F) Oscillator

All C8051T630/1/2/3/4/5 devices include a programmable internal high-frequency oscillator that defaults as the system clock after a system reset. The internal oscillator period caPara1n be adjusted via the OSCICL register as defined by SFR Definition 19.2.

On C8051T630/1/2/3/4/5 devices, OSCICL is factory calibrated to obtain a 24.5 MHz base frequency.

The system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset.

19.2.1. Internal Oscillator Suspend Mode

When software writes a logic 1 to SUSPEND (OSCICN.5), the internal oscillator is suspended. If the system clock is derived from the internal oscillator, the input clock to the peripheral or CIP-51 will be stopped until one of the following events occur:

- Port 0 Match Event.
- Port 1 Match Event.
- Comparator 0 enabled and output is logic 0.
- Timer3 Overflow Event.

When one of the oscillator awakening events occur, the internal oscillator, CIP-51, and affected peripherals resume normal operation, regardless of whether the event also causes an interrupt. The CPU resumes execution at the instruction following the write to SUSPEND.

SFR Definition 19.2. OSCICL: Internal H-F Oscillator Calibration

Bit	7	6	5	4	3	2	1	0
Name		OSCICL[6:0]						
Туре	R	R/W						
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xB3

Bit	Name	Function
7	Unused	Unused. Read = 0; Write = Don't Care
6:0	OSCICL[6:0]	Internal Oscillator Calibration Bits.
		These bits determine the internal oscillator period. When set to 0000000b, the H-F oscillator operates at its fastest setting. When set to 1111111b, the H-F oscillator operates at its slowest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.



20.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins can be assigned to various analog, digital, and external interrupt functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

20.2.1. Assigning Port I/O Pins to Analog Functions

Table 20.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 20.1 shows the potential mapping of Port I/O to each analog function.

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0-P1.7	AMX0P, AMX0N, PnSKIP
Comparator0 Input	P0.0–P1.7	CPT0MX, PnSKIP
Voltage Reference (VREF0)	P0.0	REF0CN, PnSKIP
Current DAC Output (IDA0)	P0.1	IDA0CN, PnSKIP
External Oscillator in RC or C Mode (EXTCLK)	P0.3	OSCXCN, PnSKIP

Table 20.1. Port I/O Assignment for Analog Functions

20.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital func-tions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 20.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 20.2. Port I/O Assignment for Digital Funct	ions
---	------

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, CP0, CP0A, SYSCLK, PCA0 (CEX0-2 and ECI), T0 or T1.	Any Port pin available for assignment by the Crossbar. This includes P0.0 - P1.7 pins which have their PnSKIP bit set to 0. Note: The Crossbar will always assign UART0 pins to P0.4 and P0.5.	XBR0, XBR1
Any pin used for GPIO	P0.0-P2.0	PnSKIP



21.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 21.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 21.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

21.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

21.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 21.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 21.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 21.5 for SMBus status decoding using the SMB0CN register.



Bit	Set by Hardware When:	Cleared by Hardware When:
MASTED	 A START is generated. 	 A STOP is generated.
WASTER		 Arbitration is lost.
	 START is generated. 	A START is detected.
	 SMB0DAT is written before the start of an 	 Arbitration is lost.
TAMODE	SMBus frame.	 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.
	A STOP is detected while addressed as a	A pending STOP is generated.
STO	slave.	
	 Arbitration is lost due to a detected STOP. 	
	A byte has been received and an ACK	After each ACK cycle.
ACKRQ	hardware ACK is not enabled)	
	A repeated START is detected as a	Each time SL is cleared
	MASTER when STA is low (unwanted repeated START).	
ARBLOST	 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). 	
ACK	The incoming ACK value is low	The incoming ACK value is high
	(ACKNOWLEDGE).	(NOT ACKNOWLEDGE).
	A START has been generated.	Must be cleared by software.
	Lost arbitration.	
SI	 A byte has been transmitted and an ACK/NACK received. 	
5	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. 	

Table 21.3. Sources for Hardware Changes to SMB0CN

21.4.3. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 21.4.2.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register (SFR Definition 21.3) and the SMBus Slave Address Mask register (SFR Definition 21.4). A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in bit positions of the slave address mask SLVM[6:0] enable a comparison between the received slave address and the hardware's slave address SLV[6:0] for those bits. A 0 in a bit



of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address. Additionally, if the GC bit in register SMB0ADR is set to 1, hardware will recognize the General Call Address (0x00). Table 21.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

Table 21.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV[6:0]	Slave Address Mask SLVM[6:0]	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

SFR Definition 21.3. SMB0ADR: SMBus Slave Address

Bit	7	6	5	4	3	2	1	0
Name				SLV[6:0]				GC
Туре	R/W				R/W			
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD7

Bit	Name	Function			
7:1	SLV[6:0]	SMBus Hardware Slave Address.			
		Defines the SMBus Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.			
0	GC	General Call Address Enable.			
		 When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized. 			



21.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. 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.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.

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 for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0-DAT is written while an active Master Receiver. Figure 21.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.



Figure 21.6. Typical Master Read Sequence



22.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 22.2), which is not useraccessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.



Figure 22.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section "24.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload" on page 173). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 22.1-A and Equation 22.1-B.

A) UartBaudRate =
$$\frac{1}{2} \times T1_Overflow_Rate$$

B) T1_Overflow_Rate = $\frac{T1_{CLK}}{256 - TH1}$

Equation 22.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and T1H is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section "24. Timers" on page 169. A quick reference for typical baud rates and system clock frequencies is given in Table 22.1 through Table 22.2. The internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.



24. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 3 offers the ability to be clocked from the external oscillator while the device is in Suspend mode, and can be used as a wake-up source. This allows for implementation of a very low-power system, including RTC capability.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:	
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload	
16-bit counter/timer			
8-bit counter/timer with			
auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload	
Two 8-bit counter/timers			
(Timer 0 only)			

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 24.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.



C8051T630/1/2/3/4/5



Figure 24.1. T0 Mode 0 Block Diagram

24.1.2. Mode 1: 16-bit Counter/Timer

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



24.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 may be clocked by the system clock, the system clock divided by 12, the external oscillator source divided by 8, or the internal low-frequency oscillator divided by 8. The external clock mode is ideal for realtime clock (RTC) functionality, where the internal high-frequency oscillator drives the system clock while Timer 3 is clocked by an external oscillator source. Note that the external oscillator source divided by 8 and the LFO source divided by 8 are synchronized with the system clock when in all operating modes except suspend. When the internal oscillator is placed in suspend mode, The external clock/8 signal or the LFO/8 output can directly drive the timer. This allows the use of an external clock or the LFO to wake up the device from suspend mode. The timer will continue to run in suspend mode and count up. When the timer overflow occurs, the device will wake from suspend mode, and begin executing code again. The timer value may be set prior to entering suspend, to overflow in the desired amount of time (number of clocks) to wake the device. If a wake-up source other than the timer wakes the device from suspend mode, it may take up to three timer clocks before the timer registers can be read or written. During this time, the STSYNC bit in register OSCICN will be set to 1, to indicate that it is not safe to read or write the timer registers.

Important Note: In internal LFO/8 mode, the divider for the internal LFO must be set to 1 for proper functionality. The timer will not operate if the LFO divider is not set to 1.

24.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 24.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.



Figure 24.7. Timer 3 16-Bit Mode Block Diagram



25. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and three 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled. The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflows, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8 to 11-Bit PWM, or 16-Bit PWM (each mode is described in Section "25.3. Capture/Compare Modules" on page 194). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 25.1

Important Note: The PCA Module 2 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. Access to certain PCA registers is restricted while WDT mode is enabled. See Section 25.4 for details.



Figure 25.1. PCA Block Diagram

