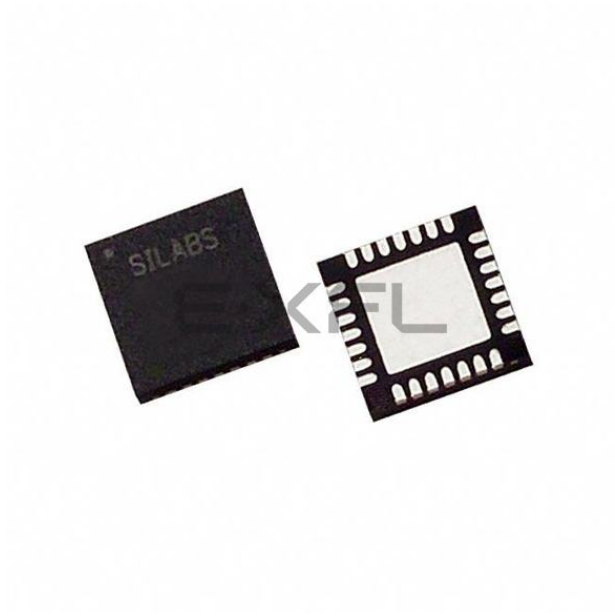

Welcome to E-XFL.COM

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	21
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/c8051t323-gm

2. Ordering Information

Table 2.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	EPROM Code Memory (Bytes)	RAM (Bytes)	Calibrated Internal 48 MHz Oscillator	Internal 80 kHz Oscillator	USB with 1k Endpoint RAM	Supply Voltage Regulator	SMBus/I2C	Enhanced SPI	UARTs	Timers (16-bit)	Programmable Counter Array	Digital Port I/Os	10-bit 500ksps ADC	Internal Voltage Reference	Temperature Sensor	Analog Comparator	Lead-free (RoHS Compliant)	Package
C8051T620-GM	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	24	Y	Y	Y	2	Y	QFN32
C8051T621-GM	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	24	—	—	—	2	Y	QFN32
C8051T626-B-GM[5]	48	64k[1]	3328	Y	Y	Y	Y	Y	Y	2	4	Y	24	Y	Y	Y	2	Y	QFN32
C8051T627-B-GM[5]	48	32k	3328	Y	Y	Y	Y	Y	Y	2	4	Y	24	Y	Y	Y	2	Y	QFN32
C8051T320-GQ[2]	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	25	Y	Y	Y	2	Y	LQFP32
C8051T321-GM[3]	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	21	Y	Y	Y	2	Y	QFN28
C8051T322-GQ[2]	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	25	—	—	—	2	Y	LQFP32
C8051T323-GM[3]	48	16k[1]	1280	Y	Y	Y	Y	Y	Y	2	4	Y	21	—	—	—	2	Y	QFN28

Notes:

1. 512 Bytes Reserved for Factory use.
2. Pin compatible with the C8051F320-GQ.
3. Pin compatible with the C8051F321-GM.
4. Lead plating material is 100% Matte Tin (Sn).
5. These ordering part numbers use a newer format that includes the silicon revision. For example, C8051T626-B-GM indicates silicon revision “B”.

Table 3.1. Pin Definitions for the C8051T620/1/6/7 & C8051T320/1/2/3(Continued)

Name	Pin Number			Type	Description
	‘T62x	‘T320/2	‘T321/3		
P0.3/	31	31	27	D I/O or A In	Port 0.3.
XTAL2				A Out	External Clock Output. This pin is the excitation driver for an external crystal or resonator.
				D In	External Clock Input. This pin is the external clock input in external CMOS clock mode.
				A In	External Clock Input. This pin is the external clock input in capacitor or RC oscillator configurations. See Oscillator Section for complete details.
P0.4	30	30	26	D I/O or A In	Port 0.4.
P0.5	29	29	25	D I/O or A In	Port 0.5.
P0.6/	28	28	24	D I/O or A In	Port 0.6.
CNVSTR				D In	ADC0 External Convert Start or IDA0 Update Source Input.
P0.7/	27	27	23	D I/O or A In	Port 0.7
VREF				A I/O	ADC Voltage Reference
P1.0	26	26	22	D I/O or A In	Port 1.0.
P1.1	25	25	21	D I/O or A In	Port 1.1.
P1.2	24	24	20	D I/O or A In	Port 1.2.
P1.3	23	23	19	D I/O or A In	Port 1.3.
P1.4	22	22	18	D I/O or A In	Port 1.4.

12. 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 30), 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 12.1 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.

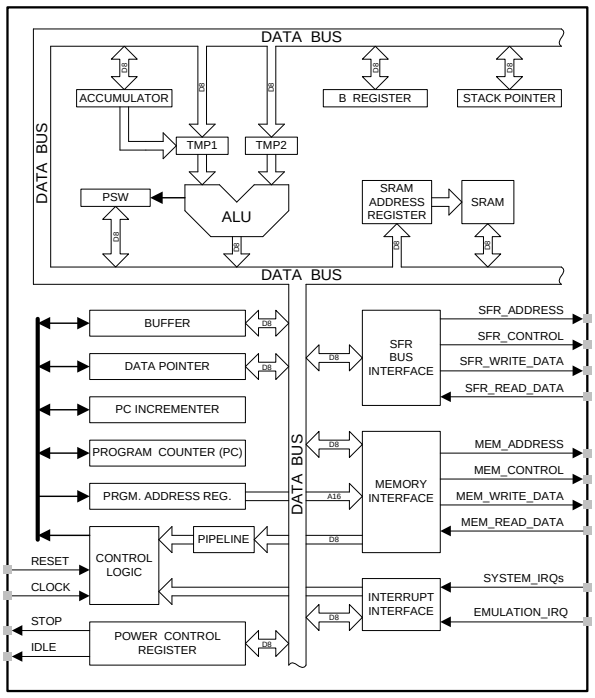

Figure 12.1. CIP-51 Block Diagram

15. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051T620/1/6/7 & C8051T320/1/2/3 device family is shown in Figure 15.1

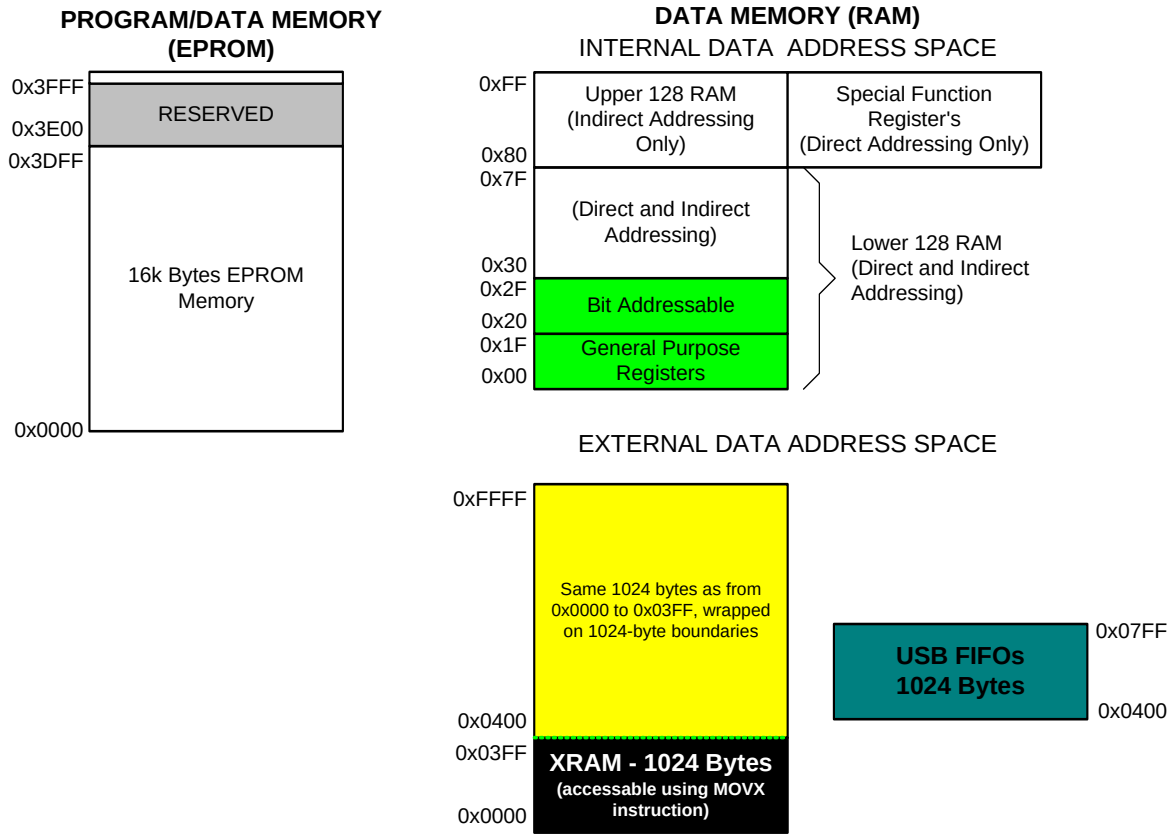

Figure 15.1. C8051T620/1 and C8051T320/1/2/3 Memory Map

15.2. Data Memory

The C8051T620/1 and C8051T320/1/2/3 device family includes 1280 bytes of RAM data memory, while the C8051T626/6 devices include 3328 bytes. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The remaining 1024 or 3072 bytes of this memory is on-chip "external" memory. The data memory map is shown in Figure 15.1 and Figure 15.2 for reference.

15.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 15.1 illustrates the data memory organization of the C8051T620/1/6/7 & C8051T320/1/2/3.

15.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 12.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

15.2.1.2. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

15.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

17.3. INT0 and INT1 External Interrupt Sources

The INT0 and INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL (INT0 Polarity) and IN1PL (INT1 Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “28.1. Timer 0 and Timer 1” on page 248) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	INT0 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	INT1 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

INT0 and INT1 are assigned to Port pins as defined in the IT01CF register (see SFR Definition 17.7). Note that INT0 and INT0 Port pin assignments are independent of any Crossbar assignments. INT0 and INT1 will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to INT0 and/or INT1, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register PnSKIP (see Section “22.3. Priority Crossbar Decoder” on page 142 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the INT0 and INT1 external interrupts, respectively. If an INT0 or INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The V_{DD} monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by code and a software reset is performed, the V_{DD} monitor will still be disabled after the reset.

I**mportant Note:** If the V_{DD} monitor is being turned on from a disabled state, it should be enabled before it is selected as a reset source. Selecting the V_{DD} monitor as a reset source before it is enabled and stabilized may cause a system reset. In some applications, this reset may be undesirable. If this is not desirable in the application, a delay should be introduced between enabling the monitor and selecting it as a reset source. The procedure for enabling the V_{DD} monitor and configuring it as a reset source from a disabled state is shown below:

1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = 1).
2. If necessary, wait for the V_{DD} monitor to stabilize (see Table 7.4 for the V_{DD} Monitor turn-on time).
3. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 20.2 for V_{DD} monitor timing; note that the power-on-reset delay is not incurred after a V_{DD} monitor reset. See Table 7.4 for complete electrical characteristics of the V_{DD} monitor.

21.5. Programmable Internal Low-Frequency (L-F) Oscillator

All C8051T620/1/6/7 & C8051T320/1/2/3 devices include a programmable low-frequency internal oscillator, which is calibrated to a nominal frequency of 80 kHz. The low-frequency oscillator circuit includes a divider that can be changed to divide the clock by 1, 2, 4, or 8, using the OSCLD bits in the OSCLCN register (see SFR Definition 21.5). Additionally, the OSCLF[3:0] bits can be used to adjust the oscillator's output frequency.

21.5.1. Calibrating the Internal L-F Oscillator

Timers 2 and 3 include capture functions that can be used to capture the oscillator frequency, when running from a known time base. When either Timer 2 or Timer 3 is configured for L-F Oscillator Capture Mode, a falling edge (Timer 2) or rising edge (Timer 3) of the low-frequency oscillator's output will cause a capture event on the corresponding timer. As a capture event occurs, the current timer value (TMRnH:TMRnL) is copied into the timer reload registers (TMRnRLH:TMRnRLL). By recording the difference between two successive timer capture values, the low-frequency oscillator's period can be calculated. The OSCLF bits can then be adjusted to produce the desired oscillator frequency.

SFR Definition 21.5. OSCLCN: Internal L-F Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	OSCLEN	OSCLRDY	OSCLF[3:0]				OSCLD[1:0]	
Type	R/W	R	R.W				R/W	
Reset	0	0	Varies	Varies	Varies	Varies	0	0

SFR Address = 0x86

Bit	Name	Function
7	OSCLEN	**Internal L-F Oscillator Enable.** 0: Internal L-F Oscillator Disabled. 1: Internal L-F Oscillator Enabled.
6	OSCLRDY	**Internal L-F Oscillator Ready.** 0: Internal L-F Oscillator frequency not stabilized. 1: Internal L-F Oscillator frequency stabilized. **Note:** OSCLRDY is only set back to 0 in the event of a device reset or a change to the OSCLD[1:0] bits.
5:2	OSCLF[3:0]	**Internal L-F Oscillator Frequency Control Bits.** Fine-tune control bits for the Internal L-F oscillator frequency. When set to 0000b, the L-F oscillator operates at its fastest setting. When set to 1111b, the L-F oscillator operates at its slowest setting.
1:0	OSCLD[1:0]	**Internal L-F Oscillator Divider Select.** 00: Divide by 8 selected. 01: Divide by 4 selected. 10: Divide by 2 selected. 11: Divide by 1 selected.

SFR Definition 22.4. P0MASK: Port 0 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P0MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAE

Bit	Name	Function
7:0	P0MASK[7:0]	**Port 0 Mask Value.** Selects P0 pins to be compared to the corresponding bits in P0MAT. 0: P0.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P0.n pin logic value is compared to P0MAT.n.

SFR Definition 22.5. P0MAT: Port 0 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P0MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x84

Bit	Name	Function
7:0	P0MAT[7:0]	**Port 0 Match Value.** Match comparison value used on Port 0 for bits in P0MASK which are set to 1. 0: P0.n pin logic value is compared with logic LOW. 1: P0.n pin logic value is compared with logic HIGH.

SFR Definition 22.6. P1MASK: Port 1 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P1MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBA

Bit	Name	Function
7:0	P1MASK[7:0]	**Port 1 Mask Value.** Selects P1 pins to be compared to the corresponding bits in P1MAT. 0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P1.n pin logic value is compared to P1MAT.n.

SFR Definition 22.7. P1MAT: Port 1 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P1MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB6

Bit	Name	Function
7:0	P1MAT[7:0]	**Port 1 Match Value.** Match comparison value used on Port 1 for bits in P1MASK which are set to 1. 0: P1.n pin logic value is compared with logic LOW. 1: P1.n pin logic value is compared with logic HIGH.

USB Register Definition 23.15. OUT1IE: USB0 OUT Endpoint Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name					OUT3E	OUT2E	OUT1E	
Type	R	R	R	R	R/W	R/W	R/W	R
Reset	0	0	0	0	1	1	1	0

USB Register Address = 0x09

Bit	Name	Function
7:4	Unused	Read = 0000b. Write = don't care.
3	OUT3E	**OUT Endpoint 3 Interrupt Enable.** 0: OUT Endpoint 3 interrupt disabled. 1: OUT Endpoint 3 interrupt enabled.
2	OUT2E	**OUT Endpoint 2 Interrupt Enable.** 0: OUT Endpoint 2 interrupt disabled. 1: OUT Endpoint 2 interrupt enabled.
1	OUT1E	**OUT Endpoint 1 Interrupt Enable.** 0: OUT Endpoint 1 interrupt disabled. 1: OUT Endpoint 1 interrupt enabled.
0	Unused	Read = 0b. Write = don't care.

USB Register Definition 23.17. E0CSR: USB0 Endpoint0 Control

Bit	7	6	5	4	3	2	1	0
Name	SSUEND	SOPRDY	SDSTL	SUEND	DATAEND	STSTL	INPRDY	OPRDY
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x11

Bit	Name	Description	Write	Read
7	SSUEND	**Serviced Setup End Bit.**	Software should set this bit to 1 after servicing a Setup End (bit SUEND) event. Hardware clears the SUEND bit when software writes 1 to SSUEND.	This bit always reads 0.
6	SOPRDY	**Serviced OPRDY Bit.**	Software should write 1 to this bit after servicing a received Endpoint0 packet. The OPRDY bit will be cleared by a write of 1 to SOPRDY.	This bit always reads 0.
5	SDSTL	**Send Stall Bit.** Software can write 1 to this bit to terminate the current transfer (due to an error condition, unexpected transfer request, etc.). Hardware will clear this bit to 0 when the STALL handshake is transmitted.		
4	SUEND	**Setup End Bit.** Hardware sets this read-only bit to 1 when a control transaction ends before software has written 1 to the DATAEND bit. Hardware clears this bit when software writes 1 to SSUEND.		
3	DATAEND	**Data End Bit.** Software should write 1 to this bit: 1) When writing 1 to INPRDY for the last outgoing data packet. 2) When writing 1 to INPRDY for a zero-length data packet. 3) When writing 1 to SOPRDY after servicing the last incoming data packet. This bit is automatically cleared by hardware.		
2	STSTL	**Sent Stall Bit.** Hardware sets this bit to 1 after transmitting a STALL handshake signal. This flag must be cleared by software.		
1	INPRDY	**IN Packet Ready Bit.** Software should write 1 to this bit after loading a data packet into the Endpoint0 FIFO for transmit. Hardware clears this bit and generates an interrupt under either of the following conditions: 1) The packet is transmitted. 2) The packet is overwritten by an incoming SETUP packet. 3) The packet is overwritten by an incoming OUT packet.		
0	OPRDY	**OUT Packet Ready Bit.** Hardware sets this read-only bit and generates an interrupt when a data packet has been received. This bit is cleared only when software writes 1 to the SOPRDY bit.		

ates a STALL condition, an interrupt will be generated and the STSTL bit (EINCSRL.5) set to 1. The STSTL bit must be reset to 0 by firmware.

Hardware will automatically reset INPRDY to 0 when a packet slot is open in the endpoint FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for firmware to load two packets into the IN FIFO at a time. In this case, hardware will reset INPRDY to 0 immediately after firmware loads the first packet into the FIFO and sets INPRDY to 1. An interrupt will not be generated in this case; an interrupt will only be generated when a data packet is transmitted.

When firmware writes 1 to the FCDT bit (EINCSRH.3), the data toggle for each IN packet will be toggled continuously, regardless of the handshake received from the host. This feature is typically used by Interrupt endpoints functioning as rate feedback communication for Isochronous endpoints. When FCDT = 0, the data toggle bit will only be toggled when an ACK is sent from the host in response to an IN packet.

23.12.2. Endpoints1-3 IN Isochronous Mode

When the ISO bit (EINCSRH.6) is set to 1, the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO IN mode, the host will send one IN token (data request) per frame; the location of data within each frame may vary. Because of this, it is recommended that double buffering be enabled for ISO IN endpoints.

Hardware will automatically reset INPRDY (EINCSRL.0) to 0 when a packet slot is open in the endpoint FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for firmware to load two packets into the IN FIFO at a time. In this case, hardware will reset INPRDY to 0 immediately after firmware loads the first packet into the FIFO and sets INPRDY to 1. An interrupt will not be generated in this case; an interrupt will only be generated when a data packet is transmitted.

If there is not a data packet ready in the endpoint FIFO when USB0 receives an IN token from the host, USB0 will transmit a zero-length data packet and set the UNDRUN bit (EINCSRL.2) to 1.

The ISO Update feature (see Section 23.7) can be useful in starting a double buffered ISO IN endpoint. If the host has already set up the ISO IN pipe (has begun transmitting IN tokens) when firmware writes the first data packet to the endpoint FIFO, the next IN token may arrive and the first data packet sent before firmware has written the second (double buffered) data packet to the FIFO. The ISO Update feature ensures that any data packet written to the endpoint FIFO will not be transmitted during the current frame; the packet will only be sent after a SOF signal has been received.

Table 24.5. 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	—
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	—
						Reschedule failed transfer.	1	0	0	1110

SFR Definition 26.2. SMOD1: UART1 Mode

Bit	7	6	5	4	3	2	1	0
Name	MCE1	S1PT[1:0]		PE1	S1DL[1:0]		XBE1	SBL1
Type	R/W	R/W		R/W	R/W		R/W	R/W
Reset	0	0	0	0	1	1	0	0

SFR Address = 0xE5

Bit	Name	Function
7	MCE1	**Multiprocessor Communication Enable.** 0: RI will be activated if stop bit(s) are 1. 1: RI will be activated if stop bit(s) and extra bit are 1 (extra bit must be enabled using XBE1). Note: This function is not available when hardware parity is enabled.
6:5	S1PT[1:0]	**Parity Type Bits.** 00: Odd 01: Even 10: Mark 11: Space
4	PE1	**Parity Enable.** This bit activates hardware parity generation and checking. The parity type is selected by bits S1PT1-0 when parity is enabled. 0: Hardware parity is disabled. 1: Hardware parity is enabled.
3:2	S1DL[1:0]	**Data Length.** 00: 5-bit data 01: 6-bit data 10: 7-bit data 11: 8-bit data
1	XBE1	**Extra Bit Enable.** When enabled, the value of TBX1 will be appended to the data field. 0: Extra Bit Disabled. 1: Extra Bit Enabled.
0	SBL1	**Stop Bit Length.** 0: Short - Stop bit is active for one bit time. 1: Long - Stop bit is active for two bit times (data length = 6, 7, or 8 bits), or 1.5 bit times (data length = 5 bits).

28.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

28.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 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 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 28.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

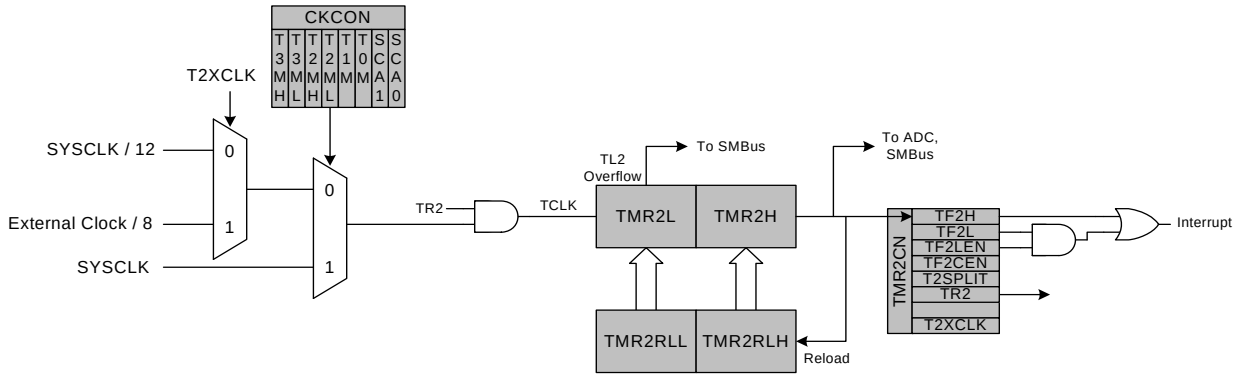

Figure 28.4. Timer 2 16-Bit Mode Block Diagram

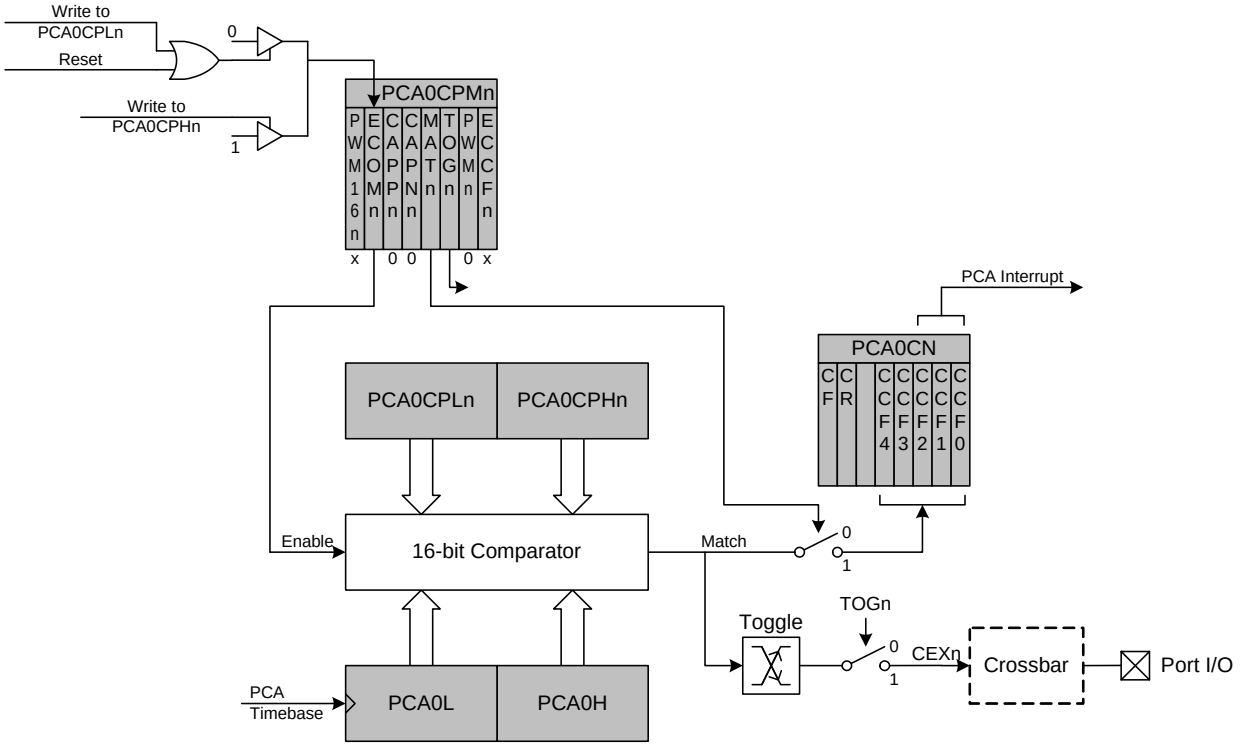

Figure 29.6. PCA High-Speed Output Mode Diagram

29.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 29.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 29.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

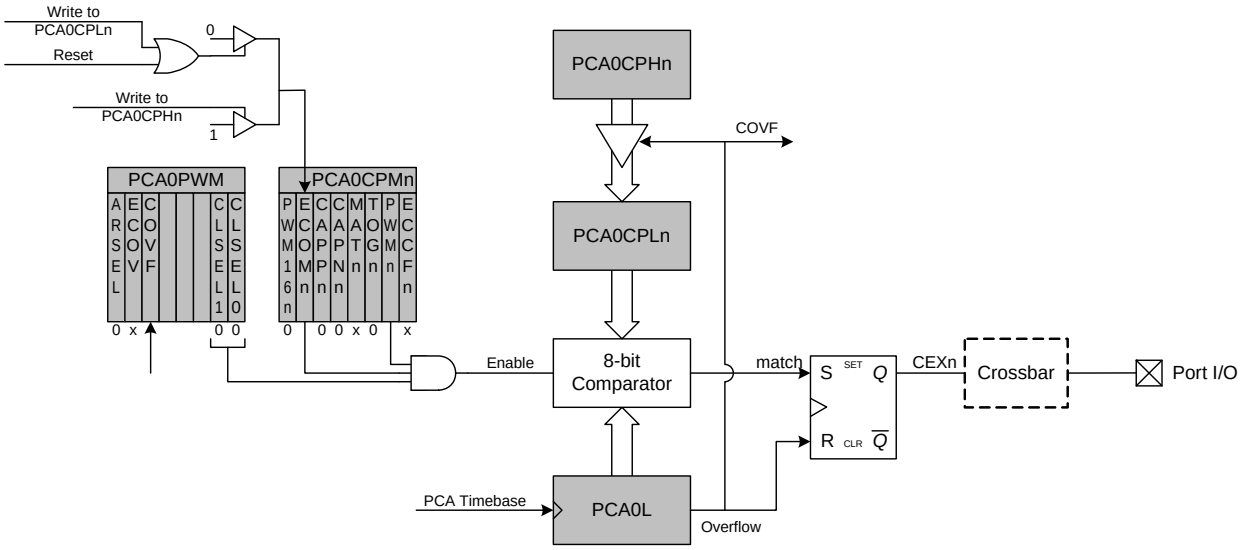

Figure 29.8. PCA 8-Bit PWM Mode Diagram

29.3.5.2. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an "Auto-Reload" Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module's capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 29.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module's auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 29.2, where N is the number of bits in the PWM cycle.

Important Note About PCA0CPHn and PCA0CPLn Registers: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(2^N - PCA0CPn)}{2^N}$$

Equation 29.3. 9, 10, and 11-Bit PWM Duty Cycle

SFR Definition 29.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF				CLSEL[1:0]	
Type	R/W	R/W	R/W	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9, 10, and 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Unused	Read = 000b; Write = Don't care.
1:0	CLSEL[1:0]	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle, between 8, 9, 10, or 11 bits. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.