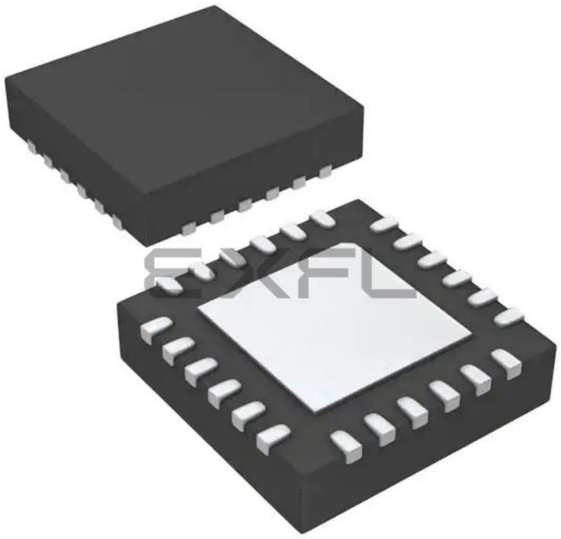

What is "Embedded - Microcontrollers"?

"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	16
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	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t622-gm

10. 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 C8051T622/3 and C8051T326/7 device family is shown in Figure 10.1

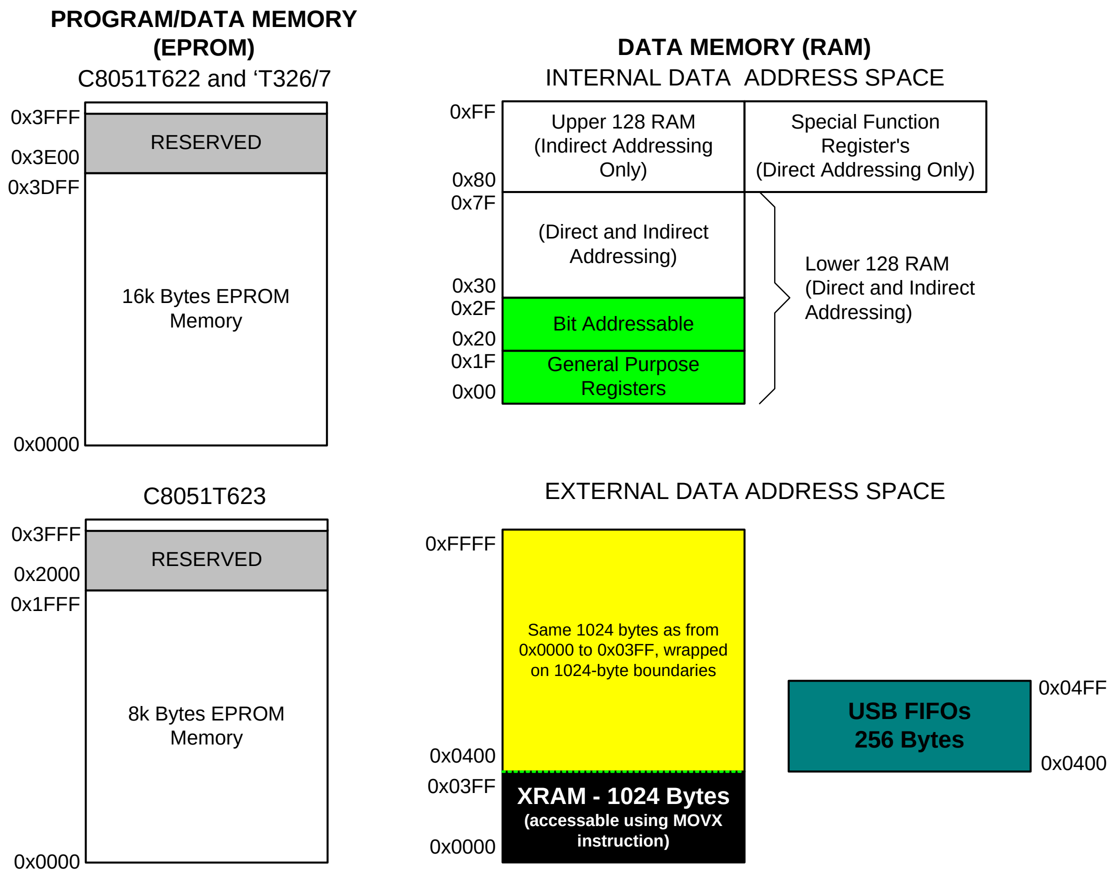

Figure 10.1. Memory Map

10.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051T622/3 and C8051T326/7 implements 16384 or 8192 bytes of this program memory space as in-system byte-programmable EPROM organized in a contiguous block from addresses 0x0000 to 0x3FFF or 0x0000 to 0x1FFF.

Note: 512 bytes (0x3E00 – 0x3FFF) of this memory are reserved for factory use and are not available for user program storage. C2 Register Definition 10.2 shows the program memory maps for C8051T622/3 and C8051T326/7 devices.

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 12.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	Reserved	Reserved	PPCA0	Reserved	Reserved	PUSB0	PSMB0
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 = 0xF6

Bit	Name	Function
7	PT3	**Timer 3 Interrupt Priority Control.** This bit sets the priority of the Timer 3 interrupt. 0: Timer 3 interrupts set to low priority level. 1: Timer 3 interrupts set to high priority level.
6:5	Reserved	Reserved. Must Write 00b.
4	PPCA0	**Programmable Counter Array (PCA0) Interrupt Priority Control.** This bit sets the priority of the PCA0 interrupt. 0: PCA0 interrupt set to low priority level. 1: PCA0 interrupt set to high priority level.
3:2	Reserved	Reserved. Must Write 00b.**.**
1	PUSB0	**USB (USB0) Interrupt Priority Control.** This bit sets the priority of the USB0 interrupt. 0: USB0 interrupt set to low priority level. 1: USB0 interrupt set to high priority level.
0	PSMB0	**SMBus (SMB0) Interrupt Priority Control.** This bit sets the priority of the SMB0 interrupt. 0: SMB0 interrupt set to low priority level. 1: SMB0 interrupt set to high priority level.

SFR Definition 13.3. IAPCN: In-Application Programming Control

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

SFR Address = 0xF5

Bit	Name	Function
7	IAPEN	**In-Application Programming Enable.** 0: In-Application Programming is disabled. 1: In-Application Programming is enabled.
6	IAPHWD	**In-Application Programming Hardware Disable.** This bit disables the In-Application Programming hardware so the V_{PP} programming pin can be used as a normal GPIO pin. **Note:** This bit should not be set less than 1 µs after the last EPROM write. 0: In-Application Programming discharge hardware enabled. 1: In-Application Programming discharge hardware disabled.
5:0	Unused	Unused. Read = 000000b. Write = don’t care.

15. Reset Sources

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

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

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

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

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source. Program execution begins at location 0x0000.

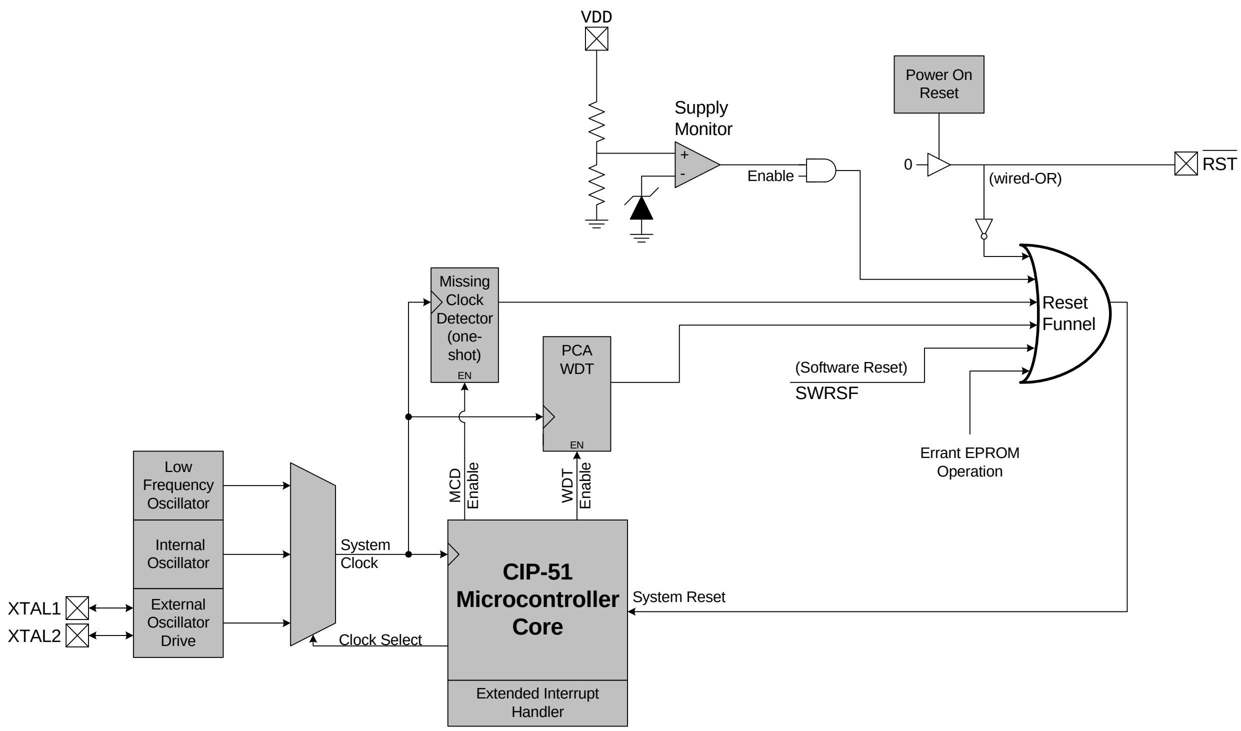

Figure 15.1. Reset Sources

15.6. EPROM Error Reset

If an EPROM program read or write targets an illegal address, a system reset is generated. This may occur due to any of the following:

- Programming hardware attempts to write or read an EPROM location which is above the user code space address limit.
- An EPROM read from firmware is attempted above user code space. This occurs when a MOVC operation is attempted above the user code space address limit.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above the user code space address limit.

The MEMERR bit (RSTSRC.6) is set following an EPROM error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

15.7. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

15.8. USB Reset

Writing 1 to the USBRSF bit in register RSTSRC selects USB0 as a reset source. With USB0 selected as a reset source, a system reset will be generated when either of the following occur:

1. RESET signaling is detected on the USB network. The USB Function Controller (USB0) must be enabled for RESET signaling to be detected. See Section “18. Universal Serial Bus Controller (USB0)” on page 116 for information on the USB Function Controller.
2. A falling or rising voltage on the VBUS pin matches the edge polarity selected by the VBPOL bit in register REG01CN. See Section “7. Voltage Regulators (REG0 and REG1)” on page 35 for details on the VBUS detection circuit.

The USBRSF bit will read 1 following a USB reset. The state of the /RST pin is unaffected by this reset.

16.4. Clock Multiplier

The C8051T622/3 and C8051T326/7 device includes a 48 MHz high-frequency oscillator instead of a 12 MHz oscillator and a 4x Clock Multiplier, so the USB0 module can be run directly from the internal high-frequency oscillator. For compatibility with the Flash development platform, however, the CLKMUL register (SFR Definition 16.4) behaves as if the Clock Multiplier is present.

SFR Definition 16.4. CLKMUL: Clock Multiplier Control

Bit	7	6	5	4	3	2	1	0
Name	MULEN	MULINIT	MULRDY				MULSEL[1:0]	
Type	R	R	R	R	R	R	R	
Reset	1	1	1	0	0	0	0	0

SFR Address = 0xB9

Bit	Name	Description	Write	Read
7	MULEN	**Clock Multiplier Enable Bit.** 0: Clock Multiplier disabled. 1: Clock Multiplier enabled. This bit always reads 1.		
6	MULINIT	**Clock Multiplier Initialize Bit.**	This bit should be a 0 when the Clock Multiplier is enabled. Once enabled, writing a 1 to this bit will initialize the Clock Multiplier.	The MULRDY bit reads 1 when the Clock Multiplier is stabilized. This bit always reads 1.
5	MULRDY	**Clock Multiplier Ready Bit.** 0: Clock Multiplier not ready. 1: Clock Multiplier ready (locked). This bit always reads 1.		
4:2	Unused	Unused. Read = 000b; Write = Don’t Care		
1:0	MULSEL[1:0]	**Clock Multiplier Input Select Bits.** These bits select the clock supplied to the Clock Multiplier. 00: Internal High-Frequency Oscillator 01: External Oscillator 10: External Oscillator/2 11: Reserved. These bits always read 00.		

SFR Definition 17.9. P0MDIN: Port 0 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF1

Bit	Name	Function
7:0	P0MDIN[7:0]	**Analog Configuration Bits for P0.7–P0.0 (respectively).** Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P0.n pin is configured for analog mode. 1: Corresponding P0.n pin is not configured for analog mode.

SFR Definition 17.10. P0MDOUT: Port 0 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA4

Bit	Name	Function
7:0	P0MDOUT[7:0]	**Output Configuration Bits for P0.7–P0.0 (respectively).** These bits are ignored if the corresponding bit in register P0MDIN is logic 0. 0: Corresponding P0.n Output is open-drain. 1: Corresponding P0.n Output is push-pull.

USB Register Definition 18.21. EINCSRH: USB0 IN Endpoint Control High

Bit	7	6	5	4	3	2	1	0
Name	DBIEN	ISO	DIRSEL		FCDT	SPLIT		
Type	R/W	R/W	R/W	R	R/W	R/W	R	R
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x12

Bit	Name	Function
7	DBIEN	**IN Endpoint Double-buffer Enable.** 0: Double-buffering disabled for the selected IN endpoint. 1: Double-buffering enabled for the selected IN endpoint.
6	ISO	**Isochronous Transfer Enable.** This bit enables/disables isochronous transfers on the current endpoint. 0: Endpoint configured for bulk/interrupt transfers. 1: Endpoint configured for isochronous transfers.
5	DIRSEL	**Endpoint Direction Select.** This bit is valid only when the selected FIFO is not split (SPLIT = 0). 0: Endpoint direction selected as OUT. 1: Endpoint direction selected as IN.
4	UNUSED	Unused. Read = 0b. Write = don't care.
3	FCDT	**Force Data Toggle Bit.** 0: Endpoint data toggle switches only when an ACK is received following a data packet transmission. 1: Endpoint data toggle forced to switch after every data packet is transmitted, regardless of ACK reception.
2	SPLIT	**FIFO Split Enable.** When SPLIT = 1, the selected endpoint FIFO is split. The upper half of the selected FIFO is used by the IN endpoint; the lower half of the selected FIFO is used by the OUT endpoint.
1:0	Unused	Unused. Read = 00b. Write = don't care.

18.13. Controlling Endpoints1-2 OUT

Endpoints1-2 OUT are managed via USB registers EOUTCSRL and EOUTCSRH. All OUT endpoints can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing 1 to the ISO bit in register EOUTCSRH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1-2 OUT interrupt may be generated by the following:

1. Hardware sets the OPRDY bit (EINCSRL.0) to 1.
2. Hardware generates a STALL condition.

USB Register Definition 18.22. EOUTCSRL: USB0 OUT Endpoint Control Low Byte

Bit	7	6	5	4	3	2	1	0
Name	CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY
Type	W	R/W	R/W	R/W	R	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x14

Bit	Name	Description	Write	Read
7	CLRDT	**Clear Data Toggle Bit.**	Software should write 1 to this bit to reset the OUT endpoint data toggle to 0.	This bit always reads 0.
6	STSTL	**Sent Stall Bit.** Hardware sets this bit to 1 when a STALL handshake signal is transmitted. This flag must be cleared by software.		
5	SDSTL	**Send Stall Bit.** Software should write 1 to this bit to generate a STALL handshake. Software should write 0 to this bit to terminate the STALL signal. This bit has no effect in ISO mode.		
4	FLUSH	**FIFO Flush Bit.** Writing a 1 to this bit flushes the next packet to be read from the OUT endpoint FIFO. The FIFO pointer is reset and the OPRDY bit is cleared. Multiple packets must be flushed individually. Hardware resets the FLUSH bit to 0 when the flush is complete. **Note:** If data for the current packet has already been read from the FIFO, the FLUSH bit should not be used to flush the packet. Instead, the FIFO should be read manually.		
3	DATERR	**Data Error Bit.** In ISO mode, this bit is set by hardware if a received packet has a CRC or bit-stuffing error. It is cleared when software clears OPRDY. This bit is only valid in ISO mode.		
2	OVRUN	**Data Overrun Bit.** This bit is set by hardware when an incoming data packet cannot be loaded into the OUT endpoint FIFO. This bit is only valid in ISO mode, and must be cleared by software. 0: No data overrun. 1: A data packet was lost because of a full FIFO since this flag was last cleared.		
1	FIFOFUL	**OUT FIFO Full.** This bit indicates the contents of the OUT FIFO. If double buffering is enabled (DBIEN = 1), the FIFO is full when the FIFO contains two packets. If DBIEN = 0, the FIFO is full when the FIFO contains one packet. 0: OUT endpoint FIFO is not full. 1: OUT endpoint FIFO is full.		
0	OPRDY	**OUT Packet Ready.** Hardware sets this bit to 1 and generates an interrupt when a data packet is available. Software should clear this bit after each data packet is unloaded from the OUT endpoint FIFO.		

Table 19.1. SMBus Clock Source Selection

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 19.1. Note that the selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section "23. Timers" on page 202.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

Equation 19.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 19.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 19.2.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 19.2. Typical SMBus Bit Rate

Figure 19.4 shows the typical SCL generation described by Equation 19.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 19.1.

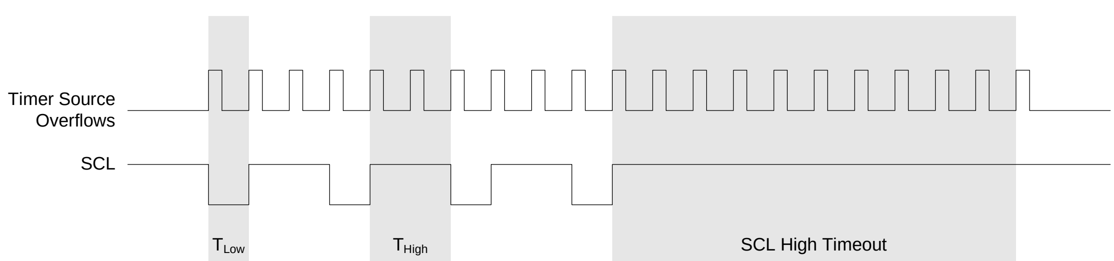

Figure 19.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 19.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 19.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} – 4 system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks

Note: Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “19.3.4. SCL Low Timeout” on page 151). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 19.4).

20.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

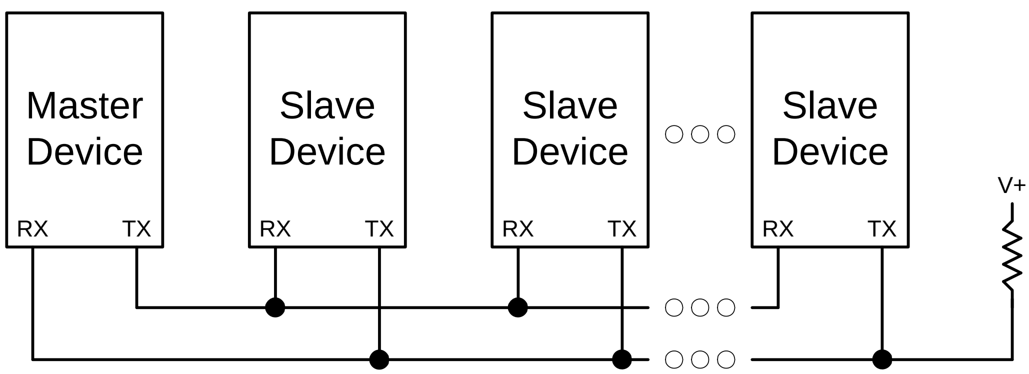

Figure 20.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 21.6. SBRLL1: UART1 Baud Rate Generator Low Byte

Bit	7	6	5	4	3	2	1	0
Name	SBRLL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xB4

Bit	Name	Function
7:0	SBRLL1[7:0]	**UART1 Baud Rate Reload Low Bits.** Low Byte of reload value for UART1 Baud Rate Generator.

23.3.3. Low-Frequency Oscillator (LFO) Capture Mode

The Low-Frequency Oscillator Capture Mode allows the LFO clock to be measured against the system clock or an external oscillator source. Timer 3 can be clocked from the system clock, the system clock divided by 12, or the external oscillator divided by 8, depending on the T3ML (CKCON.6), and T3XCLK[1:0] settings.

Setting TF3CEN to 1 enables the LFO Capture Mode for Timer 3. In this mode, T3SPLIT should be set to 0, as the full 16-bit timer is used. Upon a falling edge of the low-frequency oscillator, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set. By recording the difference between two successive timer capture values, the LFO clock frequency can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the LFO to achieve an accurate reading. This means that the LFO/8 should not be selected as the timer clock source in this mode.

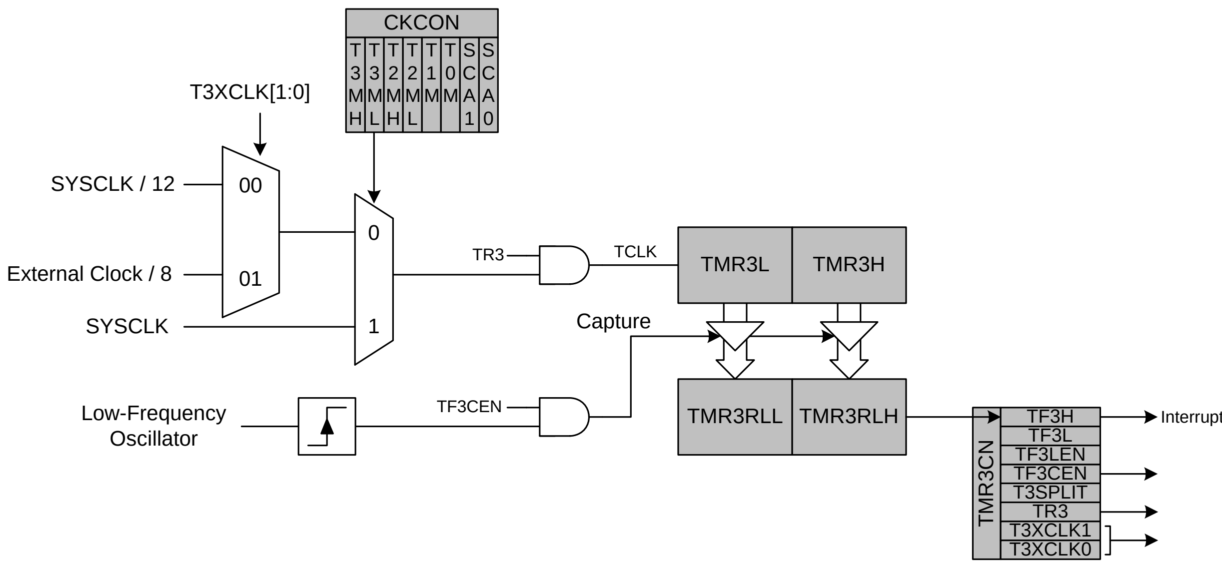

Figure 23.9. Timer 3 Low-Frequency Oscillation Capture Mode Block Diagram

SFR Definition 23.17. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x95

Bit	Name	Function
7:0	TMR3H[7:0]	**Timer 3 High Byte.** In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

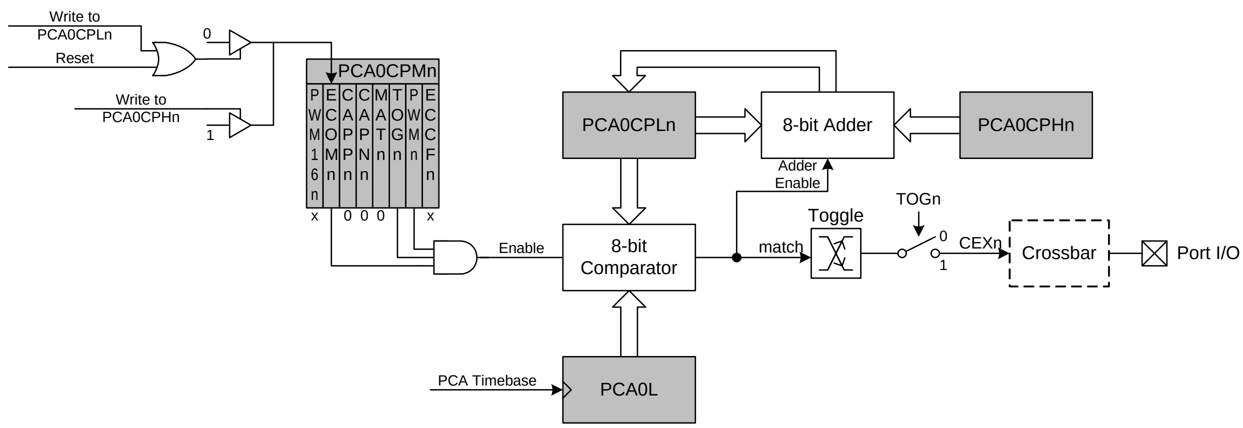

Figure 24.7. PCA Frequency Output Mode

24.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

24.3.5.1. 8-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 24.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module's capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 24.2.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare 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.

$$\text{Duty Cycle} = \frac{(256 - PCA0CPHn)}{256}$$

Equation 24.2. 8-Bit PWM Duty Cycle

Using Equation 24.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

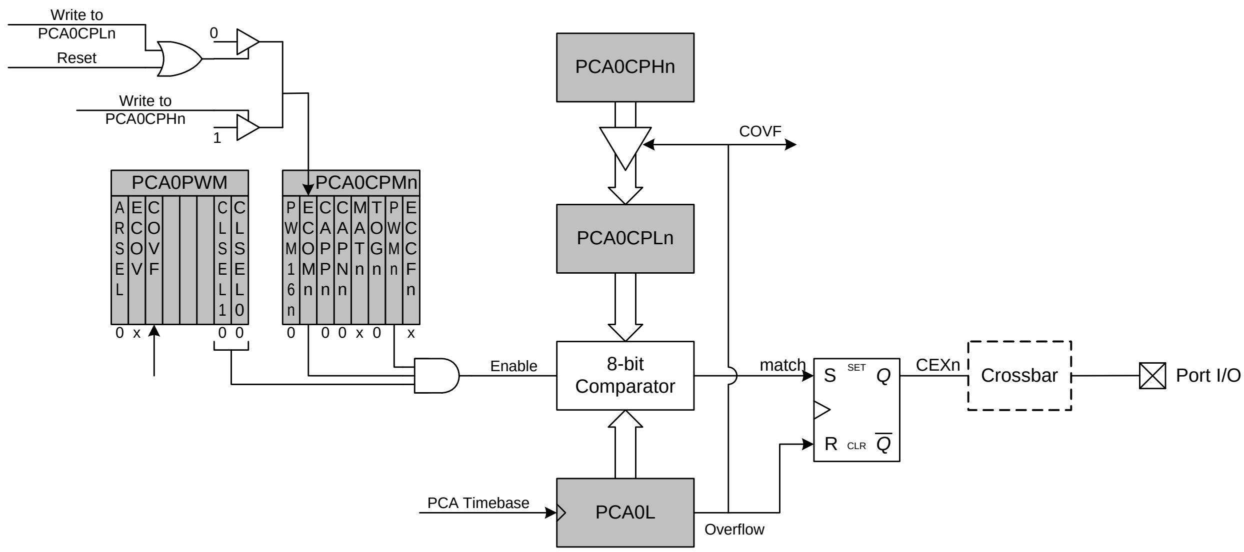

Figure 24.8. PCA 8-Bit PWM Mode Diagram

24.3.5.2. 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 24.9). 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 24.2, 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 24.3. 9, 10, and 11-Bit PWM Duty Cycle

C2 Register Definition 25.4. DEVCTL: C2 Device Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	DEVCTL[7:0]	**Device Control Register.** This register is used to halt the device for EPROM operations via the C2 interface. Refer to the EPROM chapter for more information.

C2 Register Definition 25.5. EPCTL: EPROM Programming Control Register

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

C2 Address: 0xDF

Bit	Name	Function
7:0	EPCTL[7:0]	**EPROM Programming Control Register.** This register is used to enable EPROM programming via the C2 interface. Refer to the EPROM chapter for more information.