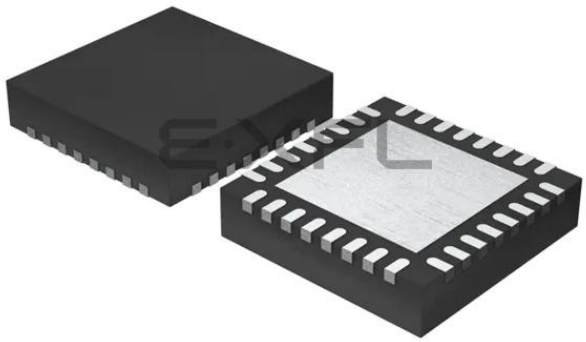

Welcome to E-XFL.COM

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, Temp Sensor, WDT
Number of I/O	24
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	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t620-gm

SFR Definition 8.1. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD0LJST	AD08BE	AMP0GN0
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	1

SFR Address = 0xBC

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Period Bits.** SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification table. $AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$ **Note:** If the Memory Power Controller is enabled (MPCE = '1'), AD0SC must be set to at least "00001" for proper ADC operation.
2	AD0LJST	**ADC0 Left Justify Select.** 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified. **Note:** The AD0LJST bit is only valid for 10-bit mode (AD08BE = 0).
1	AD08BE	**8-Bit Mode Enable.** 0: ADC operates in 10-bit mode (normal). 1: ADC operates in 8-bit mode. **Note:** When AD08BE is set to 1, the AD0LJST bit is ignored.
0	AMP0GN0	**ADC Gain Control Bit.** 0: Gain = 0.5 1: Gain = 1

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.

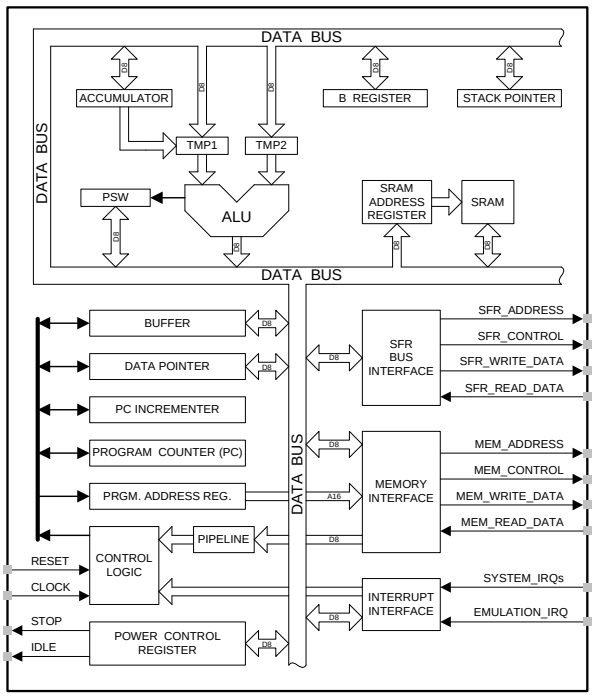

Figure 12.1. CIP-51 Block Diagram

SFR Definition 12.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD0; Bit-Addressable

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under software control.
4:3	RS[1:0]	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: • An ADD, ADDC, or SUBB instruction causes a sign-change overflow. • A MUL instruction results in an overflow (result is greater than 255). • A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

Table 16.2. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
ACC	0xE0	Accumulator	74
ADC0CF	0xBC	ADC0 Configuration	49
ADC0CN	0xE8	ADC0 Control	51
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	52
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	52
ADC0H	0xBE	ADC0 High	50
ADC0L	0xBD	ADC0 Low	50
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	53
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	53
AMX0P	0xBB	AMUX0 Positive Channel Select	56
B	0xF0	B Register	74
CKCON	0x8E	Clock Control	247
CLKMUL	0xB9	Clock Multiplier Control	132
CLKSEL	0xA9	Clock Select	129
CPT0CN	0x9B	Comparator0 Control	80
CPT0MD	0x9D	Comparator0 Mode Selection	81
CPT0MX	0x9F	Comparator0 MUX Selection	85
CPT1CN	0x9B	Comparator1 Control	82
CPT1MD	0x9D	Comparator1 Mode Selection	83
CPT1MX	0x9F	Comparator1 MUX Selection	86
DPH	0x83	Data Pointer High	73
DPL	0x82	Data Pointer Low	73
EIE1	0xE6	Extended Interrupt Enable 1	106
EIE2	0xE7	Extended Interrupt Enable 2	108
EIP1	0xF6	Extended Interrupt Priority 1	107
EIP2	0xF7	Extended Interrupt Priority 2	109
EMI0CF	0x85	External Memory Configuration	94
EMI0CN	0xAA	External Memory Interface Control	91

Table 16.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
IAPCN	0xF5	In-Application Programming Control	117
IE	0xA8	Interrupt Enable	104
IP	0xB8	Interrupt Priority	105
IT01CF	0xE4	INT0/INT1 Configuration	111
MEMKEY	0xB7	EPROM Memory Lock and Key	116
OSCICL	0xB3	Internal Oscillator Calibration	130
OSCICN	0xB2	Internal Oscillator Control	131
OSCLCN	0x86	Low-Frequency Oscillator Control	133
OSCXCN	0xB1	External Oscillator Control	137
P0	0x80	Port 0 Latch	152
P0MASK	0xAE	Port 0 Mask Configuration	150
P0MAT	0x84	Port 0 Match Configuration	150
P0MDIN	0xF1	Port 0 Input Mode Configuration	153
P0MDOUT	0xA4	Port 0 Output Mode Configuration	153
P0SKIP	0xD4	Port 0 Skip	154
P1	0x90	Port 1 Latch	154
P1MASK	0xBA	Port 1Mask Configuration	151
P1MAT	0xB6	Port 1 Match Configuration	151
P1MDIN	0xF2	Port 1 Input Mode Configuration	155
P1MDOUT	0xA5	Port 1 Output Mode Configuration	155
P1SKIP	0xD5	Port 1 Skip	156
P2	0xA0	Port 2 Latch	156
P2MDIN	0xF3	Port 2 Input Mode Configuration	157
P2MDOUT	0xA6	Port 2 Output Mode Configuration	157
P2SKIP	0xD6	Port 2 Skip	158
P3	0xB0	Port 3 Latch	158
P3MDOUT	0xA7	Port 3 Output Mode Configuration	159
PCA0CN	0xD8	PCA Control	282

17.1. MCU Interrupt Sources and Vectors

The C8051T620/1/6/7 & C8051T320/1/2/3 MCUs support 18 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 17.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

17.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 17.1.

17.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 6 system clock cycles: 1 clock cycle to detect the interrupt and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 20 system clock cycles: 1 clock cycle to detect the interrupt, 6 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Note that the CPU is stalled during EPROM write operations and USB FIFO MOVX accesses (see Section “15.2.3. Accessing USB FIFO Space” on page 91). Interrupt service latency will be increased for interrupts occurring while the CPU is stalled. The latency for these situations will be determined by the standard interrupt service procedure (as described above) and the amount of time the CPU is stalled.

17.2. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in this section. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

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.

SFR Definition 17.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7

6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

18.3.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (PSCTL.0) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write EPROM bytes.
8. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the EPROM write operation will be serviced in priority order after the EPROM operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the EPROM write pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write EPROM memory to ensure that a routine called with an illegal address does not result in modification of the EPROM.

18.3.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during EPROM write operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the EPROM operation has completed.

18.4. Program Memory CRC

A CRC engine is included on-chip which provides a means of verifying EPROM contents once the device has been programmed. The CRC engine is available for EPROM verification even if the device is fully read and write locked, allowing for verification of code contents at any time.

The CRC engine is operated through the C2 debug and programming interface, and performs 16-bit CRCs on individual 256-Byte blocks of program memory, or a 32-bit CRC on the entire memory space. To prevent hacking and extrapolation of security-locked source code, the CRC engine will only allow CRCs to be performed on contiguous 256-Byte blocks beginning on 256-Byte boundaries (lowest 8-bits of address are 0x00). For example, the CRC engine can perform a CRC for locations 0x0400 through 0x04FF, but it cannot perform a CRC for locations 0x0401 through 0x0500, or on block sizes smaller or larger than 256 Bytes.

18.4.1. Performing 32-bit CRCs on Full EPROM Content

A 32-bit CRC on the entire EPROM space is initiated by writing to the CRC1 byte over the C2 interface. The CRC calculation begins at address 0x0000, and ends at the end of user EPROM space. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 32-bit results will be available in the CRC3-0 registers. CRC3 is the MSB, and CRC0 is the LSB. The polynomial used for the 32-bit CRC calculation is 0x04C11DB7. Note: If a 16-bit CRC has been performed since the last device reset, a device reset should be initiated before performing a 32-bit CRC operation.

18.4.2. Performing 16-bit CRCs on 256-Byte EPROM Blocks

A 16-bit CRC of individual 256-byte blocks of EPROM can be initiated by writing to the CRC0 byte over the C2 interface. The value written to CRC0 is the high byte of the beginning address for the CRC. For example, if CRC0 is written to 0x02, the CRC will be performed on the 256-bytes beginning at address 0x0200, and ending at address 0x2FF. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 16-bit results will be available in the CRC1-0 registers. CRC1 is the MSB, and CRC0 is the LSB. The polynomial for the 16-bit CRC calculation is 0x1021.

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.

SFR Definition 22.3. XBR2: Port I/O Crossbar Register 2

Bit	7	6	5	4	3	2	1	0
Name							Reserved	URT1E
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE3

Bit	Name	Function
7:2	Unused	Read = 0000000b; Write = Don't Care.
1	Reserved	Must write 0.
0	URT1E	**UART1 I/O Output Enable Bit.** 0: UART1 I/O unavailable at Port pins. 1: UART1 TX1, RX1 routed to Port pins.

22.5. Port Match

Port match functionality allows system events to be triggered by a logic value change on P0 or P1. A software controlled value stored in the PnMATCH registers specifies the expected or normal logic values of P0 and P1. A Port mismatch event occurs if the logic levels of the Port's input pins no longer match the software controlled value. This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings.

The PnMASK registers can be used to individually select which P0 and P1 pins should be compared against the PnMATCH registers. A Port mismatch event is generated if (P0 & P0MASK) does not equal (P0MATCH & P0MASK) or if (P1 & P1MASK) does not equal (P1MATCH & P1MASK).

A Port mismatch event may be used to generate an interrupt or wake the device from a low power mode, such as IDLE or SUSPEND. See the Interrupts and Power Options chapters for more details on interrupt and wake-up sources.

SFR Definition 22.19. P2SKIP: Port 2 Skip

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD6

Bit	Name	Function
7:0	P2SKIP[7:0]	**Port 2 Crossbar Skip Enable Bits.** These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P2.n pin is not skipped by the Crossbar. 1: Corresponding P2.n pin is skipped by the Crossbar.

SFR Definition 22.20. P3: Port 3

Bit	7	6	5	4	3	2	1	0
Name								P3[0]
Type	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	1

SFR Address = 0xB0; Bit-Addressable

Bit	Name	Description	Write	Read
7:1	Unused	**Unused.**	Don't Care	0000000b
0	P3[0]	**Port 3 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P3.0 Port pin is logic LOW. 1: P3.0 Port pin is logic HIGH.

Table 23.2. USB0 Controller Registers

USB Register Name	USB Register Address	Description	Page Number
Interrupt Registers			
IN1INT	0x02	Endpoint0 and Endpoints1-3 IN Interrupt Flags	176
OUT1INT	0x04	Endpoints1-3 OUT Interrupt Flags	177
CMINT	0x06	Common USB Interrupt Flags	178
IN1IE	0x07	Endpoint0 and Endpoints1-3 IN Interrupt Enables	179
OUT1IE	0x09	Endpoints1-3 OUT Interrupt Enables	180
CMIE	0x0B	Common USB Interrupt Enables	181
Common Registers			
FADDR	0x00	Function Address	172
POWER	0x01	Power Management	174
FRAMEL	0x0C	Frame Number Low Byte	175
FRAMEH	0x0D	Frame Number High Byte	175
INDEX	0x0E	Endpoint Index Selection	167
CLKREC	0x0F	Clock Recovery Control	168
EENABLE	0x1E	Endpoint Enable	186
FIFOn	0x20-0x23	Endpoints0-3 FIFOs	171
Indexed Registers			
E0CSR	0x11	Endpoint0 Control / Status	184
EINCSRL		Endpoint IN Control / Status Low Byte	188
EINCSRH	0x12	Endpoint IN Control / Status High Byte	189
EOUTCSRL	0x14	Endpoint OUT Control / Status Low Byte	191
EOUTCSRH	0x15	Endpoint OUT Control / Status High Byte	192
E0CNT	0x16	Number of Received Bytes in Endpoint0 FIFO	185
EOUTCNTL		Endpoint OUT Packet Count Low Byte	192
EOUTCNTH	0x17	Endpoint OUT Packet Count High Byte	193

23.5. FIFO Management

1024 bytes of on-chip XRAM are used as FIFO space for USB0. This FIFO space is split between Endpoints0-3 as shown in Figure 23.3. FIFO space allocated for Endpoints1-3 is configurable as IN, OUT, or both (Split Mode: half IN, half OUT).

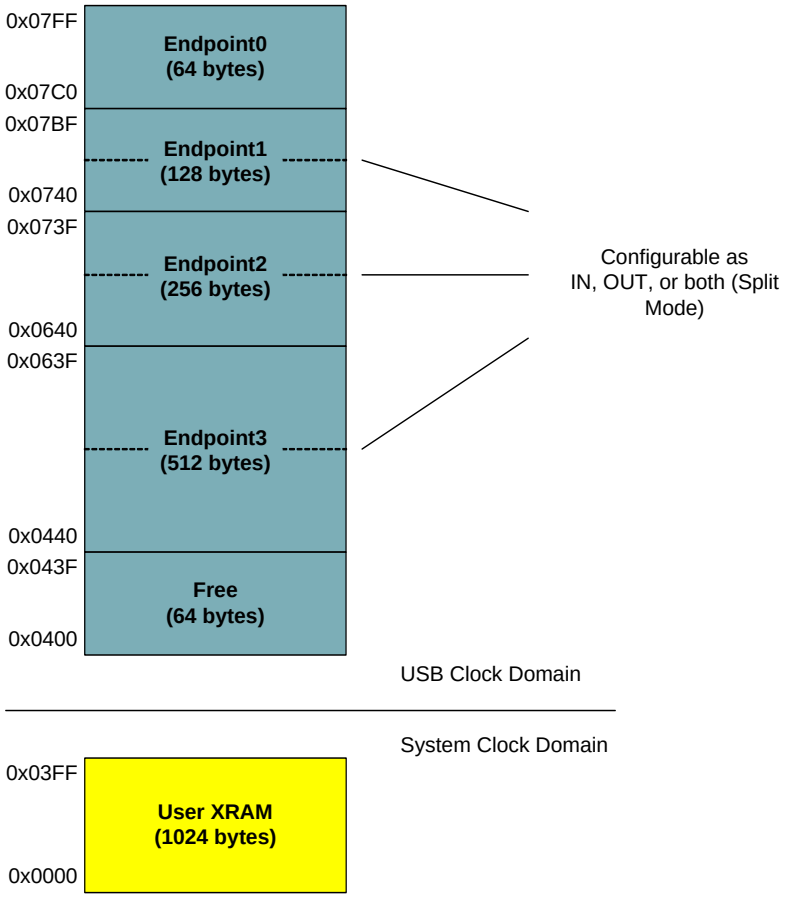

Figure 23.3. C8051T620/1 and C8051T320/1/2/3 USB FIFO Allocation

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.		

SFR Definition 26.6. 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.

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.

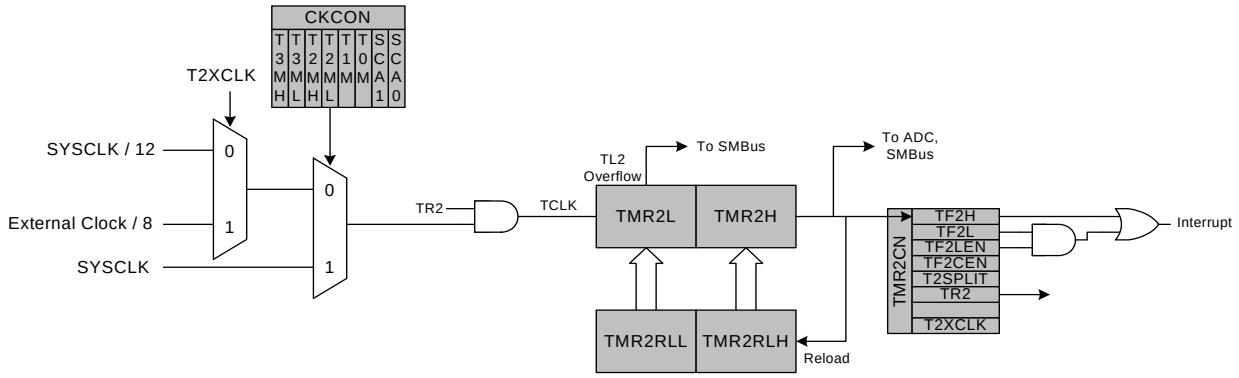

Figure 28.4. Timer 2 16-Bit Mode Block Diagram

SFR Definition 28.9. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCA

Bit	Name	Function
7:0	TMR2RLL[7:0]	**Timer 2 Reload Register Low Byte.** TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 28.10. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCB

Bit	Name	Function
7:0	TMR2RLH[7:0]	**Timer 2 Reload Register High Byte.** TMR2RLH holds the high byte of the reload value for Timer 2.

SFR Definition 28.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

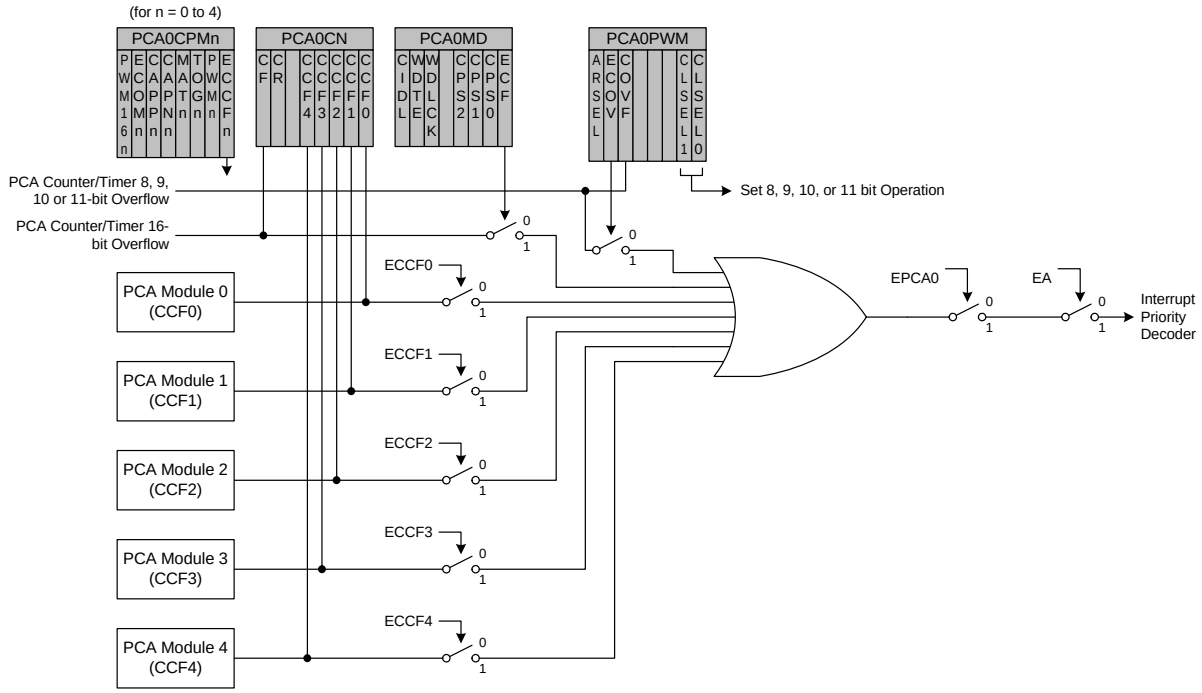

Figure 29.3. PCA Interrupt Block Diagram

29.3. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8 to 11-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation. Table 29.2 summarizes the bit settings in the PCA0CPMn and PCA0PWM registers used to select the PCA capture/compare module's operating mode. Note that all modules set to use 8, 9, 10, or 11-bit PWM mode must use the same cycle length (8-11 bits). Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt.

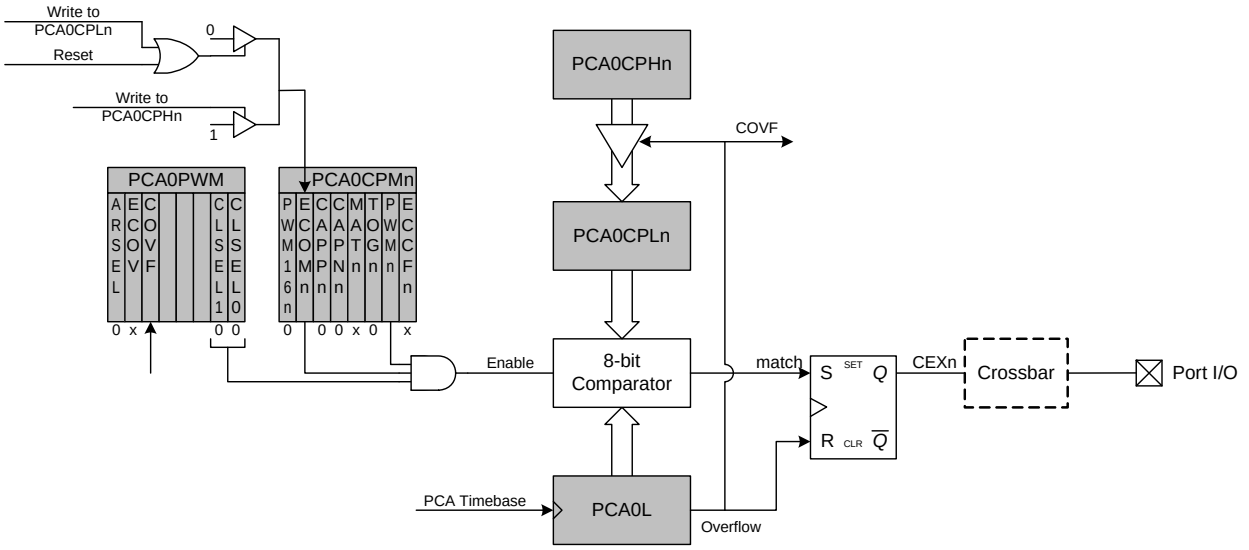

Figure 29.8. PCA 8-Bit PWM Mode Diagram

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