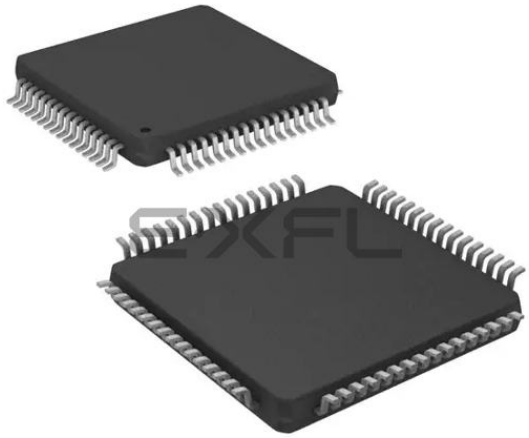


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	100MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f133-gq

1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

The devices include an on-chip 8k byte RAM block and an external memory interface (EMIF) for accessing off-chip data memory. The on-chip 8k byte block can be addressed over the entire 64k external data memory address range (overlapping 8k boundaries). External data memory address space can be mapped to on-chip memory only, off-chip memory only, or a combination of the two (addresses up to 8k directed to on-chip, above 8k directed to EMIF). The EMIF is also configurable for multiplexed or non-multiplexed address/data lines.

On the C8051F12x and C8051F130/1, the MCU's program memory consists of 128 k bytes of banked Flash memory. The 1024 bytes from addresses 0x1FC00 to 0x1FFFF are reserved. On the C8051F132/3, the MCU's program memory consists of 64 k bytes of Flash memory. This memory may be reprogrammed in-system in 1024 byte sectors, and requires no special off-chip programming voltage.

On all devices, there are also two 128 byte sectors at addresses 0x20000 to 0x200FF, which may be used by software for data storage. See Figure 1.8 for the MCU system memory map.

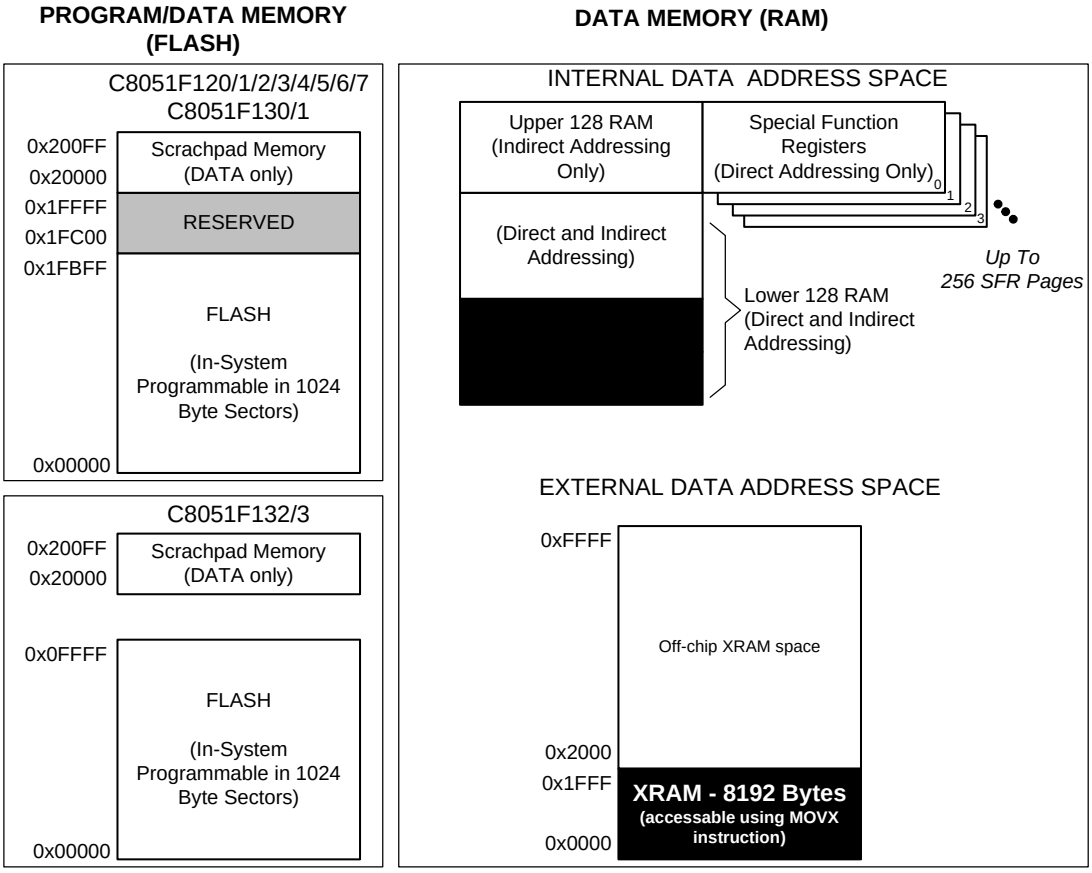
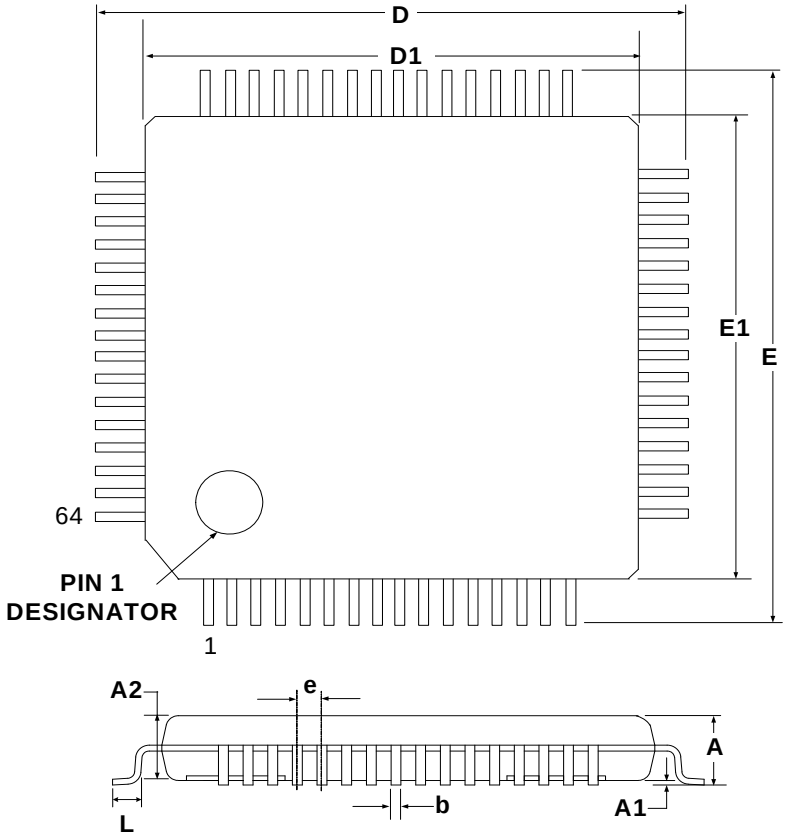


Figure 1.8. On-Chip Memory Map

4. Pinout and Package Definitions

Table 4.1. Pin Definitions

Name	Pin Numbers				Type	Description
	'F120 'F122 'F124 'F126	**'F121 'F123 'F125 'F127**	**'F130 'F132**	**'F131 'F133**		
V_{DD}	37, 64, 90	24, 41, 57	37, 64, 90	24, 41, 57		Digital Supply Voltage. Must be tied to +2.7 to +3.6 V.
DGND	38, 63, 89	25, 40, 56	38, 63, 89	25, 40, 56		Digital Ground. Must be tied to Ground.
AV+	11, 14	6	11, 14	6		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
AGND	10, 13	5	10, 13	5		Analog Ground. Must be tied to Ground.
TMS	1	58	1	58	D In	JTAG Test Mode Select with internal pullup.
TCK	2	59	2	59	D In	JTAG Test Clock with internal pullup.
TDI	3	60	3	60	D In	JTAG Test Data Input with internal pullup. TDI is latched on the rising edge of TCK.
TDO	4	61	4	61	D Out	JTAG Test Data Output with internal pullup. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver.
$\overline{RST}$	5	62	5	62	D I/O	Device Reset. Open-drain output of internal V_{DD} monitor. Is driven low when V_{DD} is < V_{RST} and MONEN is high. An external source can initiate a system reset by driving this pin low.
XTAL1	26	17	26	17	A In	Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock.
XTAL2	27	18	27	18	A Out	Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator.
MONEN	28	19	28	19	D In	V_{DD} Monitor Enable. When tied high, this pin enables the internal V_{DD} monitor, which forces a system reset when V_{DD} is < V_{RST}. When tied low, the internal V_{DD} monitor is disabled. **This pin must be tied high or low.**



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
D	-	12.00	-
D1	-	10.00	-
e	-	0.50	-
E	-	12.00	-
E1	-	10.00	-
L	0.45	0.60	0.75

Figure 4.6. TQFP-64 Package Drawing

The Temperature Sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings. Typical values for the Slope and Offset parameters can be found in Table 5.1.

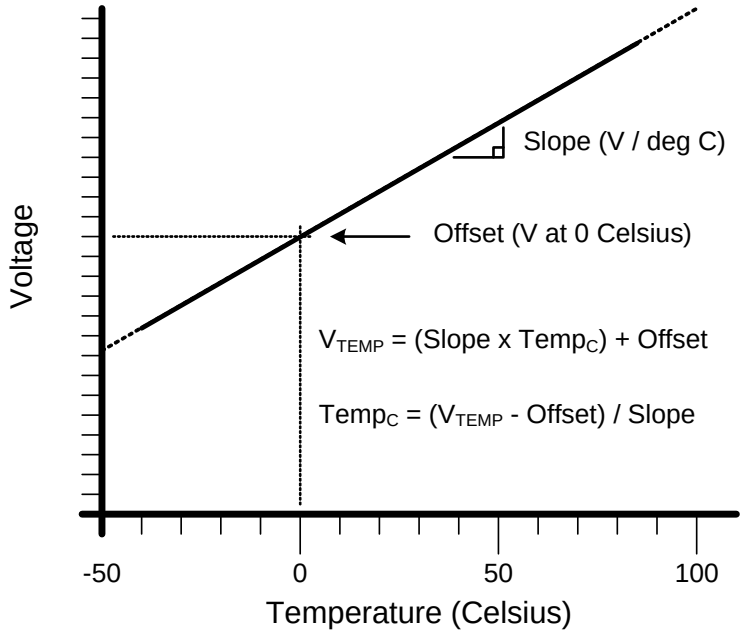


Figure 5.2. Typical Temperature Sensor Transfer Function

SFR Definition 8.1. DAC0H: DAC0 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD3
SFR Page: 0

Bits7–0: DAC0 Data Word Most Significant Byte.

SFR Definition 8.2. DAC0L: DAC0 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD2
SFR Page: 0

Bits7–0: DAC0 Data Word Least Significant Byte.

10. Comparators

Two on-chip programmable voltage comparators are included, as shown in Figure 10.1. The inputs of each comparator are available at dedicated pins. The output of each comparator is optionally available at the package pins via the I/O crossbar. When assigned to package pins, each comparator output can be programmed to operate in open drain or push-pull modes. See **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** for Crossbar and port initialization details.

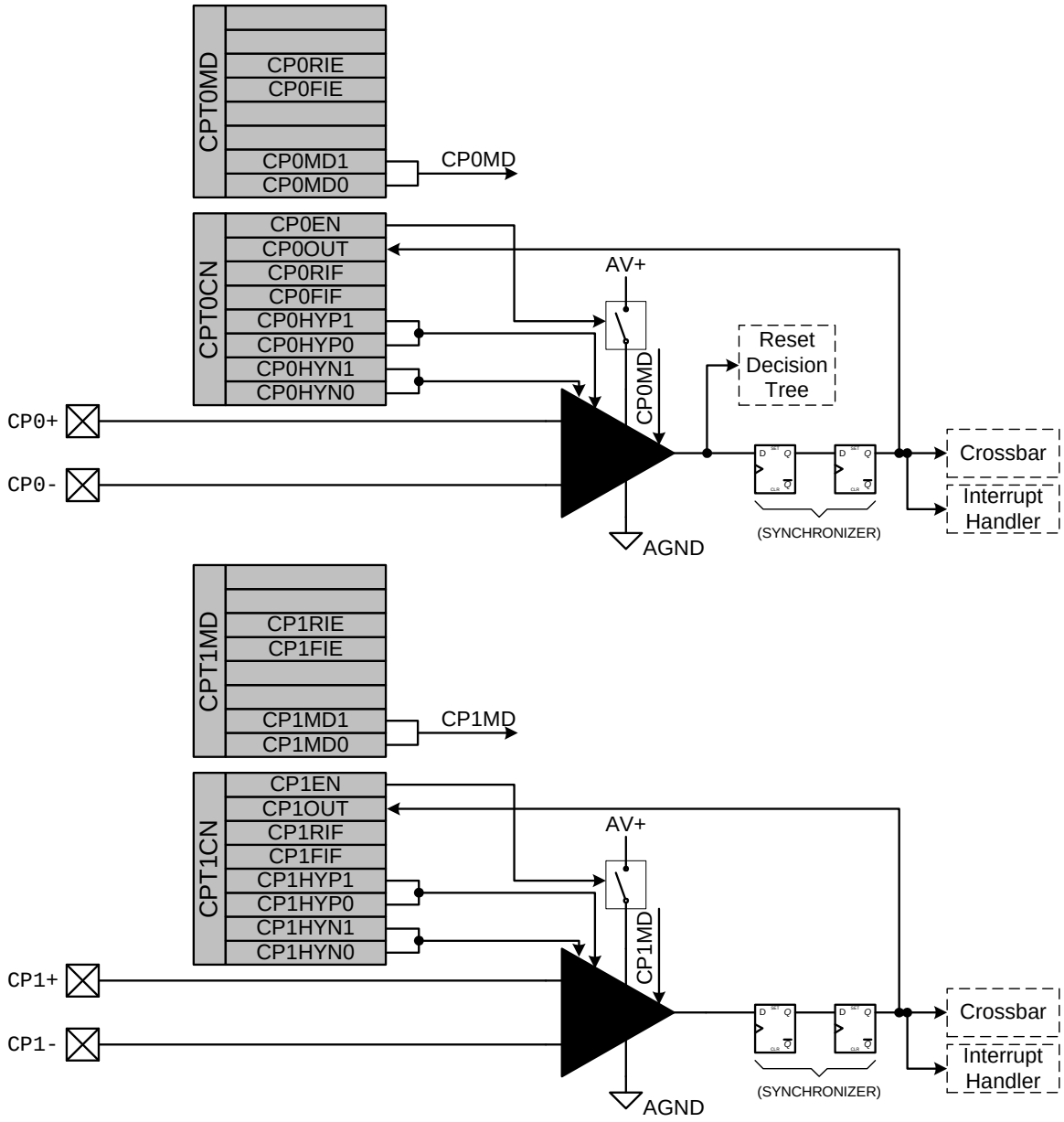


Figure 10.1. Comparator Functional Block Diagram

Table 11.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
JZ rel	Jump if A equals zero	2	2/3*
JNZ rel	Jump if A does not equal zero	2	2/3*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4*
NOP	No operation	1	1
* Branch instructions will incur a cache-miss penalty if the branch target location is not already stored in the Branch Target Cache. See **Section "16. Branch Target Cache" on page 211** for more details.			

Notes on Registers, Operands and Addressing Modes:

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

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

rel - 8-bit, signed (2s complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

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

#data - 8-bit constant

#data16 - 16-bit constant

bit - Direct-accessed bit in Data RAM or SFR

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

addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

Table 11.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
P6	0xE8	F	Port 6 Latch	page 256
P6MDOUT	0x9E	F	Port 6 Output Mode Configuration	page 256
P7	0xF8	F	Port 7 Latch	page 257
P7MDOUT	0x9F	F	Port 7 Output Mode Configuration	page 257
PCA0CN	0xD8	0	PCA Control	page 335
PCA0CPH0	0xFC	0	PCA Module 0 Capture/Compare High Byte	page 339
PCA0CPH1	0xFE	0	PCA Module 1 Capture/Compare High Byte	page 339
PCA0CPH2	0xEA	0	PCA Module 2 Capture/Compare High Byte	page 339
PCA0CPH3	0xEC	0	PCA Module 3 Capture/Compare High Byte	page 339
PCA0CPH4	0xEE	0	PCA Module 4 Capture/Compare High Byte	page 339
PCA0CPH5	0xE2	0	PCA Module 5 Capture/Compare High Byte	page 339
PCA0CPL0	0xFB	0	PCA Module 0 Capture/Compare Low Byte	page 338
PCA0CPL1	0xFD	0	PCA Module 1 Capture/Compare Low Byte	page 338
PCA0CPL2	0xE9	0	PCA Module 2 Capture/Compare Low Byte	page 338
PCA0CPL3	0xEB	0	PCA Module 3 Capture/Compare Low Byte	page 338
PCA0CPL4	0xED	0	PCA Module 4 Capture/Compare Low Byte	page 338
PCA0CPL5	0xE1	0	PCA Module 5 Capture/Compare Low Byte	page 338
PCA0CPM0	0xDA	0	PCA Module 0 Mode	page 337
PCA0CPM1	0xDB	0	PCA Module 1 Mode	page 337
PCA0CPM2	0xDC	0	PCA Module 2 Mode	page 337
PCA0CPM3	0xDD	0	PCA Module 3 Mode	page 337
PCA0CPM4	0xDE	0	PCA Module 4 Mode	page 337
PCA0CPM5	0xDF	0	PCA Module 5 Mode	page 337
PCA0H	0xFA	0	PCA Counter High Byte	page 338
PCA0L	0xF9	0	PCA Counter Low Byte	page 338
PCA0MD	0xD9	0	PCA Mode	page 336
PCON	0x87	All Pages	Power Control	page 164
PLL0CN	0x89	F	PLL Control	page 193
PLL0DIV	0x8D	F	PLL Divider	page 194
PLL0FLT	0x8F	F	PLL Filter	page 195
PLL0MUL	0x8E	F	PLL Multiplier	page 194
PSBANK	0xB1	All Pages	Flash Bank Select	page 134
PSCTL	0x8F	0	Flash Write/Erase Control	page 209
PSW	0xD0	All Pages	Program Status Word	page 152
RCAP2H	0xCB	0	Timer/Counter 2 Capture/Reload High Byte	page 323
RCAP2L	0xCA	0	Timer/Counter 2 Capture/Reload Low Byte	page 323
RCAP3H	0xCB	1	Timer 3 Capture/Reload High Byte	page 323
RCAP3L	0xCA	1	Timer 3 Capture/Reload Low Byte	page 323
RCAP4H	0xCB	2	Timer/Counter 4 Capture/Reload High Byte	page 323
RCAP4L	0xCA	2	Timer/Counter 4 Capture/Reload Low Byte	page 323

SFR Definition 11.10. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xE0
SFR Page: All Pages

Bits7–0: ACC: Accumulator.
This register is the accumulator for arithmetic operations.

SFR Definition 11.11. B: B Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF0
SFR Page: All Pages

Bits7–0: B: B Register.
This register serves as a second accumulator for certain arithmetic operations.

11.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 20 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE, EIE1, or EIE2). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Note: Any instruction that clears the EA bit should be immediately followed by an instruction that has two or more opcode bytes. For example:

```
// in 'C':
EA = 0; // clear EA bit.
EA = 0; // this is a dummy instruction with two-byte opcode.

; in assembly:
CLR EA ; clear EA bit.
CLR EA ; this is a dummy instruction with two-byte opcode.
```

If an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears the EA bit), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the EA bit will return a '0' inside the interrupt service routine. When the "CLR EA" opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

11.3.1. MCU Interrupt Sources and Vectors

The MCUs support 20 interrupt sources. Software can simulate an interrupt event 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 11.4. 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).

NOTES:

17.5.3. Split Mode with Bank Select

When EMI0CF.[3:2] are set to ‘10’, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the 8k boundary will access on-chip XRAM space.
- Effective addresses above the 8k boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in “Bank Select” mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

17.5.4. External Only

When EMI0CF[3:2] are set to ‘11’, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the 8k boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in “Split Mode without Bank Select” described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

17.6. EMIF Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, /RD and /WR strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 17.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for /RD or /WR pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for /ALE + 1 for /RD or /WR + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 17.1 lists the ac parameters for the External Memory Interface, and Figure 17.4 through Figure 17.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

17.6.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’.

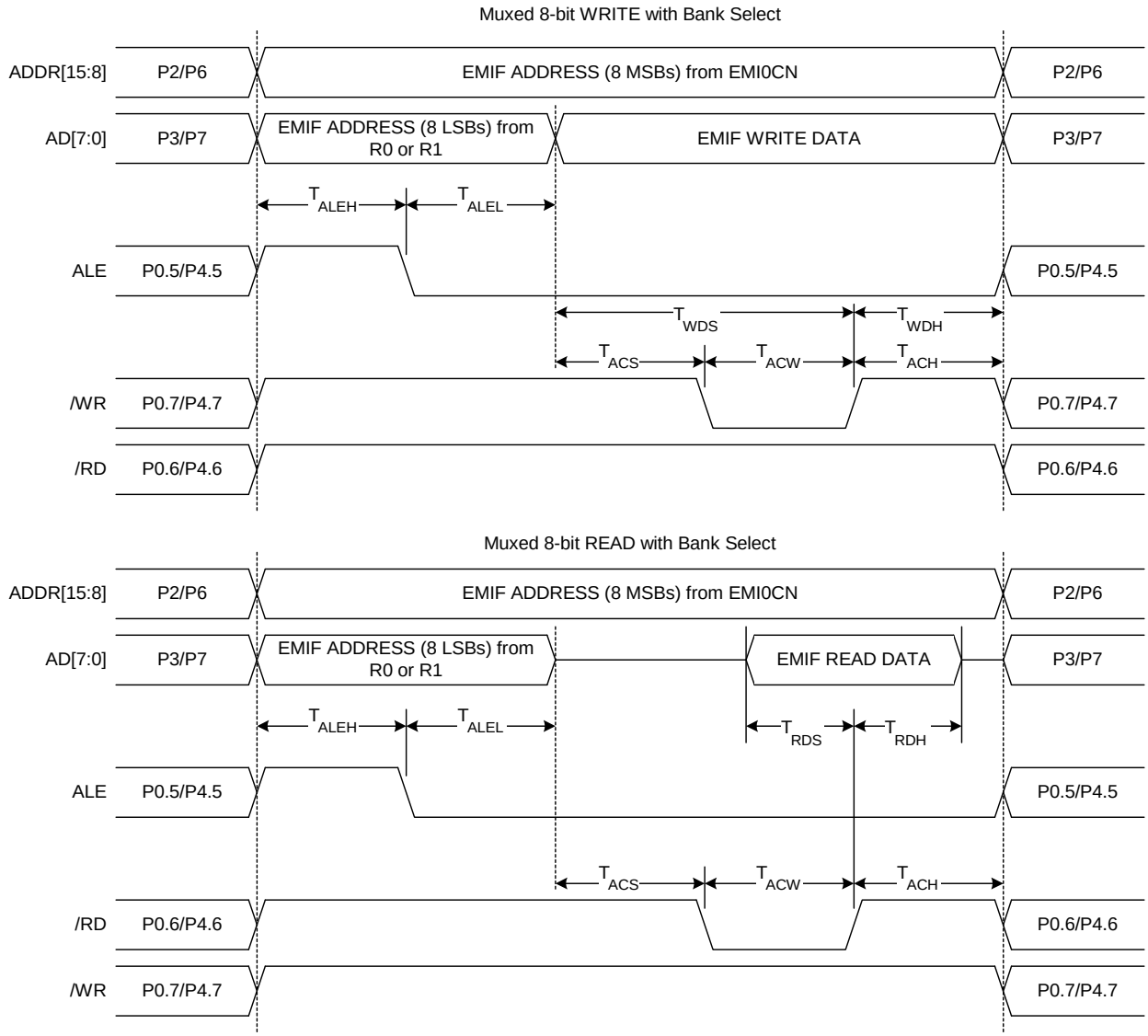


Figure 17.9. Multiplexed 8-bit MOVX with Bank Select Timing

A wide array of digital resources is available through the four lower I/O Ports: P0, P1, P2, and P3. Each of the pins on P0, P1, P2, and P3, can be defined as a General-Purpose I/O (GPIO) pin or can be controlled by a digital peripheral or function (like UART0 or /INT1 for example), as shown in Figure 18.2. The system designer controls which digital functions are assigned pins, limited only by the number of pins available. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read from its associated Data register regardless of whether that pin has been assigned to a digital peripheral or behaves as GPIO. The Port pins on Port 1 can be used as Analog Inputs to ADC2.

An External Memory Interface which is active during the execution of an off-chip MOVX instruction can be active on either the lower Ports or the upper Ports. See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

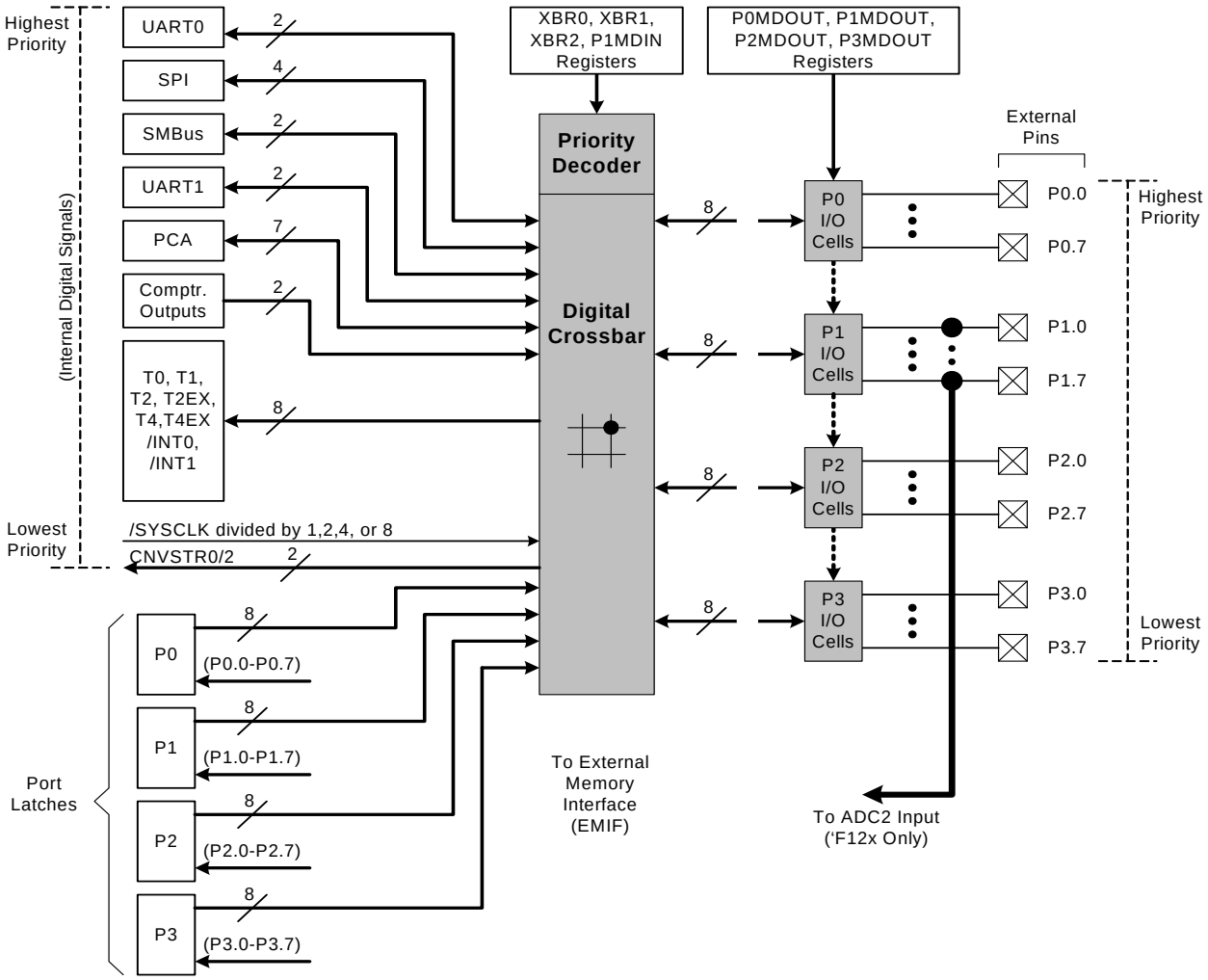


Figure 18.2. Port I/O Functional Block Diagram

SFR Definition 19.1. SMB0CN: SMBus0 Control

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
BUSY	ENSMB	STA	STO	SI	AA	FTE	TOE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xC0
SFR Page: 0

Bit7: BUSY: Busy Status Flag.
0: SMBus0 is free
1: SMBus0 is busy

Bit6: ENSMB: SMBus Enable.
This bit enables/disables the SMBus serial interface.
0: SMBus0 disabled.
1: SMBus0 enabled.

Bit5: STA: SMBus Start Flag.
0: No START condition is transmitted.
1: When operating as a master, a START condition is transmitted if the bus is free. (If the bus is not free, the START is transmitted after a STOP is received.) If STA is set after one or more bytes have been transmitted or received and before a STOP is received, a repeated START condition is transmitted.

Bit4: STO: SMBus Stop Flag.
0: No STOP condition is transmitted.
1: Setting STO to logic 1 causes a STOP condition to be transmitted. When a STOP condition is received, hardware clears STO to logic 0. If both STA and STO are set, a STOP condition is transmitted followed by a START condition. In slave mode, setting the STO flag causes SMBus to behave as if a STOP condition was received.

Bit3: SI: SMBus Serial Interrupt Flag.
This bit is set by hardware when one of 27 possible SMBus0 states is entered. (Status code 0xF8 does not cause SI to be set.) When the SI interrupt is enabled, setting this bit causes the CPU to vector to the SMBus interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: AA: SMBus Assert Acknowledge Flag.
This bit defines the type of acknowledge returned during the acknowledge cycle on the SCL line.
0: A "not acknowledge" (high level on SDA) is returned during the acknowledge cycle.
1: An "acknowledge" (low level on SDA) is returned during the acknowledge cycle.

Bit1: FTE: SMBus Free Timer Enable Bit
0: No timeout when SCL is high
1: Timeout when SCL high time exceeds limit specified by the SMB0CR value.

Bit0: TOE: SMBus Timeout Enable Bit
0: No timeout when SCL is low.
1: Timeout when SCL low time exceeds limit specified by Timer 3, if enabled.

Table 19.1. SMB0STA Status Codes and States (Continued)

	0x60	Own slave address + W received. ACK transmitted.	Wait for data.
	0x68	Arbitration lost in sending SLA + R/W as master. Own address + W received. ACK transmitted.	Save current data for retry when bus is free. Wait for data.
	0x70	General call address received. ACK transmitted.	Wait for data.
	0x78	Arbitration lost in sending SLA + R/W as master. General call address received. ACK transmitted.	Save current data for retry when bus is free.
	0x80	Data byte received. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x88	Data byte received. NACK transmitted.	Set STO to reset SMBus.
	0x90	Data byte received after general call address. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x98	Data byte received after general call address. NACK transmitted.	Set STO to reset SMBus.
	0xA0	STOP or repeated START received.	No action necessary.
	0xA8	Own address + R received. ACK transmitted.	Load SMB0DAT with data to transmit.
	0xB0	Arbitration lost in transmitting SLA + R/W as master. Own address + R received. ACK transmitted.	Save current data for retry when bus is free. Load SMB0DAT with data to transmit.
	0xB8	Data byte transmitted. ACK received.	Load SMB0DAT with data to transmit.
	0xC0	Data byte transmitted. NACK received.	Wait for STOP.
	0xC8	Last data byte transmitted (AA=0). ACK received.	Set STO to reset SMBus.
	0xD0	SCL Clock High Timer per SMB0CR timed out	Set STO to reset SMBus.
	0x00	Bus Error (illegal START or STOP)	Set STO to reset SMBus.
	0xF8	Idle	State does not set SI.

21.1. UART0 Operational Modes

UART0 provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON0 register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 21.1.

Table 21.1. UART0 Modes

Mode	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	Synchronous	SYSCLK / 12	8	None
1	Asynchronous	Timer 1, 2, 3, or 4 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK / 32 or SYSCLK / 64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1, 2, 3, or 4 Overflow	9	1 Start, 1 Stop

21.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX0 pin. The TX0 pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions (see the interconnect diagram in Figure 21.3).

Data transmission begins when an instruction writes a data byte to the SBUF0 register. Eight data bits are transferred LSB first (see the timing diagram in Figure 21.2), and the TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the eighth bit time. Data reception begins when the REN0 Receive Enable bit (SCON0.4) is set to logic 1 and the RI0 Receive Interrupt Flag (SCON0.0) is cleared. One cycle after the eighth bit is shifted in, the RI0 flag is set and reception stops until software clears the RI0 bit. An interrupt will occur if enabled when either TI0 or RI0 are set.

The Mode 0 baud rate is SYSCLK / 12. RX0 is forced to open-drain in Mode 0, and an external pullup will typically be required.

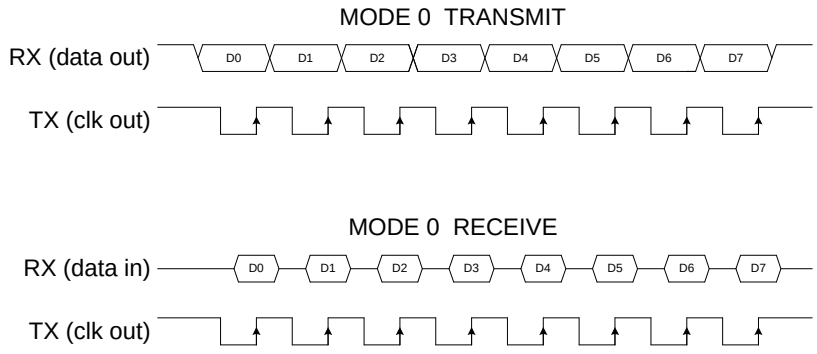


Figure 21.2. UART0 Mode 0 Timing Diagram

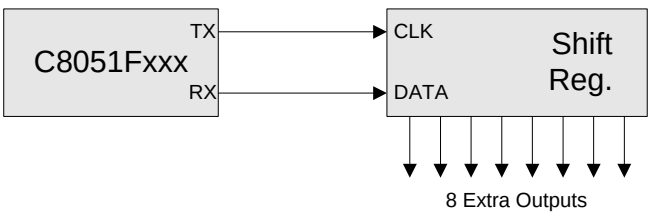


Figure 21.3. UART0 Mode 0 Interconnect

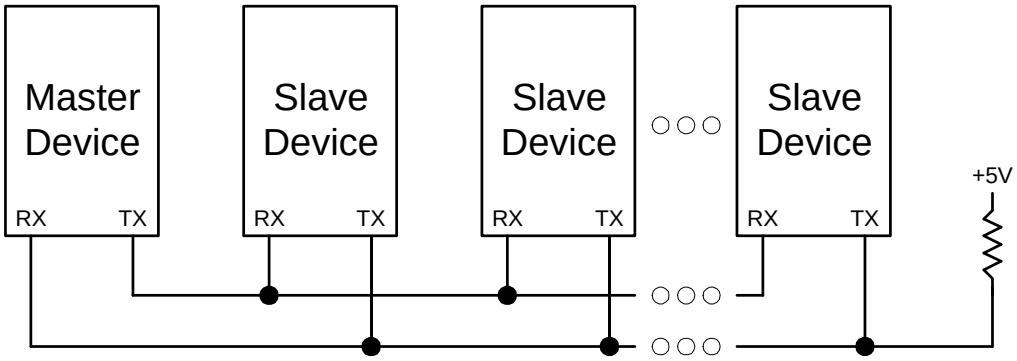


Figure 21.7. UART Multi-Processor Mode Interconnect Diagram

21.3. Frame and Transmission Error Detection

All Modes:
The Transmit Collision bit (TXCOL0 bit in register SSTA0) reads ‘1’ if user software writes data to the SBUF0 register while a transmit is in progress.

Modes 1, 2, and 3:
The Receive Overrun bit (RXOV0 in register SSTA0) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. The Frame Error bit (FE0 in register SSTA0) reads ‘1’ if an invalid (low) STOP bit is detected.

24.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 24.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic 1). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 24.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8*
***Note:** External clock divided by 8 is synchronized with the system clock.			

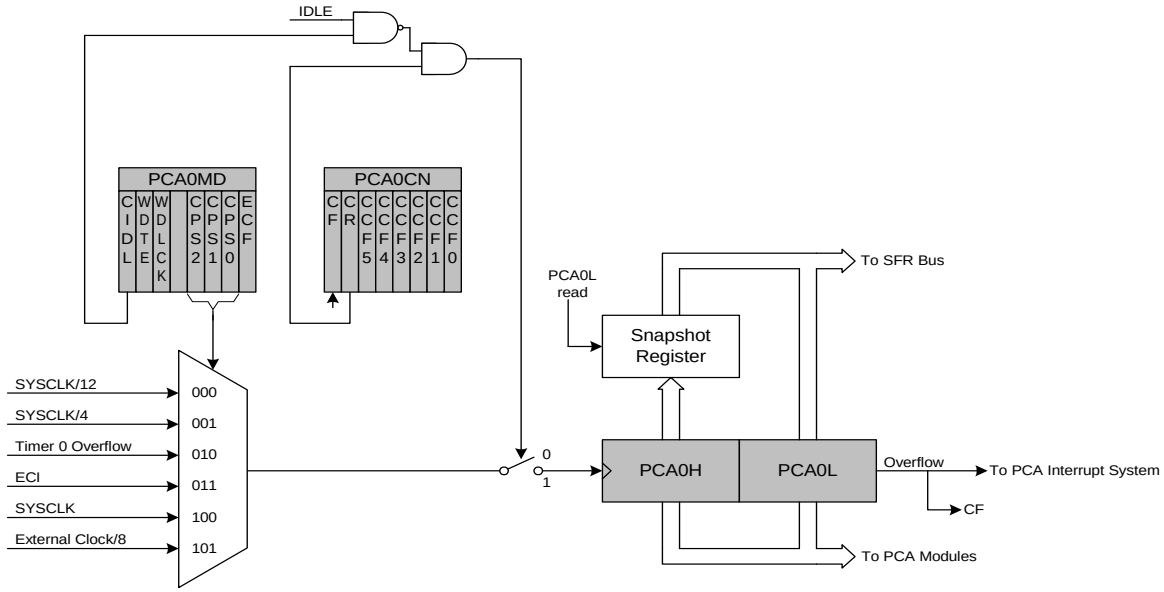


Figure 24.2. PCA Counter/Timer Block Diagram

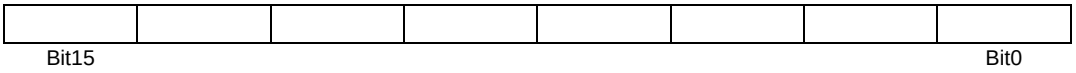
25. JTAG (IEEE 1149.1)

Each MCU has an on-chip JTAG interface and logic to support boundary scan for production and in-system testing, Flash read/write operations, and non-intrusive in-circuit debug. The JTAG interface is fully compliant with the IEEE 1149.1 specification. Refer to this specification for detailed descriptions of the Test Interface and Boundary-Scan Architecture. Access of the JTAG Instruction Register (IR) and Data Registers (DR) are as described in the Test Access Port and Operation of the IEEE 1149.1 specification.

The JTAG interface is accessed via four dedicated pins on the MCU: TCK, TMS, TDI, and TDO.

Through the 16-bit JTAG Instruction Register (IR), any of the eight instructions shown in Figure 25.1 can be commanded. There are three DR's associated with JTAG Boundary-Scan, and four associated with Flash read/write operations on the MCU.

JTAG Register Definition 25.1. IR: JTAG Instruction Register



IR Value	Instruction	Description
0x0000	EXTEST	Selects the Boundary Data Register for control and observability of all device pins
0x0002	SAMPLE/ PRELOAD	Selects the Boundary Data Register for observability and presetting the scan-path latches
0x0004	IDCODE	Selects device ID Register
0xFFFF	BYPASS	Selects Bypass Data Register
0x0082	Flash Control	Selects FLASHCON Register to control how the interface logic responds to reads and writes to the FLASHDAT Register
0x0083	Flash Data	Selects FLASHDAT Register for reads and writes to the Flash memory
0x0084	Flash Address	Selects FLASHADR Register which holds the address of all Flash read, write, and erase operations
0x0085	Flash Scale	Selects FLASHSCL Register which controls the Flash one-shot timer and read-always enable