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	20MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f001

Figure 10.5. DPH: Data Pointer High Byte 74
Figure 10.6. PSW: Program Status Word 75
Figure 10.7. ACC: Accumulator 76
Figure 10.8. B: B Register 76
10.4. INTERRUPT HANDLER 77
Table 10.4. Interrupt Summary 78
Figure 10.9. IE: Interrupt Enable 79
Figure 10.10. IP: Interrupt Priority 80
Figure 10.11. EIE1: Extended Interrupt Enable 1 81
Figure 10.12. EIE2: Extended Interrupt Enable 2 82
Figure 10.13. EIP1: Extended Interrupt Priority 1 83
Figure 10.14. EIP2: Extended Interrupt Priority 2 84
10.5. Power Management Modes 85
Figure 10.15. PCON: Power Control Register 86
11. FLASH MEMORY 87
11.1. Programming The Flash Memory 87
Table 11.1. FLASH Memory Electrical Characteristics 87
11.2. Non-volatile Data Storage 88
11.3. Security Options 88
Figure 11.1. PSCTL: Program Store RW Control 88
Figure 11.2. Flash Program Memory Security Bytes 89
Figure 11.3. FLACL: Flash Access Limit (C8051F005/06/07/15/16/17 only) 90
Figure 11.4. FLSCL: Flash Memory Timing Prescaler 91
12. EXTERNAL RAM (C8051F005/06/07/15/16/17) 92
Figure 12.1. EMI0CN: External Memory Interface Control 92
13. RESET SOURCES 93
Figure 13.1. Reset Sources Diagram 93
13.1. Power-on Reset 94
13.2. Software Forced Reset 94
Figure 13.2. VDD Monitor Timing Diagram 94
13.3. Power-fail Reset 94
13.4. External Reset 95
13.5. Missing Clock Detector Reset 95
13.6. Comparator 0 Reset 95
13.7. External CNVSTR Pin Reset 95
13.8. Watchdog Timer Reset 95
Figure 13.3. WDTCN: Watchdog Timer Control Register 96
Figure 13.4. RSTSRC: Reset Source Register 97
Table 13.1. Reset Electrical Characteristics 98
14. OSCILLATOR 99
Figure 14.1. Oscillator Diagram 99
Figure 14.2. OSCICN: Internal Oscillator Control Register 100
Table 14.1. Internal Oscillator Electrical Characteristics 100
Figure 14.3. OSCXCN: External Oscillator Control Register 101
14.1. External Crystal Example 102
14.2. External RC Example 102
14.3. External Capacitor Example 102
15. PORT INPUT/OUTPUT 103
15.1. Priority Cross Bar Decoder 103
15.2. Port I/O Initialization 103
Figure 15.1. Port I/O Functional Block Diagram 104
Figure 15.2. Port I/O Cell Block Diagram 104
Table 15.1. Crossbar Priority Decode 105
Figure 15.3. XBR0: Port I/O CrossBar Register 0 106

1. SYSTEM OVERVIEW

The C8051F000 family are fully integrated mixed-signal System on a Chip MCUs with a true 12-bit multi-channel ADC (F000/01/02/05/06/07), or a true 10-bit multi-channel ADC (F010/11/12/15/16/17). See the Product Selection Guide in Table 1.1 for a quick reference of each MCUs' feature set. Each has a programmable gain pre-amplifier, two 12-bit DACs, two voltage comparators (except for the F002/07/12/17, which have one), a voltage reference, and an 8051-compatible microcontroller core with 32kbytes of FLASH memory. There are also I2C/SMBus, UART, and SPI serial interfaces implemented in hardware (not "bit-banged" in user software) as well as a Programmable Counter/Timer Array (PCA) with five capture/compare modules. There are also 4 general-purpose 16-bit timers and 4 byte-wide general-purpose digital Port I/O. The C8051F000/01/02/10/11/12 have 256 bytes of RAM and execute up to 20MIPS, while the C8051F005/06/07/15/16/17 have 2304 bytes of RAM and execute up to 25MIPS.

With an on-board VDD monitor, WDT, and clock oscillator, the MCUs are truly stand-alone System-on-a-Chip solutions. Each MCU effectively configures and manages the analog and digital peripherals. The FLASH memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. Each MCU can also individually shut down any or all of the peripherals to conserve power.

On-board JTAG debug support allows non-intrusive (uses no on-chip resources), full speed, in-circuit debug using the production MCU installed in the final application. This debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, and run and halt commands. All analog and digital peripherals are fully functional when using JTAG debug.

Each MCU is specified for 2.7V-to-3.6V operation over the industrial temperature range (-45C to +85C). The Port I/Os, /RST, and JTAG pins are tolerant for input signals up to 5V. The C8051F000/05/10/15 are available in the 64-pin TQFP (see block diagram in Figure 1.1). The C8051F001/06/11/16 are available in the 48-pin TQFP (see block diagram in Figure 1.2). The C8051F002/07/12/17 are available in the 32-pin LQFP (see block diagram in Figure 1.3).

Table 1.1. Product Selection Guide

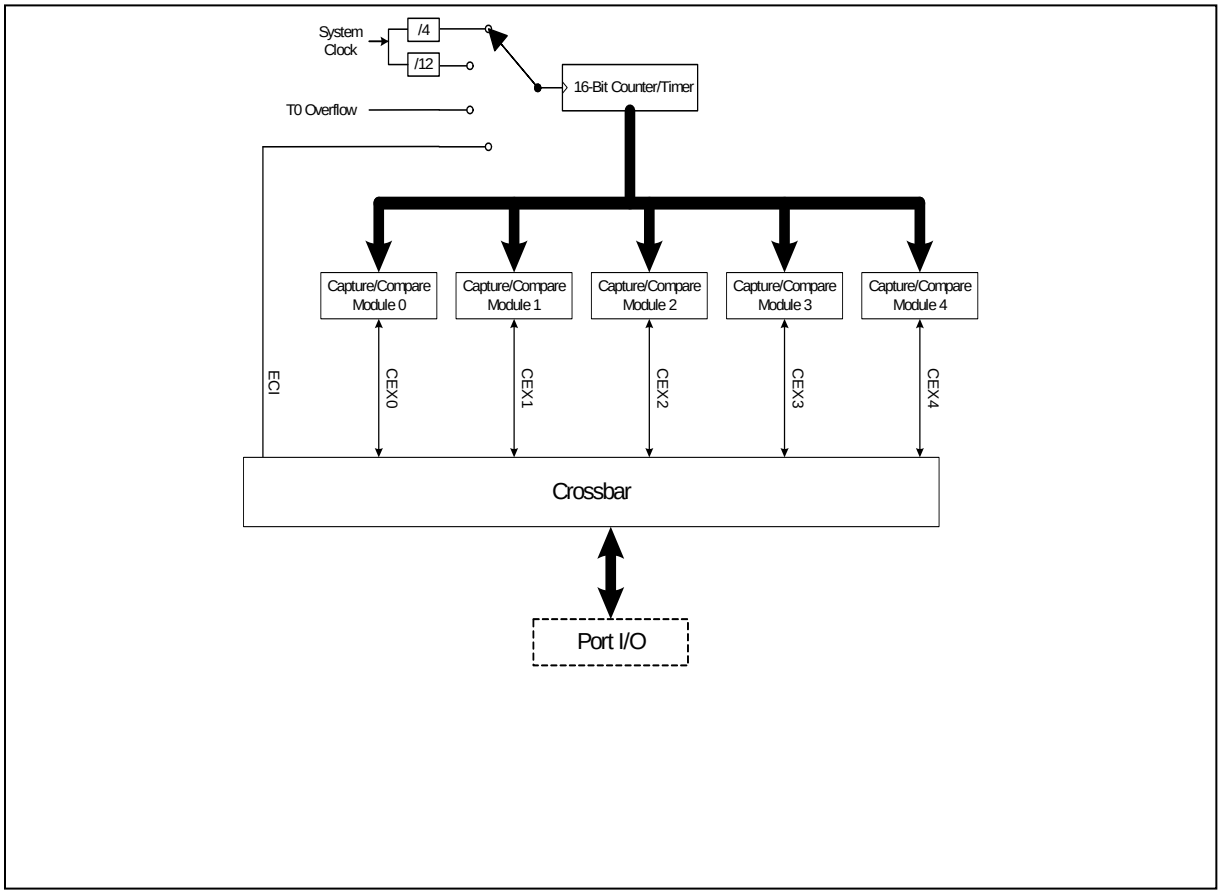
	MIPS (Peak)	FLASH Memory	RAM	SMBus/I2C	SPI	UART	Timers (16-bit)	Programmable Counter Array	Digital Port I/O's	ADC Resolution (bits)	ADC Max Speed (ksps)	ADC Inputs	Voltage Reference	Temperature Sensor	DAC Resolution	DAC Outputs	Voltage Comparators	Package
C8051F000	20	32k	256	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F001	20	32k	256	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F002	20	32k	256	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F005	25	32k	2304	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F006	25	32k	2304	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F007	25	32k	2304	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F010	20	32k	256	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F011	20	32k	256	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F012	20	32k	256	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP
C8051F015	25	32k	2304	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F016	25	32k	2304	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F017	25	32k	2304	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP

1.5. Programmable Counter Array

The C8051F000 MCU family has an on-board Programmable Counter/Timer Array (PCA) in addition to the four 16-bit general-purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer timebase with 5 programmable capture/compare modules. The timebase gets its clock from one of four sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflow, or an External Clock Input (ECI).

Each capture/compare module can be configured to operate in one of four modes: Edge-Triggered Capture, Software Timer, High Speed Output, or Pulse Width Modulator. The PCA Capture/Compare Module I/O and External Clock Input are routed to the MCU Port I/O via the Digital Crossbar.

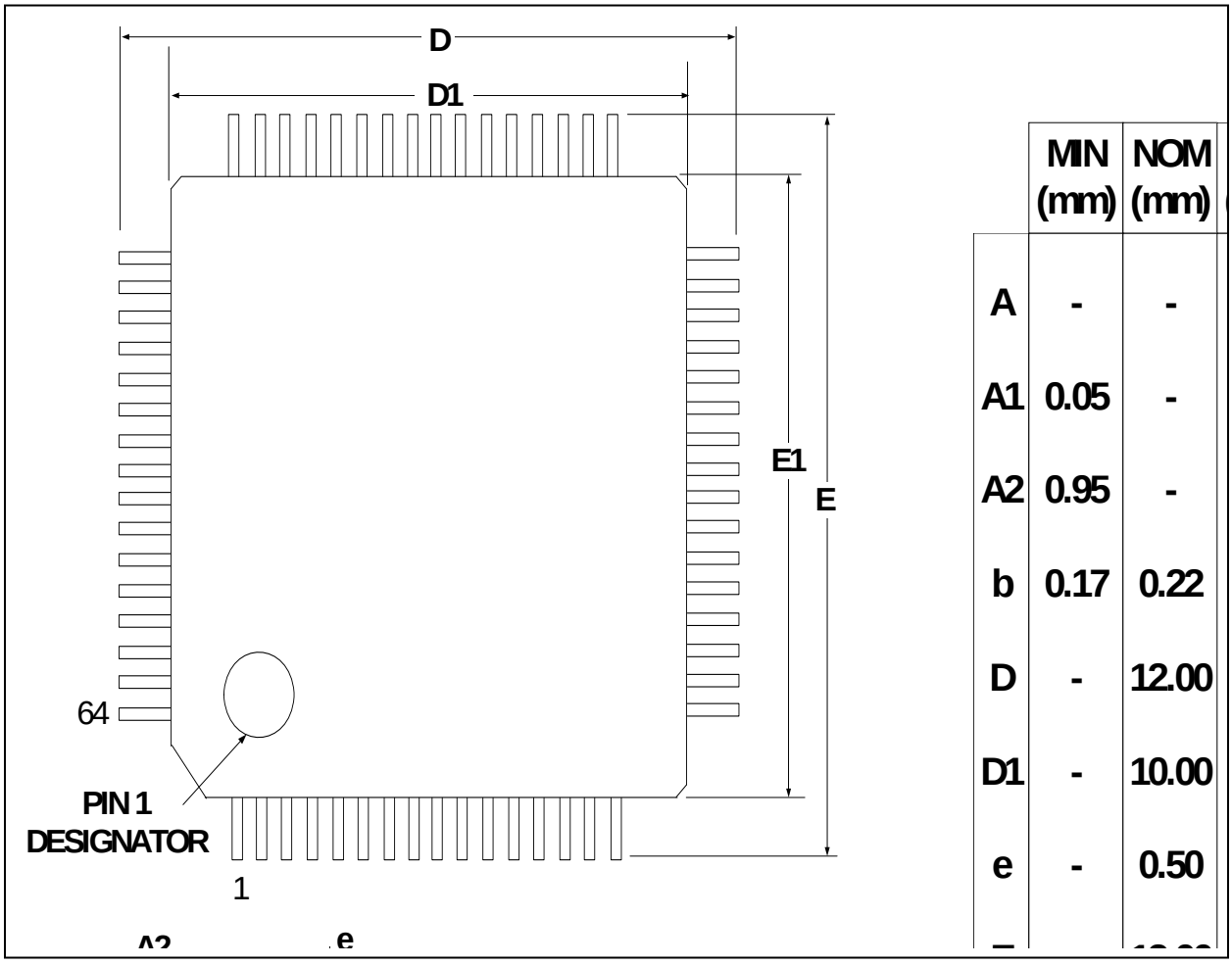
Figure 1.9. PCA Block Diagram



1.6. Serial Ports

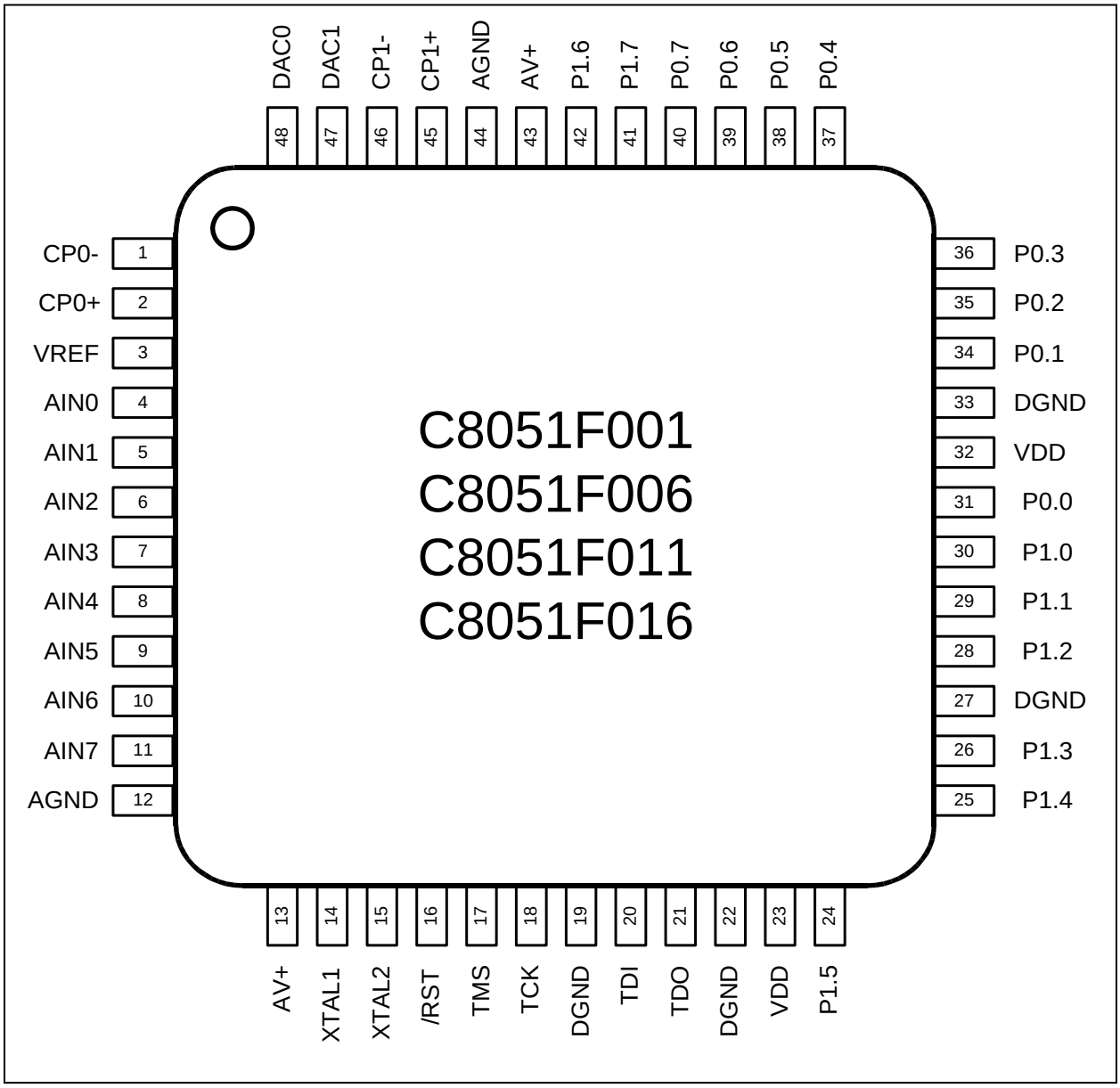
The C8051F000 MCU Family includes a Full-Duplex UART, SPI Bus, and I2C/SMBus. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little intervention by the CPU. The serial buses do not "share" resources such as timers, interrupts, or Port I/O, so any or all of the serial buses may be used together.

Figure 4.2. TQFP-64 Package Drawing



	MIN (mm)	NOM (mm)
A	-	-
A1	0.05	-
A2	0.95	-
b	0.17	0.22
D	-	12.00
D1	-	10.00
e	-	0.50

Figure 4.3. TQFP-48 Pinout Diagram



5.2. ADC Modes of Operation

The ADC uses VREF to determine its full-scale voltage, thus the reference must be properly configured before performing a conversion (see Section 9). The ADC has a maximum conversion speed of 100ksps. The ADC conversion clock is derived from the system clock. Conversion clock speed can be reduced by a factor of 2, 4, 8 or 16 via the ADCSC bits in the ADC0CF Register. This is useful to adjust conversion speed to accommodate different system clock speeds.

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC Start of Conversion Mode bits (ADSTM1, ADSTM0) in ADC0CN. Conversions may be initiated by:

1. Writing a 1 to the ADBUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR;
4. A Timer 2 overflow (i.e. timed continuous conversions).

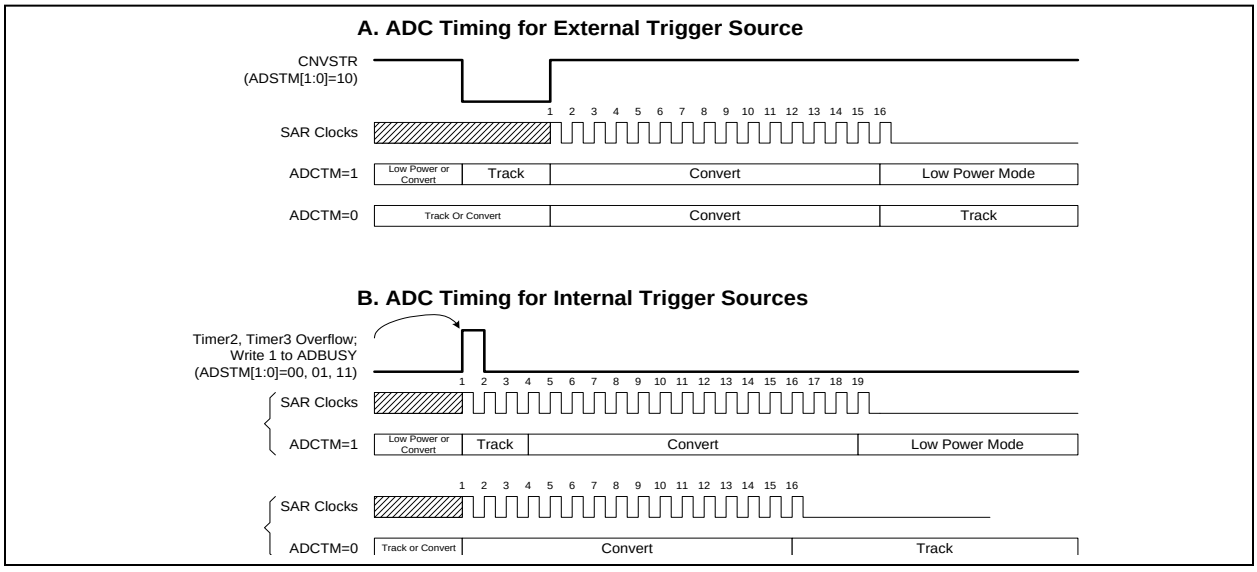
Writing a 1 to ADBUSY provides software control of the ADC whereby conversions are performed "on-demand". During conversion, the ADBUSY bit is set to 1 and restored to 0 when conversion is complete. The falling edge of ADBUSY triggers an interrupt (when enabled) and sets the ADCINT interrupt flag. **Note: When conversions are performed "on-demand", the ADCINT flag, not ADBUSY, should be polled to determine when the conversion has completed.** Converted data is available in the ADC data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 5.9) depending on the programmed state of the ADLJST bit in the ADC0CN register.

The ADCTM bit in register ADC0CN controls the ADC track-and-hold mode. In its default state, the ADC input is continuously tracked, except when a conversion is in progress. Setting ADCTM to 1 allows one of four different low power track-and-hold modes to be specified by states of the ADSTM1-0 bits (also in ADC0CN):

1. Tracking begins with a write of 1 to ADBUSY and lasts for 3 SAR clocks;
2. Tracking starts with an overflow of Timer 3 and lasts for 3 SAR clocks;
3. Tracking is active only when the CNVSTR input is low;
4. Tracking starts with an overflow of Timer 2 and lasts for 3 SAR clocks.

Modes 1, 2 and 4 (above) are useful when the start of conversion is triggered with a software command or when the ADC is operated continuously. Mode 3 is used when the start of conversion is triggered by external hardware. In this case, the track-and-hold is in its low power mode at times when the CNVSTR input is high. Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes.

Figure 5.2. 12-Bit ADC Track and Conversion Example Timing



register configured for accessing the external data memory space. Refer to Section 11 (Flash Memory) for further details.

10.2. 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. There are 256 bytes of internal data memory and 64K bytes of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 10.2.

10.2.1. Program Memory

The CIP-51 has a 64K-byte program memory space. The MCU implements 32896 bytes of this program memory space as in-system, reprogrammable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x807F. Note: 512 bytes (0x7E00 – 0x7FFF) of this memory are reserved for factory use and are not available for user program storage.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section 11 (Flash Memory) for further details.

10.2.2. Data Memory

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may be addressed as bytes or as 128 bit locations accessible with the direct-bit addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F will access the upper 128 bytes of data memory. Figure 10.2 illustrates the data memory organization of the CIP-51.

The C8051F005/06/07/15/16/17 also have 2048 bytes of RAM in the external data memory space of the CIP-51, accessible using the MOVX instruction. Refer to Section 12 (External RAM) for details.

10.2.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in Figure 10.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

10.2.4. Bit Addressable Locations

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

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

```
MOV   C, 22h.3
```

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

13. RESET SOURCES

The reset circuitry of the MCUs allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the CIP-51 halts program execution, forces the external port pins to a known state and initializes the SFRs to their defined reset values. Interrupts and timers are disabled. On exit, the program counter (PC) is reset, and program execution starts at location 0x0000.

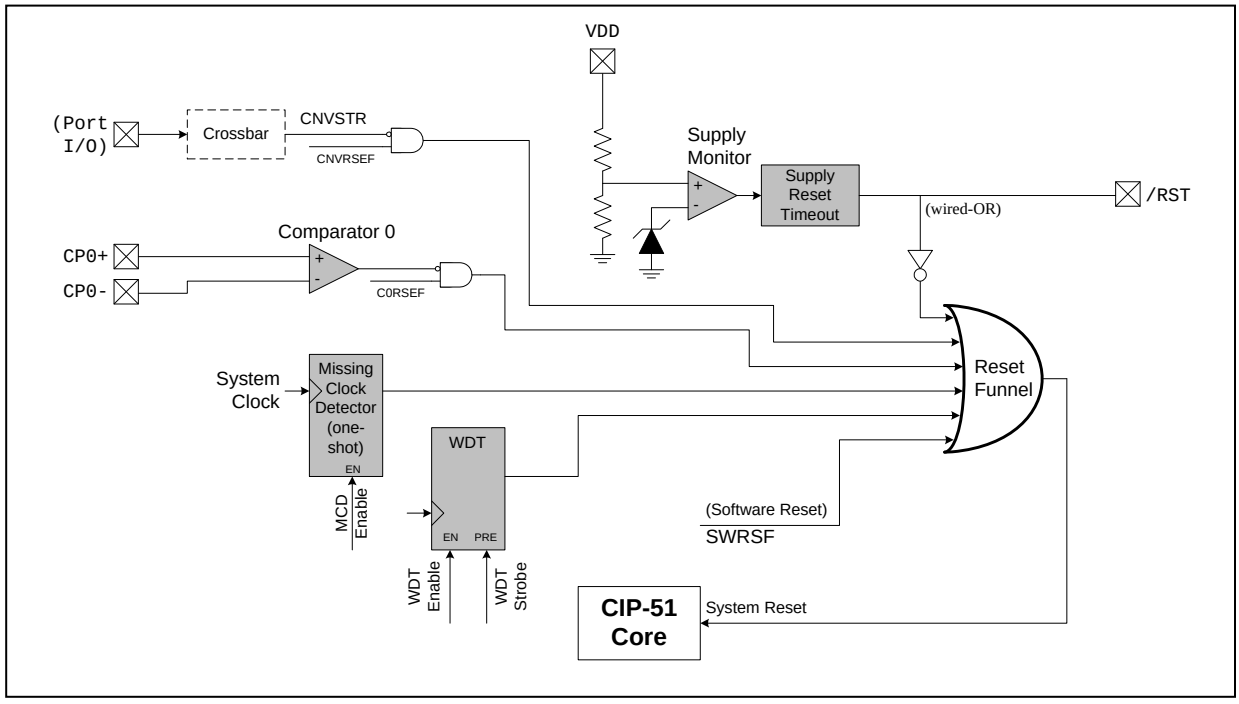
All of the SFRs are reset to predefined values. The reset values of the SFR bits are defined in the SFR detailed descriptions. The contents of internal data memory are not changed during a reset and any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively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

The I/O port latches are reset to 0xFF (all logic ones), activating internal weak pull-ups which take the external I/O pins to a high state. The weak pull-ups are enabled during and after the reset. If the source of reset is from the VDD Monitor or writing a 1 to PORSF, the /RST pin is driven low until the end of the VDD reset timeout.

On exit from the reset state, the MCU uses the internal oscillator running at 2MHz as the system clock by default. Refer to Section 14 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval. (Section 13.8 details the use of the Watchdog Timer.)

There are seven sources for putting the MCU into the reset state: power-on/power-fail, external /RST pin, external CNVSTR signal, software commanded, Comparator 0, Missing Clock Detector, and Watchdog Timer. Each reset source is described below:

Figure 13.1. Reset Sources Diagram



not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an open-drain output that is driving a 0 to avoid unnecessary power dissipation.

The third and final step is to initialize the individual resources selected using the appropriate setup registers. Initialization procedures for the various digital resources may be found in the detailed explanation of each available function. The reset state of each register is shown in the figures that describe each individual register.

Figure 15.1. Port I/O Functional Block Diagram

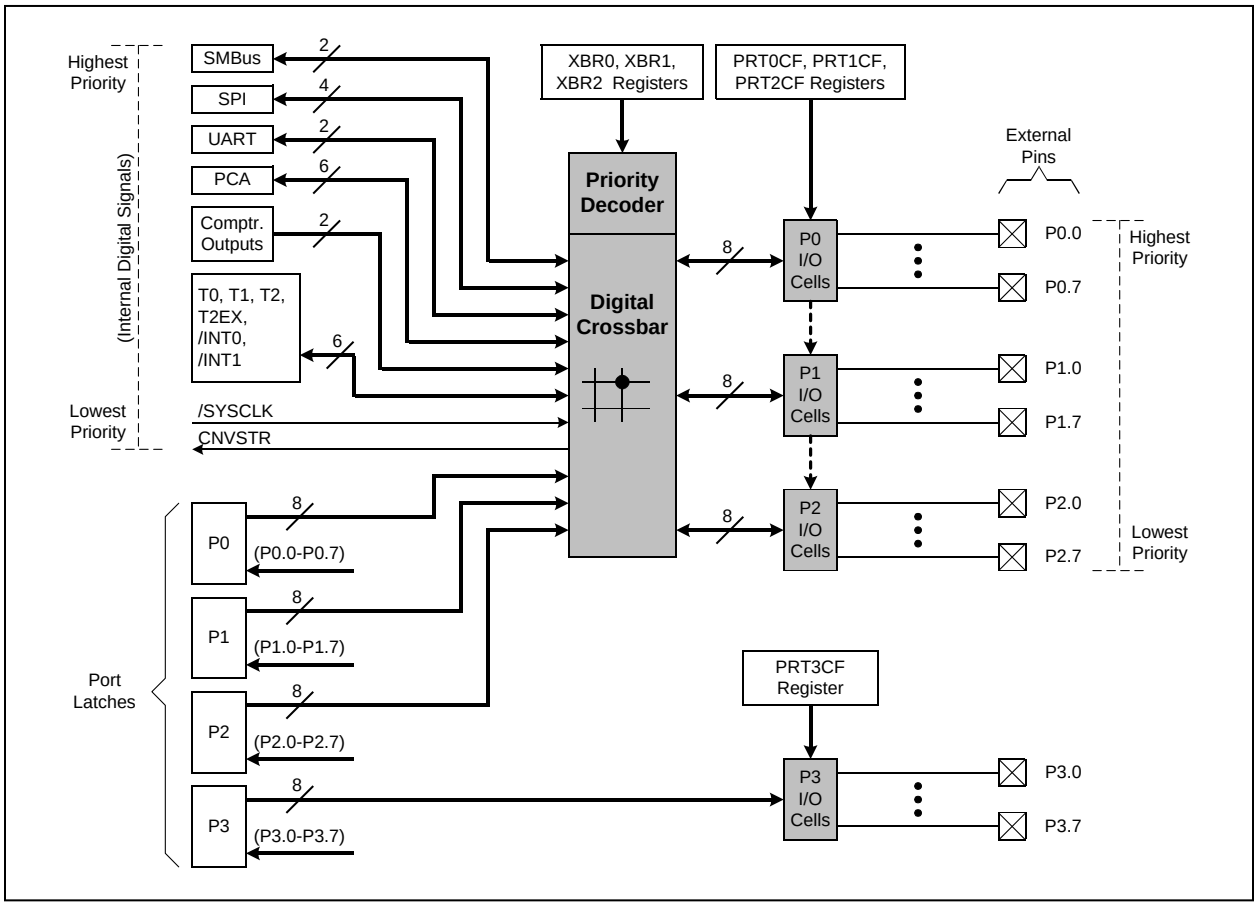


Figure 15.2. Port I/O Cell Block Diagram

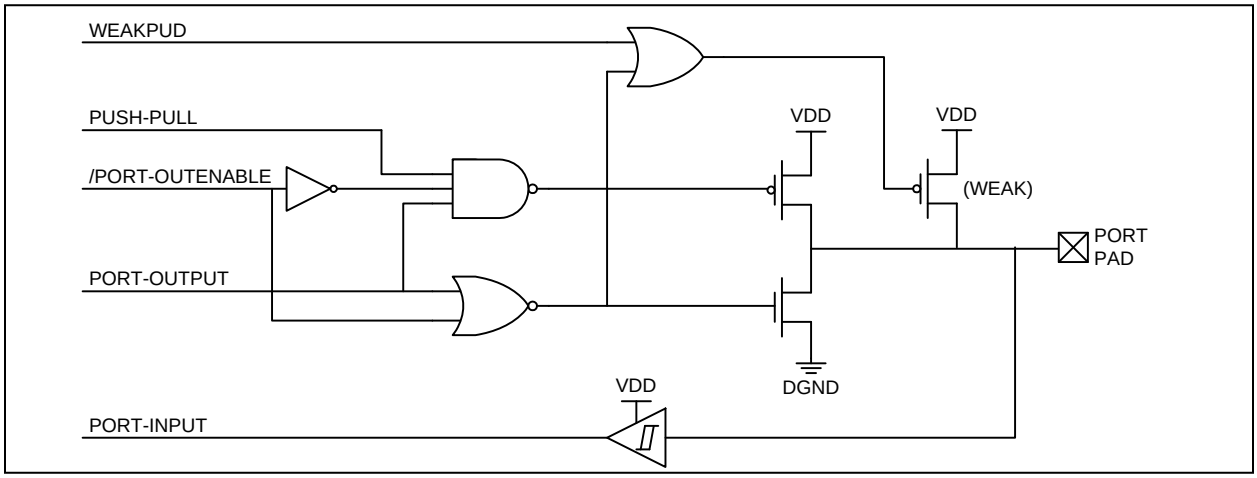


Figure 15.11. P2: Port2 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xA0

Bits7-0: P2.[7:0]
(Write – Output appears on I/O pins per XBR0, XBR1, and XBR2 registers)
0: Logic Low Output.
1: Logic High Output (high-impedance if corresponding PRT2CF.n bit = 0)
(Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P2.n is logic low.
1: P2.n is logic high.

Figure 15.12. PRT2CF: Port2 Configuration Register

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: 0xA6

Bits7-0: PRT2CF.[7:0]: Output Configuration Bits for P2.7-P2.0 (respectively)
0: Corresponding P2.n Output mode is Open-Drain.
1: Corresponding P2.n Output mode is Push-Pull.

16.6.1. Control Register

The SMBus Control register SMB0CN is used to configure and control the SMBus interface. All of the bits in the register can be read or written by software. Two of the control bits are also affected by the SMBus hardware. The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by the hardware when a valid serial interrupt condition occurs. It can only be cleared by software. The Stop flag (STO, SMB0CN.4) is cleared to logic 0 by hardware when a STOP condition is present on the bus.

Setting the ENSMB flag to logic 1 enables the SMBus interface. Clearing the ENSMB flag to logic 0 disables the SMBus interface and removes it from the bus. Momentarily clearing the ENSMB flag and then resetting it to logic 1 will reset a SMBus communication. However, ENSMB should not be used to temporarily remove a device from the bus since the bus state information will be lost. Instead, the Assert Acknowledge (AA) flag should be used to temporarily remove the device from the bus (see description of AA flag below).

Setting the Start flag (STA, SMB0CN.5) to logic 1 will put the SMBus in a master mode. If the bus is free, the SMBus hardware will generate a START condition. If the bus is not free, the SMBus hardware waits for a STOP condition to free the bus and then generates a START condition after a 5µs delay per the SMB0CR value. (In accordance with the SMBus protocol, the SMBus interface also considers the bus free if the bus is idle for 50µs and no STOP condition was recognized.) If STA is set to logic 1 while the SMBus is in master mode and one or more bytes have been transferred, a repeated START condition will be generated. To ensure proper operation, the STO flag should be explicitly cleared before setting STA to a logic 1.

When the Stop flag (STO, SMB0CN.4) is set to logic 1 while the SMBus interface is in master mode, the hardware generates a STOP condition on the SMBus. In a slave mode, the STO flag may be used to recover from an error condition. In this case, a STOP condition is not generated on the SMBus, but the SMBus hardware behaves as if a STOP condition has been received and enters the “not addressed” slave receiver mode. The SMBus hardware automatically clears the STO flag to logic 0 when a STOP condition is detected on the bus.

The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by hardware when the SMBus interface enters one of 27 possible states. If interrupts are enabled for the SMBus interface, an interrupt request is generated when the SI flag is set. The SI flag must be cleared by software. While SI is set to logic 1, the clock-low period of the serial clock will be stretched and the serial transfer is suspended.

The Assert Acknowledge flag (AA, SMB0CN.2) is used to set the level of the SDA line during the acknowledge clock cycle on the SCL line. Setting the AA flag to logic 1 will cause an ACKNOWLEDGE (low level on SDA) to be sent during the acknowledge cycle if the device has been addressed. Setting the AA flag to logic 0 will cause a NOT ACKNOWLEDGE (high level on SDA) to be sent during acknowledge cycle. After the transmission of a byte in slave mode, the slave can be temporarily removed from the bus by clearing the AA flag. The slave’s own address and general call address will be ignored. To resume operation on the bus, the AA flag must be reset to logic 1 to allow the slave’s address to be recognized.

Setting the SMBus Free Timer Enable bit (FTE, SMB0CN.1) to logic 1 enables the SMBus Free Timeout feature. If SCL and SDA remain high for the SMBus Free Timeout given in the SMBus Clock Rate Register (Figure 16.5), the bus will be considered free and a Start will be generated if pending. The bus free period should be greater than 50µs.

Setting the SMBus timeout enable bit (TOE, SMB0CN.0) to logic 1 enables Timer 3 to count up when the SCL line is low and Timer 3 is enabled. If Timer 3 overflows, a Timer 3 interrupt will be generated, which will alert the CPU that a SMBus SCL low timeout has occurred.

Figure 17.7. SPI0CKR: SPI Clock Rate Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7-0: SCR7-SCR0: SPI Clock Rate
These bits determine the frequency of the SCK output when the SPI module is configured for master mode operation. The SCK clock frequency is a divided down version of the system clock, and is given in the following equations:

$f_{SCK} = 0.5 * f_{SYSCLK} / (SPI0CKR + 1)$, for $0 \leq SPI0CKR \leq 255$,

Figure 17.8. SPI0DAT: SPI Data Register

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: 0x9B

Bits7-0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI data. Writing data to SPI0DAT places the data immediately into the shift register and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

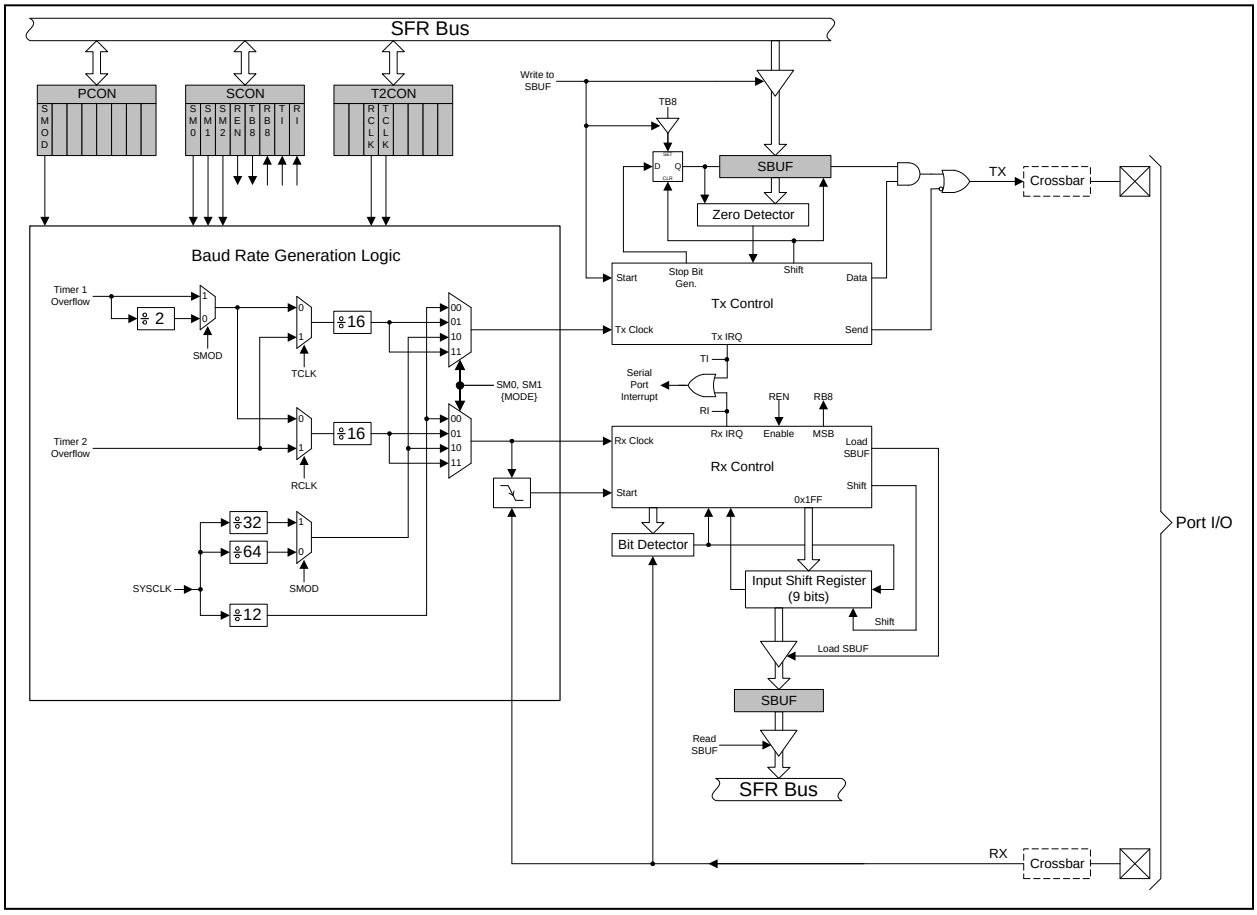
18. UART

The UART is a serial port capable of asynchronous transmission. The UART can function in full duplex mode. In all modes, receive data is buffered in a holding register. This allows the UART to start reception of a second incoming data byte before software has finished reading the previous data byte.

The UART has an associated Serial Control Register (SCON) and a Serial Data Buffer (SBUF) in the SFRs. The single SBUF location provides access to both transmit and receive registers. Reads access the Receive register and writes access the Transmit register automatically.

The UART is capable of generating interrupts if enabled. The UART has two sources of interrupts: a Transmit Interrupt flag, TI (SCON.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI (SCON.0) set when reception of a data byte is complete. The UART interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software. This allows software to determine the cause of the UART interrupt (transmit complete or receive complete).

Figure 18.1. UART Block Diagram



Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is 0 or the input signal /INT0 is logic-level one. Setting GATE0 to logic 1 allows the timer to be controlled by the external input signal /INT0, facilitating pulse width measurements.

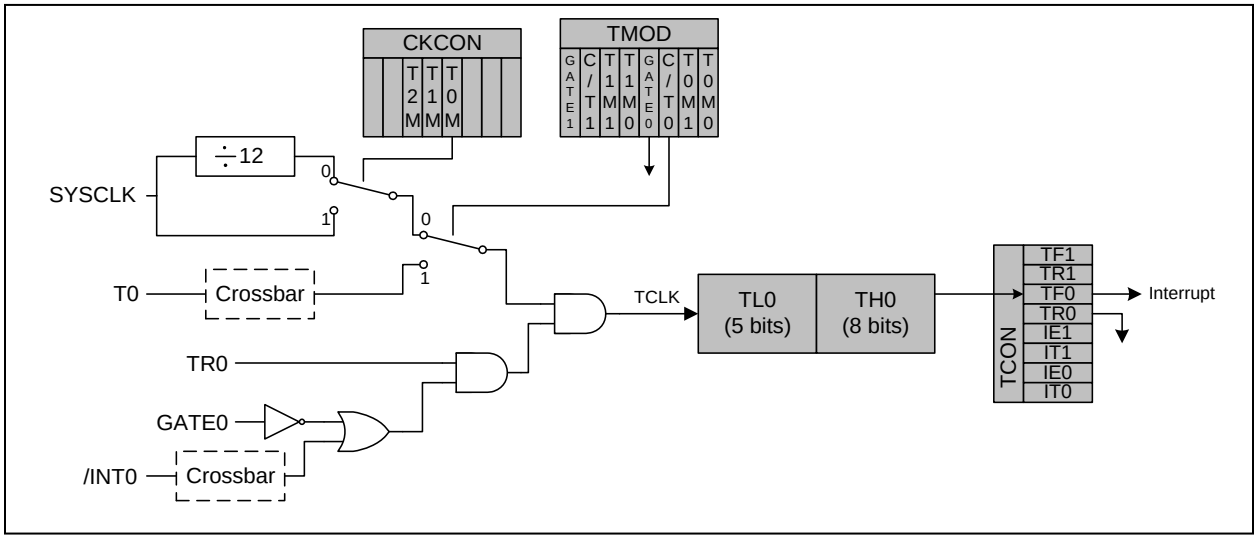
TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don't Care

Setting TR0 does not reset the timer register. The timer register should be initialized to the desired value before enabling the timer.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0.

Figure 19.1. T0 Mode 0 Block Diagram



19.1.2. Mode 1: 16-bit Counter/Timer

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

Figure 19.4. TCON: Timer Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0x88

Bit7: TF1: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: TF0: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: IE1: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. This flag is the inverse of the /INT1 input signal's logic level when IT1 = 0.

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured /INT1 signal will detect falling edge or active-low level-sensitive interrupts.
0: /INT1 is level triggered.
1: /INT1 is edge triggered.

Bit1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. This flag is the inverse of the /INT0 input signal's logic level when IT0 = 0.

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured /INT0 signal will detect falling edge or active-low level-sensitive interrupts.
0: /INT0 is level triggered.
1: /INT0 is edge triggered.

19.2.3. Mode 2: Baud Rate Generator

Timer 2 can be used as a baud rate generator for the serial port (UART) when the UART is operated in modes 1 or 3 (refer to Section 18.1 for more information on UART operational modes). In Baud Rate Generator mode, Timer 2 works similarly to the auto-reload mode. On overflow, the 16-bit value held in the two capture registers (RCAP2H, RCAP2L) is automatically loaded into the counter/timer register. However, the TF2 overflow flag is not set and no interrupt is generated. Instead, the overflow event is used as the input to the UART's shift clock. Timer 2 overflows can be used to generate baud rates for transmit and/or receive independently.

The Baud Rate Generator mode is selected by setting RCLK (T2CON.5) and/or TCLK (T2CON.4) to logic one. When RCLK or TCLK is set to logic 1, Timer 2 operates in the auto-reload mode regardless of the state of the CP/RL2 bit. The baud rate for the UART, when operating in mode 1 or 3, is determined by the Timer 2 overflow rate:

Baud Rate = Timer 2 Overflow Rate / 16.

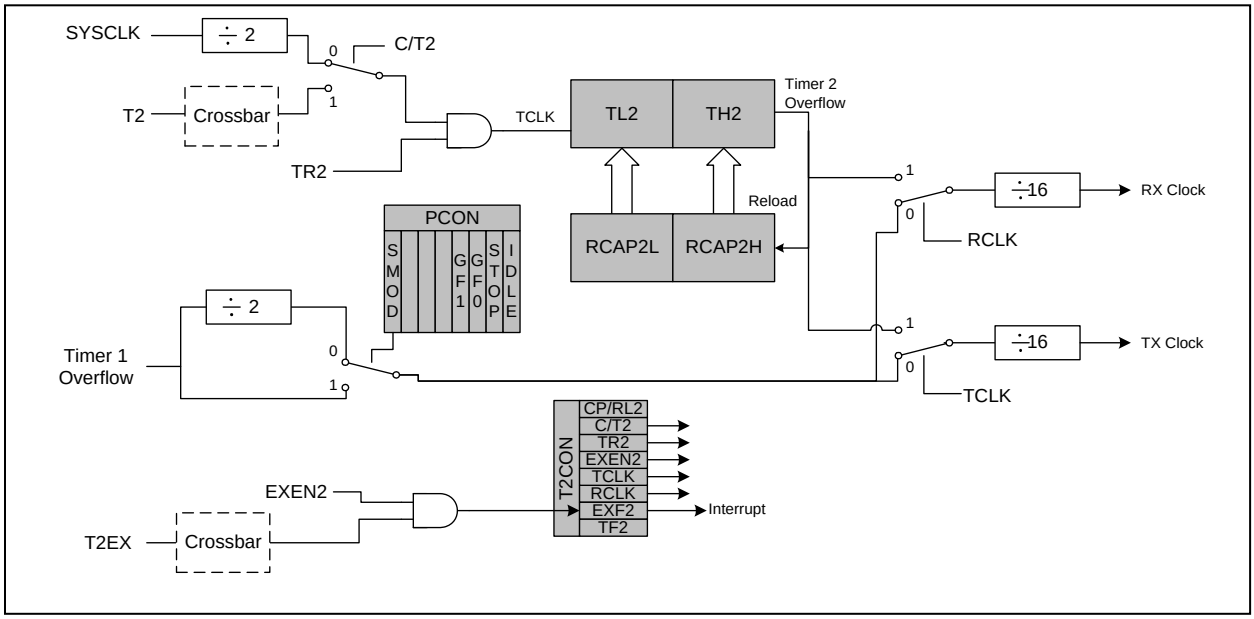
Note, in all other modes, the timebase for the timer is the system clock divided by one or twelve as selected by the T2M bit in CKCON. However, in Baud Rate Generator mode, the timebase is the system clock divided by two. No other divisor selection is possible. If a different time base is required, setting the C/T2 bit to logic 1 will allow the timebase to be derived from the external input pin T2. In this case, the baud rate for the UART is calculated as:

*Baud Rate = FCLK / [32 * (65536 – [RCAP2H:RCAP2L])]*

Where FCLK is the frequency of the signal supplied to T2 and [RCAP2H:RCAP2L] is the 16-bit value held in the capture registers.

As explained above, in Baud Rate Generator mode, Timer 2 does not set the TF2 overflow flag and therefore cannot generate an interrupt. However, if EXEN2 is set to logic 1, a high-to-low transition on the T2EX input pin will set the EXF2 flag and a Timer 2 interrupt will occur if enabled. Therefore, the T2EX input may be used as an additional external interrupt source.

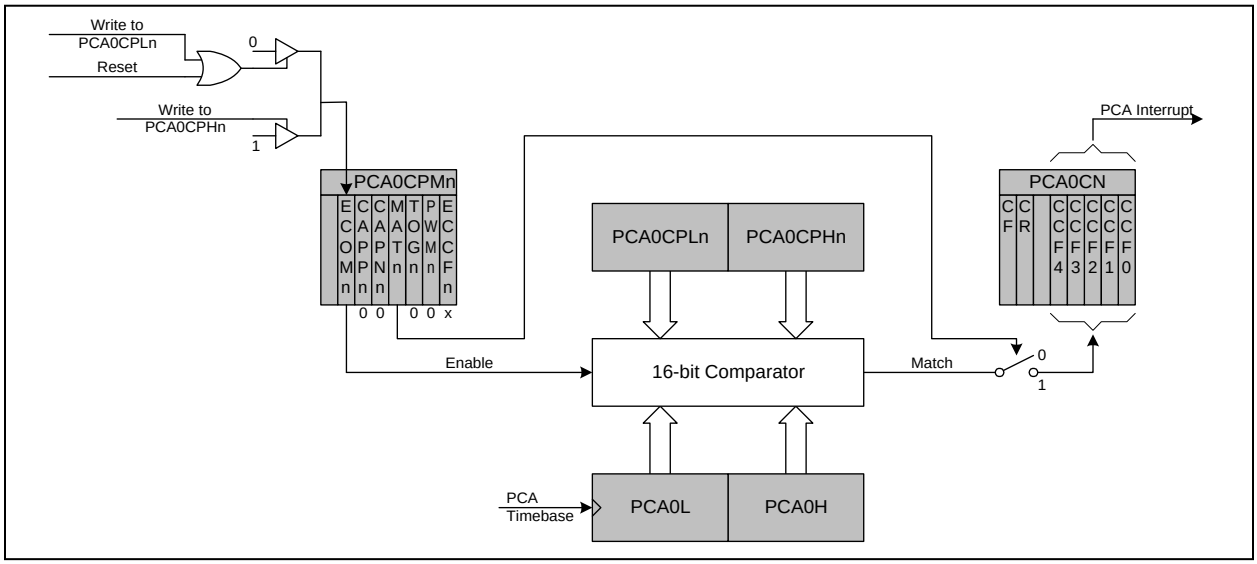
Figure 19.13. T2 Mode 2 Block Diagram



20.1.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

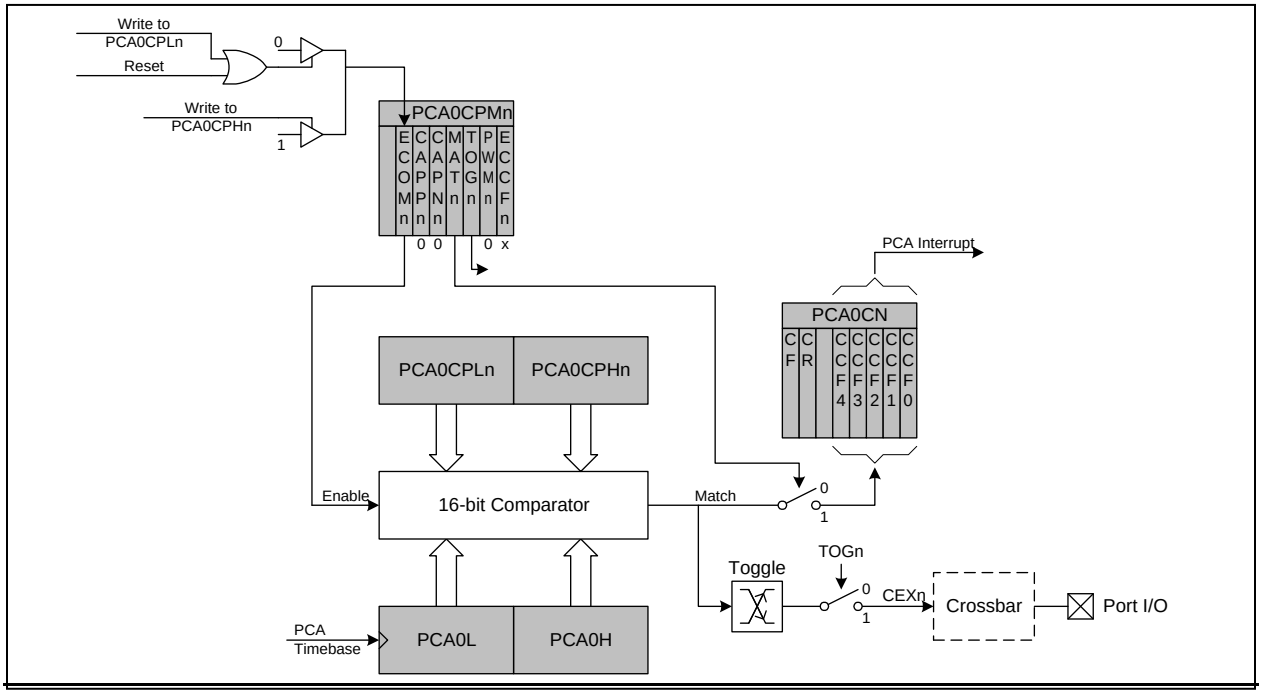
Figure 20.4. PCA Software Timer Mode Diagram



20.1.3. High Speed Output Mode

In this mode, each time a match occurs between the PCA Timer Counter and a module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn) the logic level on the module's associated CEXn pin will toggle. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Figure 20.5. PCA High Speed Output Mode Diagram



20.3. Register Descriptions for PCA

The system device may implement one or more Programmable Counter Arrays. Following are detailed descriptions of the special function registers related to the operation of the PCA. The CIP-51 System Controller section of the datasheet provides additional information on the SFRs and their use.

Figure 20.8. PCA0CN: PCA Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xD8

Bit7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the CF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: CR: PCA Counter/Timer Run Control.
This bit enables/disables the PCA Counter/Timer.
0: PCA Counter/Timer disabled.
1: PCA Counter/Timer enabled.

Bit5: UNUSED. Read = 0, Write = don't care.

Bit4: CCF4: PCA Module 4 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit3: CCF3: PCA Module 3 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: CCF2: PCA Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: CCF1: PCA Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: CCF0: PCA Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

21.1.1. EXTEST Instruction

The EXTEST instruction is accessed via the IR. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature. All inputs to on-chip logic are set to one.

21.1.2. SAMPLE Instruction

The SAMPLE instruction is accessed via the IR. The Boundary DR provides observability and presetting of the scan-path latches.

21.1.3. BYPASS Instruction

The BYPASS instruction is accessed via the IR. It provides access to the standard 1-bit JTAG Bypass data register.

21.1.4. IDCODE Instruction

The IDCODE instruction is accessed via the IR. It provides access to the 32-bit Device ID register.

Figure 21.2. DEVICEID: JTAG Device ID Register

Version	Part Number	Manufacturer ID	1	Reset Value (Varies)
Bit31 Bit28	Bit27 Bit12	Bit11 Bit1	Bit0	

Version = 0000b (Revision A) or
= 0001b (Revision B)

Part Number = 0000 0000 0000 0000b or
= 0000 0000 0000 0010b

Manufacturer ID = 0010 0100 001b (Silicon Laboratories)