

Welcome to [E-XFL.COM](https://www.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	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	8
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 4x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f007

1.4. Programmable Digital I/O and Crossbar

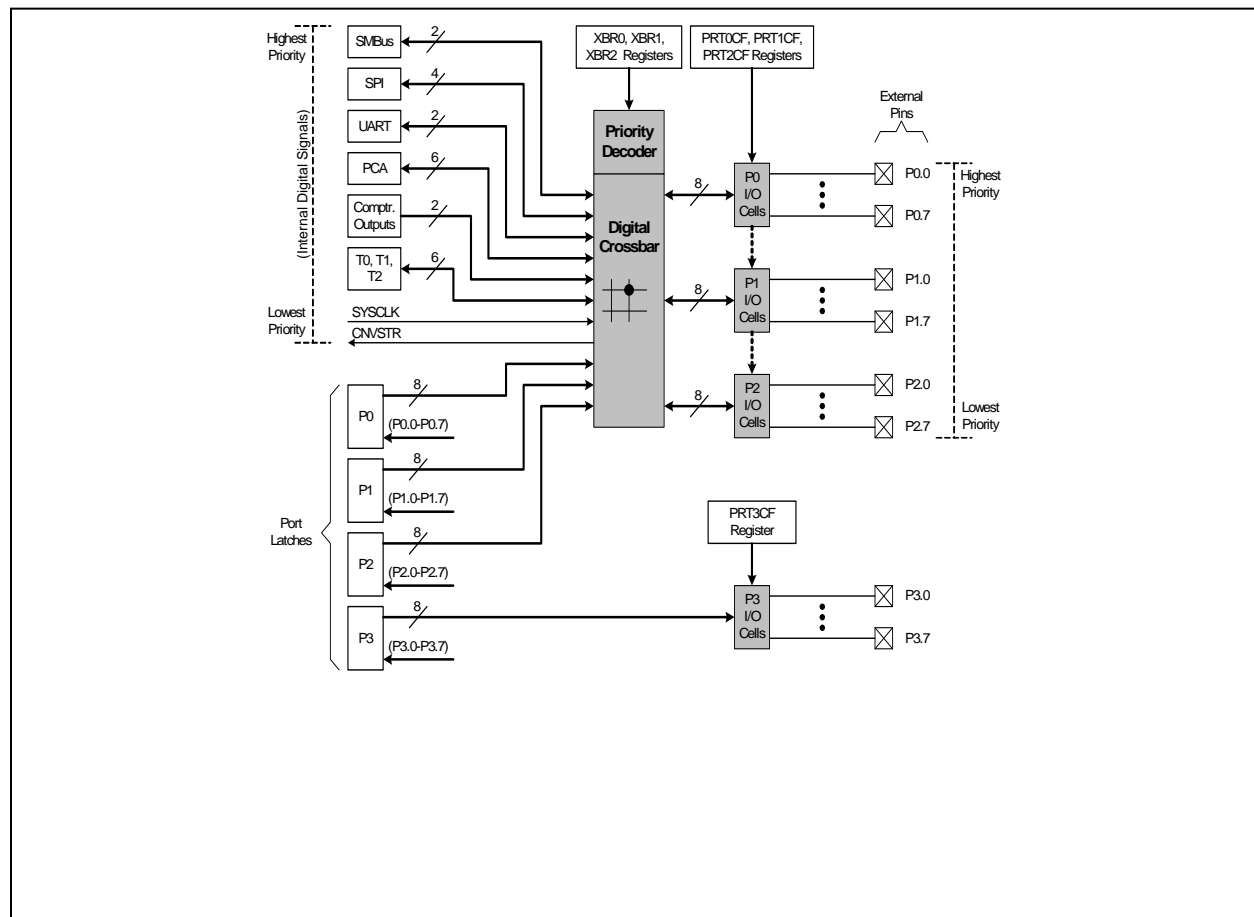
The standard 8051 Ports (0, 1, 2, and 3) are available on the MCUs. All four ports are pinned out on the F000/05/10/15. Ports 0 and 1 are pinned out on the F001/06/11/16, and only Port 0 is pinned out on the F002/07/12/17. The Ports not pinned out are still available for software use as general purpose registers. The Port I/O behave like the standard 8051 with a few enhancements.

Each Port I/O pin can be configured as either a push-pull or open-drain output. Also, the “weak pull-ups” which are normally fixed on an 8051 can be globally disabled, providing additional power saving capabilities for low power applications.

Perhaps the most unique enhancement is the Digital Crossbar. This is essentially a large digital switching network that allows mapping of internal digital system resources to Port I/O pins on P0, P1, and P2. (See Figure 1.8.) Unlike microcontrollers with standard multiplexed digital I/O, all combinations of functions are supported.

The on-board counter/timers, serial buses, HW interrupts, ADC Start of Conversion input, comparator outputs, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for his particular application.

Figure 1.8. Digital Crossbar Diagram



1.7. Analog to Digital Converter

The C8051F000/1/2/5/6/7 has an on-chip 12-bit SAR ADC with a 9-channel input multiplexer and programmable gain amplifier. With a maximum throughput of 100ksps, the ADC offers true 12-bit accuracy with an INL of ± 1 LSB. The ADC in the C8051F010/1/2/5/6/7 is similar, but with 10-bit resolution. Each ADC has a maximum throughput of 100ksps. Each ADC has an INL of ± 1 LSB, offering true 12-bit accuracy with the C8051F00x, and true 10-bit accuracy with the C8051F01x. There is also an on-board 15ppm voltage reference, or an external reference may be used via the VREF pin.

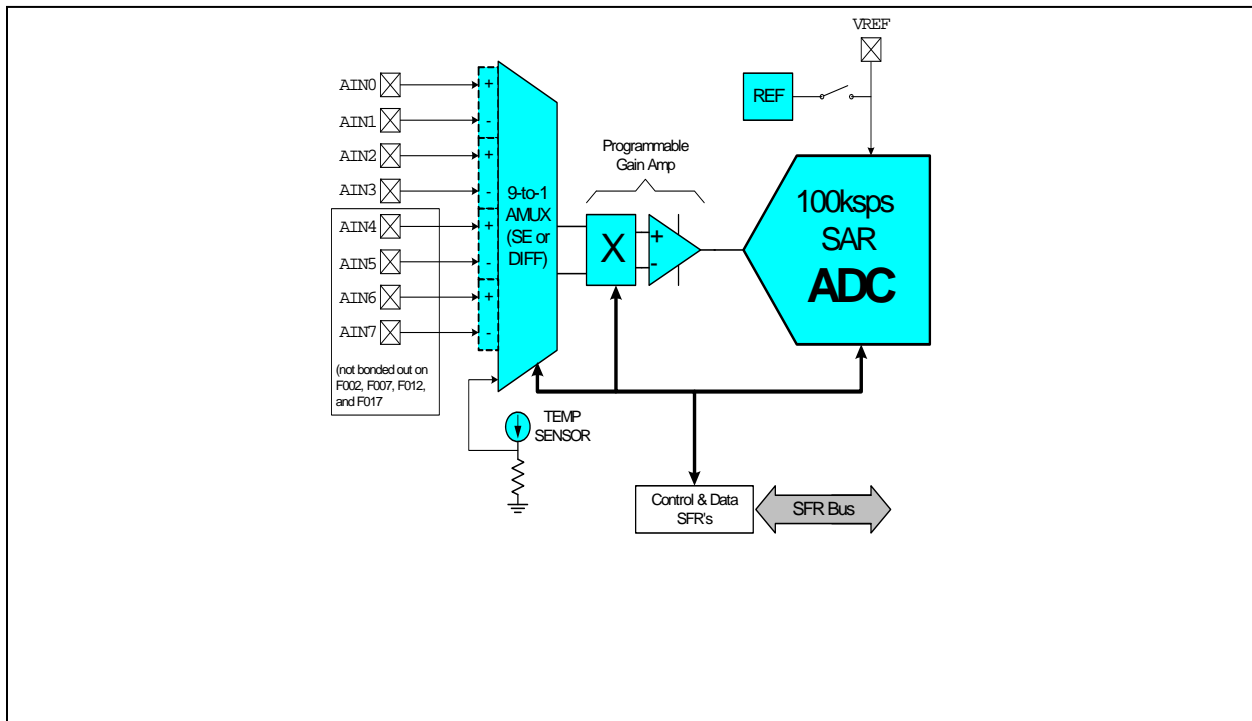
The ADC is under full control of the CIP-51 microcontroller via the Special Function Registers. One input channel is tied to an internal temperature sensor, while the other eight channels are available externally. Each pair of the eight external input channels can be configured as either two single-ended inputs or a single differential input. The system controller can also put the ADC into shutdown to save power.

A programmable gain amplifier follows the analog multiplexer. The gain can be set in software from 0.5 to 16 in powers of 2. The gain stage can be especially useful when different ADC input channels have widely varied input voltage signals, or when it is necessary to “zoom in” on a signal with a large DC offset (in differential mode, a DAC could be used to provide the DC offset).

Conversions can be started in four ways; a software command, an overflow on Timer 2, an overflow on Timer 3, or an external signal input. This flexibility allows the start of conversion to be triggered by software events, external HW signals, or convert continuously. A completed conversion causes an interrupt, or a status bit can be polled in software to determine the end of conversion. The resulting 10 or 12-bit data word is latched into two SFRs upon completion of a conversion. The data can be right or left justified in these registers under software control.

Compare registers for the ADC data can be configured to interrupt the controller when ADC data is within a specified window. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within the specified window.

Figure 1.10. ADC Diagram



1.8. Comparators and DACs

The C8051F000 MCU Family has two 12-bit DACs and two comparators on chip (the second comparator, CP1, is not bonded out on the F002, F007, F012, and F017). The MCU data and control interface to each comparator and DAC is via the Special Function Registers. The MCU can place any DAC or comparator in low power shutdown mode.

The comparators have software programmable hysteresis. Each comparator can generate an interrupt on its rising edge, falling edge, or both. The comparators' output state can also be polled in software. These interrupts are capable of waking up the MCU from idle mode. The comparator outputs can be programmed to appear on the Port I/O pins via the Crossbar.

The DACs are voltage output mode and use the same voltage reference as the ADC. They are especially useful as references for the comparators or offsets for the differential inputs of the ADC.

Figure 1.11. Comparator and DAC Diagram

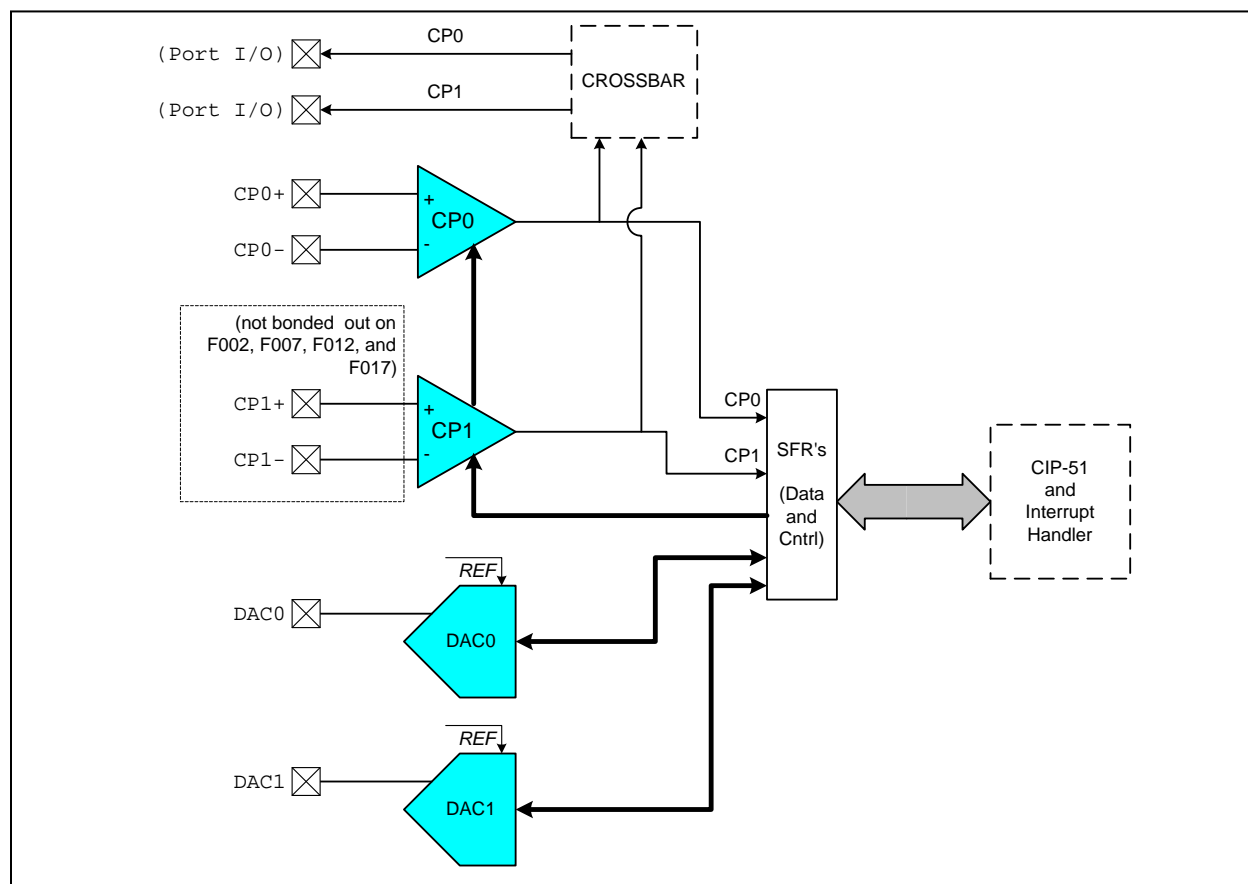


Figure 8.2. Comparator Hysteresis Plot

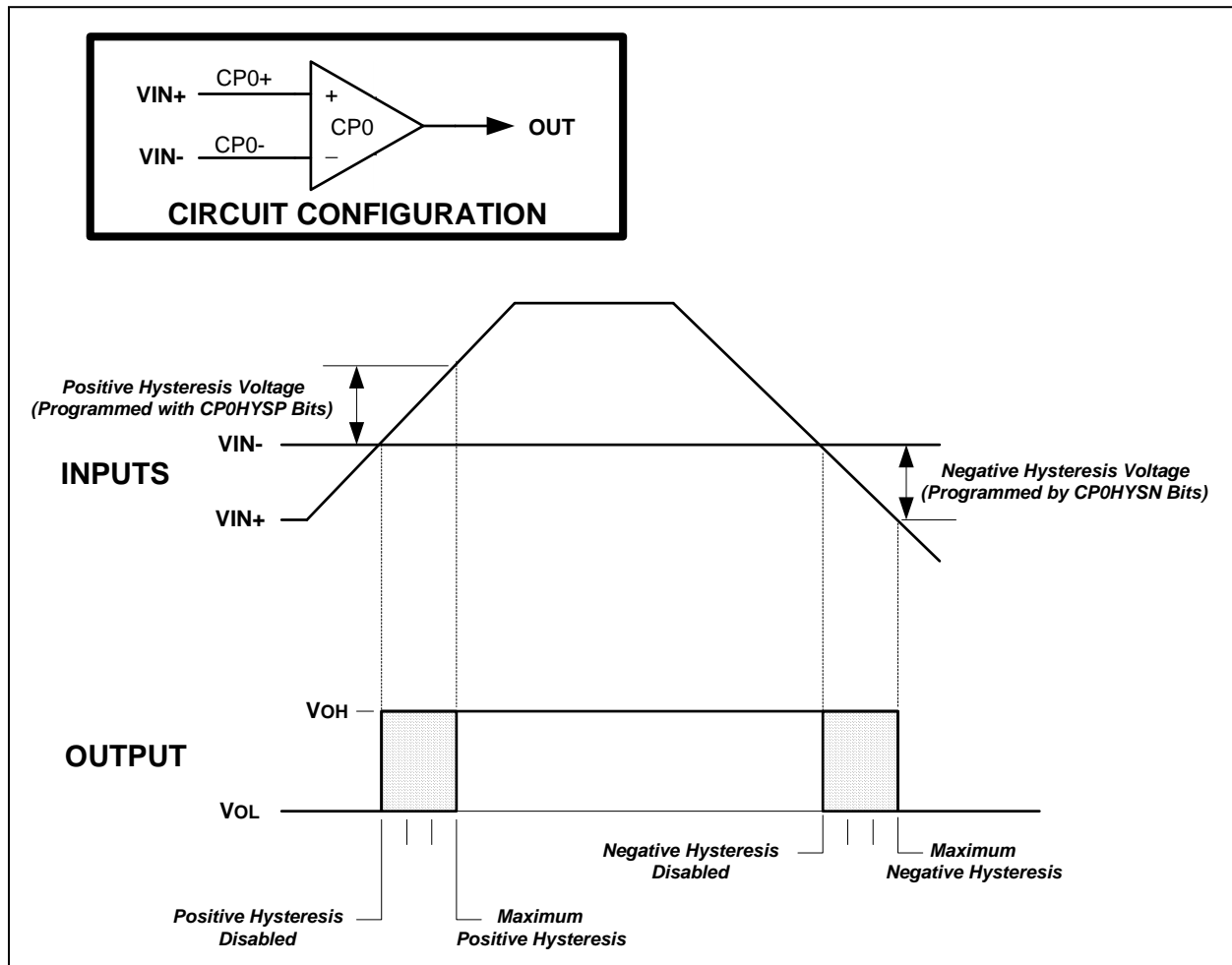


Table 8.1. Comparator Electrical Characteristics

VDD = 3.0V, AV+ = 3.0V, -40°C to +85°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Response Time1	(CP+) – (CP-) = 100mV (Note 1)		4		μs
Response Time2	(CP+) – (CP-) = 10mV (Note 1)		12		μs
Common Mode Rejection Ratio			1.5	4	mV/V
Positive Hysteresis1	CPnHYP1-0 = 00		0	1	mV
Positive Hysteresis2	CPnHYP1-0 = 01	2	4.5	7	mV
Positive Hysteresis3	CPnHYP1-0 = 10	4	9	13	mV
Positive Hysteresis4	CPnHYP1-0 = 11	10	17	25	mV
Negative Hysteresis1	CPnHYN1-0 = 00		0	1	mV
Negative Hysteresis2	CPnHYN1-0 = 01	2	4.5	7	mV
Negative Hysteresis3	CPnHYN1-0 = 10	4	9	13	mV
Negative Hysteresis4	CPnHYN1-0 = 11	10	17	25	mV
Inverting or Non-inverting Input Voltage Range		-0.25		(AV+) + 0.25	V
Input Capacitance			7		pF
Input Bias Current		-5	0.001	+5	nA
Input Offset Voltage		-10		+10	mV
POWER SUPPLY					
Power-up Time	CPnEN from 0 to 1		20		μs
Power Supply Rejection			0.1	1	mV/V
Supply Current	Operating Mode (each comparator) at DC		1.5	10	μA

Note 1: CPnHYP1-0 = CPnHYN1-0 = 00.

Table 10.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
ARITHMETIC OPERATIONS			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add indirect RAM to A	1	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add indirect RAM to A with carry	1	2
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal Adjust A	1	1
LOGICAL OPERATIONS			
ANL A,Rn	AND Register to A	1	1
ANL A,direct	AND direct byte to A	2	2
ANL A,@Ri	AND indirect RAM to A	1	2
ANL A,#data	AND immediate to A	2	2
ANL direct,A	AND A to direct byte	2	2
ANL direct,#data	AND immediate to direct byte	3	3
ORL A,Rn	OR Register to A	1	1
ORL A,direct	OR direct byte to A	2	2
ORL A,@Ri	OR indirect RAM to A	1	2
ORL A,#data	OR immediate to A	2	2
ORL direct,A	OR A to direct byte	2	2
ORL direct,#data	OR immediate to direct byte	3	3
XRL A,Rn	Exclusive-OR Register to A	1	1
XRL A,direct	Exclusive-OR direct byte to A	2	2
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL direct,A	Exclusive-OR A to direct byte	2	2
XRL direct,#data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1

Figure 10.6. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xD0

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation results in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag.
This bit is set when the last arithmetic operation results in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4-3: RS1-RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00-0x07
0	1	1	0x08-0x0F
1	0	2	0x10-0x17
1	1	3	0x18-0x1F

Note: Any instruction which changes the RS1-RS0 bits must not be immediately followed by the “MOV Rn, A” instruction.

Bit2: 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.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
(Read only)
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

Figure 10.7. 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

Bits 7-0: ACC: Accumulator
This register is the accumulator for arithmetic operations.

Figure 10.8. 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

Bits 7-0: B: B Register
This register serves as a second accumulator for certain arithmetic operations.

10.4. INTERRUPT HANDLER

The CIP-51 includes an extended interrupt system supporting a total of 22 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs 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-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.

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.

10.4.1. MCU Interrupt Sources and Vectors

The MCUs allocate 12 interrupt sources to on-chip peripherals. Up to 10 additional external interrupt sources are available depending on the I/O pin configuration of the device. 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 10.4. 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).

10.4.2. External Interrupts

Two of the external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag 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 follows the state of the external interrupt's input pin. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

The remaining four external interrupts (External Interrupts 4-7) are active-low, edge-sensitive inputs. The interrupt-pending flags for these interrupts are in the Port 1 Interrupt Flag Register shown in Figure 15.10.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to sleep for longer than the MCD timeout of 100µsec.

Figure 10.15. PCON: Power Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SMOD	GF4	GF3	GF2	GF1	GF0	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x87
<p>Bit7: SMOD: Serial Port Baud Rate Doubler Enable. 0: Serial Port baud rate is that defined by Serial Port Mode in SCON. 1: Serial Port baud rate is double that defined by Serial Port Mode in SCON.</p> <p>Bits6-2: GF4-GF0: General Purpose Flags 4-0. These are general purpose flags for use under software control.</p> <p>Bit1: STOP: Stop Mode Select. Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0. 1: Goes into power down mode. (Turns off internal oscillator).</p> <p>Bit0: IDLE: Idle Mode Select. Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0. 1: Goes into idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)</p>								

Figure 14.2. OSCICN: Internal Oscillator Control Register

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
MSCLKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bit7: MSCLKE: Missing Clock Enable Bit
 0: Missing Clock Detector Disabled
 1: Missing Clock Detector Enabled; triggers a reset if a missing clock is detected
 Bits6-5: UNUSED. Read = 00b, Write = don't care
 Bit4: IFRDY: Internal Oscillator Frequency Ready Flag
 0: Internal Oscillator Frequency not running at speed specified by the IFCN bits.
 1: Internal Oscillator Frequency running at speed specified by the IFCN bits.
 Bit3: CLKSL: System Clock Source Select Bit
 0: Uses Internal Oscillator as System Clock.
 1: Uses External Oscillator as System Clock.
 Bit2: IOSCEN: Internal Oscillator Enable Bit
 0: Internal Oscillator Disabled
 1: Internal Oscillator Enabled
 Bits1-0: IFCN1-0: Internal Oscillator Frequency Control Bits
 00: Internal Oscillator typical frequency is 2MHz.
 01: Internal Oscillator typical frequency is 4MHz.
 10: Internal Oscillator typical frequency is 8MHz.
 11: Internal Oscillator typical frequency is 16MHz.

Table 14.1. Internal Oscillator Electrical Characteristics

-40°C to +85°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Internal Oscillator Frequency	OSCICN.[1:0] = 00	1.5	2	2.4	MHz
	OSCICN.[1:0] = 01	3.1	4	4.8	
	OSCICN.[1:0] = 10	6.2	8	9.6	
	OSCICN.[1:0] = 11	12.3	16	19.2	
Internal Oscillator Current Consumption (from VDD)	OSCICN.2 = 1		200		μA
Internal Oscillator Temperature Stability			4		ppm/°C
Internal Oscillator Power Supply (VDD) Stability			6.4		%/V

15. PORT INPUT/OUTPUT

The MCUs have a wide array of digital resources, which are available through four digital I/O ports, P0, P1, P2 and P3. Each of the pins on Ports 0, 1, and 2 can be defined as either its corresponding port I/O or one of the internal digital resources assigned as shown in Figure 15.1. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins available on the selected package (the C8051F000/05/10/15 have all four ports pinned out, the F001/06/11/16 have P0 and P1, and the F002/07/12/17 have P0). 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 in the corresponding Port latch regardless of the Crossbar settings).

The CrossBar assigns the selected internal digital resources to the I/O pins based on the Priority Decode Table 15.1. The registers XBR0, XBR1, and XBR2, defined in Figure 15.3, Figure 15.4, and Figure 15.5 are used to select an internal digital function or let an I/O pin default to being a Port I/O. The crossbar functions identically for each MCU, with the caveat that P2 is not pinned out on the F001/06/11/16, and both P1 and P2 are not pinned out on the F002/07/12/17. Digital resources assigned to port pins that are not pinned out cannot be accessed.

All Port I/Os are 5V tolerant (Refer to Figure 15.2 for the port cell circuit.) The Port I/O cells are configured as either push-pull or open-drain in the Port Configuration Registers (PRT0CF, PRT1CF, PRT2CF, PRT3CF). Complete Electrical Specifications for Port I/O are given in Table 15.2.

15.1. Priority Cross Bar Decoder

One of the design goals of this MCU family was to make the entire palette of digital resources available to the designer even on reduced pin count packages. The Priority CrossBar Decoder provides an elegant solution to the problem of connecting the internal digital resources to the physical I/O pins.

The Priority CrossBar Decode (Table 15.1) assigns a priority to each I/O function, starting at the top with the SMBus. As the table illustrates, when selected, its two signals will be assigned to Pin 0 and 1 of I/O Port 0. The decoder always fills I/O bits from LSB to MSB starting with Port 0, then Port 1, finishing if necessary with Port 2. If you choose not to use a resource, the next function down on the table will fill the priority slot. In this way it is possible to choose only the functions required by the design, making full use of the available I/O pins. Also, any extra Port I/O are grouped together for more convenient use in application code.

Registers XBR0, XBR1 and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. It is important to understand that when the SMBus, SPI Bus, or UART is selected, the crossbar assigns all pins associated with the selected bus. It would be impossible for instance to assign the RX pin from the UART function without also assigning the TX function. Standard Port I/Os appear contiguously after the prioritized functions have been assigned. For example, if you choose functions that take the first 14 Port I/O (P0.[7:0], P1.[5:0]), you would have 18 Port I/O left unused by the crossbar (P1.[7:6], P2 and P3).

15.2. Port I/O Initialization

Port I/O initialization is straightforward. Registers XBR0, XBR1 and XBR2 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR2 to 1 enables the CrossBar. **Until the Crossbar is enabled, the external pins remain as standard Ports in input mode regardless of the XBRn Register settings.** For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Code Configuration Wizard function of the IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The output driver characteristics of the I/O pins are defined using the Port Configuration Registers PRT0CF, PRT1CF, PRT2CF and PRT3CF (see Figure 15.7, Figure 15.9, Figure 15.12, and Figure 15.14). Each Port Output driver can be configured as either Open Drain or Push-Pull. This is required even for the digital resources selected in the XBRn registers and is not automatic. The only exception to this is the SMBus (SDA, SCL) and UART Receive (RX, when in mode 0) pins which are Open-drain regardless of the PRTnCF settings. When the WEAKPUD bit in XBR2 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does

15.3. General Purpose Port I/O

Each MCU has four byte-wide, bi-directional parallel ports that can be used general purpose I/O. Each port is accessed through a corresponding special function register (SFR) that is both byte addressable and bit addressable. When writing to a port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the port's input pins are returned regardless of the XBRn settings (i.e. even when the pin is assigned to another signal by the Crossbar, the Port Register can always still read its corresponding Port I/O pin). The exception to this is the execution of the *read-modify-write* instructions. The *read-modify-write* instructions when operating on a port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SET, when the destination is an individual bit in a port SFR. For these instructions, the value of the port register (not the pin) is read, modified, and written back to the SFR.

15.4. Configuring Ports Which are not Pinned Out

P2 and P3 are not pinned out on the F001/06/11/16. P1, P2, and P3 are not pinned out on the F002/07/12/17. These port registers (and corresponding interrupts, where applicable) are still available for software use in these reduced pin count MCUs. Whether used or not in software, it is recommended not to let these port drivers go to high impedance state. This is prevented after reset by having the weak pull-ups enabled as described in the XBR2 register. It is recommended that each output driver for ports not pinned out should be configured as push-pull using the corresponding PRTnCF register. This will inhibit a high impedance state even if the weak pull-up is disabled.

Figure 15.6. P0: Port0 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0x80

Bits7-0: P0.[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 PRT0CF.n bit = 0)
 (Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
 0: P0.n pin is logic low.
 1: P0.n pin is logic high.

Figure 15.7. PRT0CF: Port0 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: 0xA4

Bits7-0: PRT0CF.[7:0]: Output Configuration Bits for P0.7-P0.0 (respectively)
 0: Corresponding P0.n Output mode is Open-Drain.
 1: Corresponding P0.n Output mode is Push-Pull.

(Note: When SDA, SCL, and RX appear on any of the P0 I/O, each are open-drain regardless of the value of PRT0CF).

Figure 15.8. P1: Port1 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0x90

Bits7-0: P1.[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 PRT1CF.n bit = 0)
 (Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
 0: P1.n pin is logic low.
 1: P1.n pin is logic high.

Figure 15.9. PRT1CF: Port1 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: 0xA5

Bits7-0: PRT1CF.[7:0]: Output Configuration Bits for P1.7-P1.0 (respectively)
 0: Corresponding P1.n Output mode is Open-Drain.
 1: Corresponding P1.n Output mode is Push-Pull.

Figure 15.10. PRT1IF: Port1 Interrupt Flag Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IE7	IE6	IE5	IE4	-	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAD

Bit7: IE7: External Interrupt 7 Pending Flag.
 0: No falling edge detected on P1.7.
 1: This flag is set by hardware when a falling edge on P1.7 is detected.

Bit6: IE6: External Interrupt 6 Pending Flag.
 0: No falling edge detected on P1.6.
 1: This flag is set by hardware when a falling edge on P1.6 is detected.

Bit5: IE5: External Interrupt 5 Pending Flag.
 0: No falling edge detected on P1.5.
 1: This flag is set by hardware when a falling edge on P1.5 is detected.

Bit4: IE4: External Interrupt 4 Pending Flag.
 0: No falling edge detected on P1.4.
 1: This flag is set by hardware when a falling edge on P1.4 is detected.

Bits3-0: UNUSED. Read = 0000b, Write = don't care.

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 5 μ s 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 50 μ s 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 50 μ s.

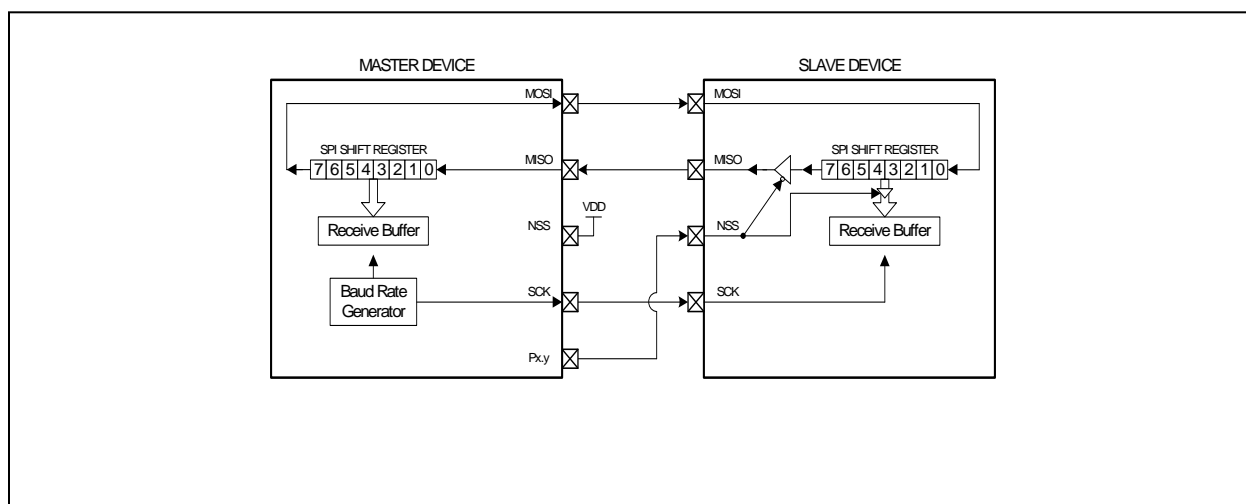
Setting the SMBus timeout enable bit (TOE, SMB0CN.0) to logic 1 enables Timer 3 to count up when the SCL line is low and Timer 3 is enabled. If Timer 3 overflows, a Timer 3 interrupt will be generated, which will alert the CPU that a SMBus SCL low timeout has occurred.

17.2. Operation

Only a SPI master device can initiate a data transfer. The SPI is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.1). Writing a byte of data to the SPI data register (SPI0DAT) when in Master Mode starts a data transfer. The SPI master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. The SPI master can be configured to shift in/out from one to eight bits in a transfer operation in order to accommodate slave devices with different word lengths. The SPIFRS bits in the SPI Configuration Register (SPI0CFG.[2:0]) are used to select the number of bits to shift in/out in a transfer operation.

While the SPI master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. The data byte received from the slave replaces the data in the master's data register. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data transfer in both directions is synchronized with the serial clock generated by the master. Figure 17.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

Figure 17.3. Full Duplex Operation



The SPI data register is double buffered on reads, but not on a write. If a write to SPI0DAT is attempted during a data transfer, the WCOL flag (SPI0CN.6) will be set to logic 1 and the write is ignored. The current data transfer will continue uninterrupted. A read of the SPI data register by the system controller actually reads the receive buffer. If the receive buffer still holds unread data from a previous transfer when the last bit of the current transfer is shifted into the SPI shift register, a receive overrun occurs and the RXOVRN flag (SPI0CN.4) is set to logic 1. The new data is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte causing the overrun is lost.

When the SPI is enabled and not configured as a master, it will operate as an SPI slave. Another SPI device acting as a master will initiate a transfer by driving the NSS signal low. The master then shifts data out of the shift register on the MOSI pin using its serial clock. The SPIF flag is set to logic 1 at the end of a data transfer (when the NSS signal goes high). The slave can load its shift register for the next data transfer by writing to the SPI data register. The slave must make the write to the data register at least one SPI serial clock cycle before the master starts the next transmission. Otherwise, the byte of data already in the slave's shift register will be transferred.

Multiple masters may reside on the same bus. A Mode Fault flag (MODE, SPI0CN.5) is set to logic 1 when the SPI is configured as a master (MSTEN = 1) and its slave select signal NSS is pulled low. When the Mode Fault flag is set, the MSTEN and SPIEN bits of the SPI control register are cleared by hardware, thereby placing the SPI module

18.1. UART Operational Modes

The UART provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 18.1 below. Detailed descriptions follow.

Table 18.1. UART Modes

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

18.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX pin. The TX 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 18.2).

Eight data bits are transmitted/received, LSB first (see the timing diagram in Figure 18.3). Data transmission begins when an instruction writes a data byte to the SBUF register. The TI Transmit Interrupt Flag (SCON.1) is set at the end of the eighth bit time. Data reception begins when the REN Receive Enable bit (SCON.4) is set to logic 1 and the RI Receive Interrupt Flag (SCON.0) is cleared. One cycle after the eighth bit is shifted in, the RI flag is set and reception stops until software clears the RI bit. An interrupt will occur if enabled when either TI or RI is set.

The Mode 0 baud rate is the system clock frequency divided by twelve. RX is forced to open-drain in mode 0, and an external pull-up will typically be required.

Figure 18.2. UART Mode 0 Interconnect

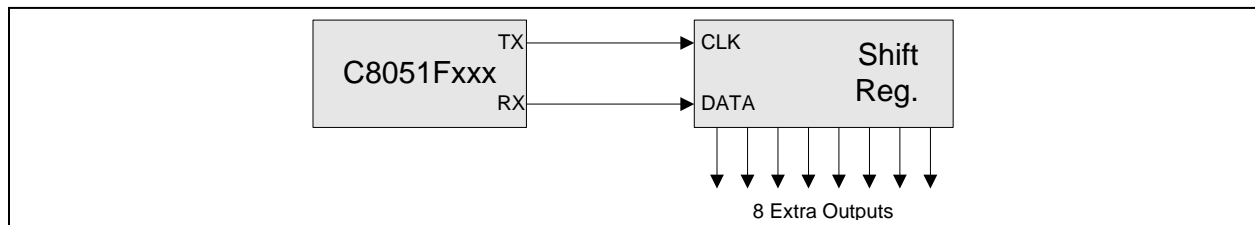
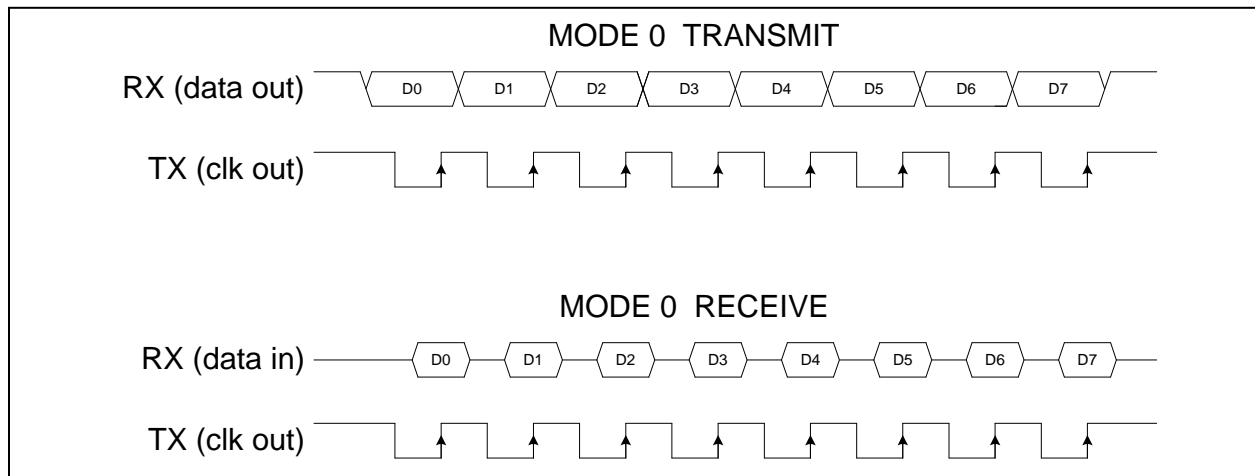


Figure 18.3. UART Mode 0 Timing Diagram



18.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate

Mode 2 provides asynchronous, full-duplex communication using a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (see timing diagram in Figure 18.6). On transmit, the ninth data bit is determined by the value in TB8 (SCON.3). It can be assigned the value of the parity flag P in the PSW or used in multiprocessor communications. On receive, the ninth data bit goes into RB8 (SCON.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF register. The TI Transmit Interrupt Flag (SCON.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit (SCON.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF receive register if the following conditions are met: RI must be logic 0, and if SM2 is logic 1, the 9th bit must be logic 1.

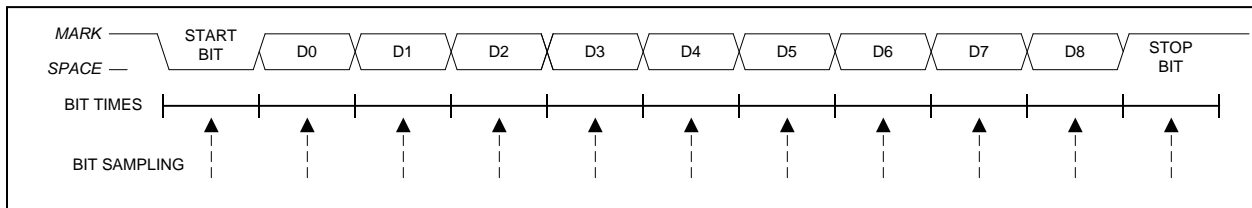
If these conditions are met, the eight bits of data are stored in SBUF, the ninth bit is stored in RB8 and the RI flag is set. If these conditions are not met, SBUF and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI are set.

The baud rate in Mode 2 is a direct function of the system clock frequency as follows:

$$\text{Mode 2 Baud Rate} = 2^{\text{SMOD}} * (\text{SYSCLK} / 64).$$

The SMOD bit (PCON.7) selects whether to divide SYSCLK by 32 or 64. In the formula, 2 is raised to the power SMOD, resulting in a baud rate of either 1/32 or 1/64 of the system clock frequency. On reset, the SMOD bit is logic 0, thus selecting the lower speed baud rate by default.

Figure 18.6. UART Modes 2 and 3 Timing Diagram



18.1.4. Mode 3: 9-Bit UART, Variable Baud Rate

Mode 3 is the same as Mode 2 in all respects except the baud rate is variable. The baud rate is determined in the same manner as for Mode 1. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Timer 1 or Timer 2 overflows generate the baud rate just as with Mode 1. In summary, Mode 3 transmits using the same protocol as Mode 2 but with Mode 1 baud rate generation.

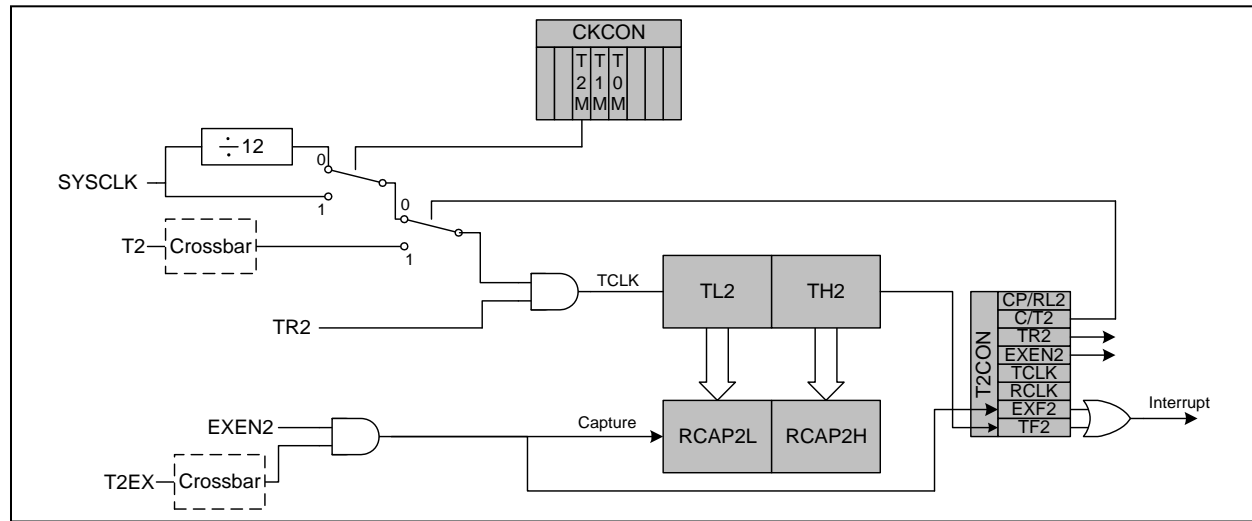
19.2.1. Mode 0: 16-bit Counter/Timer with Capture

In this mode, Timer 2 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T2EX input pin causes the 16-bit value in Timer 2 (TH2, TL2) to be loaded into the capture registers (RCAP2H, RCAP2L).

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the external T2 pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/T2 bit (T2CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set to logic 1, a high-to-low transition at the T2 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF2 timer overflow flag (T2CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL2 (T2CON.0) and the Timer 2 Run Control bit TR2 (T2CON.2) to logic 1. The Timer 2 External Enable EXEN2 (T2CON.3) must also be set to logic 1 to enable a capture. If EXEN2 is cleared, transitions on T2EX will be ignored.

Figure 19.11. T2 Mode 0 Block Diagram



21.1. Boundary Scan

The Data Register in the Boundary Scan path is an 87-bit shift register. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

Table 21.1. Boundary Data Register Bit Definitions

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Reset Enable from MCU
	Update	Reset Enable to /RST pin
1	Capture	Reset input from /RST pin
	Update	Reset output to /RST pin
2	Capture	External Clock from XTAL1 pin
	Update	Not used
3	Capture	Weak pullup enable from MCU
	Update	Weak pullup enable to Port Pins
4-11	Capture	SFR Address Bus bit from CIP-51 (e.g. Bit4=SFRA0, Bit5=SFRA1...)
	Update	SFR Address Bus bit to SFR Address Bus (e.g. Bit4=XSFA0, Bit5=XSFA1)
12-19	Capture	SFR Data Bus bit read from SFR (e.g. Bit12=SFDR0, Bit13=SFDR1...)
	Update	SFR Data Bus bit written to SFR (e.g. Bit12=SFDR0, Bit13=SFDR1...)
20	Capture	SFR Write Strobe from CIP-51
	Update	SFR Write Strobe to SFR Bus
21	Capture	SFR Read Strobe from CIP-51
	Update	SFR Read Strobe to SFR Bus
22	Capture	SFR Read/Modify/Write Strobe from CIP-51
	Update	SFR Read/Modify/Write Strobe to SFR Bus
23,25,27,29, 31,33,35,37	Capture	P0.n output enable from MCU (e.g. Bit23=P0.0, Bit25=P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit23=P0.0oe, Bit25=P0.1oe, etc.)
24,26,28,30, 32,34,36,38	Capture	P0.n input from pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
39,41,43,45, 47,49,51,53	Capture	P1.n output enable from MCU (e.g. Bit39=P1.0, Bit41=P1.1, etc.)
	Update	P1.n output enable to pin (e.g. Bit39=P1.0oe, Bit41=P1.1oe, etc.)
40,42,44,46, 48,50,52,54	Capture	P1.n input from pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
	Update	P1.n output to pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
55,57,59,61, 63,65,67,69	Capture	P2.n output enable from MCU (e.g. Bit55=P2.0, Bit57=P2.1, etc.)
	Update	P2.n output enable to pin (e.g. Bit55=P2.0oe, Bit57=P2.1oe, etc.)
56,58,60,62, 64,66,68,70	Capture	P2.n input from pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
	Update	P2.n output to pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
71,73,75,77, 79,81,83,85	Capture	P3.n output enable from MCU (e.g. Bit71=P3.0, Bit73=P3.1, etc.)
	Update	P3.n output enable to pin (e.g. Bit71=P3.0oe, Bit73=P3.1oe, etc.)
72,74,76,78, 80,82,84,86	Capture	P3.n input from pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)
	Update	P3.n output to pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)