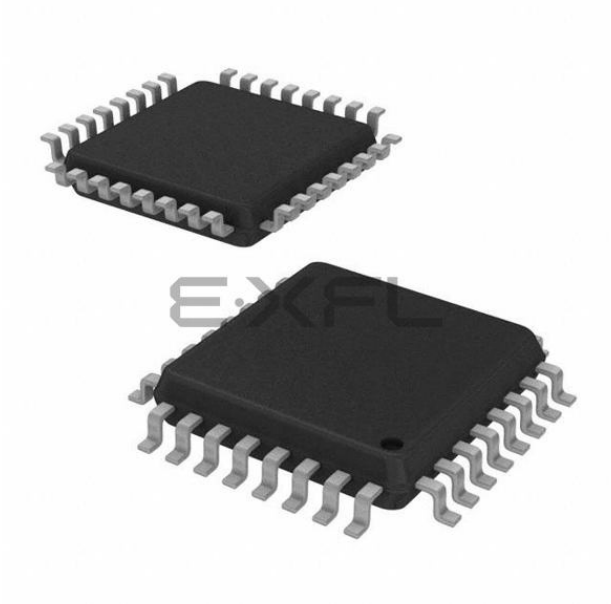


Welcome to E-XFL.COM

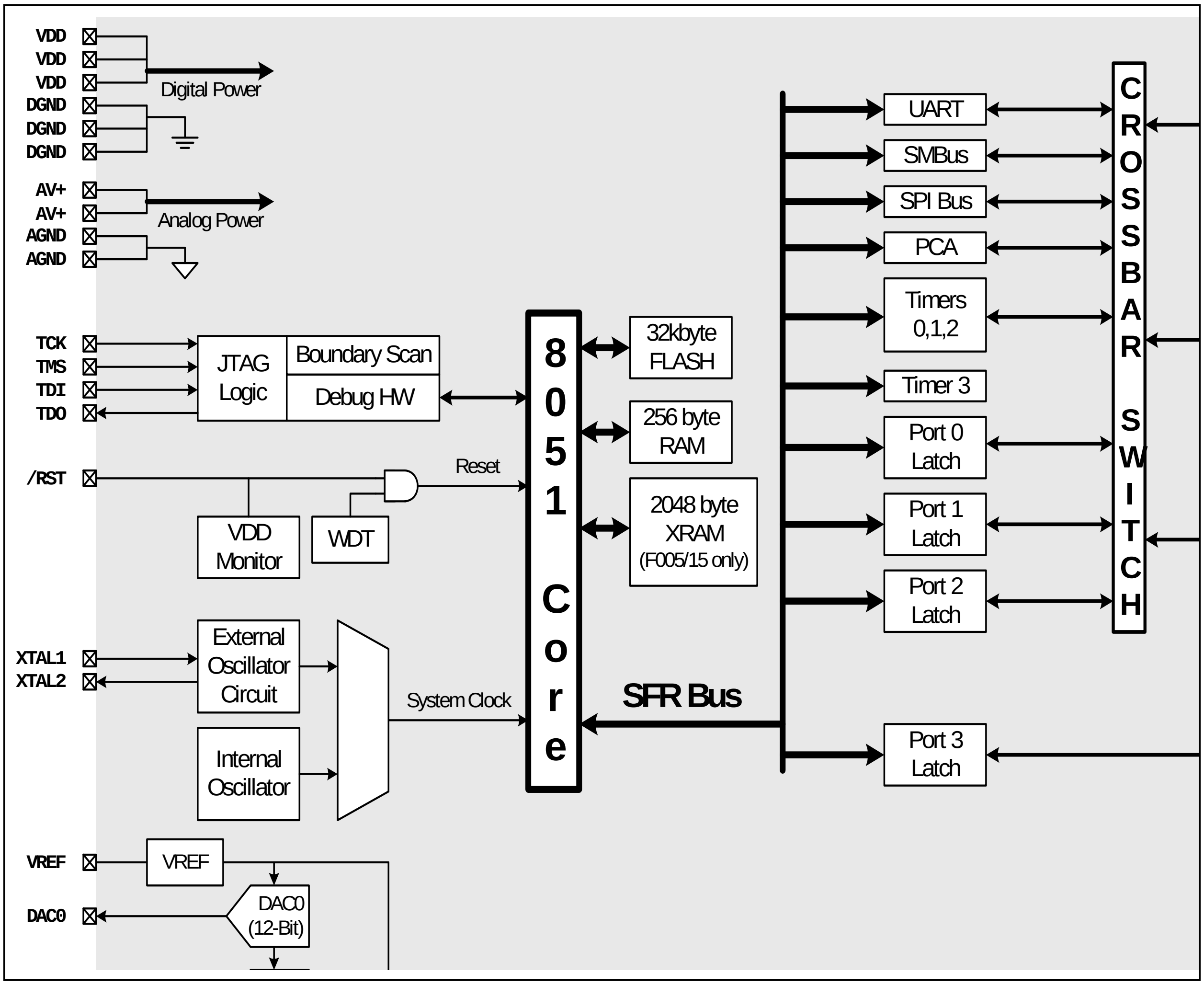
What is "Embedded - Microcontrollers"?

"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	8
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 4x10b; 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/c8051f012

Figure 1.1. C8051F000/05/10/15 Block Diagram

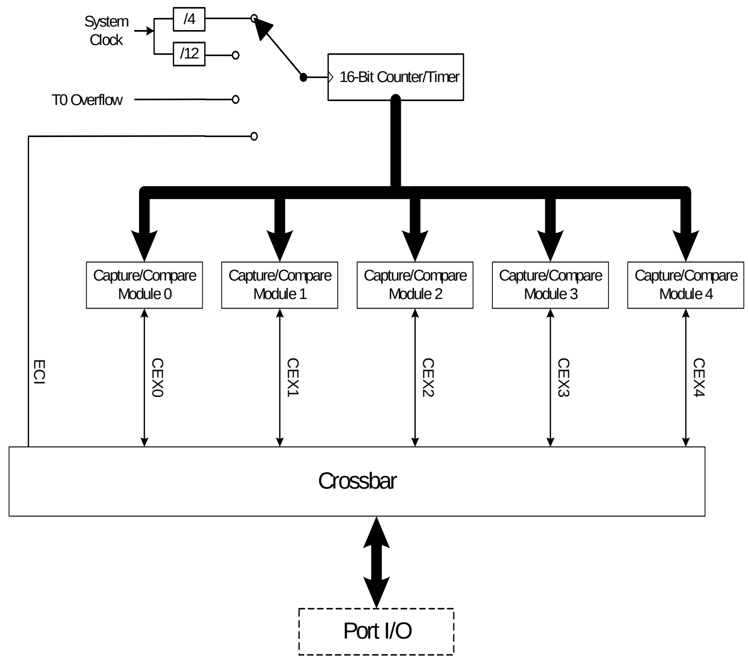


1.5. Programmable Counter Array

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.

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 ±1LSB. 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 ±1LSB, 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

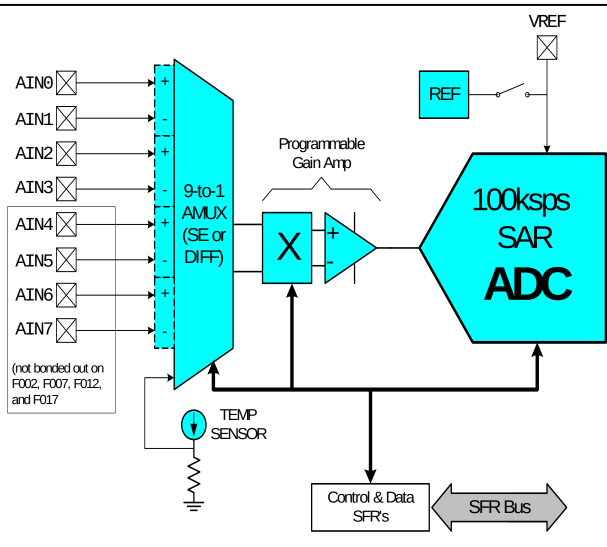
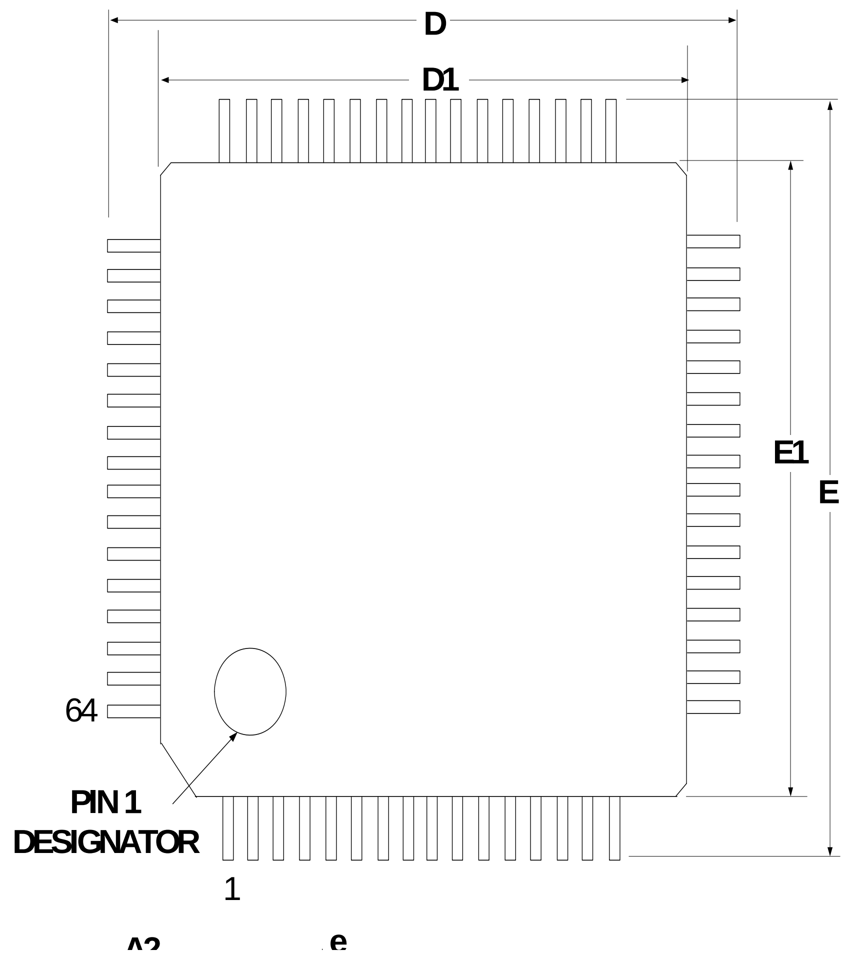


Figure 4.2. TQFP-64 Package Drawing



	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 5.6. ADC0CF: ADC Configuration Register (C8051F00x)

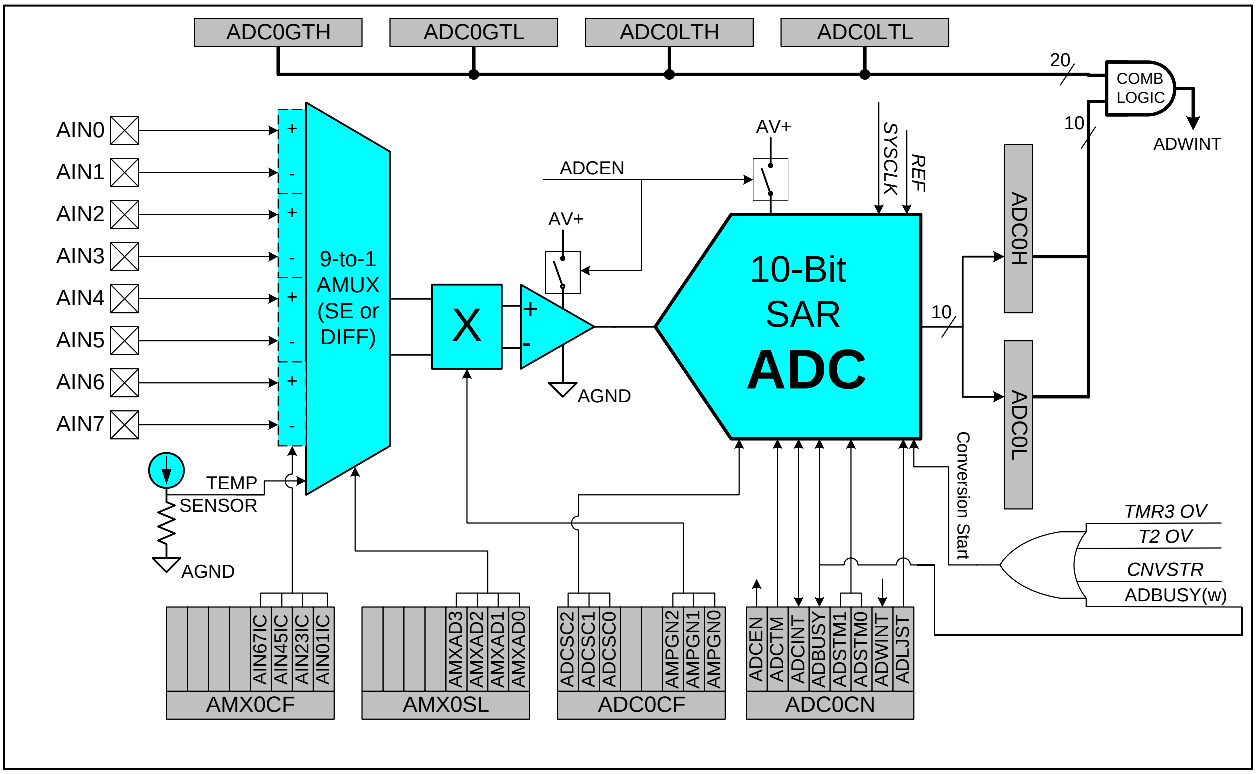
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	01100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7-5: ADCSC2-0: ADC SAR Conversion Clock Period Bits
000: SAR Conversion Clock = 1 System Clock
001: SAR Conversion Clock = 2 System Clocks
010: SAR Conversion Clock = 4 System Clocks
011: SAR Conversion Clock = 8 System Clocks
1xx: SAR Conversion Clock = 16 Systems Clocks
(Note: the SAR Conversion Clock should be ≤ 2MHz)
Bits4-3: UNUSED. Read = 00b; Write = don't care
Bits2-0: AMPGN2-0: ADC Internal Amplifier Gain
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
11x: Gain = 0.5

6. ADC (10-Bit, C8051F010/1/2/5/6/7 Only)

The ADC subsystem for the C8051F010/1/2/5/6/7 consists of a 9-channel, configurable analog multiplexer (AMUX), a programmable gain amplifier (PGA), and a 100ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 6.1). The AMUX, PGA, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Register's shown in Figure 6.1. The ADC subsystem (ADC, track-and-hold and PGA) is enabled only when the ADCEN bit in the ADC Control register (ADC0CN, Figure 6.7) is set to 1. The ADC subsystem is in low power shutdown when this bit is 0. The Bias Enable bit (BIASE) in the REF0CN register (see Figure 9.2) must be set to 1 in order to supply bias to the ADC.

Figure 6.1. 10-Bit ADC Functional Block Diagram



6.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-board temperature sensor (temperature transfer function is shown in Figure 6.3). Note that the PGA gain is applied to the temperature sensor reading. AMUX input pairs can be programmed to operate in either the differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (Figure 6.5), and the Configuration register AMX0CF (Figure 6.4). The table in Figure 6.5 shows AMUX functionality by channel for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the AMPGN2-0 bits in the ADC Configuration register, ADC0CF (Figure 6.6). The PGA can be software-programmed for gains of 0.5, 1, 2, 4, 8 or 16. It defaults to unity gain on reset.

Figure 6.7. ADC0CN: ADC Control Register (C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCEN	ADCTM	ADCINT	ADBUSY	ADSTM1	ADSTM0	ADWINT	ADLJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE8

Bit7: ADCEN: ADC Enable Bit
0: ADC Disabled. ADC is in low power shutdown.
1: ADC Enabled. ADC is active and ready for data conversions.

Bit6: ADCTM: ADC Track Mode Bit
0: When the ADC is enabled, tracking is always done unless a conversion is in process
1: Tracking Defined by ADSTM1-0 bits
ADSTM1-0:
00: Tracking starts with the write of 1 to ADBUSY and lasts for 3 SAR clocks
01: Tracking started by the overflow of Timer 3 and last for 3 SAR clocks
10: ADC tracks only when CNVSTR input is logic low
11: Tracking started by the overflow of Timer 2 and last for 3 SAR clocks

Bit5: ADCINT: ADC Conversion Complete Interrupt Flag
(Must be cleared by software)
0: ADC has not completed a data conversion since the last time this flag was cleared
1: ADC has completed a data conversion

Bit4: ADBUSY: ADC Busy Bit
Read
0: ADC Conversion complete or no valid data has been converted since a reset. The falling edge of ADBUSY generates an interrupt when enabled.
1: ADC Busy converting data
Write
0: No effect
1: Starts ADC Conversion if ADSTM1-0 = 00b

Bits3-2: ADSTM1-0: ADC Start of Conversion Mode Bits
00: ADC conversion started upon every write of 1 to ADBUSY
01: ADC conversions taken on every overflow of Timer 3
10: ADC conversion started upon every rising edge of CNVSTR
11: ADC conversions taken on every overflow of Timer 2

Bit1: ADWINT: ADC Window Compare Interrupt Flag
(Must be cleared by software)
0: ADC Window Comparison Data match has not occurred
1: ADC Window Comparison Data match occurred

Bit0: ADLJST: ADC Left Justify Data Bit
0: Data in ADC0H:ADC0L Registers is right justified
1: Data in ADC0H:ADC0L Registers is left justified

Figure 7.2. DAC0H: DAC0 High Byte 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: 0xD3

Bits7-0: DAC0 Data Word Most Significant Byte.

Figure 7.3. DAC0L: DAC0 Low Byte 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: 0xD2

Bits7-0: DAC0 Data Word Least Significant Byte.

Figure 7.4. DAC0CN: DAC0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
DAC0EN	-	-	-	-	DAC0DF2	DAC0DF1	DAC0DF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD4

Bit7: DAC0EN: DAC0 Enable Bit
0: DAC0 Disabled. DAC0 Output pin is disabled; DAC0 is in low power shutdown mode.
1: DAC0 Enabled. DAC0 Output is pin active; DAC0 is operational.

Bits6-3: UNUSED. Read = 0000b; Write = don't care

Bits2-0: DAC0DF2-0: DAC0 Data Format Bits

000: The most significant nybble of the DAC0 Data Word is in DAC0H[3:0], while the least significant byte is in DAC0L.

DAC0H								DAC0L							
				MSB											LSB

001: The most significant 5-bits of the DAC0 Data Word is in DAC0H[4:0], while the least significant 7-bits is in DAC0L[7:1].

DAC0H								DAC0L							
			MSB											LSB	

010: The most significant 6-bits of the DAC0 Data Word is in DAC0H[5:0], while the least significant 6-bits is in DAC0L[7:2].

DAC0H								DAC0L							
		MSB											LSB		

011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits is in DAC0L[7:3].

DAC0H								DAC0L							
	MSB											LSB			

1xx: The most significant byte of the DAC0 Data Word is in DAC0H, while the least significant nybble is in DAC0L[7:4].

DAC0H								DAC0L							
MSB											LSB				

8. COMPARATORS

The MCU family has two on-chip analog voltage comparators as shown in Figure 8.1. The inputs of each Comparator are available at the package pins. The output of each comparator is optionally available at the package pins via the I/O crossbar (see Section 15.1). When assigned to package pins, each comparator output can be programmed to operate in open drain or push-pull modes (see section 15.3).

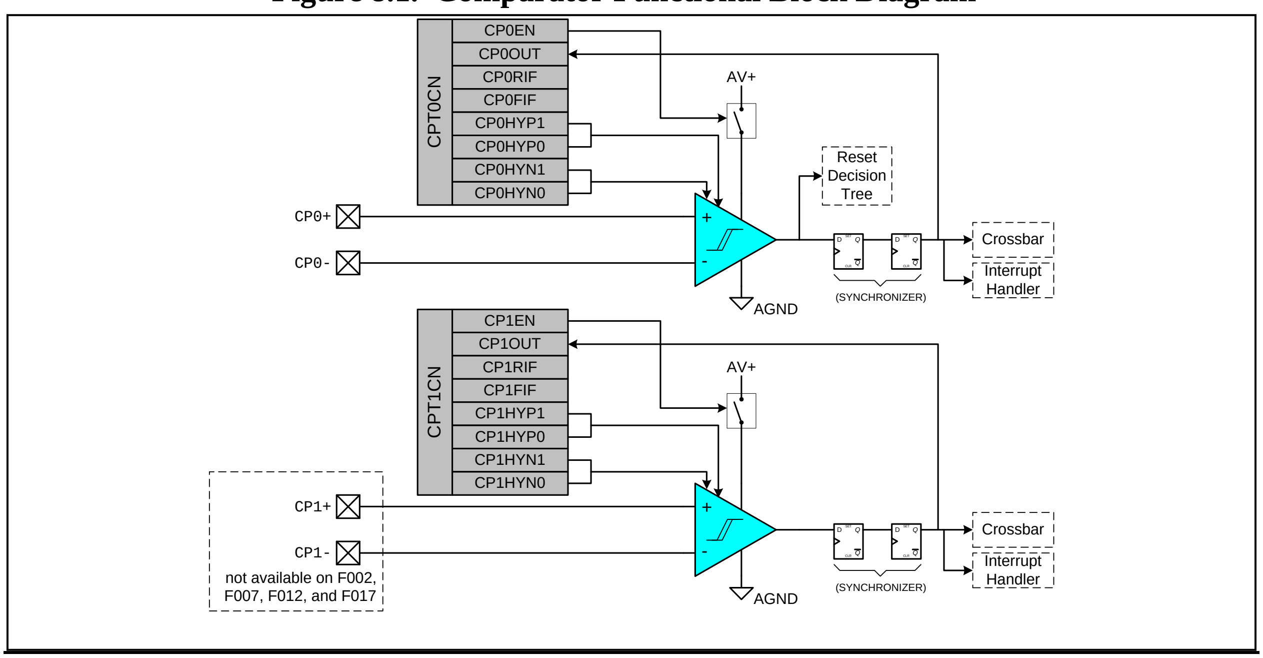
The hysteresis of each comparator is software-programmable via its respective Comparator control register (CPT0CN, CPT1CN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage. The output of the comparator can be polled in software, or can be used as an interrupt source. Each comparator can be individually enabled or disabled (shutdown). When disabled, the comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, its interrupt capability is suspended and its supply current falls to less than 1µA. Comparator 0 inputs can be externally driven from -0.25V to (AV+) + 0.25V without damage or upset.

The Comparator 0 hysteresis is programmed using bits 3-0 in the Comparator 0 Control Register CPT0CN (shown in Figure 8.3). The amount of *negative* hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 8.2, settings of 10, 4 or 2mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of *positive* hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section 10.4). The CP0FIF flag is set upon a Comparator 0 falling-edge interrupt, and the CP0RIF flag is set upon the Comparator 0 rising-edge interrupt. Once set, these bits remain set until cleared by the CPU. The Output State of Comparator 0 can be obtained at any time by reading the CP0OUT bit. Note the comparator output and interrupt should be ignored until the comparator settles after power-up. Comparator 0 is enabled by setting the CP0EN bit, and is disabled by clearing this bit. Note there is a 20usec settling time for the comparator output to stabilize after setting the CP0EN bit or a power-up. Comparator 0 can also be programmed as a reset source. For details, see Section 13.

The operation of Comparator 1 is identical to that of Comparator 0, except the Comparator 1 is controlled by the CPT1CN Register (Figure 8.4). Comparator 1 can not be programmed as a reset source. Also, the input pins for Comparator 1 are not pinned out on the F002, F007, F012, or F017 devices. The complete electrical specifications for the Comparators are given in Table 8.1.

Figure 8.1. Comparator Functional Block Diagram



10.3.1. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic l. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

Figure 10.3. SP: Stack Pointer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x81

Bits 7-0: SP: Stack Pointer.
The stack pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

Figure 10.4. DPL: Data Pointer 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: 0x82

Bits 7-0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed RAM and Flash Memory.

Figure 10.5. DPH: Data Pointer 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: 0x83

Bits 7-0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed RAM and Flash Memory.

13.1. Power-on Reset

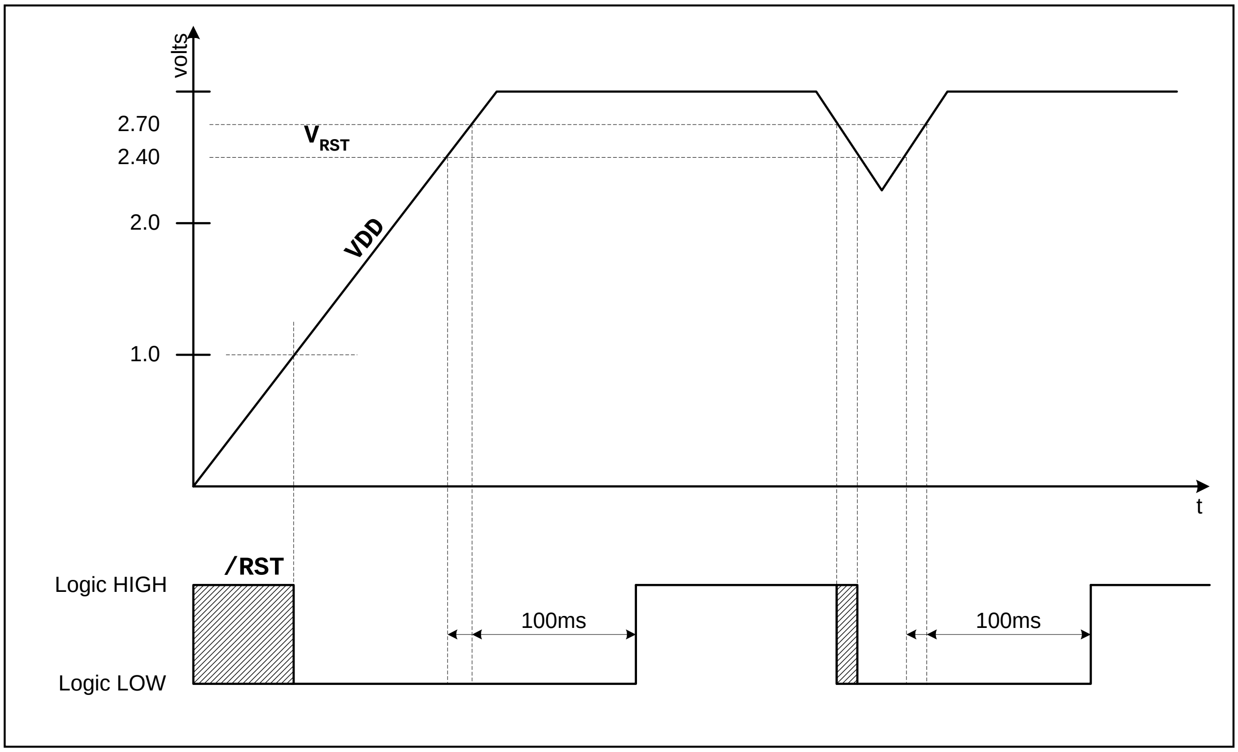
The C8051F000 family incorporates a power supply monitor that holds the MCU in the reset state until VDD rises above the V_{RST} level during power-up. (See Figure 13.2 for timing diagram, and refer to Table 13.1 for the Electrical Characteristics of the power supply monitor circuit.) The /RST pin is asserted (low) until the end of the 100ms VDD Monitor timeout in order to allow the VDD supply to become stable.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. All of the other reset flags in the RSTSRC Register are indeterminate. PORSF is cleared by a reset from any other source. Since all resets cause program execution to begin at the same location (0x0000), software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset.

13.2. Software Forced Reset

Writing a 1 to the PORSF bit forces a Power-On Reset as described in Section 13.1.

Figure 13.2. VDD Monitor Timing Diagram



13.3. Power-fail Reset

When a power-down transition or power irregularity causes VDD to drop below V_{RST}, the power supply monitor will drive the /RST pin low and return the CIP-51 to the reset state (see Figure 13.2). When VDD returns to a level above V_{RST}, the CIP-51 will leave the reset state in the same manner as that for the power-on reset. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if VDD dropped below the level required for data retention. If the PORSF flag is set, the data may no longer be valid.

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

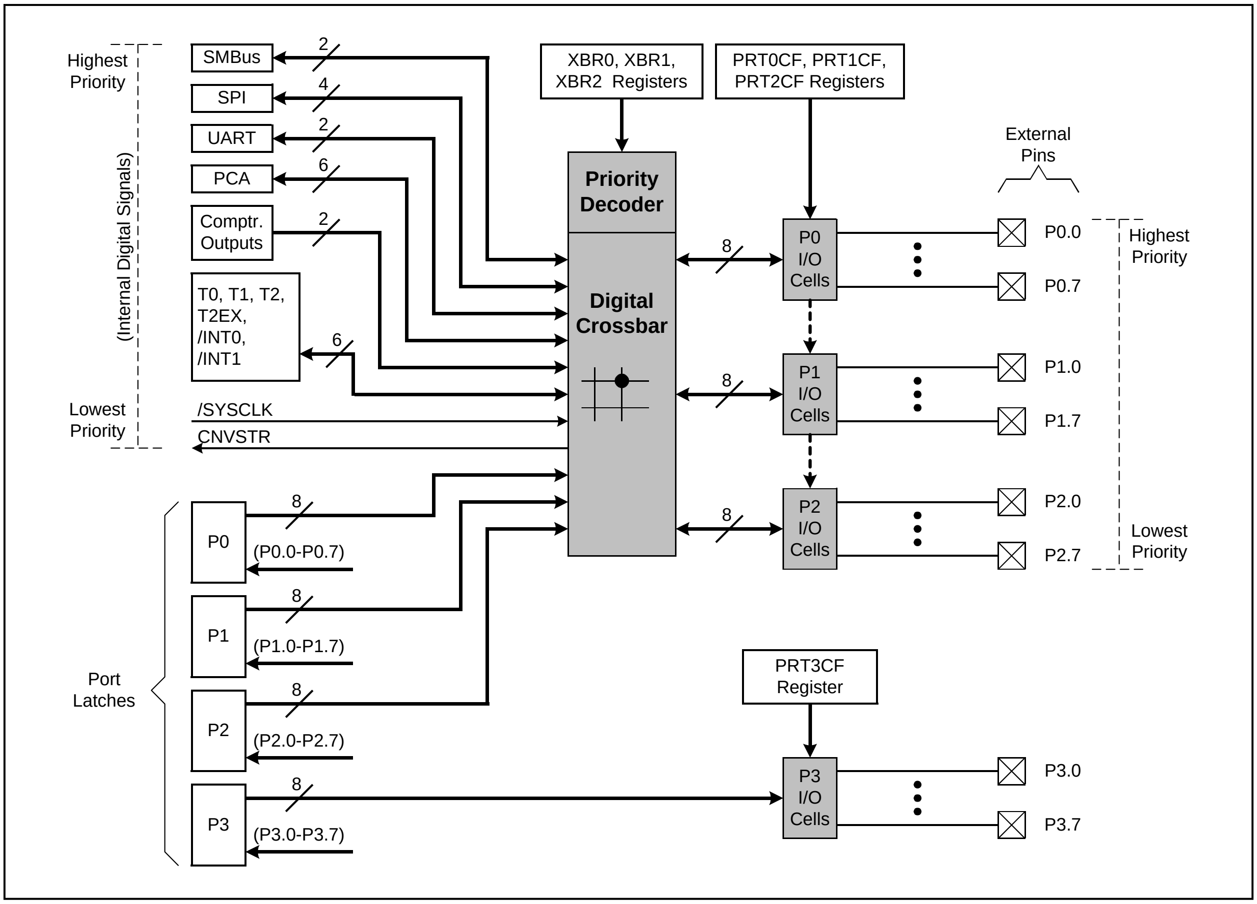


Figure 15.2. Port I/O Cell Block Diagram

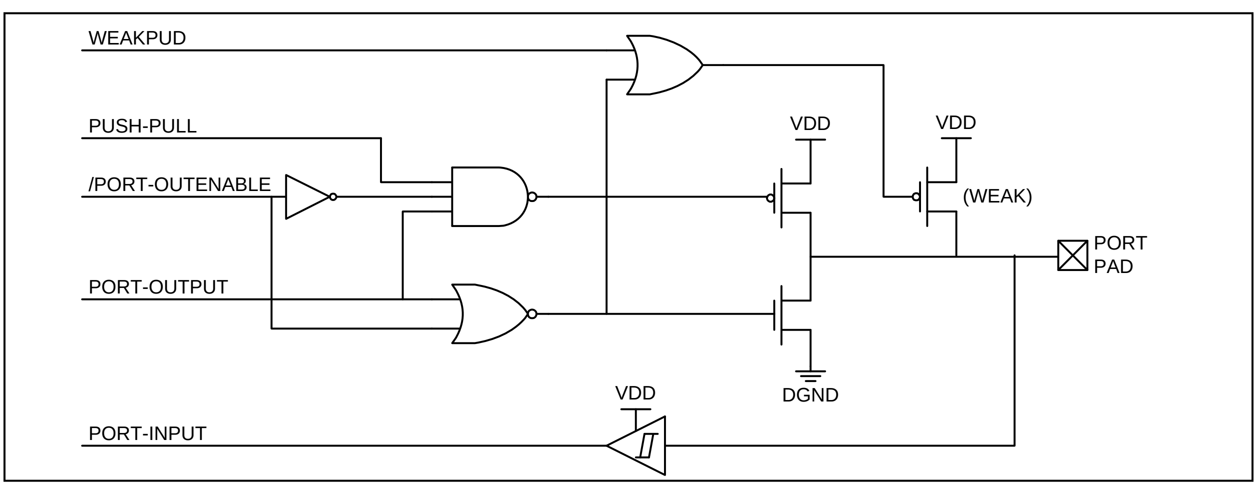
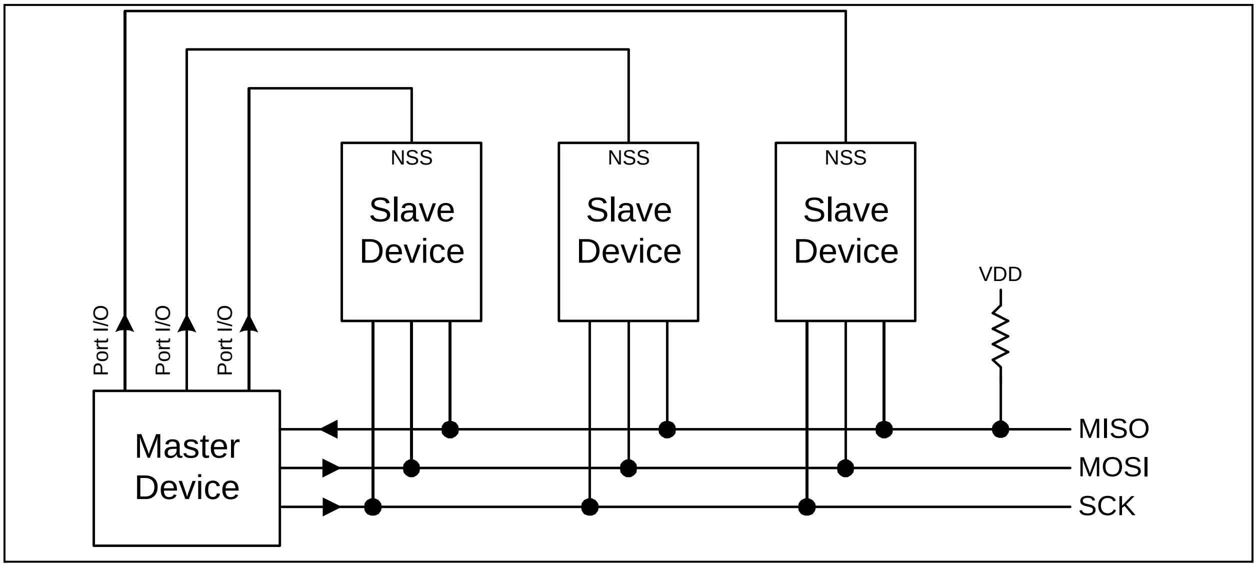


Figure 17.2. Typical SPI Interconnection



17.1. Signal Descriptions

The four signals used by the SPI (MOSI, MISO, SCK, NSS) are described below.

17.1.1. Master Out, Slave In

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. Data is transferred most-significant bit first.

17.1.2. Master In, Slave Out

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. Data is transferred most-significant bit first. A SPI slave places the MISO pin in a high-impedance state when the slave is not selected.

17.1.3. Serial Clock

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines.

17.1.4. Slave Select

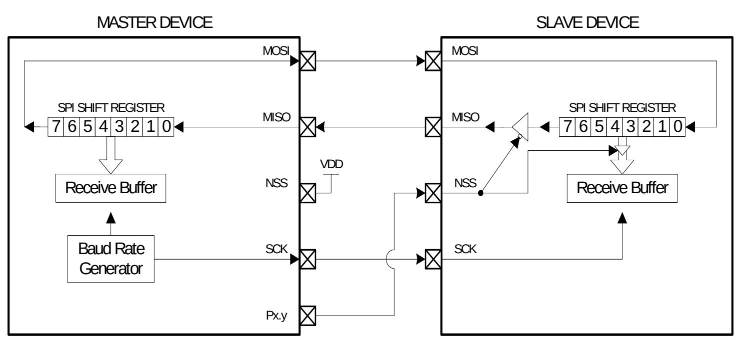
The slave select (NSS) signal is an input used to select the SPI module when in slave mode by a master, or to disable the SPI module when in master mode. When in slave mode, it is pulled low to initiate a data transfer and remains low for the duration of the transfer.

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 the 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 (MODF, 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

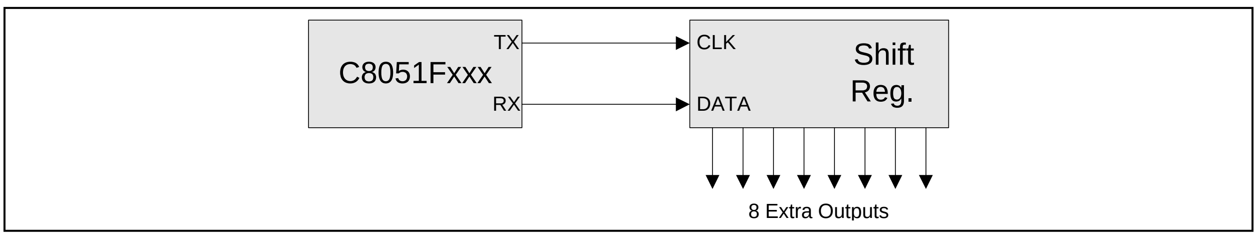
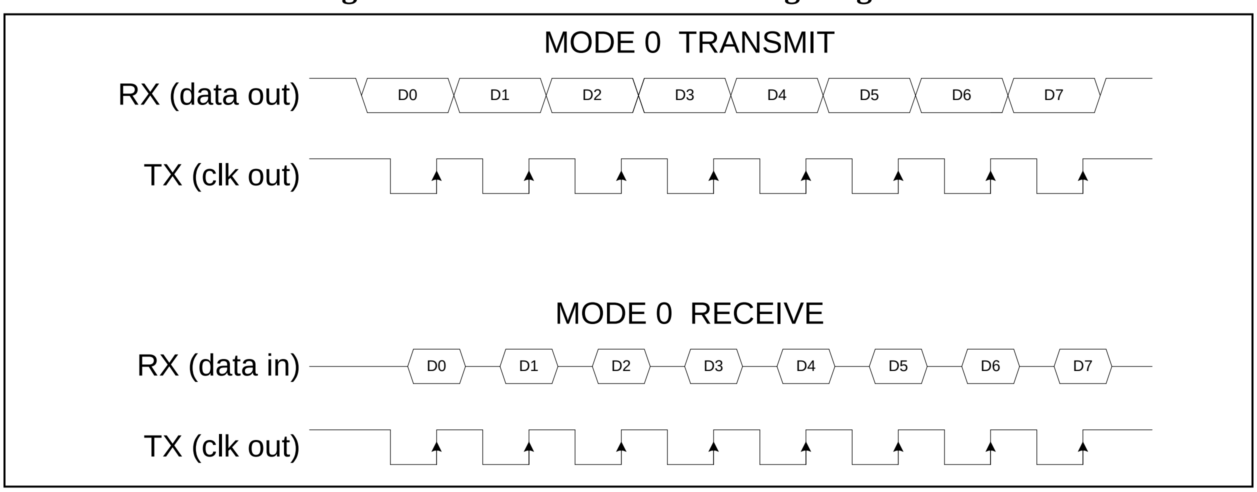


Figure 18.3. UART Mode 0 Timing Diagram

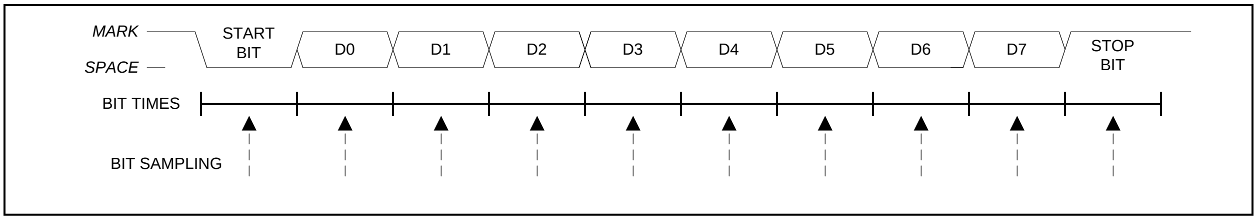


Mode 1 provides standard asynchronous, full duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit (see the timing diagram in Figure 18.4). Data are transmitted from the TX pin and received at the RX pin (see the interconnection diagram in Figure 18.5). On receive, the eight data bits are stored in SBUF and the stop bit goes into RB8 (SCON.2).

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 stop bit must be logic 1.

If these conditions are met, the eight bits of data are stored in SBUF, the stop 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 is set.

Figure 18.4. UART Mode 1 Timing Diagram



The baud rate generated in Mode 1 is a function of timer overflow. The UART can use Timer 1 operating in *8-bit Counter/Timer with Auto-Reload Mode*, or Timer 2 operating in *Baud Rate Generator Mode* to generate the baud rate (note that the TX and RX clock sources are selected separately). On each timer overflow event (a rollover from all ones (0xFF for Timer 1, 0xFFFF for Timer 2) to zero), a clock is sent to the baud rate logic.

When Timer 1 is selected as a baud rate source, the SMOD bit (PCON.7) selects whether or not to divide the Timer 1 overflow rate by two. On reset, the SMOD bit is logic 0, thus selecting the lower speed baud rate by default. The SMOD bit affects the baud rate generated by Timer 1 as follows:

*Mode 1 Baud Rate = (1 / 32) * T1_OVERFLOWRATE (when the SMOD bit is set to logic 0).*
*Mode 1 Baud Rate = (1 / 16) * T1_OVERFLOWRATE (when the SMOD bit is set to logic 1).*

When Timer 2 is selected as a baud rate source, the baud rate generated by Timer 2 is as follows:

*Mode 1 Baud Rate = (1 / 16) * T2_OVERFLOWRATE.*

The Timer 1 overflow rate is determined by the Timer 1 clock source (T1CLK) and reload value (TH1). The frequency of T1CLK can be selected as SYSCLK, SYSCLK/12, or an external clock source. The Timer 1 overflow rate can be calculated as follows:

T1_OVERFLOWRATE = T1CLK / (256 – TH1).

For example, assume TMOD = 0x20.
If T1M (CKCON.4) is logic 1, then the above equation becomes:

T1_OVERFLOWRATE = (SYSCLK) / (256 – TH1).

If T1M (CKCON.4) is logic 0, then the above equation becomes:

T1_OVERFLOWRATE = (SYSCLK/12) / (256 – TH1).

Table 18.2. Oscillator Frequencies for Standard Baud Rates

Oscillator Frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate**
24.0	208	0xF3	115200 (115384)
23.592	205	0xF3	115200 (113423)
22.1184	192	0xF4	115200
18.432	160	0xF6	115200
16.5888	144	0xF7	115200
14.7456	128	0xF8	115200
12.9024	112	0xF9	115200
11.0592	96	0xFA	115200
9.216	80	0xFB	115200
7.3728	64	0xFC	115200
5.5296	48	0xFD	115200
3.6864	32	0xFE	115200
1.8432	16	0xFF	115200
24.576	320	0xEC	76800
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	848	0xCB	28800 (28921)
24.0	833	0xCC	28800 (28846)
23.592	819	0xCD	28800 (28911)
22.1184	768	0xD0	28800
18.432	640	0xD8	28800
16.5888	576	0xDC	28800
14.7456	512	0xE0	28800
12.9024	448	0xE4	28800
11.0592	384	0xE8	28800
9.216	320	0xEC	28800
7.3728	256	0xF0	28800
5.5296	192	0xF4	28800
3.6864	128	0xF8	28800
1.8432	64	0xFC	28800

* Assumes SMOD=1 and T1M=1.
** Numbers in parenthesis show the actual baud rate.

Figure 18.8. SBUF: Serial (UART) Data Buffer 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: 0x99

Bits7-0: SBUF.[7:0]: Serial Data Buffer Bits 7-0 (MSB-LSB)
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF is what initiates the transmission. When data is moved from SBUF, it comes from the receive buffer.

Figure 19.5. TMOD: Timer Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 = logic level one.

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5-4: T1M1-T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	**T1M0**	**Mode**
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 Inactive/stopped

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 = logic level one.

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

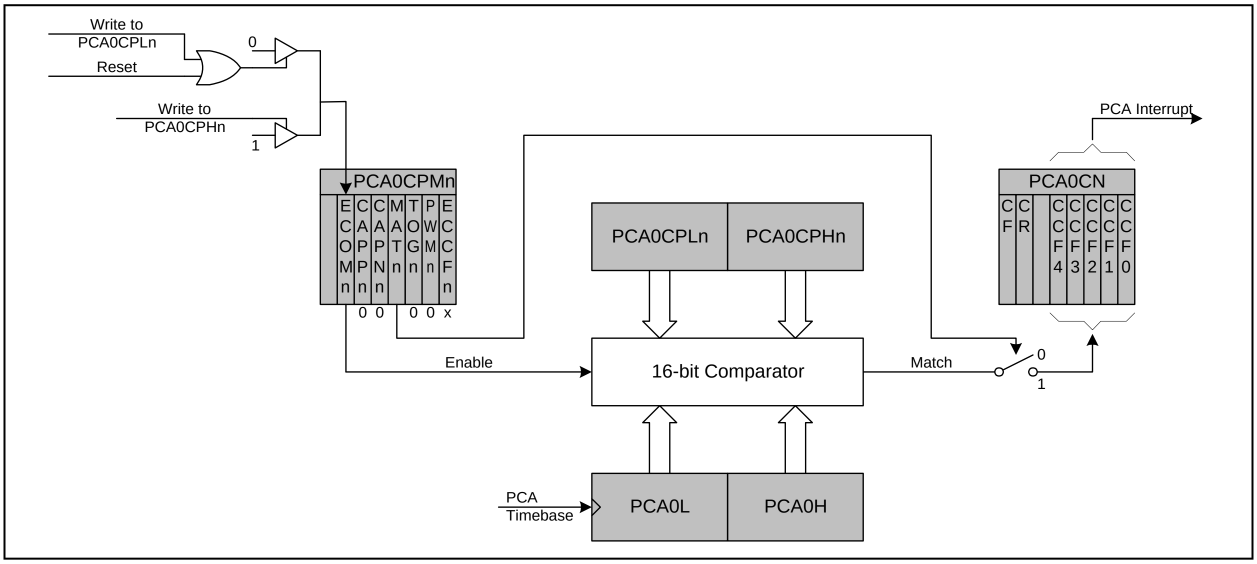
Bits1-0: T0M1-T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	**T0M0**	**Mode**
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

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

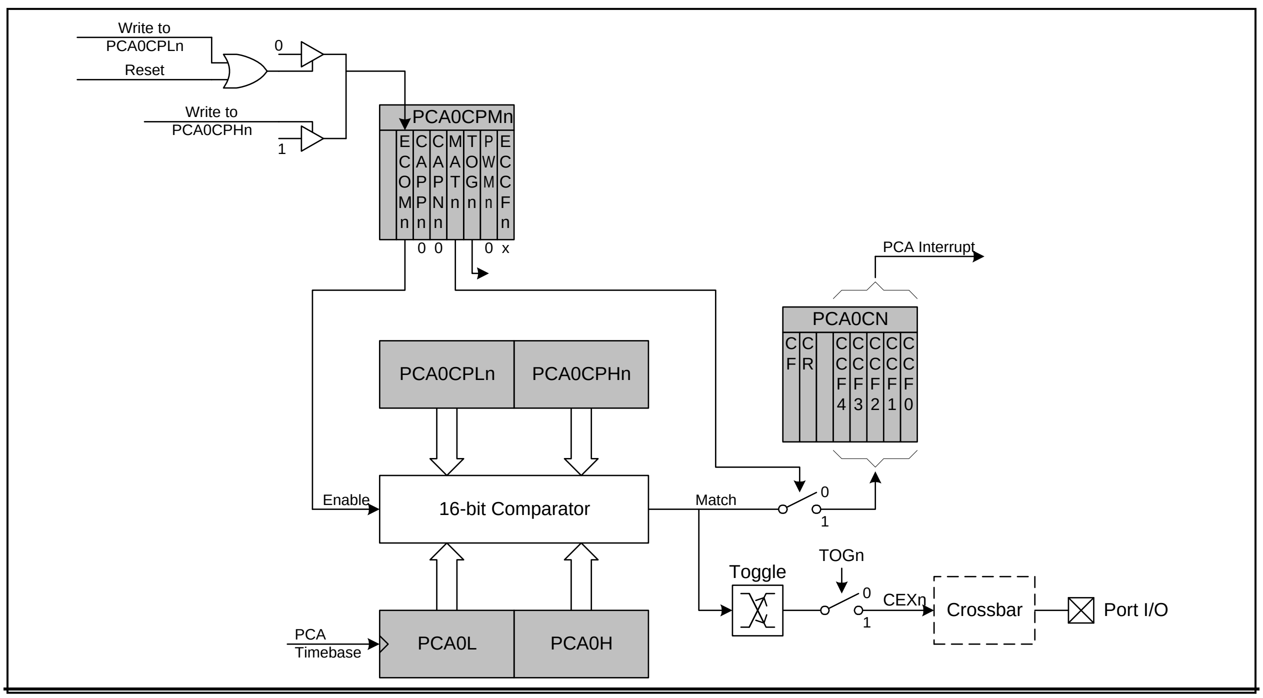


Figure 21.3. FLASHCON: JTAG Flash Control Register

WRMD3	WRMD2	WRMD1	WRMD0	RDMD3	RDMD2	RDMD1	RDMD0	Reset Value
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bits7-4: WRMD3-0: Write Mode Select Bits.
The Write Mode Select Bits control how the interface logic responds to writes to the FLASHDAT Register per the following values:
0000: A FLASHDAT write replaces the data in the FLASHDAT register, but is otherwise ignored.
0001: A FLASHDAT write initiates a write of FLASHDAT into the memory location addressed by the FLASHADR register. FLASHADR is incremented by one when complete.
0010: A FLASHDAT write initiates an erasure (sets all bytes to 0xFF) of the Flash page containing the address in FLASHADR. FLASHDAT must be 0xA5 for the erase to occur. FLASHADR is not affected. If FLASHADR = 0x7DFE – 0x7DFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x7E00 – 0x7FFF).
(All other values for WRMD3-0 are reserved.)

Bits3-0: RDMD3-0: Read Mode Select Bits.
The Read Mode Select Bits control how the interface logic responds to reads to the FLASHDAT Register per the following values:
0000: A FLASHDAT read provides the data in the FASHDAT register, but is otherwise ignored.
0001: A FLASHDAT read initiates a read of the byte addressed by the FLASHADR register if no operation is currently active. This mode is used for block reads.
0010: A FLASHDAT read initiates a read of the byte addressed by FLASHADR only if no operation is active and any data from a previous read has already been read from FLASHDAT. This mode allows single bytes to be read (or the last byte of a block) without initiating an extra read.
(All other values for RDMD3-0 are reserved.)

Figure 21.4. FLASHADR: JTAG Flash Address Register

																Reset Value
Bit15															Bit0	0x0000

This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15-0: Flash Operation 16-bit Address.