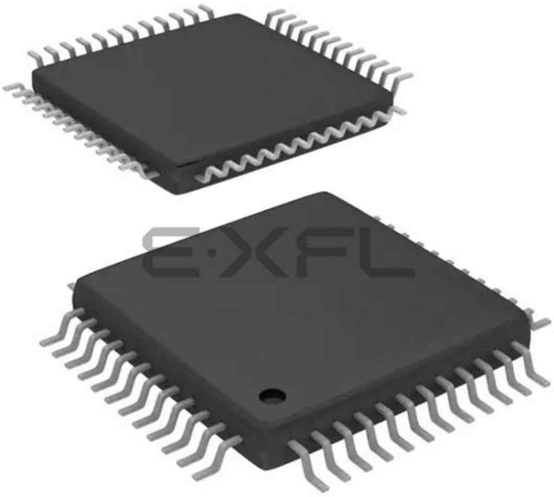


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
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	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.8V ~ 3.6V
Data Converters	A/D 8x10b
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/c8051f019-gqr

1. SYSTEM OVERVIEW

The C8051F018/9 are fully integrated mixed-signal System on a Chip MCUs with a true 10-bit multi-channel ADC. See the Product Selection Guide in Table 1.1 for a quick reference of each MCUs' feature set. Each has two voltage comparators, a voltage reference, and an 8051-compatible microcontroller core with 16kbytes of FLASH memory and 1.25kbytes of RAM. 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.

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 2.8V-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 C8051F018 is available in the 64-pin TQFP (see block diagram in Figure 1.1). The C8051F019 is available in the 48-pin TQFP (see block diagram in Figure 1.2).

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	Voltage Comparators	Package
C8051F018	25	16k	1280	√	√	√	4	√	32	10	100	8	√	√	2	64TQFP
C8051F019	25	16k	1280	√	√	√	4	√	16	10	100	8	√	√	2	48TQFP

1.2. On-Board Memory

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

The CIP-51 additionally has a 1024 byte RAM block in the external data memory address space. This 1024 byte block can be addressed over the entire 64k external data memory address range (see Figure 1.5).

The MCU's program memory consists of 16k + 128 bytes of FLASH. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. The 512 bytes from addresses 0x3E00 to 0x3FFF are reserved for factory use. The additional 128 byte block is located at address 0x8000. See Figure 1.5 for the MCU system memory map.

Figure 1.5. On-Board Memory Map

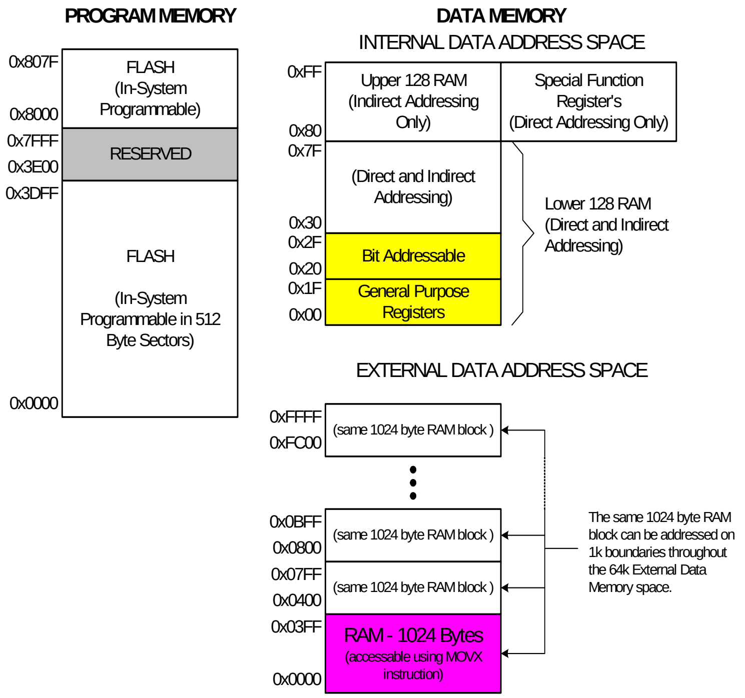


Figure 4.3. TQFP-48 Pinout Diagram

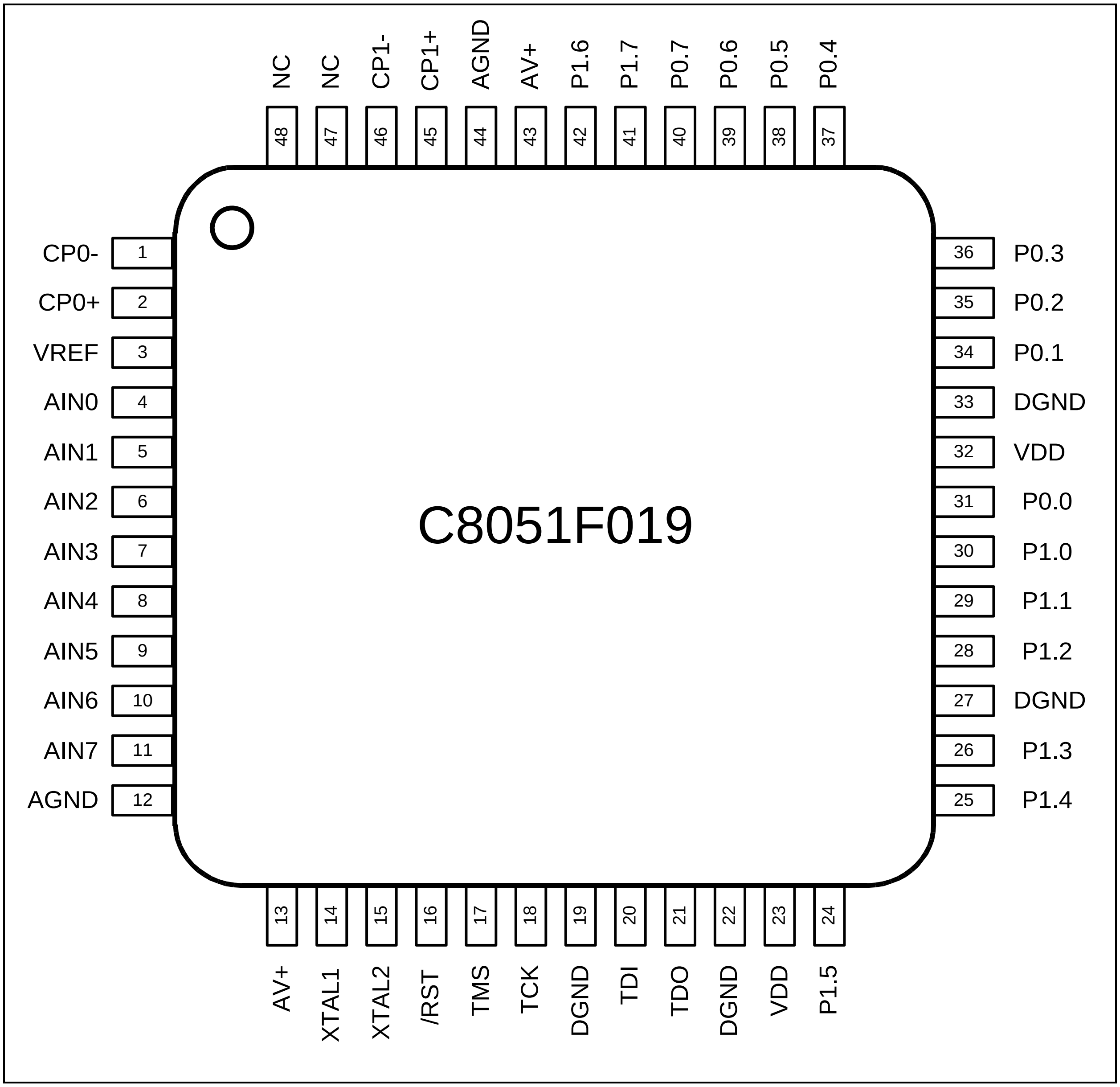


Figure 5.2. 10-Bit ADC Track and Conversion Example Timing

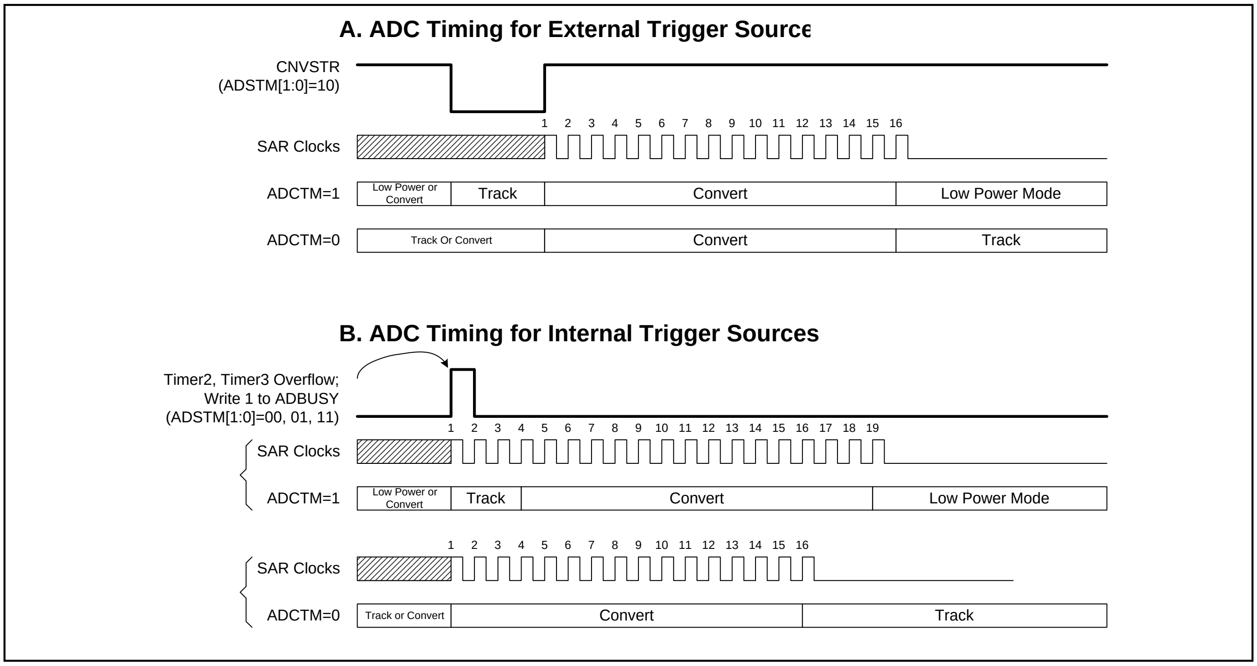


Figure 5.3. Temperature Sensor Transfer Function

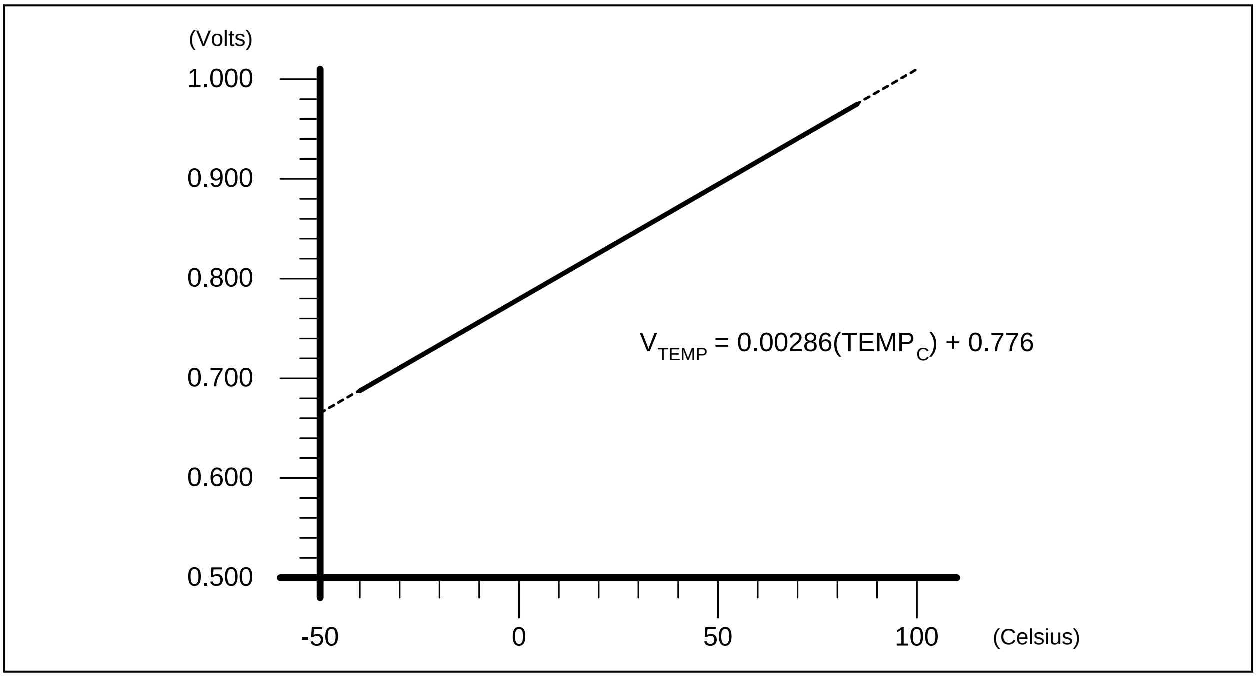


Figure 5.4. AMX0CF: AMUX Configuration Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBA

Bits7-4: UNUSED. Read = 0000b; Write = don’t care
Bit3: AIN67IC: AIN6, AIN7 Input Pair Configuration Bit
0: AIN6 and AIN7 are independent singled-ended inputs
1: AIN6, AIN7 are (respectively) +, - differential input pair
Bit2: AIN45IC: AIN4, AIN5 Input Pair Configuration Bit
0: AIN4 and AIN5 are independent singled-ended inputs
1: AIN4, AIN5 are (respectively) +, - differential input pair
Bit1: AIN23IC: AIN2, AIN3 Input Pair Configuration Bit
0: AIN2 and AIN3 are independent singled-ended inputs
1: AIN2, AIN3 are (respectively) +, - differential input pair
Bit0: AIN01IC: AIN0, AIN1 Input Pair Configuration Bit
0: AIN0 and AIN1 are independent singled-ended inputs
1: AIN0, AIN1 are (respectively) +, - differential input pair

NOTE: The ADC Data Word is in 2’s complement format for channels configured as differential.

Figure 5.7. ADC0CN: ADC Control Register

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 6.4. CPT1CN: Comparator 1 Control Register

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bit7: CP1EN: Comparator 1 Enable Bit
0: Comparator 1 Disabled.
1: Comparator 1 Enabled.

Bit6: CP1OUT: Comparator 1 Output State Flag
0: Voltage on CP1+ < CP1-
1: Voltage on CP1+ > CP1-

Bit5: CP1RIF: Comparator 1 Rising-Edge Interrupt Flag
0: No Comparator 1 Rising-Edge Interrupt has occurred since this flag was cleared
1: Comparator 1 Rising-Edge Interrupt has occurred since this flag was cleared

Bit4: CP1FIF: Comparator 1 Falling-Edge Interrupt Flag
0: No Comparator 1 Falling-Edge Interrupt has occurred since this flag was cleared
1: Comparator 1 Falling-Edge Interrupt has occurred since this flag was cleared

Bit3-2: CP1HYP1-0: Comparator 1 Positive Hysteresis Control Bits
00: Positive Hysteresis Disabled
01: Positive Hysteresis = 2mV
10: Positive Hysteresis = 4mV
11: Positive Hysteresis = 10mV

Bit1-0: CP1HYN1-0: Comparator 1 Negative Hysteresis Control Bits
00: Negative Hysteresis Disabled
01: Negative Hysteresis = 2mV
10: Negative Hysteresis = 4mV
11: Negative Hysteresis = 10mV

Figure 8.14. EIP2: Extended Interrupt Priority 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PXVLD	-	PX7	PX6	PX5	PX4	PADC0	PT3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF7

Bit7: PXVLD: External Clock Source Valid (XTLVLD) Interrupt Priority Control.
This bit sets the priority of the XTLVLD interrupt.
0: XTLVLD interrupt set to low priority level.
1: XTLVLD interrupt set to high priority level.

Bit6: Reserved: Must write 0. Reads 0.

Bit5: PX7: External Interrupt 7 Priority Control.
This bit sets the priority of the External Interrupt 7.
0: External Interrupt 7 set to low priority level.
1: External Interrupt 7 set to high priority level.

Bit4: PX6: External Interrupt 6 Priority Control.
This bit sets the priority of the External Interrupt 6.
0: External Interrupt 6 set to low priority level.
1: External Interrupt 6 set to high priority level.

Bit3: PX5: External Interrupt 5 Priority Control.
This bit sets the priority of the External Interrupt 5.
0: External Interrupt 5 set to low priority level.
1: External Interrupt 5 set to high priority level.

Bit2: PX4: External Interrupt 4 Priority Control.
This bit sets the priority of the External Interrupt 4.
0: External Interrupt 4 set to low priority level.
1: External Interrupt 4 set to high priority level.

Bit1: PADC0: ADC End of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC0 End of Conversion Interrupt.
0: ADC0 End of Conversion interrupt set to low priority level.
1: ADC0 End of Conversion interrupt set to high priority level.

Bit0: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupts.
0: Timer 3 interrupt set to low priority level.
1: Timer 3 interrupt set to high priority level.

10. EXTERNAL RAM

The C8051F018/9 includes 1024 bytes of RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN as shown in Figure 10.1). **Note: the MOVX instruction is also used for writes to the Flash memory. See Section 9 for details. The MOVX instruction accesses XRAM by default (i.e. PSTCL.0 = 0).**

For any of the addressing modes the upper 5-bits of the 16-bit external data memory address word are “don’t cares”. As a result, the 1024-byte RAM is mapped modulo style over the entire 64k external data memory address range. For example, the XRAM byte at address 0x0000 is also at address 0x0400, 0x0800, 0x0C00, 0x1000, etc. This is a useful feature when doing a linear memory fill, as the address pointer doesn’t have to be reset when reaching the RAM block boundary.

Figure 10.1. EMI0CN: External Memory Interface Control

R	R	R	R	R	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	PGSEL1	PGSEL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAF

Bits 7-2: Not Used – reads 000000b

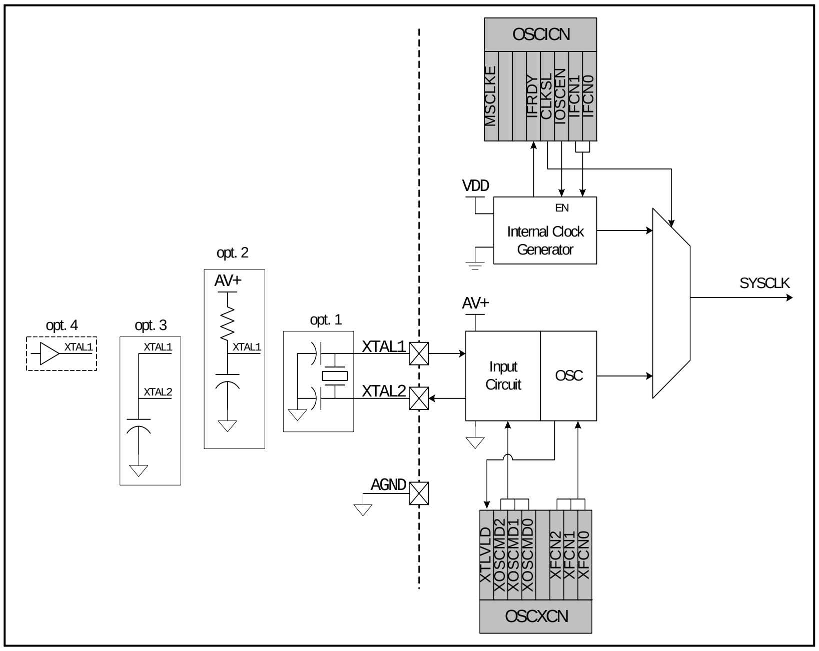
Bits 1-0: PGSEL[1:0]: XRAM Page Select Bits
The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. The upper 6-bits are “don’t cares”, so the 1k address blocks are repeated modulo over the entire 64k external data memory address space.
00: xxxxxx00b
01: xxxxxx01b
10: xxxxxx10b
11: xxxxxx11b

12. OSCILLATOR

Each MCU includes an internal oscillator and an external oscillator drive circuit, either of which can generate the system clock. The MCUs boot from the internal oscillator after any reset. The internal oscillator starts up instantly. It can be enabled/disabled and its frequency can be changed using the Internal Oscillator Control Register (OSCICN) as shown in Figure 12.2. The internal oscillator's electrical specifications are given in Table 12.1.

Both oscillators are disabled when the /RST pin is held low. The MCUs can run from the internal oscillator or external oscillator, and switch between the two at will using the CLKSL bit in the OSCICN Register. The external oscillator requires an external resonator, parallel-mode crystal, capacitor, or RC network connected to the XTAL1/XTAL2 pins (see Figure 12.1). The oscillator circuit must be configured for one of these sources in the OSCXCN register. An external CMOS clock can also provide the system clock via overdriving the XTAL1 pin. The XTAL1 and XTAL2 pins are 3.6V (not 5V) tolerant. The external oscillator can be left enabled and running even when the MCU has switched to using the internal oscillator.

Figure 12.1. Oscillator Diagram



15. SERIAL PERIPHERAL INTERFACE BUS

The Serial Peripheral Interface (SPI) provides access to a four-wire, full-duplex, serial bus. SPI supports the connection of multiple slave devices to a master device on the same bus. A separate slave-select signal (NSS) is used to select a slave device and enable a data transfer between the master and the selected slave. Multiple masters on the same bus are also supported. Collision detection is provided when two or more masters attempt a data transfer at the same time. The SPI can operate as either a master or a slave. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency.

When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS, and the serial input data synchronously with the system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less that 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of ¼ the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the system clock.

Figure 15.1. SPI Block Diagram

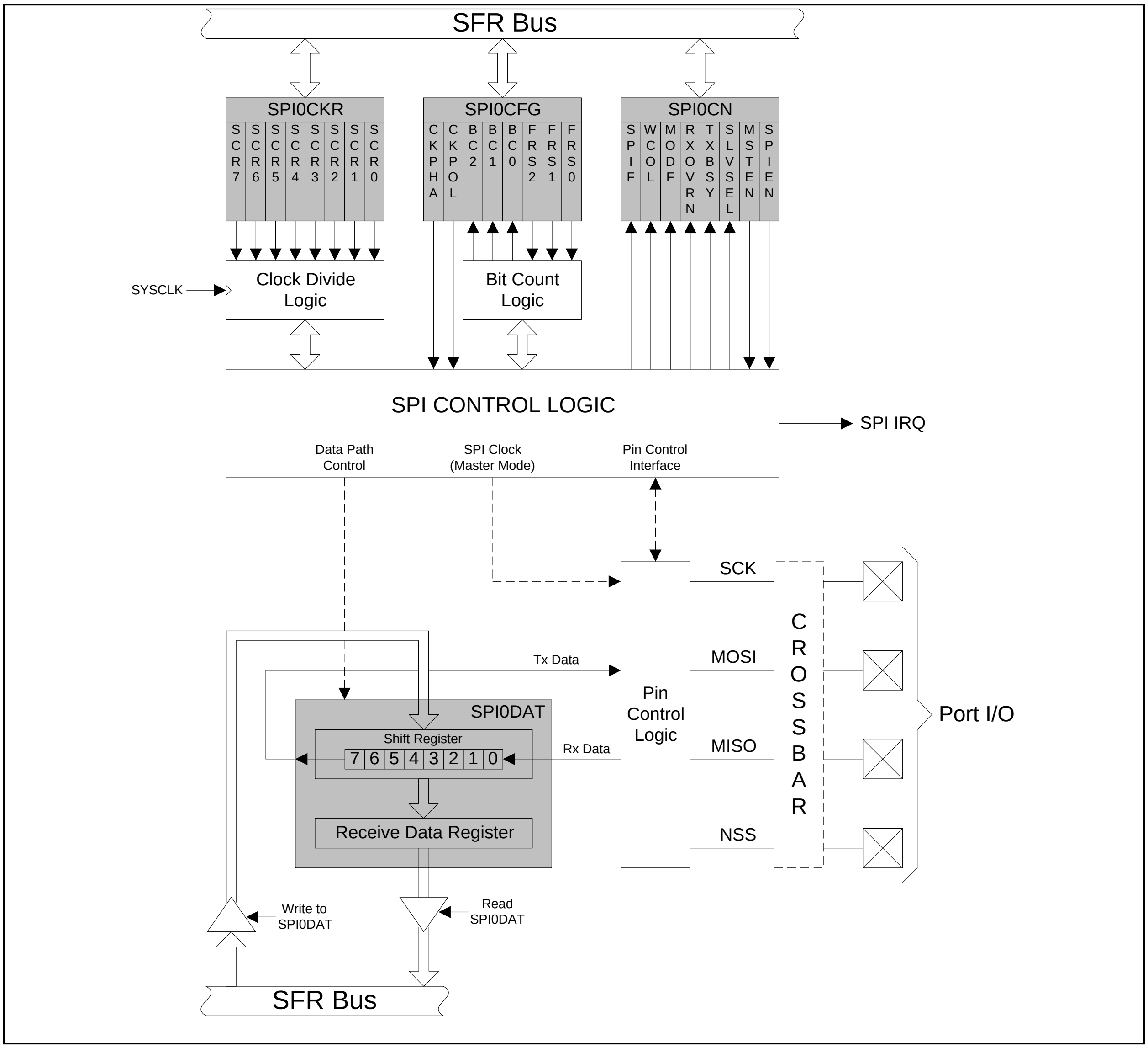
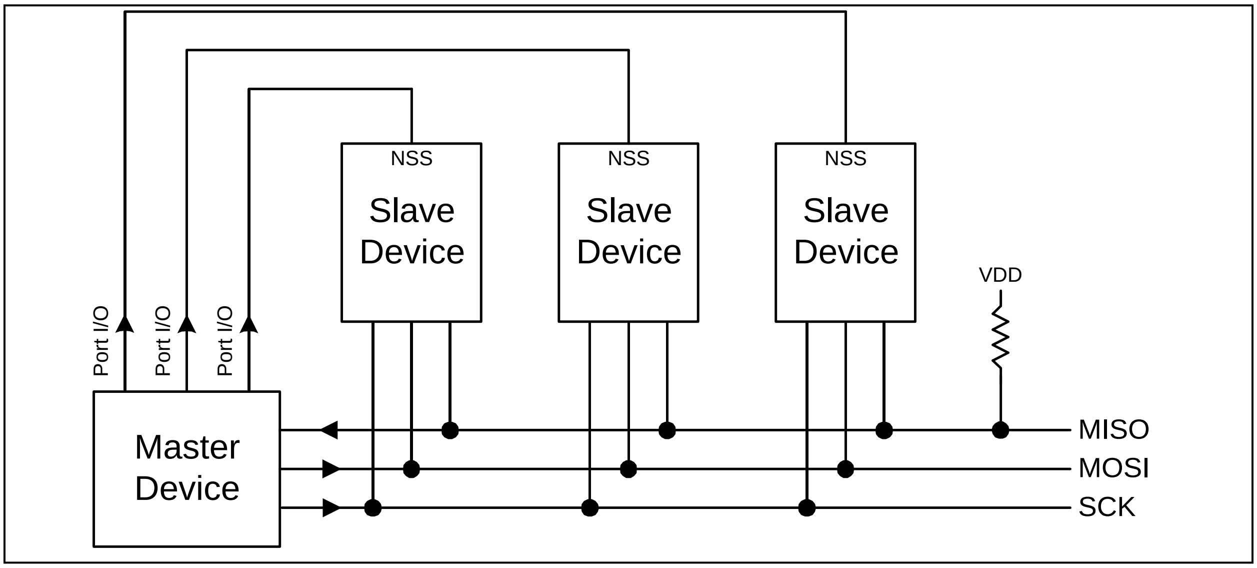


Figure 15.2. Typical SPI Interconnection



15.1. Signal Descriptions

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

15.1.1. 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.

15.1.2. 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.

15.1.3. 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.

15.1.4. Slave Select

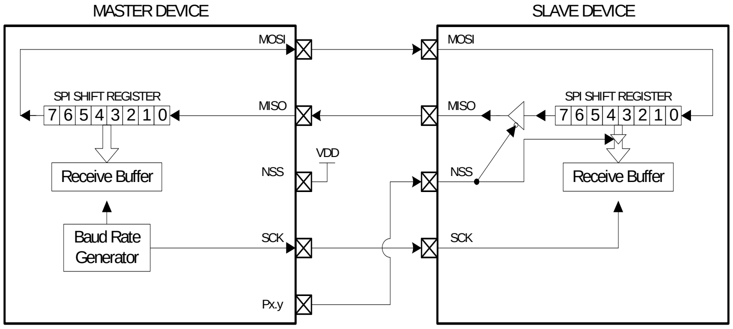
The slave select (NSS) signal is an input used to select the SPI module when in slave mode by a master, or to disable the SPI module when in master mode. When in slave mode, it is pulled low to initiate a data transfer and remains low for the duration of the transfer.

15.2. 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 15.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

Figure 15.3. 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

15.4. SPI Special Function Registers

The SPI is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI Bus are described in the following section.

Figure 15.5. SPI0CFG: SPI Configuration Register

R/W	R/W	R	R	R	R/W	R/W	R/W	Reset Value
CKPHA	CKPOL	BC2	BC1	BC0	SPIFRS2	SPIFRS1	SPIFRS0	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9A

Bit7: CKPHA: SPI Clock Phase.
This bit controls the SPI clock phase.
0: Data sampled on first edge of SCK period.
1: Data sampled on second edge of SCK period.

Bit6: CKPOL: SPI Clock Polarity.
This bit controls the SPI clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bits5-3: BC2-BC0: SPI Bit Count.
Indicates which of the up to 8 bits of the SPI word have been transmitted.

BC2-BC0			Bit Transmitted
0	0	0	Bit 0 (LSB)
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7 (MSB)

Bits2-0: SPIFRS2-SPIFRS0: SPI Frame Size.
These three bits determine the number of bits to shift in/out of the SPI shift register during a data transfer in master mode. They are ignored in slave mode.

SPIFRS			Bits Shifted
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

.

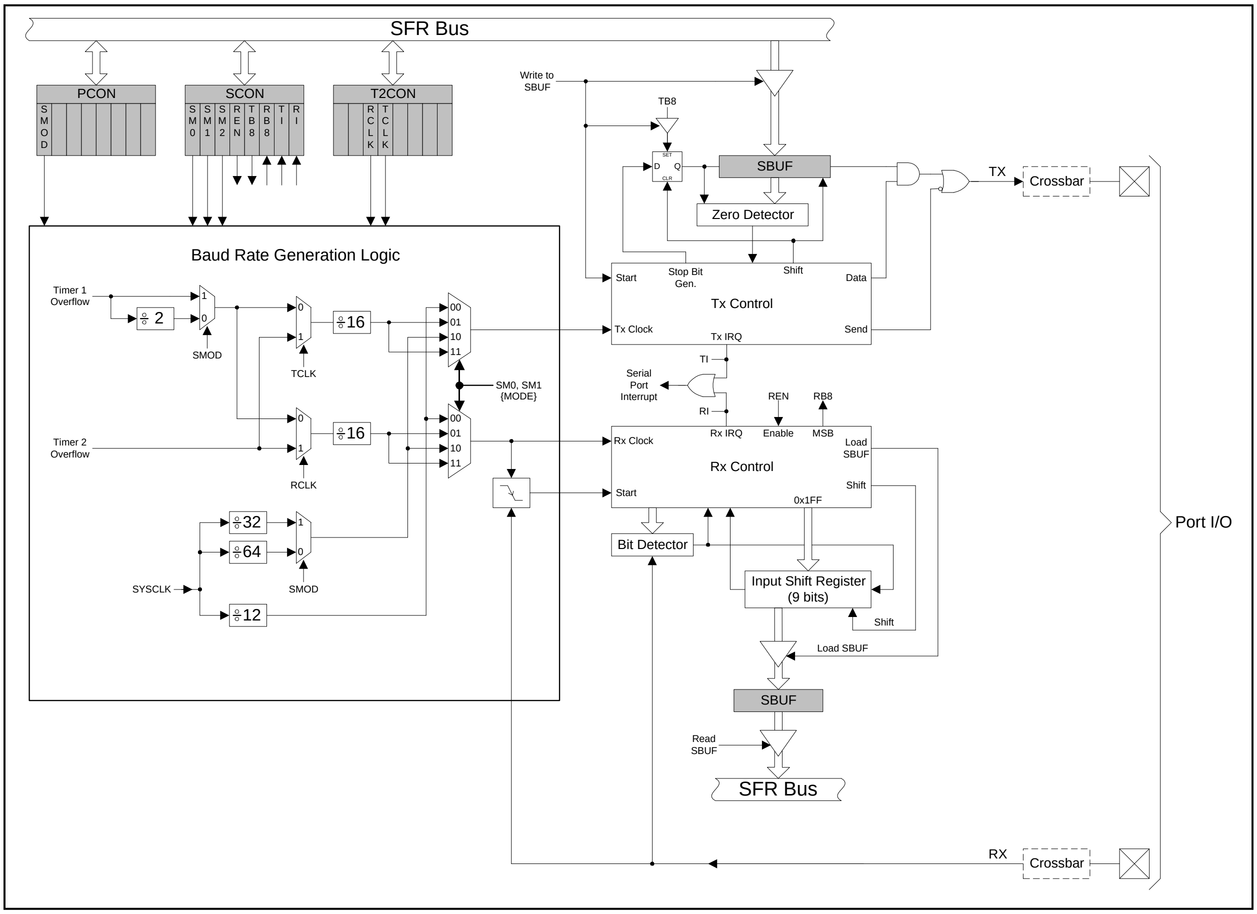
16. UART

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 16.1. UART Block Diagram



17.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

Timer 0 and Timer 1 behave differently in Mode 3. Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. It can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3, so with Timer 0 in Mode 3, Timer 1 can be turned off and on by switching it into and out of its Mode 3. When Timer 0 is in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used for baud rate generation. Refer to Section 16 (UART) for information on configuring Timer 1 for baud rate generation.

Figure 17.3. T0 Mode 3 Block Diagram

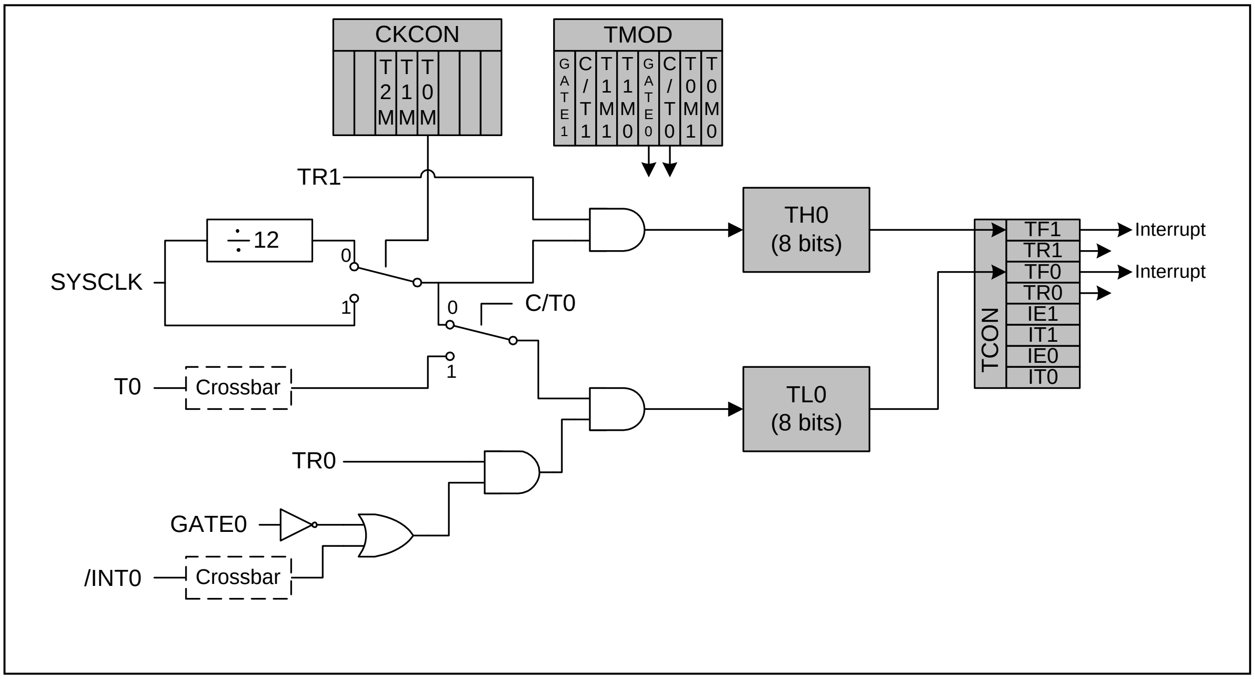


Figure 17.5. TMOD: Timer Mode Register

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

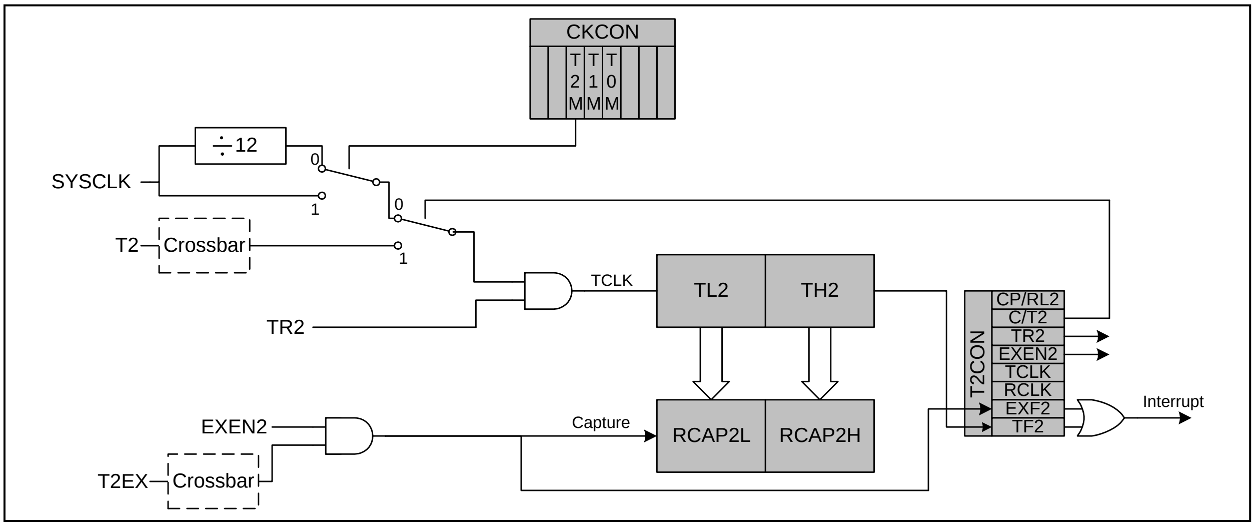
17.2.1. Mode 0: 16-bit Counter/Timer with Capture

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 17.11. T2 Mode 0 Block Diagram



18.1. Capture/Compare Modules

Each module can be configured to operate independently in one of four operation modes: Edge-triggered Capture, Software Timer, High Speed Output, or Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

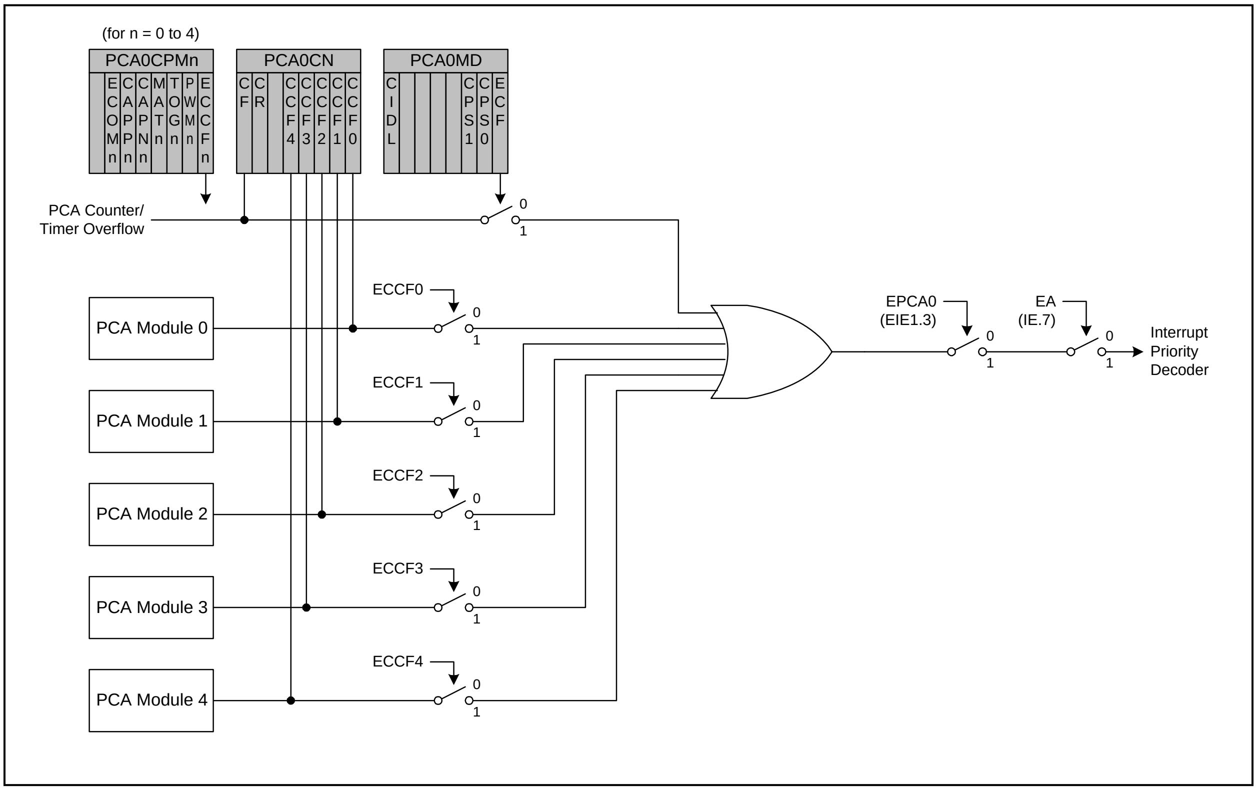
Table 18.1 summarizes the bit settings in the PCA0CPMn registers used to place the PCA capture/compare modules into different operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic 1. See Figure 18.2 for details on the PCA interrupt configuration.

Table 18.1. PCA0CPM Register Settings for PCA Capture/Compare Modules

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Operation Mode
X	1	0	0	0	0	X	Capture triggered by positive edge on CEXn
X	0	1	0	0	0	X	Capture triggered by negative edge on CEXn
X	1	1	0	0	0	X	Capture triggered by transition on CEXn
1	0	0	1	0	0	X	Software Timer
1	0	0	1	1	0	X	High Speed Output
1	0	0	X	0	1	X	Pulse Width Modulator

X = Don't Care

Figure 18.2. PCA Interrupt Block Diagram



18.2. PCA Counter/Timer

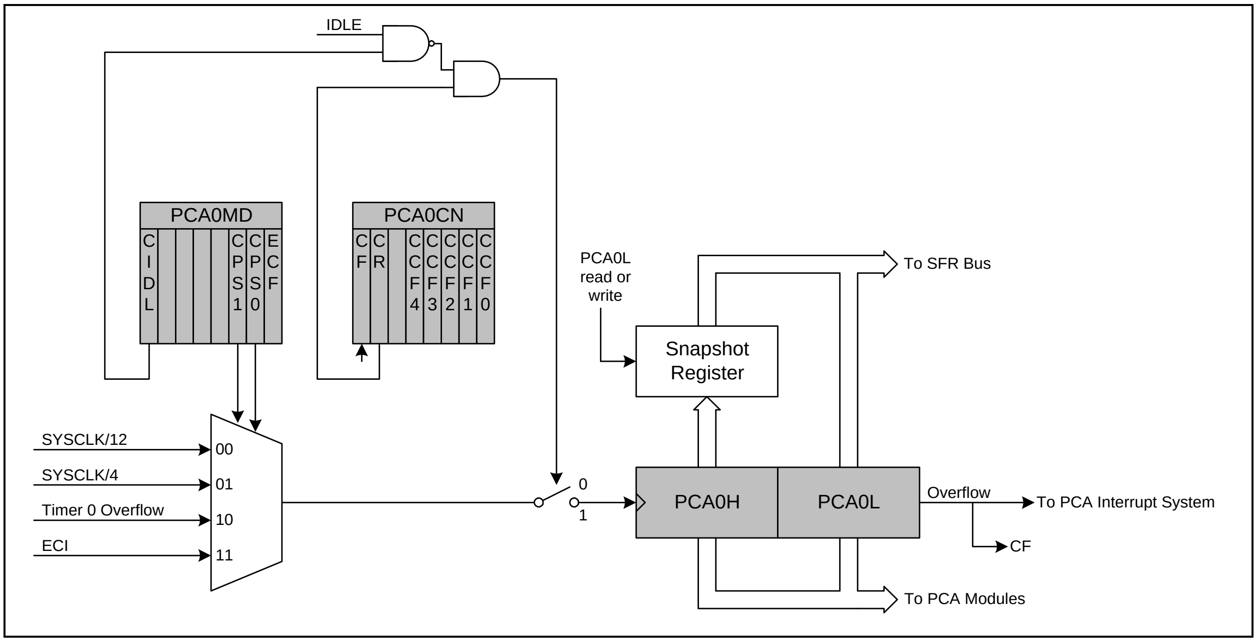
The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H at the same time. By reading the PCA0L Register first, this allows the PCA0H value to be held (at the time PCA0L was read) until the user reads the PCA0H Register. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS1 and CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 18.2.

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

Table 18.2. PCA Timebase Input Options

CPS1	CPS0	Timebase
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	Timer 0 overflow
1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)

Figure 18.7. PCA Counter/Timer Block Diagram


IDLE
PCA0MD
C I D L
C P S 1
C P S 0
E C F
PCA0CN
C F
C R
C C F 4
C C F 3
C C F 2
C C F 1
C C F 0
PCA0L read or write
To SFR Bus
Snapshot Register
SYSCLK/12
SYSCLK/4
Timer 0 Overflow
ECI
00
01
10
11
0
1
PCA0H
PCA0L
Overflow
To PCA Interrupt System
CF
To PCA Modules