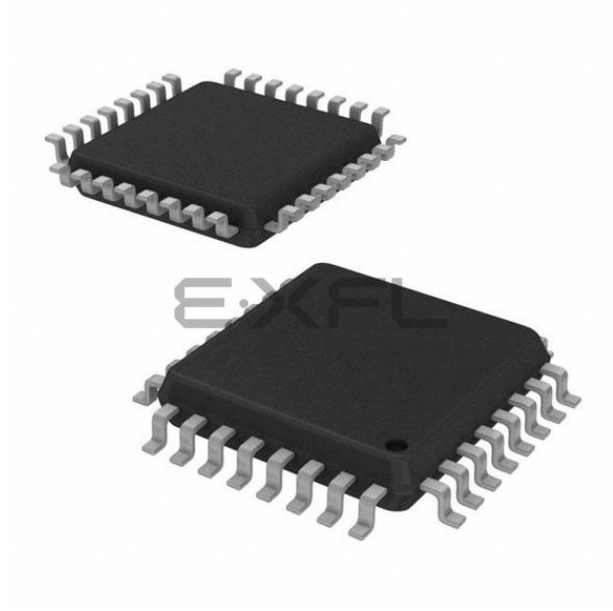


Welcome to E-XFL.COM

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	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 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/c8051f017-gq

1.1.3. Additional Features

The C8051F000 MCU family has several key enhancements both inside and outside the CIP-51 core to improve its overall performance and ease of use in the end applications.

The extended interrupt handler provides 21 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

There are up to seven reset sources for the MCU: an on-board VDD monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator 0, a forced software reset, the CNVSTR pin, and the /RST pin. The /RST pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the /RST pin. Each reset source except for the VDD monitor and Reset Input Pin may be disabled by the user in software. The WDT may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand alone clock generator which is used by default as the system clock after any reset. If desired, the clock source may be switched on the fly to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the fast (up to 16MHz) internal oscillator as needed.

Figure 1.5. On-Board Clock and Reset

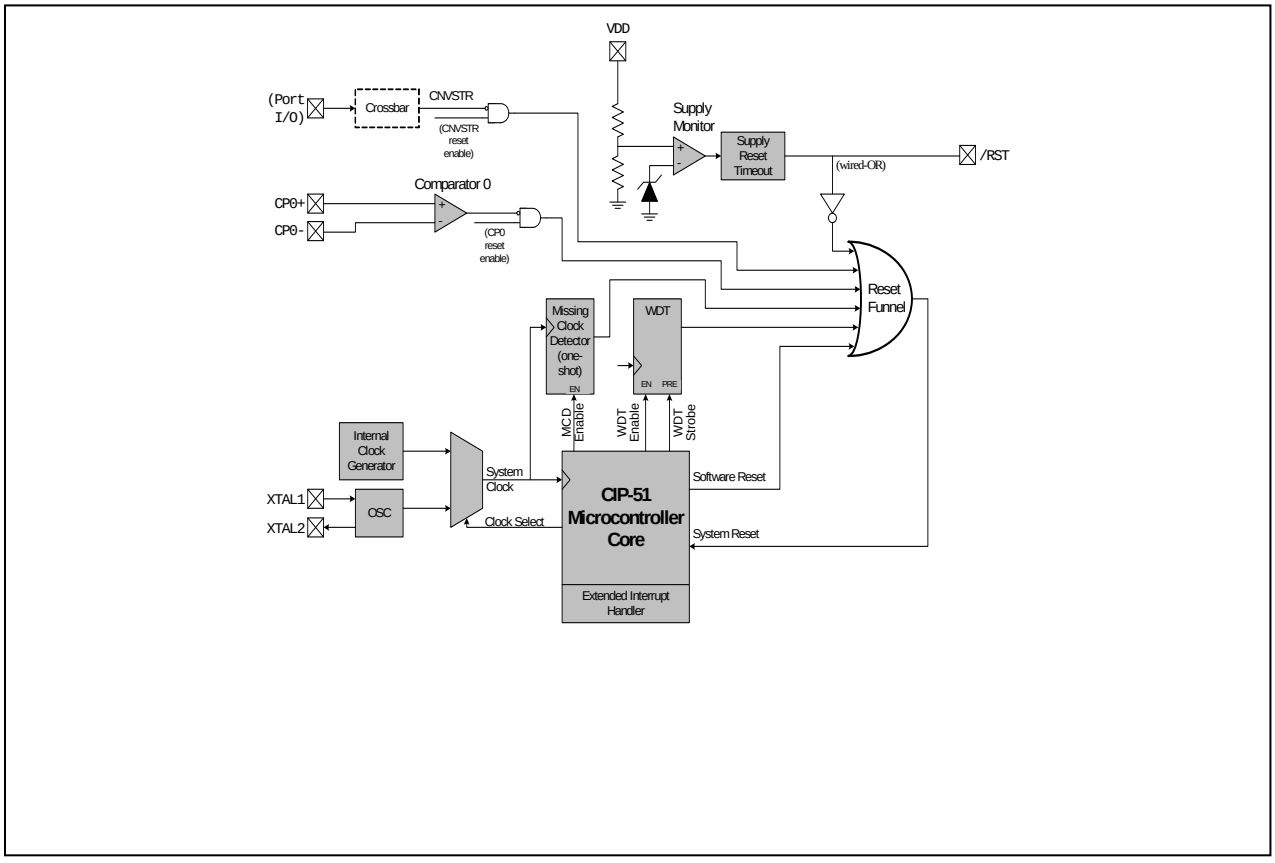


Figure 5.6. ADC0CF: ADC Configuration Register (C8051F00x)

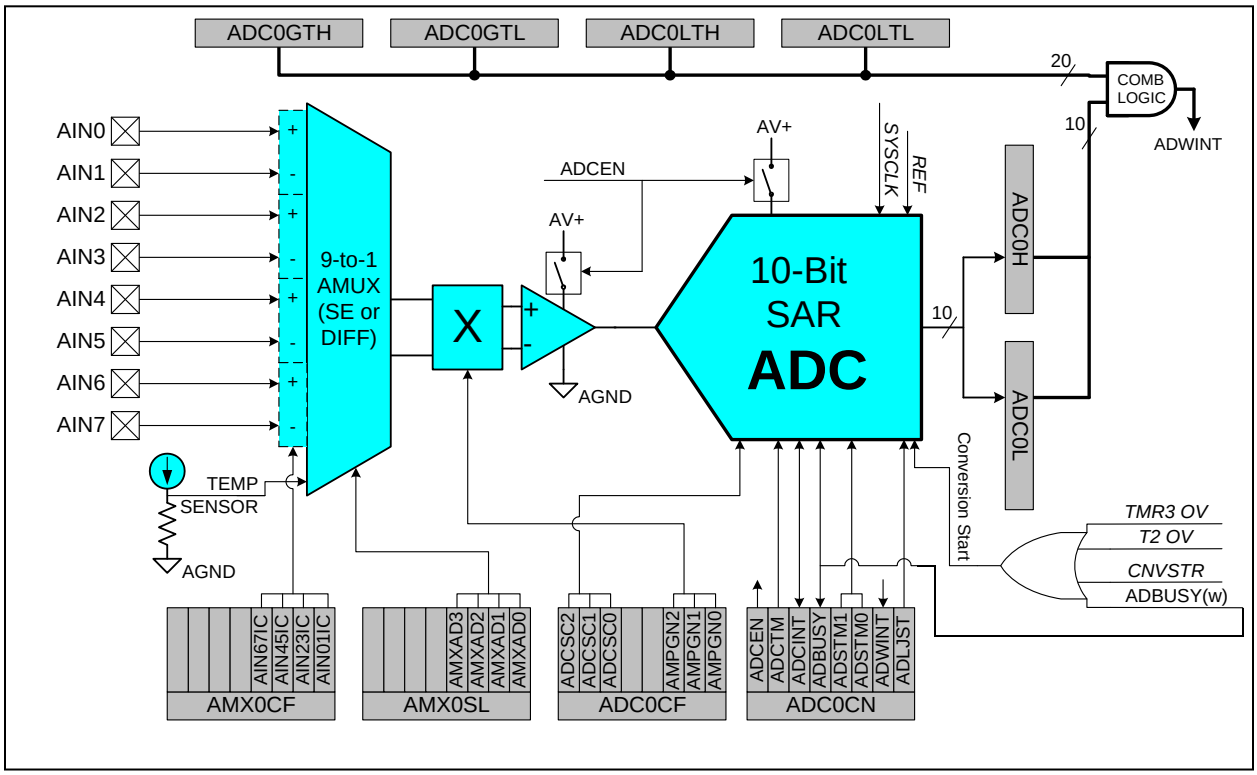
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	01100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

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 6.3. Temperature Sensor Transfer Function

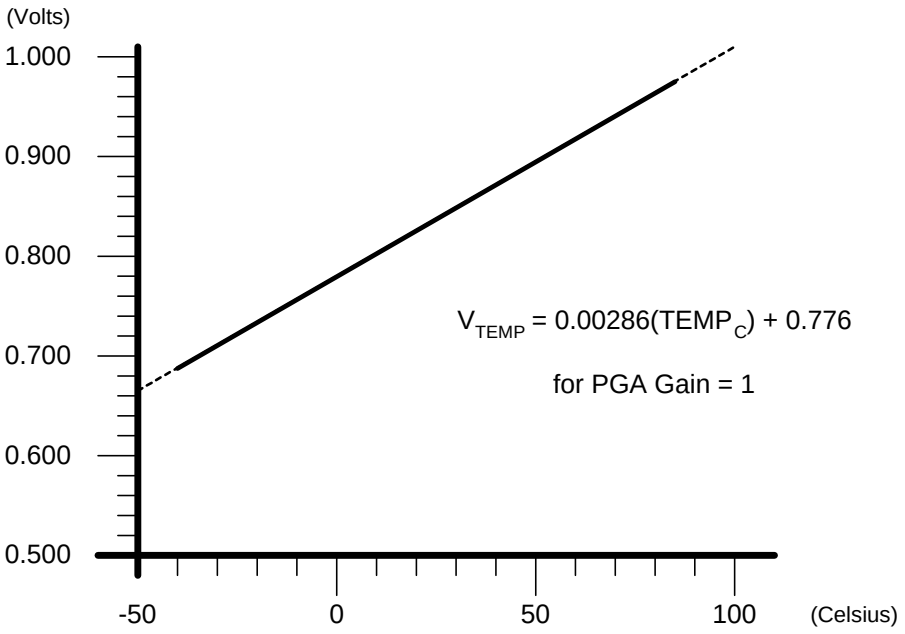


Figure 6.4. AMX0CF: AMUX Configuration Register (C8051F01x)

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 6.8. ADC0H: ADC Data Word MSB Register (C8051F01x)

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: 0xBF

Bits7-0: ADC Data Word Bits
For ADLJST = 1: Upper 8-bits of the 10-bit ADC Data Word.
For ADLJST = 0: Bits7-2 are the sign extension of Bit1. Bits 1-0 are the upper 2-bits of the 10-bit ADC Data Word.

Figure 6.9. ADC0L: ADC Data Word LSB Register (C8051F01x)

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: 0xBE

Bits7-0: ADC Data Word Bits
For ADLJST = 1: Bits7-6 are the lower 2-bits of the 10-bit ADC Data Word. Bits5-0 will always read 0.
For ADLJST = 0: Bits7-0 are the lower 8-bits of the 10-bit ADC Data Word.

NOTE: Resulting 10-bit ADC Data Word appears in the ADC Data Word Registers as follows:
ADC0H[1:0]:ADC0L[7:0], if ADLJST = 0
(ADC0H[7:2] will be sign extension of ADC0H.1 if a differential reading, otherwise = 000000b)

ADC0H[7:0]:ADC0L[7:6], if ADLJST = 1
(ADC0L[5:0] = 000000b)

EXAMPLE: ADC Data Word Conversion Map, AIN0 Input in Single-Ended Mode (AMX0CF=0x00, AMX0SL=0x00)

AIN0 – AGND (Volts)	**ADC0H:ADC0L** (ADLJST = 0)	**ADC0H:ADC0L** (ADLJST = 1)
REF x (1023/1024)	0x03FF	0xFFC0
REF x ½	0x0200	0x8000
REF x (511/1024)	0x01FF	0x7FC0
0	0x0000	0x0000

EXAMPLE: ADC Data Word Conversion Map, AIN0-AIN1 Differential Input Pair (AMX0CF=0x01, AMX0SL=0x00)

AIN0 – AIN1 (Volts)	**ADC0H:ADC0L** (ADLJST = 0)	**ADC0H:ADC0L** (ADLJST = 1)
REF x (511/512)	0x01FF	0x7FC0
0	0x0000	0x0000
-REF x (1/512)	0xFFFF	0xFFC0
-REF	0xFE00	0x8000

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

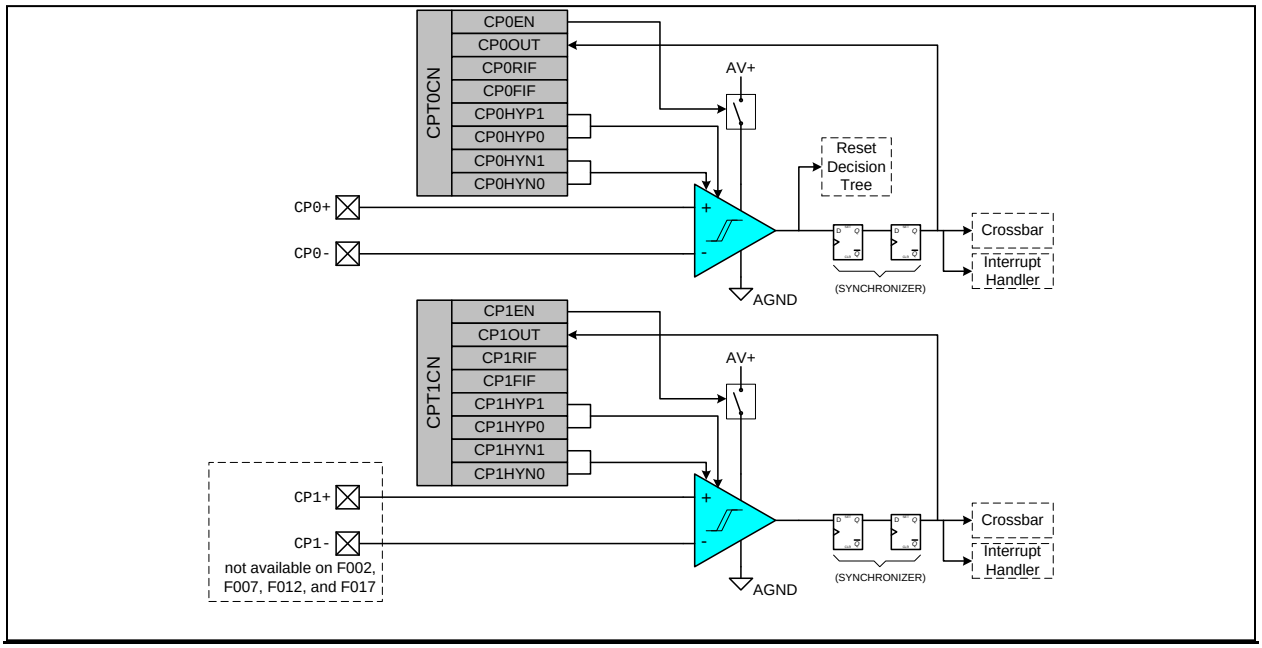


Figure 10.10. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS	PT1	PX1	PT0	PX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xB8

Bits7-6: UNUSED. Read = 11b, Write = don't care.

Bit5: PT2 Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupts.
0: Timer 2 interrupts set to low priority level.
1: Timer 2 interrupts set to high priority level.

Bit4: PS: Serial Port (UART) Interrupt Priority Control.
This bit sets the priority of the Serial Port (UART) interrupts.
0: UART interrupts set to low priority level.
1: UART interrupts set to high priority level.

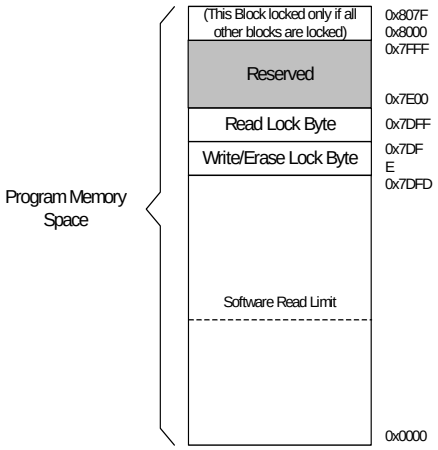
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupts.
0: Timer 1 interrupts set to low priority level.
1: Timer 1 interrupts set to high priority level.

Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupts.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.

Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupts.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.

Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupts.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

Figure 11.2. Flash Program Memory Security Bytes



Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

Bit	Memory Block
7	0x7000 - 0x7DFD
6	0x6000 - 0x6FFF
5	0x5000 - 0x5FFF
4	0x4000 - 0x4FFF
3	0x3000 - 0x3FFF
2	0x2000 - 0x2FFF
1	0x1000 - 0x1FFF
0	0x0000 - 0x0FFF

FLASH Read Lock Byte

Bits7-0: Each bit locks a corresponding block of memory. (Bit 7 is MSB.)

0: Read operations are locked (disabled) for corresponding block across the JTAG interface.

1: Read operations are unlocked (enabled) for corresponding block across the JTAG interface.

FLASH Write/Erase Lock Byte

Bits7-0: Each bit locks a corresponding block of memory.

0: Write/Erase operations are locked (disabled) for corresponding block across the JTAG interface.

1: Write/Erase operations are unlocked (enabled) for corresponding block across the JTAG interface.

FLASH Access Limit Register (FLACL)

The content of this register is used as the high byte of the 16-bit software read limit address. The 16-bit read limit address value is calculated as 0xNN00 where NN is replaced by the contents of this register. Software running at or above this address is prohibited from using the MOVX or MOVC instructions to read, write, or erase, locations below this address. Any attempts to read locations below this limit will return the value 0x00.

The lock bits can always be read and cleared to logic 0 regardless of the security setting applied to the block containing the security bytes. This allows additional blocks to be protected after the block containing the security bytes has been locked. However, the only means of removing a lock once set is to erase the entire program memory space by performing a JTAG erase operation (i.e. cannot be done in user firmware). **NOTE: Addressing either security byte while performing a JTAG erase operation will automatically initiate erasure of the entire program memory space (except for the reserved area). This erasure can only be performed via JTAG. If a non-security byte in the 0x7C00-0x7DFF page is addressed during erasure, only that page (including the security bytes) will be erased.**

The Flash Access Limit security feature (see Figure 11.3) protects proprietary program code and data from being read by software running on the C8051F005/06/07/15/16/17 MCUs. This feature provides support for OEMs that wish to program the MCU with proprietary value-added firmware before distribution. The value-added firmware can be protected while allowing additional code to be programmed in remaining program memory space later.

The Software Read Limit (SRL) is a 16-bit address that establishes two logical partitions in the program memory space. The first is an upper partition consisting of all the program memory locations at or above the SRL address, and the second is a lower partition consisting of all the program memory locations starting at 0x0000 up to (but excluding) the SRL address. Software in the upper partition can execute code in the lower partition, but is

prohibited from reading locations in the lower partition using the MOVC instruction. (Executing a MOVC instruction from the upper partition with a source address in the lower partition will always return a data value of 0x00.) Software running in the lower partition can access locations in both the upper and lower partition without restriction.

The Value-added firmware should be placed in the lower partition. On reset, control is passed to the value-added firmware via the reset vector. Once the value-added firmware completes its initial execution, it branches to a predetermined location in the upper partition. If entry points are published, software running in the upper partition may execute program code in the lower partition, but it cannot read the contents of the lower partition. Parameters may be passed to the program code running in the lower partition either through the typical method of placing them on the stack or in registers before the call or by placing them in prescribed memory locations in the upper partition.

The SRL address is specified using the contents of the Flash Access Register. The 16-bit SRL address is calculated as 0xNN00, where NN is the contents of the SRL Security Register. Thus, the SRL can be located on 256-byte boundaries anywhere in program memory space. However, the 512-byte erase sector size essentially requires that a 512 boundary be used. The contents of a non-initialized SRL security byte is 0x00, thereby setting the SRL address to 0x0000 and allowing read access to all locations in program memory space by default.

Figure 11.3. FLACL: Flash Access Limit (C8051F005/06/07/15/16/17 only)

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: 0xB7

Bits 7-0: FLACL: Flash Access Limit.
This register holds the high byte of the 16-bit program memory read/write/erase limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. A write to this register sets the Flash Access Limit. **This register can only be written once after any reset. Any subsequent writes are ignored until the next reset.**

14.1. External Crystal Example

If a crystal or ceramic resonator were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 1. For an ECS-110.5-20-4 crystal, the resonate frequency is 11.0592MHz, the intrinsic capacitance is 7pF, and the ESR is 60Ω. The compensation capacitors should be 33pF each, and the PWB parasitic capacitance is estimated to be 2pF. The appropriate External Oscillator Frequency Control value (XFCN) from the Crystal column in the table in Figure 14.3 (OSCXCN Register) should be 111b.

Because the oscillator detect circuitry needs time to settle after the crystal oscillator is enabled, software should wait at least 1ms between enabling the crystal oscillator and polling the XTLVLD bit. The recommend procedure is:
1. Enable the external oscillator
2. Wait at least 1 ms
3. Poll for XTLVLD '0' ==> '1'
4. Switch to the external oscillator

Switching to the external oscillator before the crystal oscillator has stabilized could result in unpredictable behavior.

NOTE: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device, keeping the traces as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

14.2. External RC Example

If an external RC network were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 2. The capacitor must be no greater than 100pF, but using a very small capacitor will increase the frequency drift due to the PWB parasitic capacitance. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100kHz, let R = 246kΩ and C = 50pF:

$$f = 1.23(10^3)/RC = 1.23(10^3) / [246 * 50] = 0.1MHz = 100kHz$$

$$XFCN \geq \log_2(f/25kHz)$$
$$XFCN \geq \log_2(100kHz/25kHz) = \log_2(4)$$
$$XFCN \geq 2, \text{ or code } 010$$

14.3. External Capacitor Example

If an external capacitor were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 3. The capacitor must be no greater than 100pF, but using a very small capacitor will increase the frequency inaccuracy due to the PWB parasitic capacitance. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation from the equations below. Assume AV+ = 3.0V and C = 50pF:

$$f = KF / (C * VDD) = KF / (50 * 3)$$
$$f = KF / 150$$

If a frequency of roughly 90kHz is desired, select the K Factor from the table in Figure 14.3 as KF = 13:

$$f = 13 /150 = 0.087MHz, \text{ or } 87kHz$$

Therefore, the XFCN value to use in this example is 011.

Figure 15.3. XBR0: Port I/O CrossBar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0OEN	ECIE	PCA0ME			UARTEN	SPI0OEN	SMB0OEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1

Bit7: CP0OEN: Comparator 0 Output Enable Bit
0: CP0 unavailable at Port pin.
1: CP0 routed to Port Pin.

Bit6: ECIE: PCA0 Counter Input Enable Bit
0: ECI unavailable at Port pin.
1: ECI routed to Port Pin.

Bits3-5: PCA0ME: PCA Module I/O Enable Bits
000: All PCA I/O unavailable at Port pins.
001: CEX0 routed to Port Pin.
010: CEX0, CEX1 routed to 2 Port Pins.
011: CEX0, CEX1, CEX2 routed to 3 Port Pins.
100: CEX0, CEX1, CEX2, CEX3 routed to 4 Port Pins.
101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to 5 Port Pins.
110: RESERVED
111: RESERVED

Bit2: UARTEN: UART I/O Enable Bit
0: UART I/O unavailable at Port pins.
1: RX, TX routed to 2 Port Pins.

Bit1: SPI0OEN: SPI Bus I/O Enable Bit
0: SPI I/O unavailable at Port pins.
1: MISO, MOSI, SCK, and NSS routed to 4 Port Pins.

Bit0: SMB0OEN: SMBus Bus I/O Enable Bit
0: SMBus I/O unavailable at P0.0, P0.1.
1: SDA routed to P0.0, SCL routed to P0.1.

Figure 15.4. XBR1: Port I/O CrossBar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1OEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: SYSCKE: SYSCLK Output Enable Bit
0: SYSCLK unavailable at Port pin.
1: SYSCLK output routed to Port Pin.

Bit6: T2EXE: T2EX Enable Bit
0: T2EX unavailable at Port pin.
1: T2EX routed to Port Pin.

Bit5: T2E: T2 Enable Bit
0: T2 unavailable at Port pin.
1: T2 routed to Port Pin.

Bit4: INT1E: /INT1 Enable Bit
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port Pin.

Bit3: T1E: T1 Enable Bit
0: T1 unavailable at Port pin.
1: T1 routed to Port Pin.

Bit2: INT0E: /INT0 Enable Bit
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port Pin.

Bit1: T0E: T0 Enable Bit
0: T0 unavailable at Port pin.
1: T0 routed to Port Pin.

Bit0: CP1OEN: Comparator 1 Output Enable Bit
0: CP1 unavailable at Port pin.
1: CP1 routed to Port Pin.

Figure 15.5. XBR2: Port I/O CrossBar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	-	-	-	-	-	CNVSTE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3

Bit7: WEAKPUD: Port I/O Weak Pull-up Disable Bit
0: Weak Pull-ups Enabled (except for Ports whose I/O are configured as push-pull)
1: Weak Pull-ups Disabled
Bit6: XBARE: Crossbar Enable Bit
0: Crossbar Disabled
1: Crossbar Enabled
Bits5-1: UNUSED. Read = 00000b, Write = don't care.
Bit0: CNVSTE: ADC Convert Start Input Enable Bit
0: CNVSTR unavailable at Port pin.
1: CNVSTR routed to Port Pin.

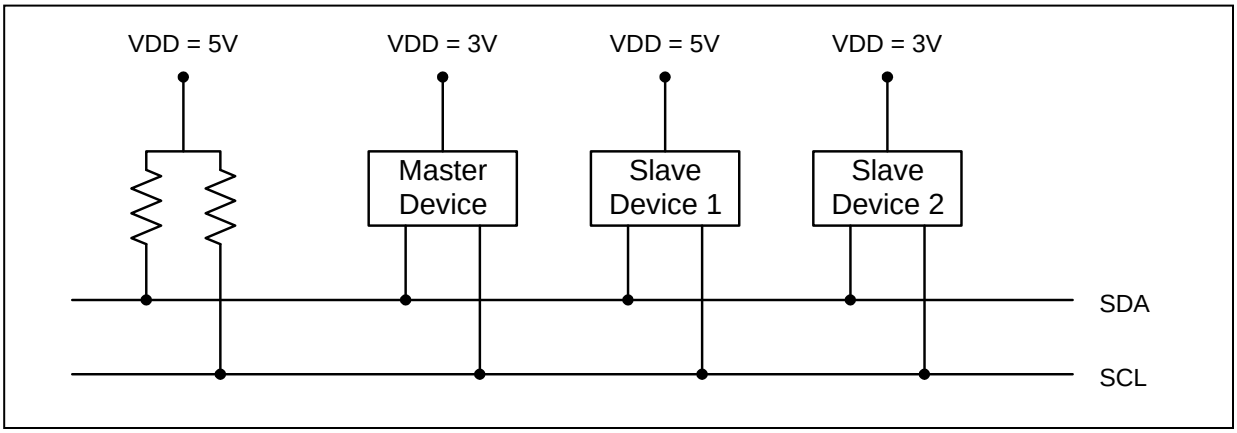
Example Usage of XBR0, XBR1, XBR2:
When selected, the digital resources fill the Port I/O pins in order (top to bottom as shown in Table 15.1) starting with P0.0 through P0.7, and then P1.0 through P1.7, and finally P2.0 through P2.7. If the digital resources are not mapped to the Port I/O pins, they default to their matching internal Port Register bits.

Example1: If XBR0 = 0x11, XBR1 = 0x00, and XBR2 = 0x40:
P0.0=SDA, P0.1=SCL, P0.2=CEX0, P0.3=CEX1, P0.4 … P2.7 map to corresponding Port I/O.

Example2: If XBR0 = 0x80, XBR1 = 0x04, and XBR2 = 0x41:
P0.0=CP0, P0.1=/INT0, P0.2 = CNVSTR, P0.3 … P2.7 map to corresponding Port I/O.

Figure 16.2 shows a typical SMBus configuration. The SMBus interface will work at any voltage between 3.0V and 5.0V and different devices on the bus may operate at different voltage levels. The SCL (serial clock) and SDA (serial data) lines are bi-directional. They must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. When the bus is free, both lines are pulled high. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus will not exceed 300ns and 1000ns, respectively.

Figure 16.2. Typical SMBus Configuration



16.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. *The I²C-bus and how to use it (including specifications)*, Philips Semiconductor.
2. *The I²C-Bus Specification -- Version 2.0*, Philips Semiconductor.
3. *System Management Bus Specification -- Version 1.1*, SBS Implementers Forum.

Figure 16.4. SMB0CN: SMBus Control Register

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
BUSY	ENSMB	STA	STO	SI	AA	FTE	TOE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xC0

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

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

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

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

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

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

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

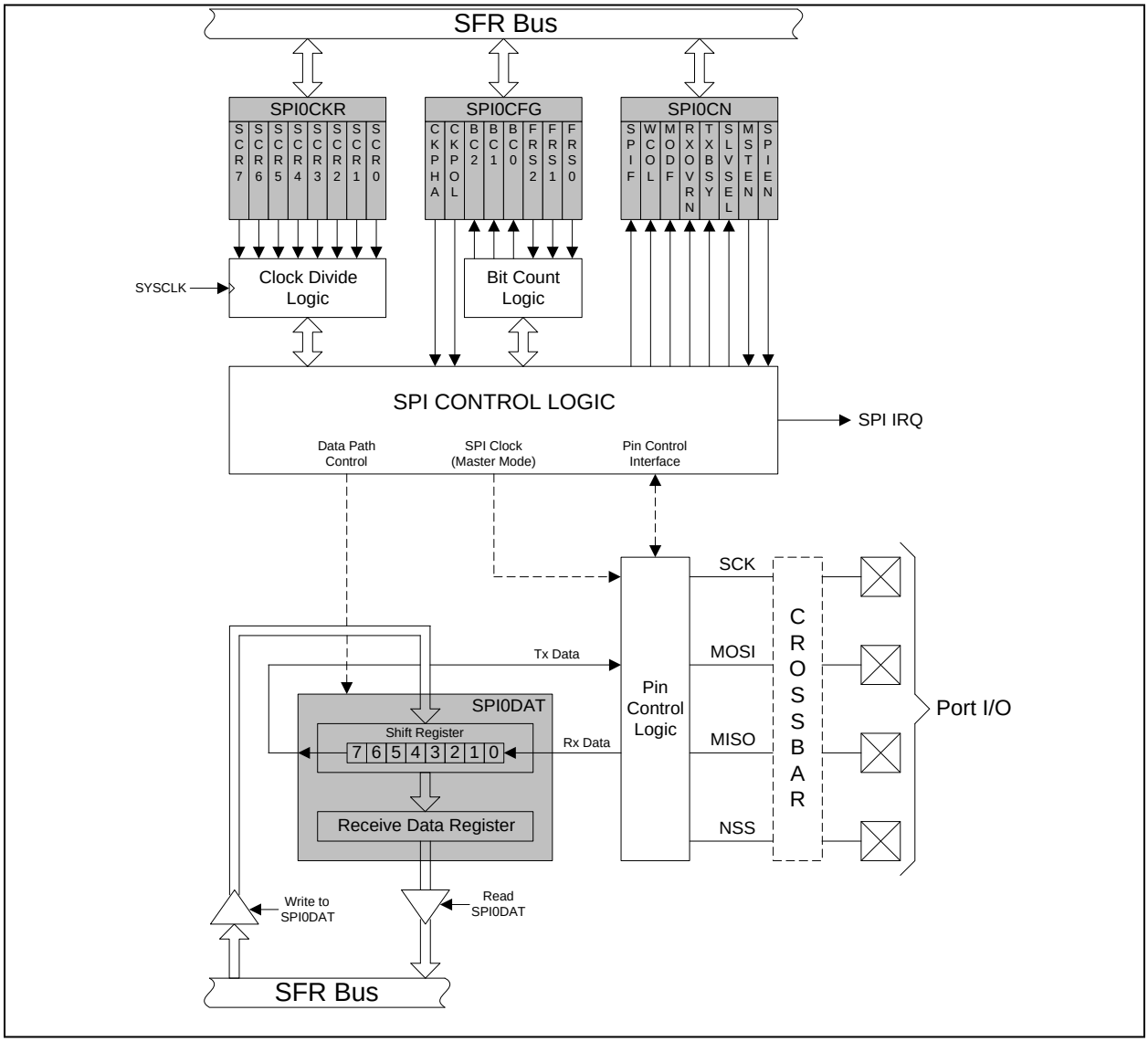
Bit0: TOE: SMBus Timeout Enable Bit
0: No timeout when SCL is low.
1: Timeout when SCL low time exceeds limit specified by Timer 3, if enabled.

17. 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 17.1. SPI Block Diagram



19. TIMERS

Each MCU implements four counter/timers: three are 16-bit counter/timers compatible with those found in the standard 8051, and one is a 16-bit timer for use with the ADC, SMBus, or for general purpose use. These can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers additional capabilities not available in Timers 0 and 1. Timer 3 is similar to Timer 2, but without the capture or Baud Rate Generator modes.

Timer 0 and Timer 1:	Timer 2:	Timer 3:
13-bit counter/timer	16-bit counter/timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer	16-bit counter/timer with capture	
8-bit counter/timer with auto-reload	Baud rate generator	
Two 8-bit counter/timers (Timer 0 only)		

When functioning as a timer, the counter/timer registers are incremented on each clock tick. Clock ticks are derived from the system clock divided by either one or twelve as specified by the Timer Clock Select bits (T2M-T0M) in CKCON. The twelve-clocks-per-tick option provides compatibility with the older generation of the 8051 family. Applications that require a faster timer can use the one-clock-per-tick option.

When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin for T0, T1, or T2. Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is sampled.

19.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are accessed and controlled through SFRs. Each counter/timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control (TCON) register is used to enable Timer 0 and Timer 1 as well as indicate their status. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits M1-M0 in the Counter/Timer Mode (TMOD) register. Each timer can be configured independently. Following is a detailed description of each operating mode.

19.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as a 13-bit counter/timer in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. Clearing C/T selects the system clock as the input for the timer. When C/T0 is set to logic 1, high-to-low transitions at the selected input pin increment the timer register. (Refer to Port I/O Section 15.1 for information on selecting and configuring external I/O pins.)

Figure 19.6. CKCON: Clock Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	T2M	T1M	T0M	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8E

Bits7-6: UNUSED. Read = 00b, Write = don't care.

Bit5: T2M: Timer 2 Clock Select.
This bit controls the division of the system clock supplied to Timer 2. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T2 = 1).
0: Timer 2 uses the system clock divided by 12.
1: Timer 2 uses the system clock.

Bit4: T1M: Timer 1 Clock Select.
This bit controls the division of the system clock supplied to Timer 1.
0: Timer 1 uses the system clock divided by 12.
1: Timer 1 uses the system clock.

Bit3: T0M: Timer 0 Clock Select.
This bit controls the division of the system clock supplied to Counter/Timer 0.
0: Counter/Timer uses the system clock divided by 12.
1: Counter/Timer uses the system clock.

Bits2-0: Reserved. Read = 000b, Must Write = 000.

20.1. 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.

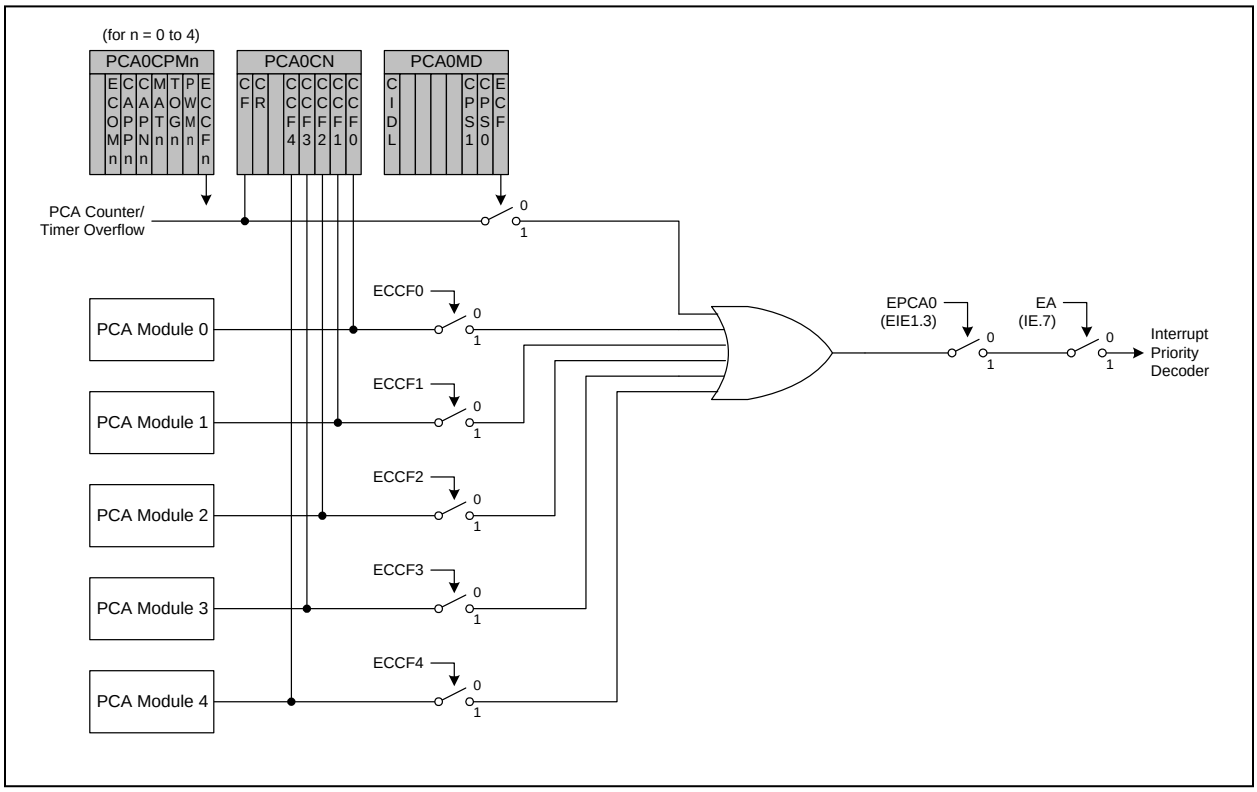
Table 20.1 summarizes the bit settings in the PCA0CPMn registers used to place the PCA capture/compare modules into different operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts 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 20.2 for details on the PCA interrupt configuration.

Table 20.1. 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 20.2. PCA Interrupt Block Diagram



20.2. 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 20.2.

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 20.2. 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 20.7. PCA Counter/Timer Block Diagram