

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	15
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f855-c-imr

1.2. Typical Performance Curves

1.2.1. Operating Supply Current

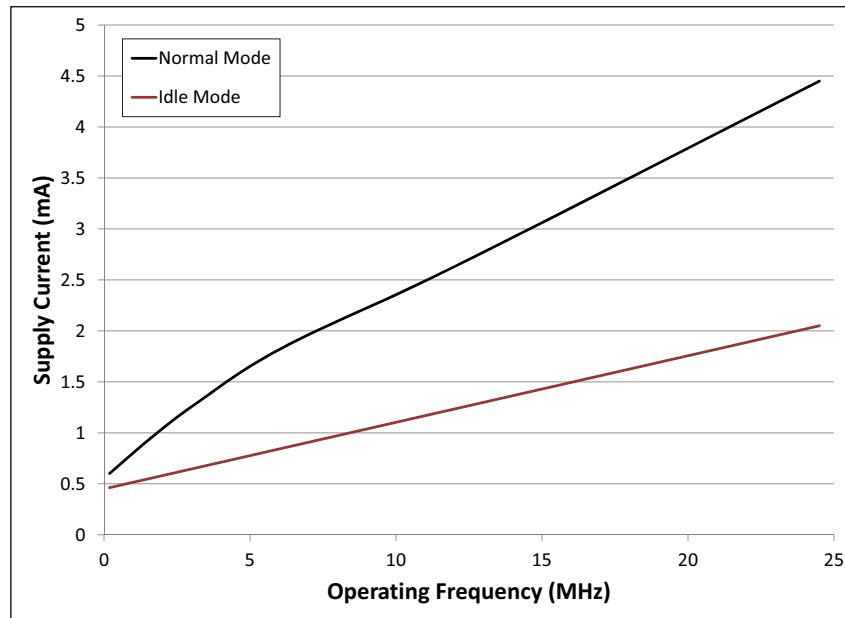


Figure 1.1. Typical Operating Current Running From 24.5 MHz Internal Oscillator

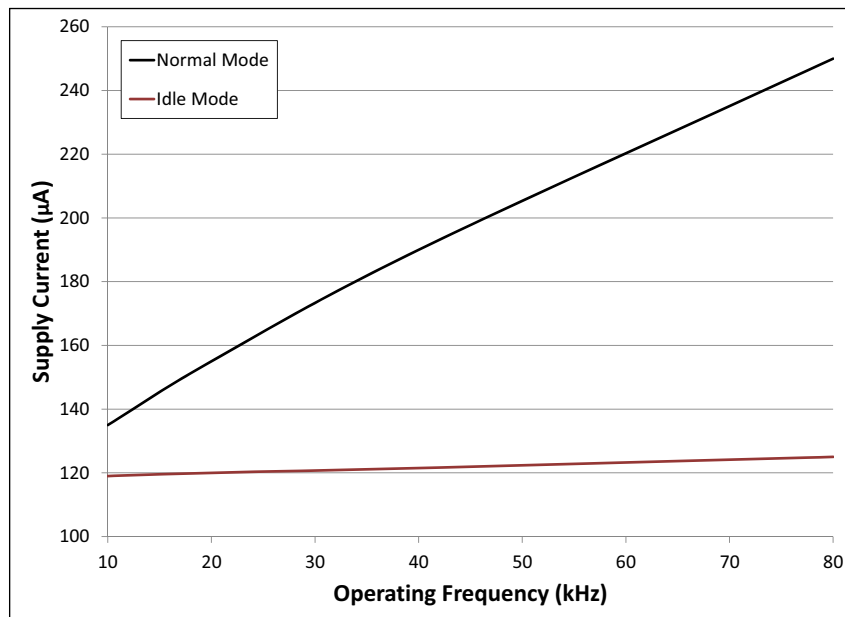


Figure 1.2. Typical Operating Current Running From 80 kHz Internal Oscillator

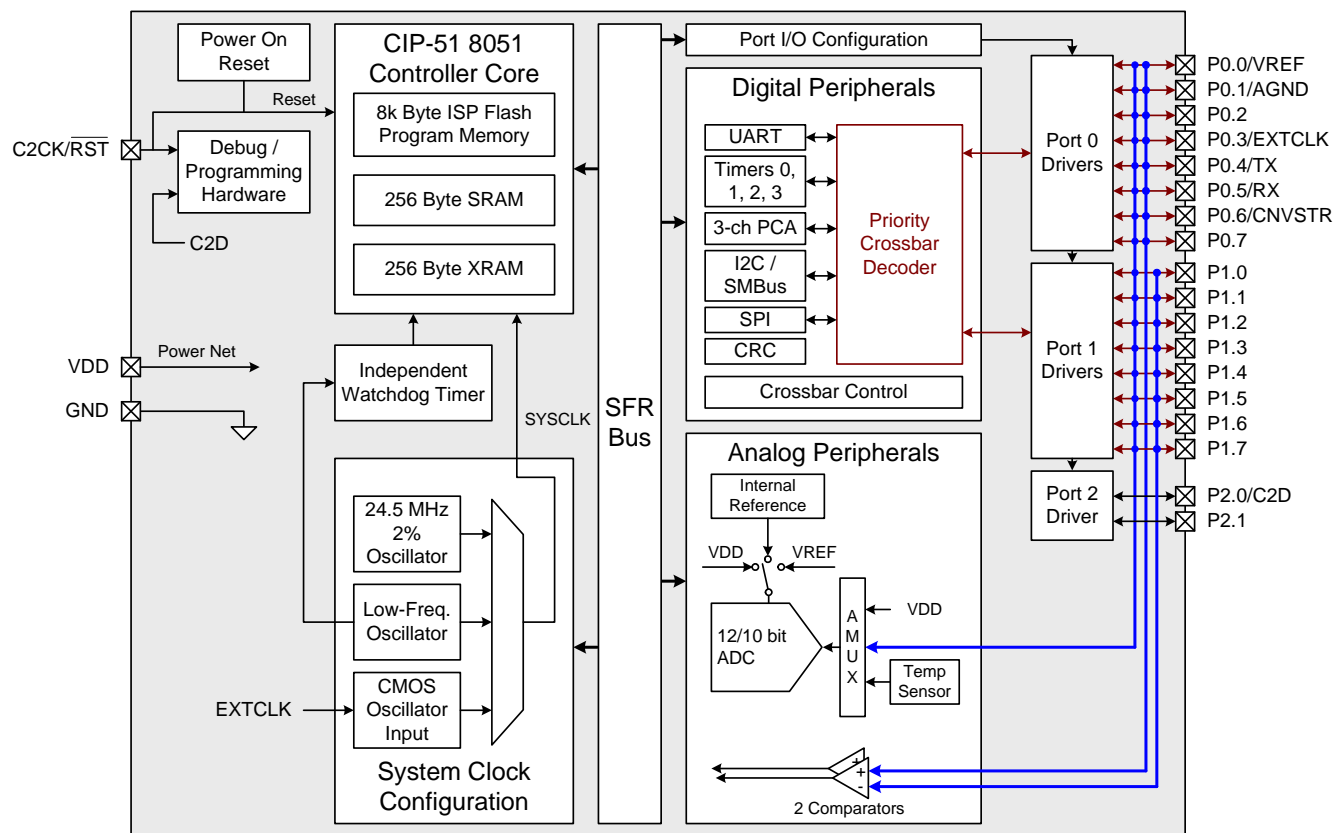


Figure 2.1. C8051F85x/86x Family Block Diagram (QSOP-24 Shown)

Register 11.2. DERIVID: Derivative Identification

Bit	7	6	5	4	3	2	1	0
Name	DERIVID							
Type	R							
Reset	X	X	X	X	X	X	X	X
SFR Address: 0xAD								

Table 11.3. DERIVID Register Bit Descriptions

Bit	Name	Function
7:0	DERIVID	<p>Derivative ID.</p> <p>This read-only register returns the 8-bit derivative ID, which can be used by firmware to identify which device in the product family the code is executing on. The '{R}' tag in the part numbers below indicates the device revision letter in the ordering code.</p> <p>0xD0: C8051F850-{R}-GU 0xD1: C8051F851-{R}-GU 0xD2: C8051F852-{R}-GU 0xD3: C8051F853-{R}-GU 0xD4: C8051F854-{R}-GU 0xD5: C8051F855-{R}-GU 0xE0: C8051F860-{R}-GS 0xE1: C8051F861-{R}-GS 0xE2: C8051F862-{R}-GS 0xE3: C8051F863-{R}-GS 0xE4: C8051F864-{R}-GS 0xE5: C8051F865-{R}-GS 0xF0: C8051F850-{R}-GM 0xF1: C8051F851-{R}-GM 0xF2: C8051F852-{R}-GM 0xF3: C8051F853-{R}-GM 0xF4: C8051F854-{R}-GM 0xF5: C8051F855-{R}-GM</p>

Register 12.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PMAT	PSMB0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

SFR Address: 0xF3

Table 12.5. EIP1 Register Bit Descriptions

Bit	Name	Function
7	PT3	Timer 3 Interrupt Priority Control. This bit sets the priority of the Timer 3 interrupt. 0: Timer 3 interrupts set to low priority level. 1: Timer 3 interrupts set to high priority level.
6	PCP1	Comparator1 (CP1) Interrupt Priority Control. This bit sets the priority of the CP1 interrupt. 0: CP1 interrupt set to low priority level. 1: CP1 interrupt set to high priority level.
5	PCP0	Comparator0 (CP0) Interrupt Priority Control. This bit sets the priority of the CP0 interrupt. 0: CP0 interrupt set to low priority level. 1: CP0 interrupt set to high priority level.
4	PPCA0	Programmable Counter Array (PCA0) Interrupt Priority Control. This bit sets the priority of the PCA0 interrupt. 0: PCA0 interrupt set to low priority level. 1: PCA0 interrupt set to high priority level.
3	PADC0	ADC0 Conversion Complete Interrupt Priority Control. This bit sets the priority of the ADC0 Conversion Complete interrupt. 0: ADC0 Conversion Complete interrupt set to low priority level. 1: ADC0 Conversion Complete interrupt set to high priority level.
2	PWADC0	ADC0 Window Comparator Interrupt Priority Control. This bit sets the priority of the ADC0 Window interrupt. 0: ADC0 Window interrupt set to low priority level. 1: ADC0 Window interrupt set to high priority level.
1	PMAT	Port Match Interrupt Priority Control. This bit sets the priority of the Port Match Event interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.

Table 12.5. EIP1 Register Bit Descriptions

Bit	Name	Function
0	PSMB0	SMBus (SMB0) Interrupt Priority Control. This bit sets the priority of the SMB0 interrupt. 0: SMB0 interrupt set to low priority level. 1: SMB0 interrupt set to high priority level.

Register 14.4. ADC0AC: ADC0 Accumulator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD12BE	ADAE	ADSJST			ADRPT		
Type	RW	RW	RW			RW		
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xB3								

Table 14.7. ADC0AC Register Bit Descriptions

Bit	Name	Function
7	AD12BE	12-Bit Mode Enable. Enables 12-bit Mode. In 12-bit mode, the ADC throughput is reduced by a factor of 4. 0: 12-bit Mode Disabled. 1: 12-bit Mode Enabled.
6	ADAE	Accumulate Enable. Enables multiple conversions to be accumulated when burst mode is disabled. 0: ADC0H:ADC0L contain the result of the latest conversion when Burst Mode is disabled. 1: ADC0H:ADC0L contain the accumulated conversion results when Burst Mode is disabled. Software must write 0x0000 to ADC0H:ADC0L to clear the accumulated result.
5:3	ADSJST	Accumulator Shift and Justify. Specifies the format of data read from ADC0H:ADC0L. All remaining bit combinations are reserved. 000: Right justified. No shifting applied. 001: Right justified. Shifted right by 1 bit. 010: Right justified. Shifted right by 2 bits. 011: Right justified. Shifted right by 3 bits. 100: Left justified. No shifting applied. 101-111: Reserved.
2:0	ADRPT	Repeat Count. Selects the number of conversions to perform and accumulate in Burst Mode. This bit field must be set to 000 if Burst Mode is disabled. 000: Perform and Accumulate 1 conversion (not used in 12-bit mode). 001: Perform and Accumulate 4 conversions (1 conversion in 12-bit mode). 010: Perform and Accumulate 8 conversions (2 conversions in 12-bit mode). 011: Perform and Accumulate 16 conversions (4 conversions in 12-bit mode). 100: Perform and Accumulate 32 conversions (8 conversions in 12-bit mode). 101: Perform and Accumulate 64 conversions (16 conversions in 12-bit mode). 110-111: Reserved.

20.4.4. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8 through 11-bit) PWM modes. The entire PCA0CP register is used to determine the duty cycle in 16-bit PWM mode.

To output a varying duty cycle, new value writes should be synchronized with the PCA CCFn match flag to ensure seamless updates.

16-Bit PWM mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, the match interrupt flag should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a match edge or up edge occurs. The CF flag in PCA0CN can be used to detect the overflow or down edge.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

20.6. PCA Control Registers

Register 20.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	Reserved			CCF2	CCF1	CCF0
Type	RW	RW	R			RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD8 (bit-addressable)								

Table 20.3. PCA0CN Register Bit Descriptions

Bit	Name	Function
7	CF	PCA Counter/Timer Overflow Flag. Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	PCA Counter/Timer Run Control. This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled. 1: PCA Counter/Timer enabled.
5:3	Reserved	Must write reset value.
2	CCF2	PCA Module 2 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	PCA Module 1 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	PCA Module 0 Capture/Compare Flag. This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

21. Port I/O (Port 0, Port 1, Port 2, Crossbar, and Port Match)

Digital and analog resources on the C8051F85x/86x family are externally available on the device's multi-purpose I/O pins. Port pins P0.0-P1.7 can be defined as general-purpose I/O (GPIO), assigned to one of the internal digital resources through the crossbar, or assigned to an analog function. Port pins P2.0 and P2.1 can be used as GPIO. Port pin P2.0 is shared with the C2 Interface Data signal (C2D). The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a priority crossbar decoder. Note that the state of a port I/O pin can always be read in the corresponding port latch, regardless of the crossbar settings.

The crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 21.2 and Figure 21.3). The registers XBR0, XBR1 and XBR2 are used to select internal digital functions.

The port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1). Additionally, each bank of port pins (P0, P1, and P2) has two selectable drive strength settings.

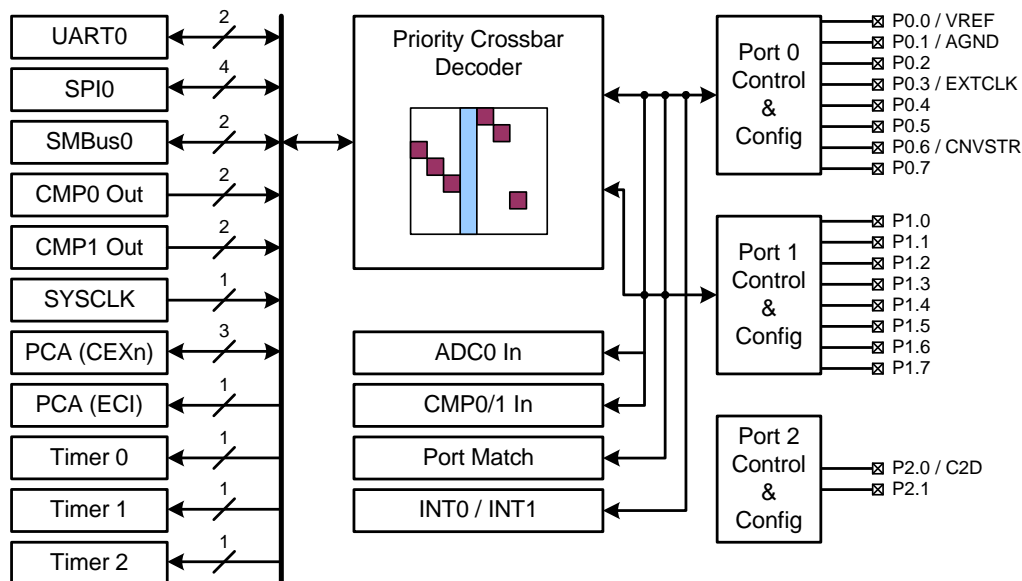


Figure 21.1. Port I/O Functional Block Diagram

21.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins can be assigned to various analog, digital, and external interrupt functions. The port pins assigned to analog functions should be configured for analog I/O, and port pins assigned to digital or external interrupt functions should be configured for digital I/O.

21.2.1. Assigning Port I/O Pins to Analog Functions

Table 21.1 shows all available analog functions that require port I/O assignments. Table 21.1 shows the potential mapping of port I/O to each analog function.

Table 21.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0 - P1.7	ADC0MX, PnSKIP, PnMDIN
Comparator0 Input	P0.0 - P1.7	CPT0MX, PnSKIP, PnMDIN
Comparator1 Input	P0.0 - P1.7	CPT1MX, PnSKIP, PnMDIN
Voltage Reference (VREF)	P0.0	REF0CN, PnSKIP, PnMDIN
Reference Ground (AGND)	P0.1	REF0CN, PnSKIP, PnMDIN

21.2.2. Assigning Port I/O Pins to Digital Functions

Any port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the crossbar for pin assignment; however, some digital functions bypass the crossbar in a manner similar to the analog functions listed above. Table 21.2 shows all digital functions available through the crossbar and the potential mapping of port I/O to each function.

Table 21.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) Used for Assignment
UART0, SPI0, SMBus0, CP0, CP0A, CP1, CP1A, SYSCLK, PCA0 (CEX0-2 and ECI), T0, T1 or T2.	Any port pin available for assignment by the crossbar. This includes P0.0 - P1.7 pins which have their PnSKIP bit set to '0'. Note: The crossbar will always assign UART0 pins to P0.4 and P0.5.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0 - P2.1	P0SKIP, P1SKIP, P2SKIP

21.7. Port I/O and Pin Configuration Control Registers

Register 21.1. XBR0: Port I/O Crossbar 0

Bit	7	6	5	4	3	2	1	0
Name	SYSCKE	CP1AE	CP1E	CP0AE	CP0E	SMB0E	SPI0E	URT0E
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xE1								

Table 21.4. XBR0 Register Bit Descriptions

Bit	Name	Function
7	SYSCKE	SYSCCLK Output Enable. 0: SYSCCLK unavailable at Port pin. 1: SYSCCLK output routed to Port pin.
6	CP1AE	Comparator1 Asynchronous Output Enable. 0: Asynchronous CP1 unavailable at Port pin. 1: Asynchronous CP1 routed to Port pin.
5	CP1E	Comparator1 Output Enable. 0: CP1 unavailable at Port pin. 1: CP1 routed to Port pin.
4	CP0AE	Comparator0 Asynchronous Output Enable. 0: Asynchronous CP0 unavailable at Port pin. 1: Asynchronous CP0 routed to Port pin.
3	CP0E	Comparator0 Output Enable. 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.
2	SMB0E	SMBus0 I/O Enable. 0: SMBus0 I/O unavailable at Port pins. 1: SMBus0 I/O routed to Port pins.
1	SPI0E	SPI I/O Enable. 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins. The SPI can be assigned either 3 or 4 GPIO pins.
0	URT0E	UART I/O Output Enable. 0: UART I/O unavailable at Port pin. 1: UART TX, RX routed to Port pins P0.4 and P0.5.

-
1. Enable the VDD supply monitor (VMONEN = 1).
 2. Wait for the VDD supply monitor to stabilize (optional).
 3. Enable the VDD monitor as a reset source in the RSTSRC register.

22.4. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. The PINRSF flag is set on exit from an external reset.

22.5. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than the MCD time window, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag will read 1, signifying the MCD as the reset source; otherwise, this bit reads 0. Writing a 1 to the MCDRSF bit enables the Missing Clock Detector; writing a 0 disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.6. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a 1 to the C0RSEF flag. Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag will read 1 signifying Comparator0 as the reset source; otherwise, this bit reads '0'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.7. Watchdog Timer Reset

The programmable Watchdog Timer (WDT) can be used to prevent software from running out of control during a system malfunction. The WDT function can be enabled or disabled by software as described in the watchdog timer section. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit is set to '1'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.8. Flash Error Reset

If a flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A flash write or erase is attempted above user code space.
- A flash read is attempted above user code space.
- A program read is attempted above user code space (i.e. a branch instruction to the reserved area).
- A flash read, write or erase attempt is restricted due to a flash security setting.

The FERROR bit is set following a flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.9. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit. The SWRSF bit will read 1 following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

23.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

23.1.1. Master Out, Slave In (MOSI)

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. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

23.1.2. Master In, Slave Out (MISO)

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. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

23.1.3. Serial Clock (SCK)

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. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected ($NSS = 1$) in 4-wire slave mode.

23.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 23.2, Figure 23.3, and Figure 23.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device.

Table 24.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—

Register 24.4. SMB0ADR: SMBus0 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV							GC
Type	RW							RW
Reset	0	0	0	0	0	0	0	0

SFR Address: 0xD7

Table 24.10. SMB0ADR Register Bit Descriptions

Bit	Name	Function
7:1	SLV	SMBus Hardware Slave Address. Defines the SMBus0 Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC	General Call Address Enable. When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

25. Timers (Timer0, Timer1, Timer2 and Timer3)

Each MCU in the C8051F85x/86x family includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timers for timing peripherals or for general purpose use. These timers 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 and Timer 3 are also identical and offer both 16-bit and split 8-bit timer functionality with auto-reload capabilities. Timer 2 and Timer 3 both offer a capture function, but are different in their system-level connections. Timer 2 is capable of performing a capture function on an external signal input routed through the crossbar, while the Timer 3 capture is dedicated to the low-frequency oscillator output. Table 25.1 summarizes the modes available to each timer.

Table 25.1. Timer Modes

Timer 0 and Timer 1 Modes	Timer 2 Modes	Timer 3 Modes
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
8-bit counter/timer with auto-reload	Input pin capture	Low-frequency oscillator capture
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked.

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it must be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

All four timers are capable of clocking other peripherals and triggering events in the system. The individual peripherals select which timer to use for their respective functions. Table 25.2 summarizes the peripheral connections for each timer. Note that the Timer 2 and Timer 3 high overflows apply to the full timer when operating in 16-bit mode or the high-byte timer when operating in 8-bit split mode.

Table 25.2. Timer Peripheral Clocking / Event Triggering

Function	T0 Overflow	T1 Overflow	T2 High Overflow	T2 Low Overflow	T3 High Overflow	T3 Low Overflow
UART0 Baud Rate		X				
SMBus0 Clock Rate	X	X	X	X		
SMBus0 SCL Low Timeout					X	
PCA0 Clock	X					

25.1.2. Mode 1: 16-bit Counter/Timer

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

25.2.2. 8-bit Timers with Auto-Reload

When TnSPLIT is set, the timer operates as two 8-bit timers (TMRnH and TMRnL). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.5. TMRnRLL holds the reload value for TMRnL; TMRnRLH holds the reload value for TMRnH. The TRn bit in TMRnCN handles the run control for TMRnH. TMRnL is always running when configured for 8-bit auto-reload mode.

Each 8-bit timer may be configured to clock from SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Clock Select bits (TnMH and TnML in CKCON) select either SYSCLK or the clock defined by the External Clock Select bit (TnXCLK in TMRnCN), as follows:

TnMH	TnXCLK	TMRnH Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

TnML	TnXCLK	TMRnL Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TFnH bit is set when TMRnH overflows from 0xFF to 0x00; the TFnL bit is set when TMRnL overflows from 0xFF to 0x00. When timer interrupts are enabled, an interrupt is generated each time TMRnH overflows. If timer interrupts are enabled and TFnLEN is set, an interrupt is generated each time either TMRnL or TMRnH overflows. When TFnLEN is enabled, software must check the TFnH and TFnL flags to determine the source of the timer interrupt. The TFnH and TFnL interrupt flags are not cleared by hardware and must be manually cleared by software.

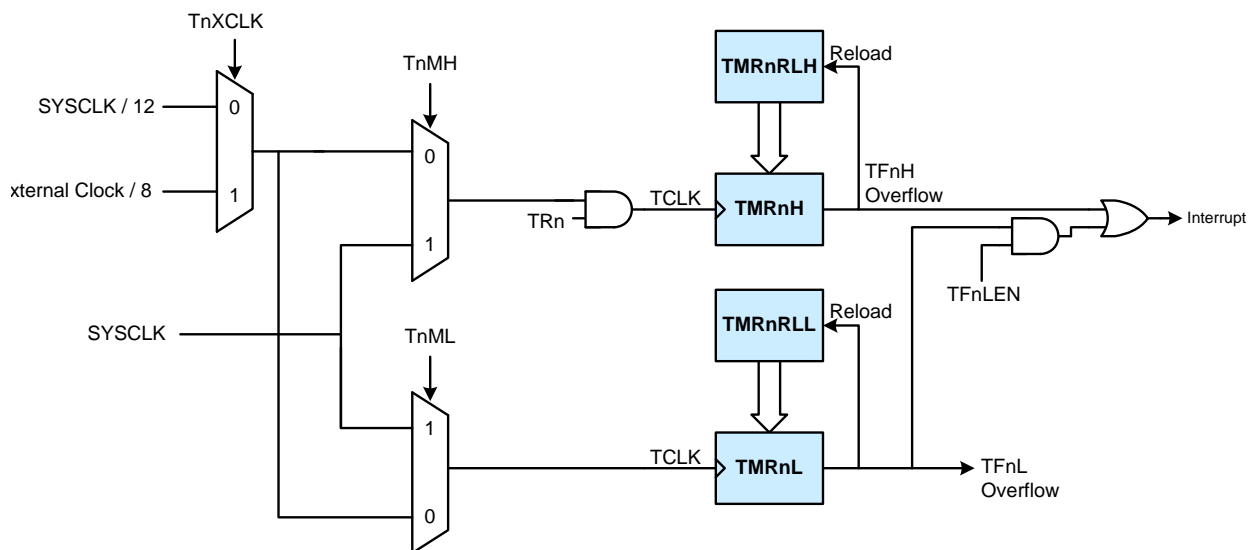


Figure 25.5. 8-Bit Mode Block Diagram

26.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE bit of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB8 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

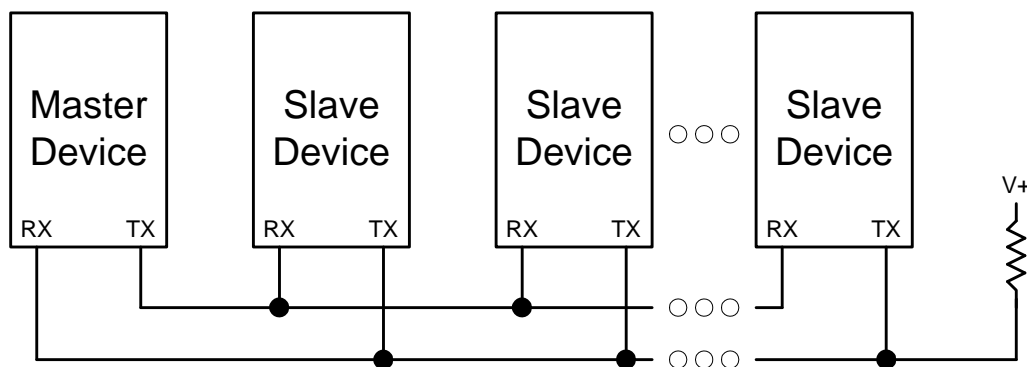


Figure 26.5. UART Multi-Processor Mode Interconnect Diagram

26.4. UART Control Registers

Register 26.1. SCON0: UART0 Serial Port Control

Bit	7	6	5	4	3	2	1	0
Name	SMODE	Reserved	MCE	REN	TB8	RB8	TI	RI
Type	RW	R	RW	RW	RW	RW	RW	RW
Reset	0	1	0	0	0	0	0	0
SFR Address: 0x98 (bit-addressable)								

Table 26.2. SCON0 Register Bit Descriptions

Bit	Name	Function
7	SMODE	Serial Port 0 Operation Mode. Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate (Mode 0). 1: 9-bit UART with Variable Baud Rate (Mode 1).
6	Reserved	Must write reset value.
5	MCE	Multiprocessor Communication Enable. This bit enables checking of the stop bit or the 9th bit in multi-drop communication buses. The function of this bit is dependent on the UART0 operation mode selected by the SMODE bit. In Mode 0 (8-bits), the peripheral will check that the stop bit is logic 1. In Mode 1 (9-bits) the peripheral will check for a logic 1 on the 9th bit. 0: Ignore level of 9th bit / Stop bit. 1: RI is set and an interrupt is generated only when the stop bit is logic 1 (Mode 0) or when the 9th bit is logic 1 (Mode 1).
4	REN	Receive Enable. 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB8	Ninth Transmission Bit. The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB8	Ninth Receive Bit. RB8 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI	Transmit Interrupt Flag. Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.