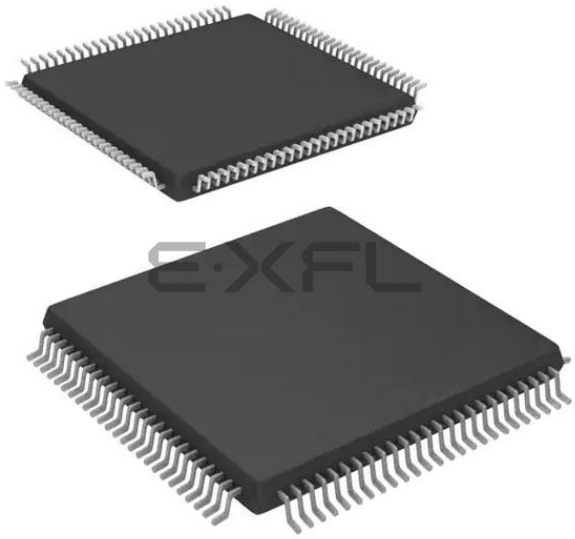


Welcome to E-XFL.COM

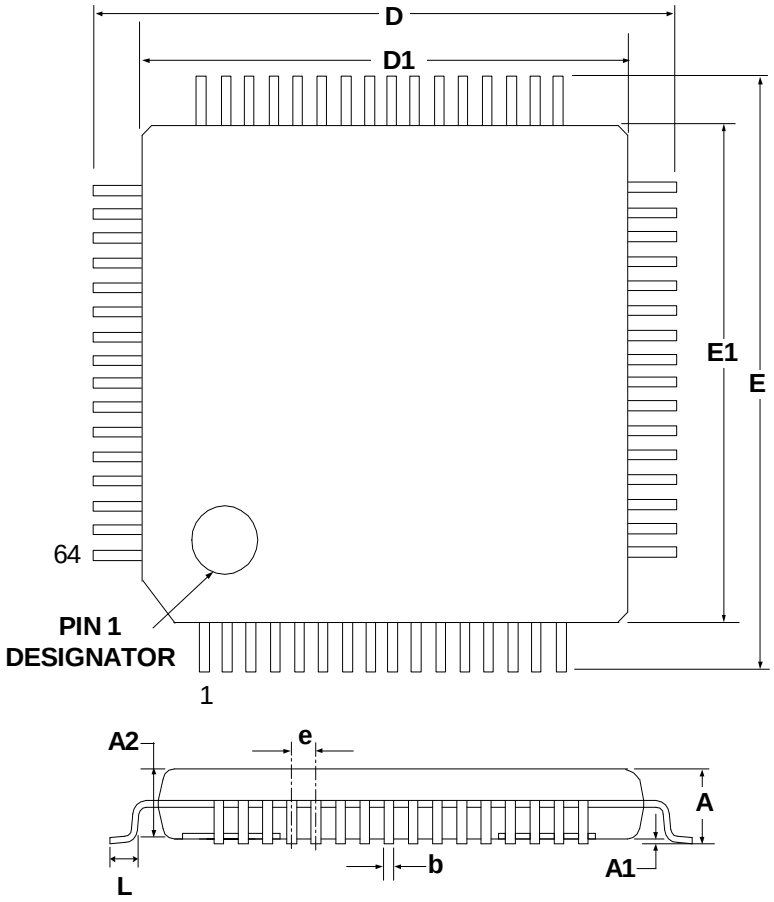
What is "Embedded - Microcontrollers"?

"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	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x16b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f062-gq

List of Figures



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	-	1.05
b	0.17	0.22	0.27
D	-	12.00	-
D1	-	10.00	-
e	-	0.50	-
E	-	12.00	-
E1	-	10.00	-
L	0.45	0.60	0.75

Figure 4.6. TQFP-64 Package Drawing

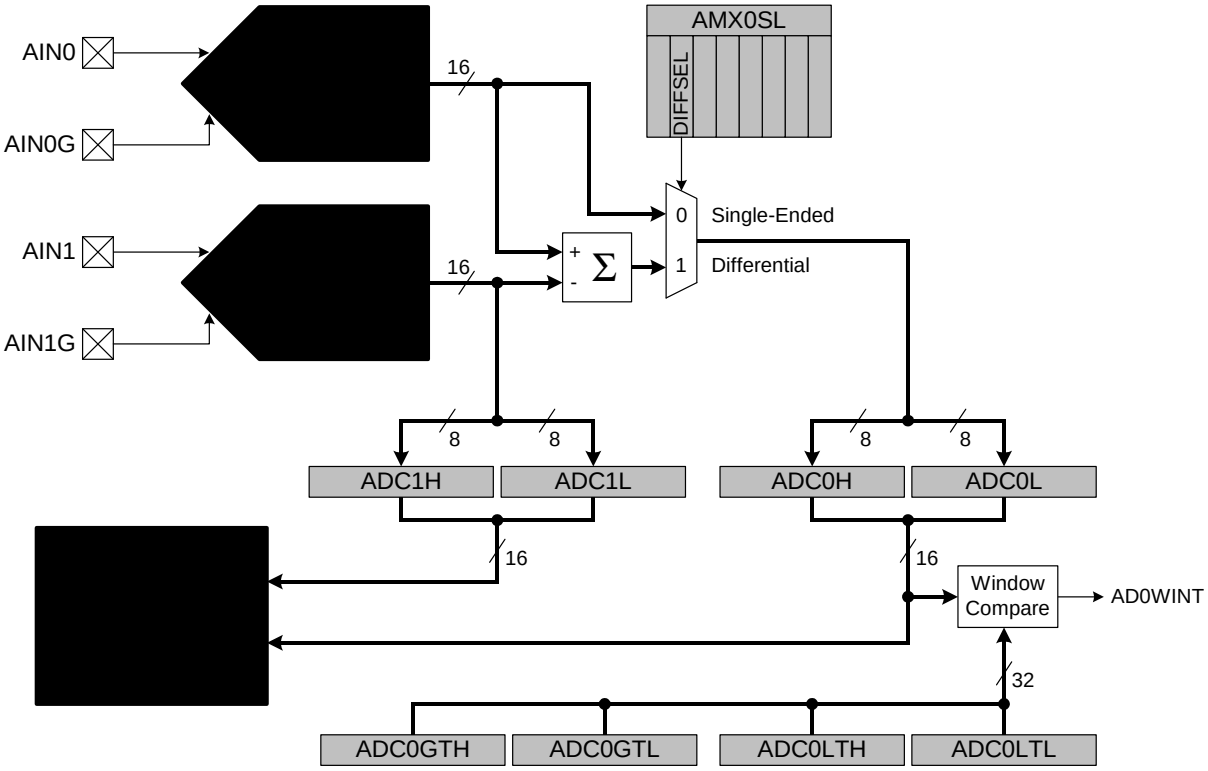


Figure 5.2. 16-bit ADC0 and ADC1 Data Path Diagram

5.1. Single-Ended or Differential Operation

ADC0 and ADC1 can be programmed to operate independently as single-ended ADCs, or together to accept a differential input. In single-ended mode, the ADCs can be configured to sample simultaneously, or to use different conversion speeds. In differential mode, ADC1 is a slave to ADC0, and its configuration is based on ADC0 settings, except during offset or gain calibrations. The DIFFSEL bit in the Channel Select Register AMX0SL (Figure 5.6) selects between single-ended and differential mode.

5.1.1. Pseudo-Differential Inputs

The inputs to the ADCs are pseudo-differential. The actual voltage measured by each ADC is equal to the voltage between the AINn pin and the AINnG pin. AINnG must be a DC signal between -0.2 and 0.6 V. In most systems, AINnG will be connected to AGND. If not tied to AGND, the AINnG signal can be used to negate a limited amount of fixed offset, but it is recommended that the internal offset calibration features of the device be used for this purpose. When operating in differential mode, AIN0G and AIN1G should be tied together. AINn must remain above AINnG in both modes for accurate conversion results.

5.2. Voltage Reference

The voltage reference circuitries for ADC0 and ADC1 allow for many different voltage reference configurations. Each ADC has the capability to use its own dedicated, on-chip voltage reference, or an off-chip reference circuit. A block diagram of the reference circuitry for one ADC is shown in Figure 5.3.

The internal voltage reference circuit for each ADC consists of an independent, temperature stable 1.2 V bandgap voltage reference generator, with an output buffer amplifier which multiplies the bandgap reference by 2. The maximum load seen by the VREFn (VREF0 or VREF1) pin must be less than 100 µA to AGND. Bypass capacitors of 0.1 µF and 47 µF are recommended from the VREFn pin to VRGNDn.

The voltage reference circuitry for each ADC is controlled in the Reference Control Registers. REF0CN (defined in Figure 5.11) is the Reference Control Register for ADC0, and REF1CN (defined in Figure 5.12) is the Reference Control Register for ADC1. The REFnCN registers are used to enable/disable the internal reference and bias generator circuitry for each ADC independently. The BIASEn bits enable the on-board bias generators for each ADC, while the REFBEn bits enable the 2x buffer amplifiers which drive the VREFn pins. When disabled, the supply current drawn by the bandgap and buffer amplifier falls to less than 1 µA (typical) and the output of the buffer amplifier enters a high impedance state (approximately 25 k Ohms). If the internal voltage reference for an ADC is used, the BIASEn and REFBEn bits for that ADC must both be set to logic 1. If an external reference is used, the REFBEn bit should be set to logic 0. Note that the BIASEn bit for an ADC must be set to logic 1 to enable that ADC, regardless of the voltage reference that is used. If an ADC is not being used, the BIASEn bit can be set to logic 0 to conserve power. The electrical specifications for the Voltage References are given in Table 5.3.

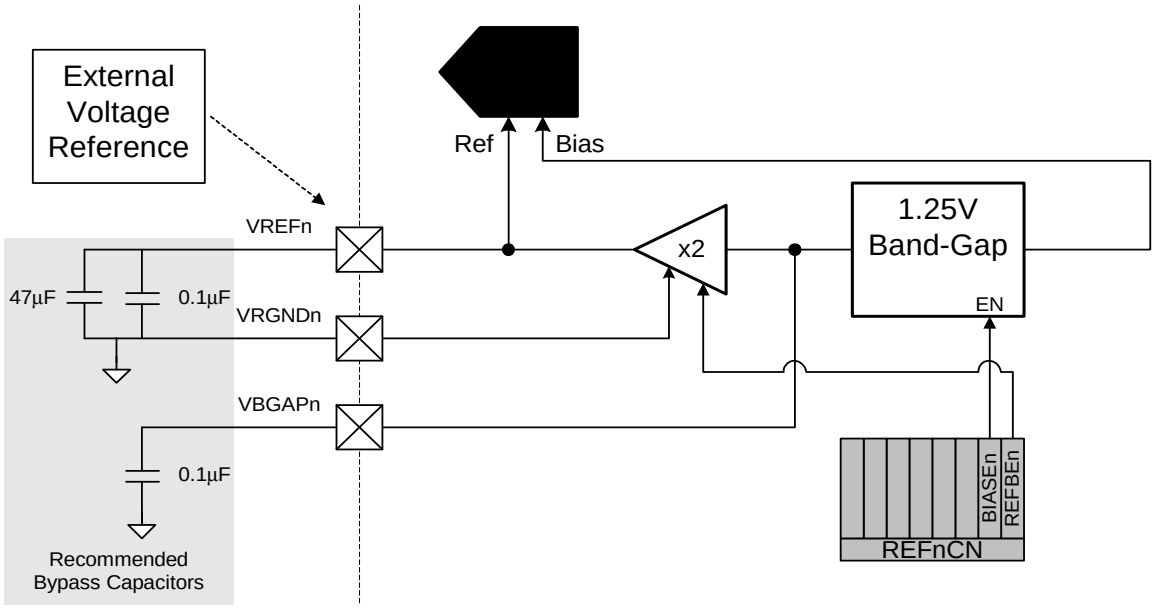
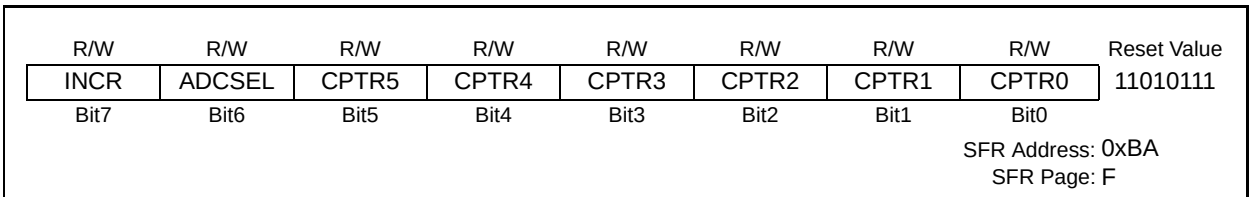


Figure 5.3. Voltage Reference Block Diagram

Figure 5.22. ADC0CPT: ADC Calibration Pointer Register



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
INCR	ADCSEL	CPTR5	CPTR4	CPTR3	CPTR2	CPTR1	CPTR0	11010111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

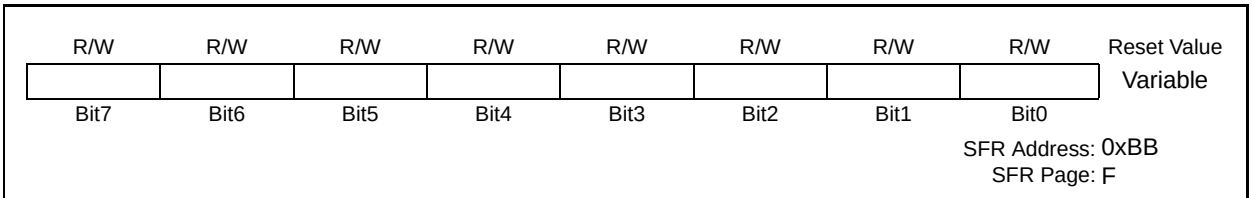
SFR Address: 0xBA
SFR Page: F

Bit 7: INCR: Pointer Address Automatic Increment.
0: Disable Auto-Increment.
1: Enable Auto-Increment. CPTR5-0 will automatically be incremented after each read or write to ADC0CCF.

Bit 6: ADCSEL: ADC Calibration Coefficient Select.
0: Reads and Writes of ADC0CCF will access ADC0 Calibration Coefficients.
1: Reads and Writes of ADC0CCF will access ADC1 Calibration Coefficients.

Bits 5-0: CPTR5-0: Calibration Coefficent Pointer.
Select which Calibration Coefficient location will be accessed when ADC0CCF is read or written.

Figure 5.23. ADC0CCF: ADC Calibration Coefficient Register



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBB
SFR Page: F

Bits 7-0: Calibration Coefficients at the location specified in ADC0CPT. See Table 5.19.

Figure 6.8. DMA0BND: DMA0 Instruction Boundary Register

SFR Page: 3
SFR Address: 0xFD

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	

Bits 7-6: Unused.
Bits 5-0: DMA0 instruction address to begin with when executing DMA instructions.

Figure 6.9. DMA0ISW: DMA0 Instruction Status Register

SFR Page: 3
SFR Address: 0xFE

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	

Bits 7-6: Unused.
Bits 5-0: Contains the address of the current DMA0 Instruction to be executed.

7. 10-Bit ADC (ADC2, C8051F060/1/2/3)

The ADC2 subsystem for the C8051F060/1/2/3 consists of an analog multiplexer (referred to as AMUX2), and a 200 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 7.1). The AMUX2, data conversion modes, and window detector can all be configured from within software via the Special Function Registers shown in Figure 7.1. ADC2 operates in both Single-ended and Differential modes, and may be configured to measure any of the pins on Port 1, or the Temperature Sensor output. The ADC2 subsystem is enabled only when the AD2EN bit in the ADC2 Control register (ADC2CN) is set to logic 1. The ADC2 subsystem is in low power shutdown when this bit is logic 0.

Figure 7.1. ADC2 Functional Block Diagram

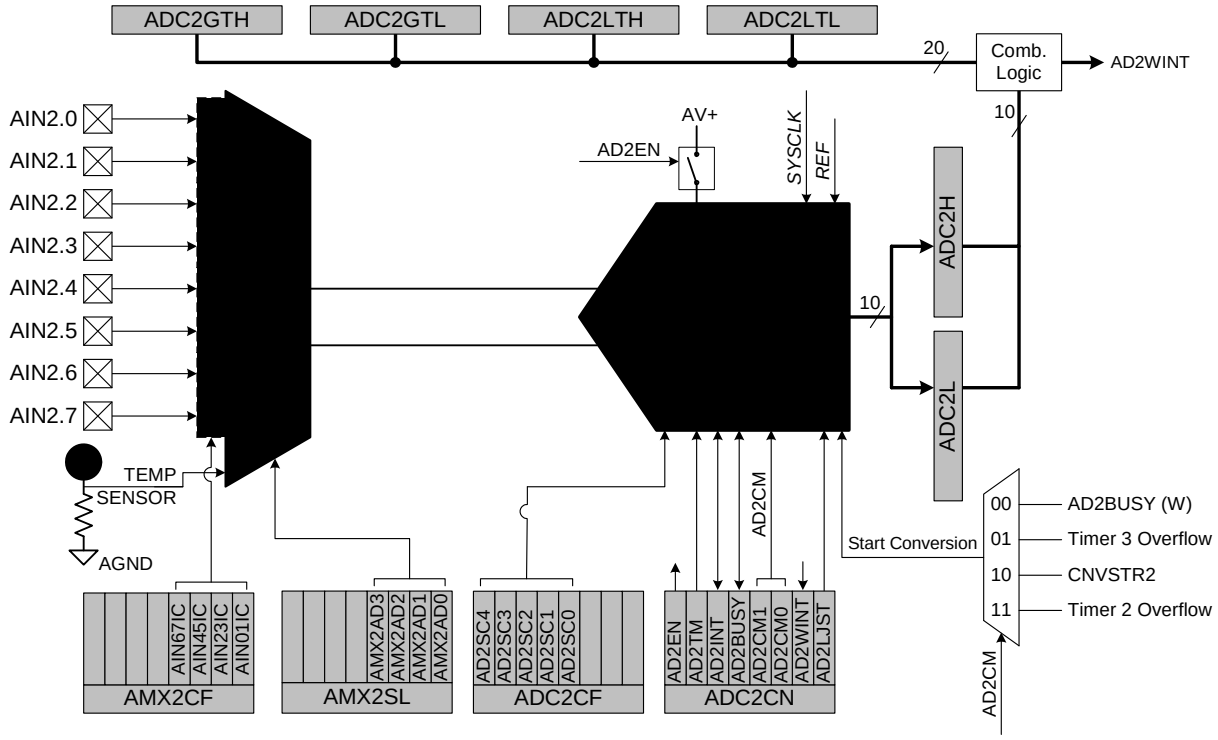


Table 13.2. Special Function Register (SFR) Memory Map

ADDRESS	SFR PAGE	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 1 2 3 F	SPI0CN CAN0CN DMA0CF P7	PCA0L DMA0CTL	PCA0H DMA0CTH	PCA0CPL0 DMA0CSL	PCA0CPH0 DMA0CSH	PCA0CPL1 DMA0BND	PCA0CPH1 DMA0ISW	WDTCN **(ALL PAGES)**
F0	0 1 2 3 F	B **(ALL PAGES)**						EIP1 **(ALL PAGES)**	EIP2 **(ALL PAGES)**
E8	0 1 2 3 F	ADC0CN ADC1CN ADC2CN P6	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	PCA0CPL4	PCA0CPH4	RSTSRC
E0	0 1 2 3 F	ACC **(ALL PAGES)**	PCA0CPL5 XBR0	PCA0CPH5 XBR1	XBR2	XBR3		EIE1 **(ALL PAGES)**	EIE2 **(ALL PAGES)**
D8	0 1 2 3 F	PCA0CN CAN0DATL DMA0CN P5	PCA0MD CAN0DATH DMA0DAL	PCA0CPM0 CAN0ADR DMA0DAH	PCA0CPM1 CAN0TST DMA0DSL	PCA0CPM2 DMA0DSH	PCA0CPM3 DMA0IPT	PCA0CPM4 DMA0IDT	PCA0CPM5
D0	0 1 2 3 F	PSW **(ALL PAGES)**	REF0CN REF1CN REF2CN	DAC0L DAC1L	DAC0H DAC1H	DAC0CN DAC1CN			
C8	0 1 2 3 F	TMR2CN TMR3CN TMR4CN P4	TMR2CF TMR3CF TMR4CF	RCAP2L RCAP3L RCAP4L	RCAP2H RCAP3H RCAP4H	TMR2L TMR3L TMR4L	TMR2H TMR3H TMR4H		SMB0CR
C0	0 1 2 3 F	SMB0CN CAN0STA	SMB0STA	SMB0DAT	SMB0ADR	ADC0GTL ADC2GTL	ADC0GTH ADC2GTH	ADC0LTL ADC2LTL	ADC0LTH ADC2LTH
B8	0 1 2 3 F	IP **(ALL PAGES)**	SADEN0	AMX2CF ADC0CPT	AMX0SL AMX2SL ADC0CCF	ADC0CF ADC1CF ADC2CF		ADC0L ADC1L ADC2L	ADC0H ADC1H ADC2H
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

13.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 22 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE-EIE2). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

13.3.1. MCU Interrupt Sources and Vectors

The MCUs support 22 interrupt sources. Software can simulate an interrupt event by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 13.4. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

13.3.2. External Interrupts

The external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of bits IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag follows the state of the external interrupt's input pin. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

17.6. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, /RD and /WR strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in Figure 17.6, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for /RD or /WR pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for /ALE + 1 for /RD or /WR + 4). The programmable setup and hold times default to the maximum delay settings after a reset.

Figure 17.6. EMI0TC: External Memory Timing Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	ERW3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA1
SFR Page: 0

Bits7-6: EAS1-0: EMIF Address Setup Time Bits.
00: Address setup time = 0 SYSCLK cycles.
01: Address setup time = 1 SYSCLK cycle.
10: Address setup time = 2 SYSCLK cycles.
11: Address setup time = 3 SYSCLK cycles.

Bits5-2: EWR3-0: EMIF /WR and /RD Pulse-Width Control Bits.
0000: /WR and /RD pulse width = 1 SYSCLK cycle.
0001: /WR and /RD pulse width = 2 SYSCLK cycles.
0010: /WR and /RD pulse width = 3 SYSCLK cycles.
0011: /WR and /RD pulse width = 4 SYSCLK cycles.
0100: /WR and /RD pulse width = 5 SYSCLK cycles.
0101: /WR and /RD pulse width = 6 SYSCLK cycles.
0110: /WR and /RD pulse width = 7 SYSCLK cycles.
0111: /WR and /RD pulse width = 8 SYSCLK cycles.
1000: /WR and /RD pulse width = 9 SYSCLK cycles.
1001: /WR and /RD pulse width = 10 SYSCLK cycles.
1010: /WR and /RD pulse width = 11 SYSCLK cycles.
1011: /WR and /RD pulse width = 12 SYSCLK cycles.
1100: /WR and /RD pulse width = 13 SYSCLK cycles.
1101: /WR and /RD pulse width = 14 SYSCLK cycles.
1110: /WR and /RD pulse width = 15 SYSCLK cycles.
1111: /WR and /RD pulse width = 16 SYSCLK cycles.

Bits1-0: EAH1-0: EMIF Address Hold Time Bits.
00: Address hold time = 0 SYSCLK cycles.
01: Address hold time = 1 SYSCLK cycle.
10: Address hold time = 2 SYSCLK cycles.
11: Address hold time = 3 SYSCLK cycles.

18. Port Input/Output

The C8051F06x family of devices are fully integrated mixed-signal System on a Chip MCUs with 59 digital I/O pins (C8051F060/2/4/6) or 24 digital I/O pins (C8051F061/3/5/7), organized as 8-bit Ports. All ports are both bit- and byte-addressable through their corresponding Port Data registers. All Port pins support configurable Open-Drain or Push-Pull output modes and weak pull-ups. Additionally, Port 0 pins are 5 V-tolerant. A block diagram of the Port I/O cell is shown in Figure 18.1. Complete Electrical Specifications for the Port I/O pins are given in Table 18.1.

Figure 18.1. Port I/O Cell Block Diagram

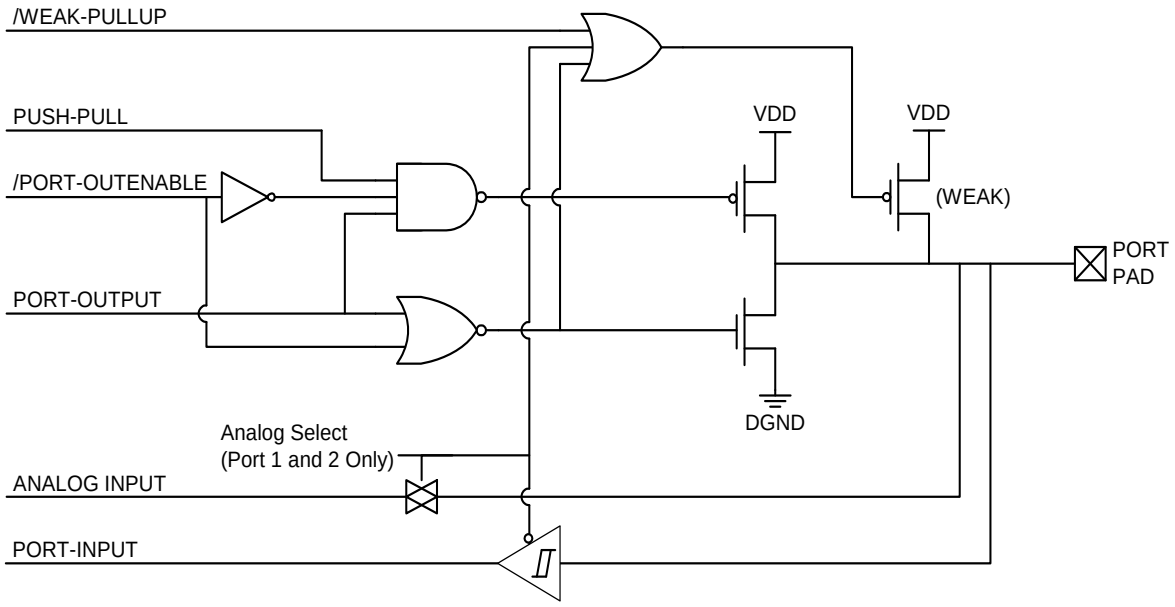


Table 18.1. Port I/O DC Electrical Characteristics

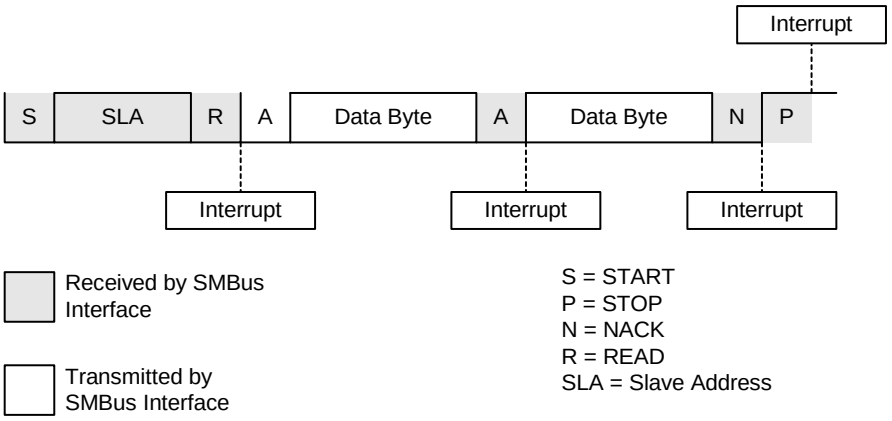
VDD = 2.7 to 3.6 V, -40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Output High Voltage (V_{OH})	I_{OH} = -3 mA, Port I/O Push-Pull I_{OH} = -10 µA, Port I/O Push-Pull	VDD - 0.7 VDD - 0.1			V
Output Low Voltage (V_{OL})	I_{OL} = 8.5 mA I_{OL} = 10 µA			0.6 0.1	V
Input High Voltage (VIH)		0.7 x VDD			
Input Low Voltage (VIL)				0.3 x VDD	
Input Leakage Current	DGND < Port Pin < VDD, Pin Tri-state Weak Pull-up Off Weak Pull-up On		 10	± 1	 µA µA
Input Capacitance			5		pF

20.3.3. Slave Transmitter Mode

Serial data is transmitted on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the SMBus0 interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 1 to indicate a "READ" operation. The SMBus0 interface receives the clock on SCL and transmits one or more bytes of serial data, waiting for an ACK from the master after each byte. SMBus0 exits slave mode after receiving a STOP condition from the master.

Figure 20.6. Typical Slave Transmitter Sequence



20.3.4. Slave Receiver Mode

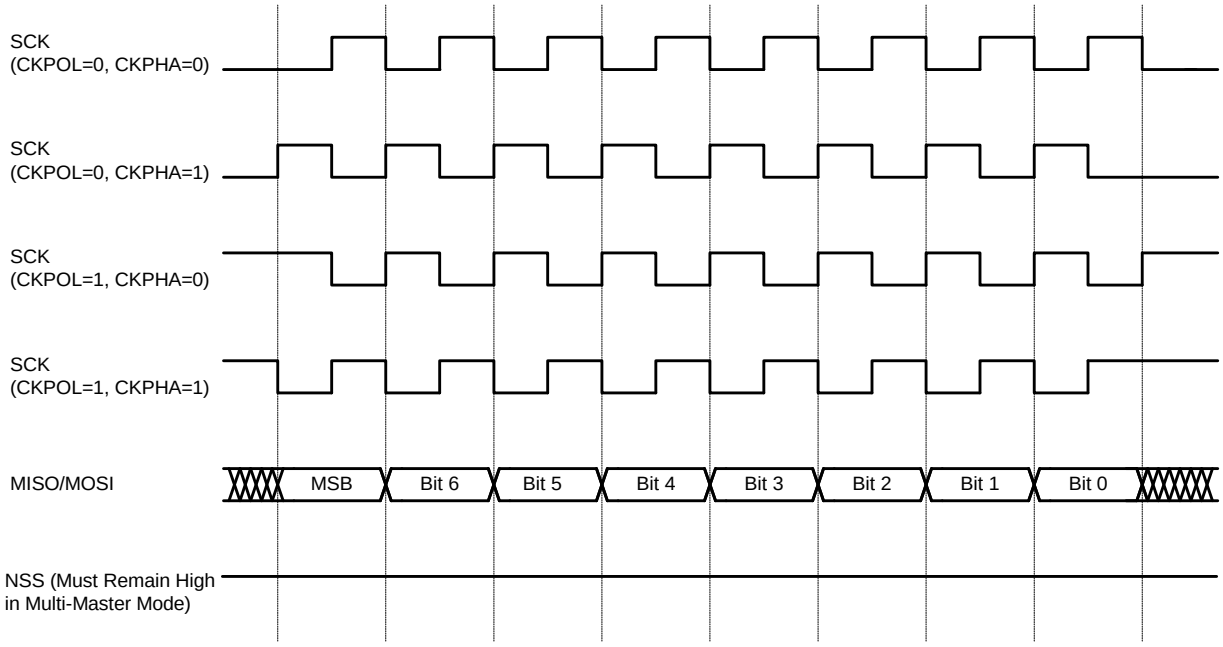
Serial data is received on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. The SMBus0 interface receives one or more bytes of serial data; after each byte is received, the interface transmits an ACK or NACK depending on the state of the AA bit in SMB0CN. SMBus0 exits Slave Receiver

21.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 21.5. For slave mode, the clock and data relationships are shown in Figure 21.6 and Figure 21.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in Figure 21.10 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. 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 (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 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 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

Figure 21.5. Master Mode Data/Clock Timing



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

Mode 2 provides asynchronous, full-duplex communication using a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Mode 2 supports multiprocessor communications and hardware address recognition (see Section 22.2). On transmit, the ninth data bit is determined by the value in TB80 (SCON0.3). It can be assigned the value of the parity flag P in the PSW or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if RI0 is logic 0 and one of the following requirements are met:

1. SM20 is logic 0
2. SM20 is logic 1, the received 9th bit is logic 1, and the received address matches the UART0 address as described in Section 22.2.

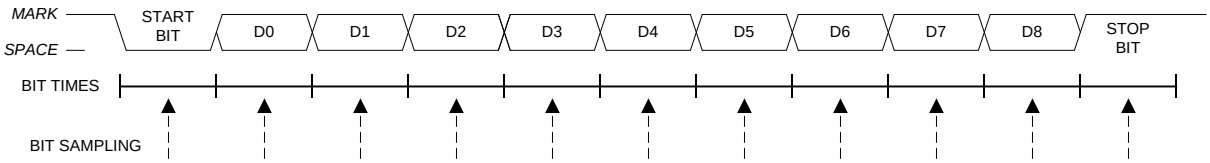
If the above conditions are satisfied, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 are set.

The baud rate in Mode 2 is either SYSCLK / 32 or SYSCLK / 64, according to the value of the SMOD0 bit in register SSTA0.

Equation 22.5. Mode 2 Baud Rate

$$BaudRate = 2^{SMOD0} \times \left(\frac{SYSCLK}{64}\right)$$

Figure 22.5. UART0 Modes 2 and 3 Timing Diagram



22.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit and the built-in UART0 address recognition hardware. 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. UART0 will recognize as “valid” (i.e., capable of causing an interrupt) **two** types of addresses: (1) a *masked* address and (2) a *broadcast* address **at any given time**. Both are described below.

22.2.1. Configuration of a Masked Address

The UART0 address is configured via two SFRs: SADDR0 (Serial Address) and SADEN0 (Serial Address Enable). SADEN0 sets the bit mask for the address held in SADDR0: bits set to logic 1 in SADEN0 correspond to bits in SADDR0 that are checked against the received address byte; bits set to logic 0 in SADEN0 correspond to “don’t care” bits in SADDR0.

Example 1, SLAVE #1	
SADDR0	= 00110101
SADEN0	= 00001111
UART0 Address	= xxxx0101

Example 2, SLAVE #2	
SADDR0	= 00110101
SADEN0	= 11110011
UART0 Address	= 0011xx01

Example 3, SLAVE #3	
SADDR0	= 00110101
SADEN0	= 11000000
UART0 Address	= 00xxxxxx

Setting the SM20 bit (SCON0.5) configures UART0 such that when a stop bit is received, UART0 will generate an interrupt only if the ninth bit is logic 1 (RB80 = ‘1’) and the received data byte matches the UART0 slave address. Following the received address interrupt, the slave will clear its SM20 bit to enable interrupts on the reception of the following data byte(s). Once the entire message is received, the addressed slave resets its SM20 bit to ignore all transmissions until it receives the next address byte. While SM20 is logic 1, UART0 ignores all bytes that do not match the UART0 address and include a ninth bit that is logic 1.

22.2.2. Broadcast Addressing

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 broadcast address is the logical OR of registers SADDR0 and SADEN0, and ‘0’s of the result are treated as “don’t cares”. Typically a broadcast address of 0xFF (hexadecimal) is acknowledged by all slaves, assuming “don’t care” bits as ‘1’s. 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)..

Example 4, SLAVE #1	
SADDR0	= 00110101
SADEN0	= 00001111
Broadcast Address	= 00111111

Example 5, SLAVE #2	
SADDR0	= 00110101
SADEN0	= 11110011
Broadcast Address	= 11110111

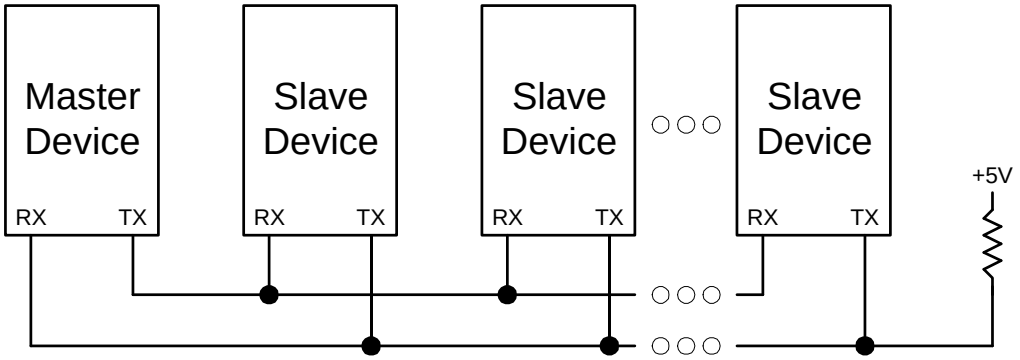
Example 6, SLAVE #3	
SADDR0	= 00110101
SADEN0	= 11000000
Broadcast Address	= 11110101

Where all ZEROES in the Broadcast address are don’t cares.

Note in the above examples 4, 5, and 6, each slave would recognize as “valid” an address of 0xFF as a broadcast address. Also note that examples 4, 5, and 6 uses the same SADDR0 and SADEN0 register values as shown in the examples 1, 2, and 3 respectively (slaves #1, 2, and 3). Thus, a master could address each slave device individually using a masked address, and also broadcast to all three slave devices. For example, if a Master were to send an address “11110101”, only slave #1 would recognize the

address as valid. If a master were to then send an address of “11111111”, all three slave devices would recognize the address as a valid broadcast address.

Figure 22.7. UART Multi-Processor Mode Interconnect Diagram



22.3. Frame and Transmission Error Detection

All Modes:
The Transmit Collision bit (TXCOL0 bit in register SCON0) reads ‘1’ if user software writes data to the SBUF0 register while a transmit is in progress. Note that the TXCOL0 bit is also used as the SM20 bit when written by user software. This bit does not generate an interrupt.

Modes 1, 2, and 3:

The Receive Overrun bit (RXOV0 in register SCON0) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. Note that the RXOV0 bit is also used as the SM10 bit when written by user software. The Frame Error bit (FE0 in register SSTA0) reads ‘1’ if an invalid (low) STOP bit is detected. Note that the FE0 bit is also used as the SM00 bit when written by user software. The RXOV0 and FE0 bits do not generate interrupts.

23.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB81 (SCON1.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB81 (SCON1.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: (1) RI1 must be logic 0, and (2) if MCE1 is logic 1, the 9th bit must be logic 1 (when MCE1 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF1, the ninth bit is stored in RB81, and the RI1 flag is set to '1'. If the above conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set to '1'. A UART1 interrupt will occur if enabled when either TI1 or RI1 is set to '1'.

Figure 23.5. 9-Bit UART Timing Diagram

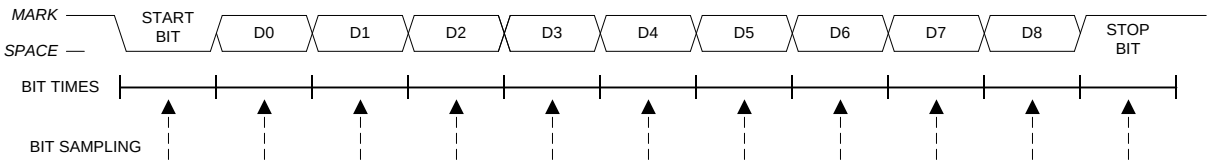


Figure 23.7. SCON1: Serial Port 1 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S1MODE	-	MCE1	REN1	TB81	RB81	TI1	RI1	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98
SFR Page: 1

Bit7: S1MODE: Serial Port 1 Operation Mode.
This bit selects the UART1 Operation Mode.
0: 8-bit UART with Variable Baud Rate.
1: 9-bit UART with Variable Baud Rate.

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: MCE1: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
S1MODE = 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI1 will only be activated if stop bit is logic level 1.
S1MODE = 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI1 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART receiver.
0: UART1 reception disabled.
1: UART1 reception enabled.

Bit3: TB81: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB81: Ninth Receive Bit.
RB81 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI1: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART1 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART1 interrupt is enabled, setting this bit causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI1: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART1 (set at the STOP bit sampling time). When the UART1 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

25.1. 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 into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2-CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 25.1.

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 CPU is in Idle mode.

Table 25.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8 (synchronized with system clock)

Figure 25.2. PCA Counter/Timer Block Diagram