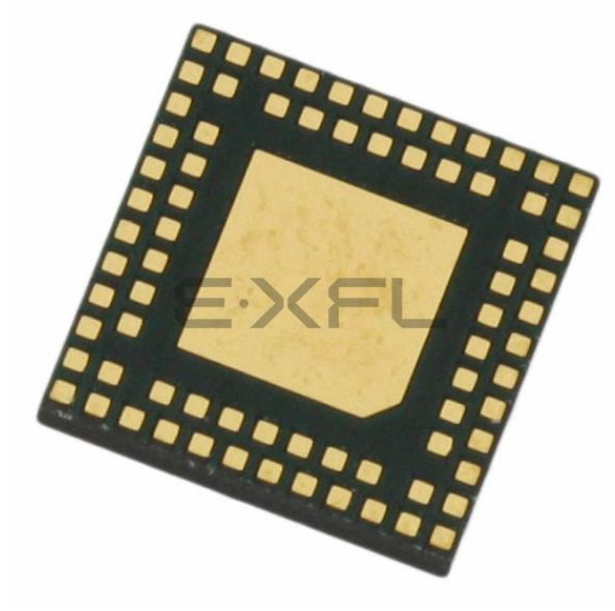

Welcome to E-XFL.COM

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	57
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.8V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	76-VFQFN Dual Rows, Exposed Pad
Supplier Device Package	76-DQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f964-b-gmr

List of Tables

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Condition	Min	Typ	Max	Unit
Digital Supply Current— Low Power Idle Mode, All peripheral clocks enabled (PCLKEN = 0x0F) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 6]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	1.5	1.9	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	1.07	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	270 280	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	232[5]	—	µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	47[5]	—	µA/MHz
Digital Supply Current— Low Power Idle Mode, All Peripheral Clocks Disabled (PCLKEN = 0x00) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 7]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	487	—	µA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	340	—	µA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	90 94	— —	µA µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	11[5]	—	µA/MHz
Digital Supply Current—Suspend Mode					
Digital Supply Current (Suspend Mode)	V_{BAT} = 1.8 V V_{BAT} = 3.8 V	— —	77 84	— —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

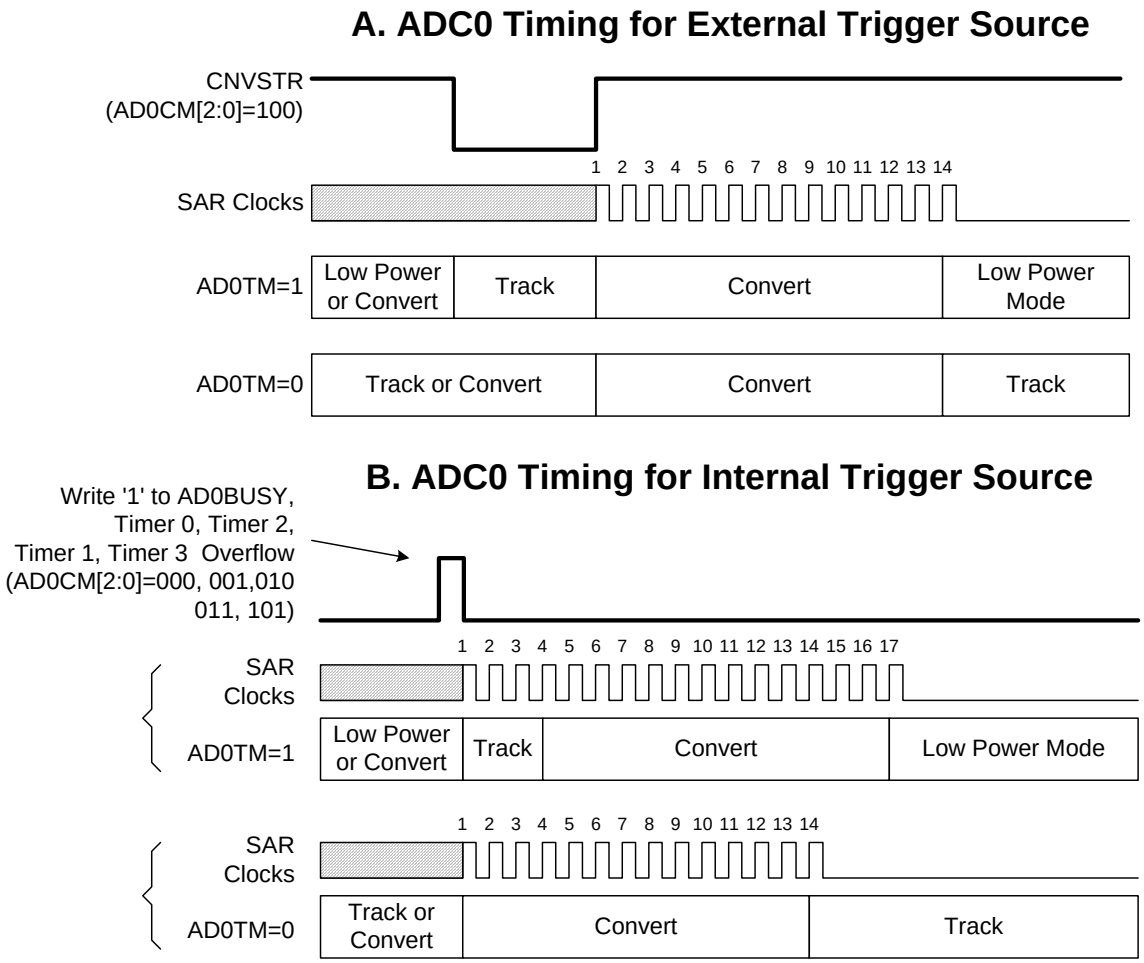

Figure 5.2. 10-Bit ADC Track and Conversion Example Timing (BURSTEN = 0)

5.5. Low Power Mode

The SAR converter provides a low power mode that allows a significant reduction in operating current when operating at low SAR clock frequencies. Low power mode is enabled by setting the AD0LPM bit (ADC0PWR.7) to 1. In general, low power mode is recommended when operating with SAR conversion clock frequency at 4 MHz or less. See the Electrical Characteristics chapter for details on power consumption and the maximum clock frequencies allowed in each mode. Setting the Low Power Mode bit reduces the bias currents in both the SAR converter and in the High-Speed Voltage Reference.

Table 5.1. Representative Conversion Times and Energy Consumption for the SAR ADC with 1.65 V High-Speed VREF

	Normal Power Mode			Low Power Mode		
	8 bit	**10 bit**	**12 bit**	**8 bit**	**10 bit**	**12 bit**
Highest nominal SAR clock frequency	8.17 MHz (24.5/3)	8.17 MHz (24.5/3)	6.67 MHz (20.0/3)	4.08 MHz (24.5/6)	4.08 MHz (24.5/6)	4.00 MHz (20.0/5)
Total number of conversion clocks required	11	13	52 (13 x 4)	11	13	52 (13*4)
Total tracking time (min)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)
Total time for one conversion	2.85 µs	3.09 µs	12.6 µs	4.19 µs	4.68 µs	17.8 µs
ADC Throughput	351 ksps	323 ksps	79 ksps	238 ksps	214 ksps	56 ksps
Energy per conversion	8.2 nJ	8.9 nJ	36.5 nJ	6.5 nJ	7.3 nJ	27.7 nJ

Note: This table assumes that the 24.5 MHz precision oscillator is used for 8- and 10-bit modes, and the 20 MHz low power oscillator is used for 12-bit mode. The values in the table assume that the oscillators run at their nominal frequencies. The maximum SAR clock values given in Table 4.12 allow for maximum oscillation frequencies of 25.0 MHz and 22 MHz for the precision and low-power oscillators, respectively, when using the given SAR clock divider values. Energy calculations are for the ADC subsystem only and do not include CPU current.

SFR Definition 5.4. ADC0PWR: ADC0 Burst Mode Power-Up Time

Bit	7	6	5	4	3	2	1	0
Name	AD0LPM				AD0PWR[3:0]			
Type	R/W	R	R	R	R/W			
Reset	0	0	0	0	1	1	1	1

SFR Page = 0xF; SFR Address = 0xBA

Bit	Name	Function
7	AD0LPM	**ADC0 Low Power Mode Enable.** Enables Low Power Mode Operation. 0: Low Power Mode disabled. 1: Low Power Mode enabled.
6:4	Unused	Read = 0000b; Write = Don't Care.
3:0	AD0PWR[3:0]	**ADC0 Burst Mode Power-Up Time.** Sets the time delay required for ADC0 to power up from a low power state. For BURSTEN = 0: ADC0 power state controlled by AD0EN. For BURSTEN = 1 and AD0EN = 1: ADC0 remains enabled and does not enter a low power state after all conversions are complete. Conversions can begin immediately following the start-of-conversion signal. For BURSTEN = 1 and AD0EN = 0: ADC0 enters a low power state after all conversions are complete. Conversions can begin a programmed delay after the start-of-conversion signal. The ADC0 Burst Mode Power-Up time is programmed according to the following equation: $$AD0PWR = \frac{Tstartup}{400ns} - 1$$ or $$Tstartup = (AD0PWR + 1)400ns$$ **Note:** Setting AD0PWR to 0x04 provides a typical tracking time of 2 us for the first sample taken after the start of conversion.

SFR Definition 7.5. CPT0MX: Comparator0 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9F

Bit	Name	Function
7:4	CMX0N	**Comparator0 Negative Input Selection.** Selects the negative input channel for Comparator0. 0000: P0.1 0001: P0.3 0010: P0.5 0011: Reserved 0100: Reserved 0101: Reserved 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 1010: Reserved 1011: Reserved 1100: Compare 1101: VBAT divided by 2 1110: Digital Supply Voltage 1111: Ground
3:0	CMX0P	**Comparator0 Positive Input Selection.** Selects the positive input channel for Comparator0. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: Reserved 0101: Reserved 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 1010: Reserved 1011: Reserved 1100: Compare 1101: VBAT divided by 2 1110: VBAT Supply Voltage 1111: VBAT Supply Voltage

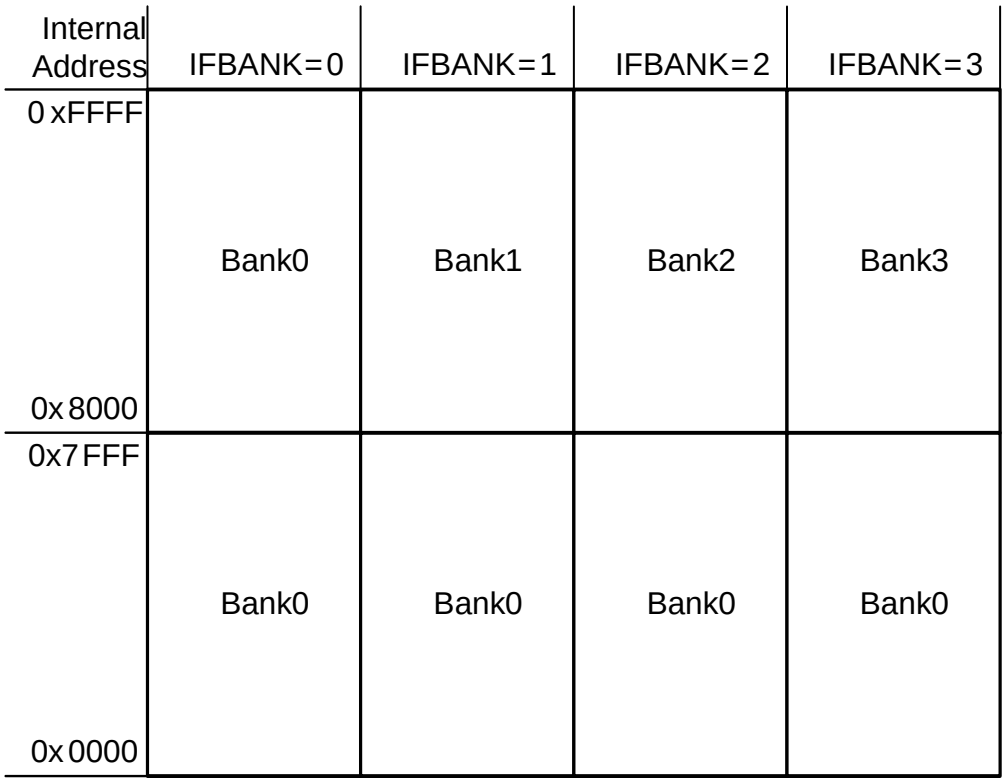

Figure 9.3. Address Memory Map for Instruction Fetches

SFR Definition 10.3. EMI0TC: External Memory Timing Control

Bit	7	6	5	4	3	2	1	0
Name	EAS[1:0]		EWR[3:0]				EAH[1:0]	
Type	R/W		R/W				R/W	
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xAF

Bit	Name	Function
7:6	EAS[1:0]	**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.
5:2	EWR[3:0]	**EMIF $\overline{WR}$ and $\overline{RD}$ Pulse-Width Control Bits.** 0000: $\overline{WR}$ and $\overline{RD}$ pulse width = 1 SYSCLK cycle. 0001: $\overline{WR}$ and $\overline{RD}$ pulse width = 2 SYSCLK cycles. 0010: $\overline{WR}$ and $\overline{RD}$ pulse width = 3 SYSCLK cycles. 0011: $\overline{WR}$ and $\overline{RD}$ pulse width = 4 SYSCLK cycles. 0100: $\overline{WR}$ and $\overline{RD}$ pulse width = 5 SYSCLK cycles. 0101: $\overline{WR}$ and $\overline{RD}$ pulse width = 6 SYSCLK cycles. 0110: $\overline{WR}$ and $\overline{RD}$ pulse width = 7 SYSCLK cycles. 0111: $\overline{WR}$ and $\overline{RD}$ pulse width = 8 SYSCLK cycles. 1000: $\overline{WR}$ and $\overline{RD}$ pulse width = 9 SYSCLK cycles. 1001: $\overline{WR}$ and $\overline{RD}$ pulse width = 10 SYSCLK cycles. 1010: $\overline{WR}$ and $\overline{RD}$ pulse width = 11 SYSCLK cycles. 1011: $\overline{WR}$ and $\overline{RD}$ pulse width = 12 SYSCLK cycles. 1100: $\overline{WR}$ and $\overline{RD}$ pulse width = 13 SYSCLK cycles. 1101: $\overline{WR}$ and $\overline{RD}$ pulse width = 14 SYSCLK cycles. 1110: $\overline{WR}$ and $\overline{RD}$ pulse width = 15 SYSCLK cycles. 1111: $\overline{WR}$ and $\overline{RD}$ pulse width = 16 SYSCLK cycles.
1:0	EAH[1:0]	**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.

13.6. Using CRC1 with SFR Access

The steps to perform a CRC using SFR access with the CRC1 module is as follow:

1. If desired, set the SEED bit in the CRC1CN SFR to seed with 0xFFFF.
2. Clear the CRC module by setting the CLR bit in the CRC1CN SFR.
3. Clear the SEED bit, if set previously in step 1.
4. Write the polynomial to CRC1POLH:L.
5. Write all data bytes to CRC1IN.
6. If desired, invert and/or flip the final results using the INV and FLIP bits.
7. Read the final CRC results from CRC1OUTH:L.
8. Clear the INV and/or FLIP bits, if set previously in step 6.

Note that all of the CRC1 SFRs are on SFR page 0x2.

13.7. Using the CRC1 module with the DMA

The steps to computing a CRC using the DMA are as follows.

1. If desired, set the SEED bit in CRC1CN to seed with 0xFFFF.
2. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
3. Clear the SEED bit, if set previously in step 1.
4. Write the polynomial to CRC1POLH:L.
5. Configure the DMA for the CRC operation:
   a. Disable the desired DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the desired DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the CRC1IN peripheral request by writing 0x2 to DMA0NCF.
   d. Enable the DMA interrupt on the selected channel by setting bit 7 of DMA0NCF.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address of the first byte of CRC data to DMA0NBAH:L.
   g. Write the size of the CRC data in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs DMA0A0H:L.
   i. Enable the interrupt on the desired channel by setting the corresponding bit in DMA0INT.
   j. Enable the desired channel by setting the corresponding bit in DMA0EN.
   k. Enable DMA interrupts by setting bit 5 of EIE2.
6. Set the DMA mode bit (bit 3) in the CRC1CN SFR to initiate the CRC operation.
7. Wait on the DMA interrupt.
8. If desired, invert and/or flip the final results using the INV and FLIP bits.
9. Read the final results from CRC1OUTH:L.
10. Clear the INV and/or FLIP bits, if set previously in step 8.

SFR Definition 20.4. DC0RDY: DC-DC Converter Ready Indicator

Bit	7	6	5	4	3	2	1	0
Name	RDYH	RDYL	Reserved					
Type	R	R	R/W					
Reset	0	0	0	1	1	1	1	1

SFR Page = 0x2; SFR Address = 0xFD

Bit	Name	Function
7	RDYH	**DC0 Ready Indicator (High Threshold).** Indicates when VDC is 100 mV higher than the target output value. 0: VDC pin voltage is less than the DC0 High Threshold. 1: VDC pin voltage is higher than the DC0 High Threshold.
6	RDYL	**DC0 Ready Indicator (Low Threshold).** Indicates when VDC is 100 mV lower than the target output value. 0: VDC pin voltage is less than the DC0 Low Threshold. 1: VDC pin voltage is higher than the DC0 Low Threshold.
5:0	Reserved	Read = 011111b; Must write 011111b.

20.9. DC-DC Converter Specifications

See Table 4.20 on page 77 for a detailed listing of dc-dc converter specifications.

25.4. Automatic Pull-Up Resistor Calibration

The Pulse Counter includes an automatic calibration engine which can automatically determine the minimum pull-up current for a particular application. The automatic calibration is especially useful when the load capacitance of field wiring varies from one installation to another.

The automatic calibration uses one of the Pulse Counter inputs (PC0 or PC1) for calibration. The CALPORT bit in the PC0PCF SFR selects either PC0 or PC1 for calibration. The reed switch on the selected input should be in the open state to allow the signal to charge during calibration. The calibration engine can calibrate the pull-ups with the meter connected normally, provided that the reed switch is open during calibration. During calibration, the integrators will ignore the input comparators, and the counters will not be incremented. Using a 250 µs sample rate and a 32 kHz RTCCLK, the calibration time will be 21 ms (28 tests @ 750 µs each) or shorter depending on the pull up strength selected. The calibration will fail if the reed switch remains closed during this entire period. If the reed switch is both opened and closed during the calibration period, the value written into PCCF[4:0] may be larger than what is actually required. The transition flag in the PC0INT1 can detect when the reed switch opens, and most systems with a wheel rotation of 10 Hz or slower should have sufficient high time for the calibration to complete before the next closing of the reed switch. Slowing the sample rate will also increase the calibration time. The same drive strength will used for both PC0 and PC1.

The example code for the Pulse Counter includes code for managing the automatic calibration engine.

25.5. Sample Rate

The Pulse Counter has a programmable sampling rate. The Pulse Counter samples the state of the reed switches at discrete time intervals based on the RTC clock. The PC0MD SFR sets the sampling rate. The system designer should carefully consider the maximum pulse rate for the particular application when setting the sampling rate and debounce time. Sample rates from 250 µs to 2 ms can be selected to either further reduce power consumption or work with shorter pulse widths. The slowest sampling rate (2 ms) will provide the lowest possible power consumption.

25.6. Debounce

Like most mechanical switches, reed switches exhibit switch bouncing that could potentially result in false counts or quadrature errors. The Pulse Counter includes digital debounce logic using a digital integrator that can eliminate false counts due to switch bounce. The input of the integrator connects to the Pulse Counter inputs with the programmable pull-ups. The output connects to the counters.

The debounce integrator has two independent programmable thresholds: one for the rising edge (Debounce High) and one for the falling edge (Debounce Low). The PC0DCH (PC0 Debounce Config High) SFR sets the threshold for the rising edge. This SFR sets the number of cumulative high samples required to output a logic high to the counter. The PC0DCL (PC0 Debounce Config Low) SFR sets the threshold for the falling edge. This SFR sets the number of cumulative high samples required to output a logic low to the counter.

Note that the debounce does count consecutive samples. Requiring consecutive samples would be susceptible to noise. The digital integrator inherently filters out noise.

The system designer should carefully consider the maximum anticipated counter frequency and duty-cycle when setting the debounce time. If the debounce configuration is set too large, the Pulse Counter will not count short pulses. The debounce-high configuration should be set to less than one-half the minimum input pulse high-time. Similarly, the debounce-low configuration should be set to less than one-half the minimum input pulse low-time.

The Debounce Timing diagram (Figure 25.4) illustrates the operation of the debounce integrator. The top waveform is the representation of the reed switch (high: open, low: closed) which shows some random switch bounce. The bottom waveform is the final signal that goes into the counter which has the switch bounce removed. Based on the actual reed switch used and sample rate, the switch bounce time may appear shorter in duration than the example in Figure 25.4. The second waveform is the pull-up resistor

SFR Definition 26.14. LCD0CHPCF: LCD0 Charge Pump Configuration

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAD

Bit	Name	Function
7:0	Reserved	Must write 0x60.

SFR Definition 26.15. LCD0CHPMD: LCD0 Charge Pump Mode

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	0	0	1

SFR Page = 0x2; SFR Address = 0xAE

Bit	Name	Function
7:0	Reserved	Must write 0xE9.

SFR Definition 26.16. LCD0BUFCN: LCD0 Buffer Control

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Page = 0xF; SFR Address = 0x9C

Bit	Name	Function
7:0	Reserved	Must write 0x44.

28.4.4. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 28.5. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC2

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

28.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. Note that the position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

28.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt.

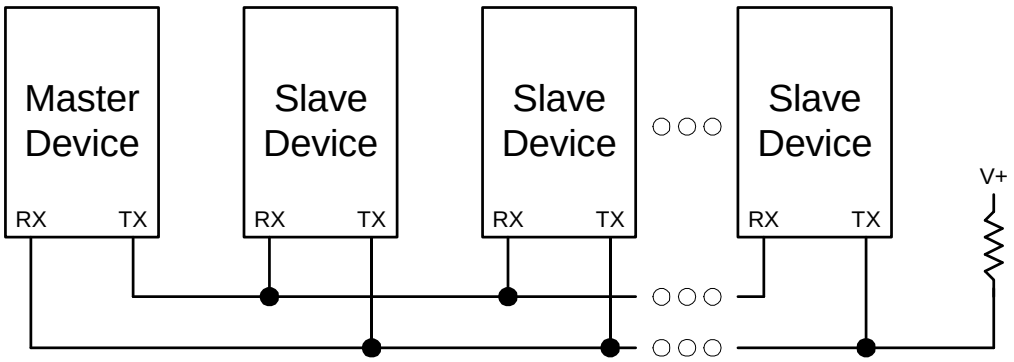

Figure 29.6. UART Multi-Processor Mode Interconnect Diagram

30.1. Signal Descriptions

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

30.1.1. 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.

30.1.2. 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.

30.1.3. 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.

30.1.4. 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 30.2, Figure 30.3, and Figure 30.4 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. See Section “27. Port Input/Output” on page 351 for general purpose port I/O and crossbar information.

30.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag

SFR Definition 32.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 17.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 17.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

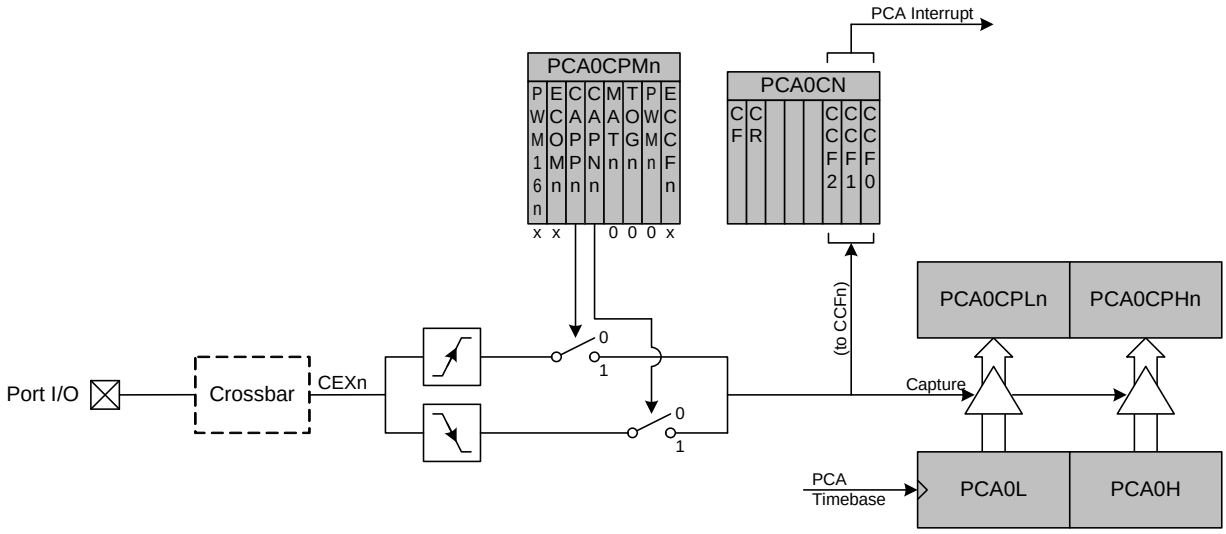

Figure 33.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

33.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

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.