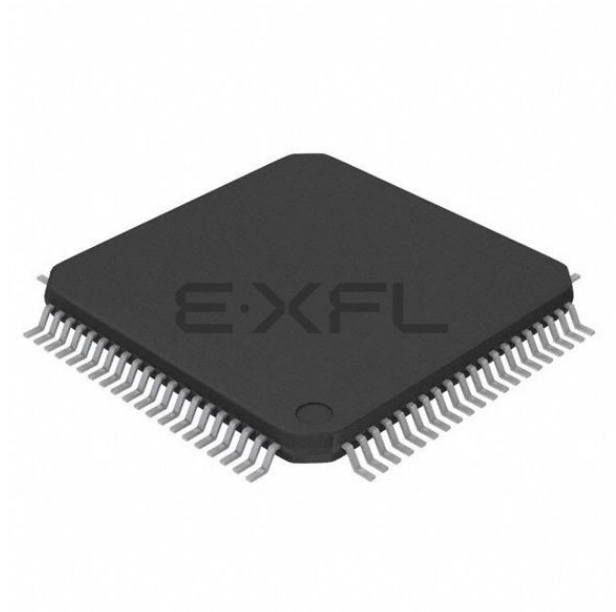


Welcome to E-XFL.COM

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, POR, PWM, WDT
Number of I/O	57
Program Memory Size	128KB (128K 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	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f962-a-gq

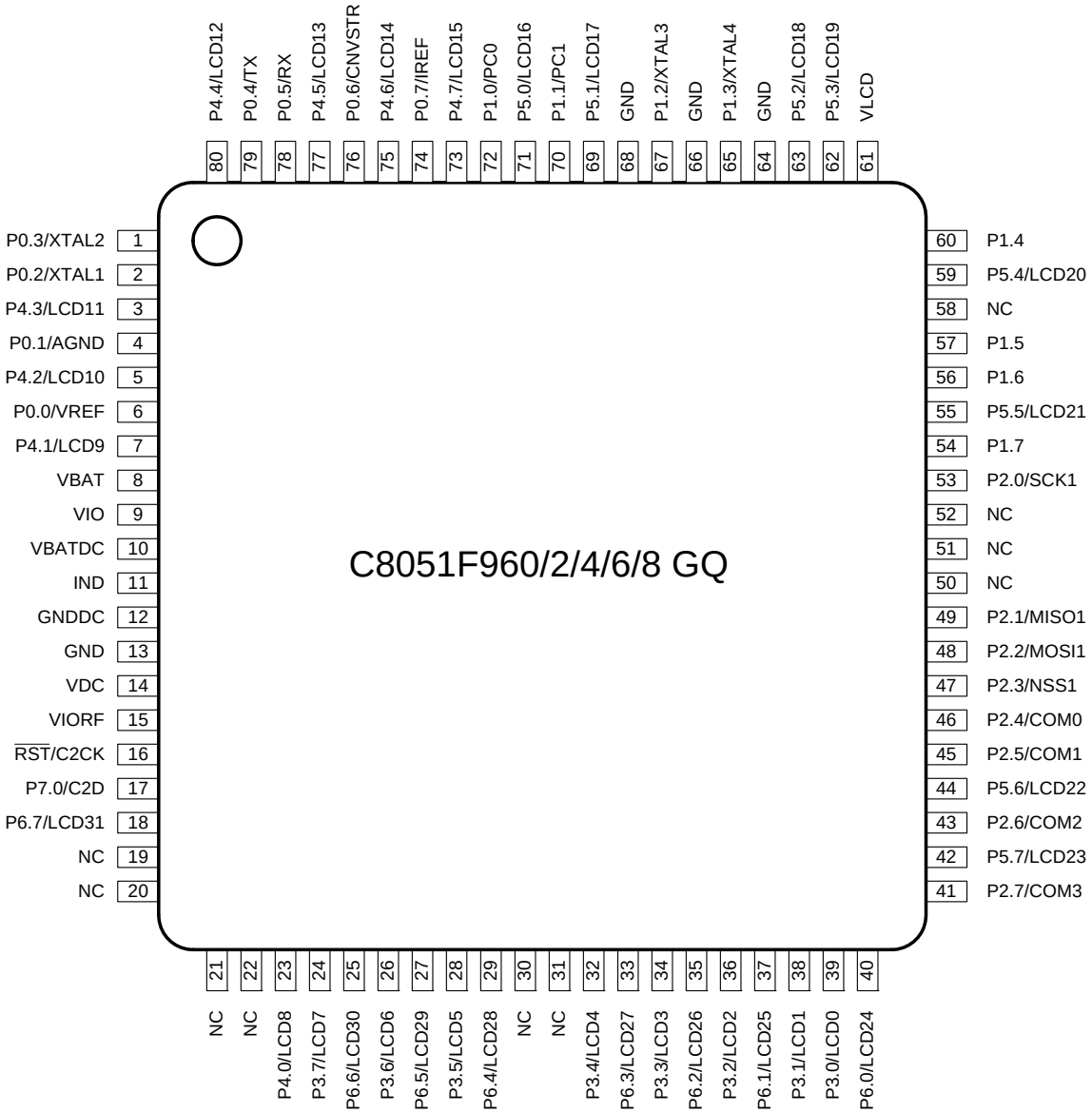


Figure 3.3. TQFP-80 Pinout Diagram (Top View)

5.2.4. Settling Time Requirements

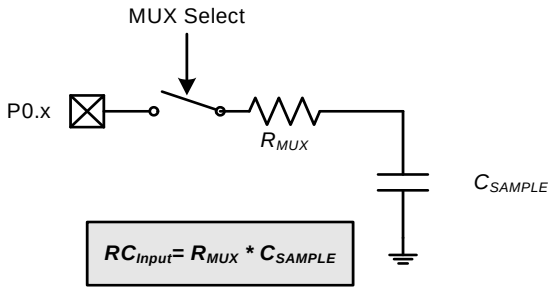
A minimum amount of tracking time is required before each conversion can be performed, to allow the sampling capacitor voltage to settle. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For many applications, these three SAR clocks will meet the minimum tracking time requirements, and higher values for the external source impedance will increase the required tracking time.

Figure 5.4 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation . When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 4.12 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).



Note: The value of CSAMPLE depends on the PGA Gain. See Table 4.12 for details.

Figure 5.4. ADC0 Equivalent Input Circuits

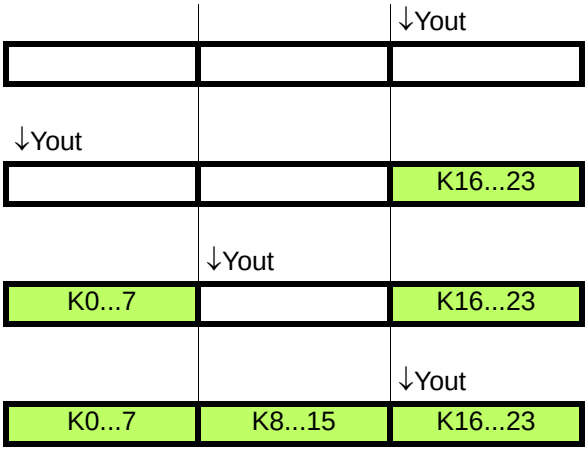
5.2.5. Gain Setting

The ADC has gain settings of 1x and 0.5x. In 1x mode, the full scale reading of the ADC is determined directly by V_{REF}. In 0.5x mode, the full-scale reading of the ADC occurs when the input voltage is V_{REF} x 2. The 0.5x gain setting can be useful to obtain a higher input Voltage range when using a small V_{REF} voltage, or to measure input voltages that are between V_{REF} and V_{DD}. Gain settings for the ADC are controlled by the AMP0GN bit in register ADC0CF.

14.2.4. Using the DMA to unwrap the extended Key

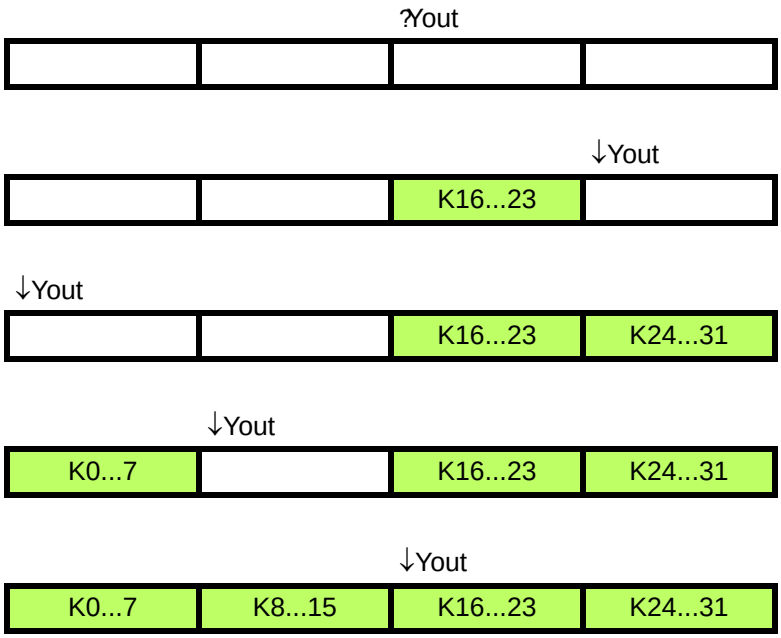
When used with the DMA, the address offset sfr DMANAOH/L may be manipulated to store the extended key in the desired order. This requires two DMA transfers for the AES0YOUT channel. When using a 192-bit key, the DMA0NSZ can be set to 24 bytes and the DMA0NA0 set to 16. This will place the last 8 bytes of the 192-bit key in the desired location as shown in Table 14.2. The Yout arrow indicated the address offset position after each 8-bytes are transferred. Enabling the WRAP bit in DMA0NMD will reset the DMA0NAO value after byte 23. Then the DMA0NZ can be reset to 16 for the remaining sixteen bytes.

Table 14.2. 192-Bit Key DMA Usage



When using a 256-bit key, the DMA0NSZ can be set to 32 and the DMA0NAOL set to 16 This will place the last16 bytes of the 256-bit key in the desired location as shown in Table 14.3.Enabling the WRAP bit in DMA0NMD will reset the DMA0NAO value after byte 31. Then the DMA0NZ can be set to 16 for the remaining sixteen bytes.

Table 14.3. 256-bit Key DMA Usage



15. Encoder/Decoder

The Encoder/Decoder consists of three 8-bit data registers, a control register and an encoder/decoder logic block.

The size of the input data depends on the mode. The input data for Manchester encoding is one byte. For Manchester decoding it is two bytes. Three-out-of-Six encoding is two bytes. Three-out-of six decoding is three bytes.

The output size also depends on the mode selected. The input and output data size are shown below:

Table 15.1. Encoder Input and Output Data Sizes

	Input Data Size	Output Data Size
Operation	**Bytes**	**Bytes**
Manchester Encode	1	2
Manchester Decode	2	1
Three out of Six Encode	2	3
Three out of Six Decode	3	2

The input and output data is always right justified. So for Manchester mode the input uses only ENC0L and the output data is only in ENC0M and ENC0L. ENC0H is not used for Manchester mode

Table 16.2. SFR Map (0x80–0xBF)

Addr.	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xB8	0x0	IP	IREF0CN	ADC0AC	ADC0MX	ADC0CF	ADC0L	ADC0H	P1MASK
	0x2		CRC1IN	CRC1OUTL	CRC1OUTH	CRC1POLL	CRC1POLH	CRC1CN	
	0xF		IREF0CF	ADC0PWR	ADC0TK		TOFFL	TOFFH	
0xB0	0x0	P3	OSCXCN	OSCICN	PMU0MD		PMU0CF	PMU0FL	FLKEY
	0x2		DC0CN	DC0CF	DC0MD		LCD0CHPCN	LCD0BUFMD	
	0xF		P3MDOUT	OSCIFL	OSCICL			FLSCL	
0xA8	0x0	IE	CLKSEL	EMI0CN	EMI0CF	RTC0ADR	RTC0DAT	RTC0KEY	EMI0TC
	0x2		LCD0CLKDIVL	LCD0CLKDIVH	LCD0MSCN	LCD0MSCF	LCD0CHPCF	LCD0CHPMD	LCD0VBMCF
	0xF		CLKSEL	P6DRV	P7DRV	LCD0BUFCF			
0xA0	0x0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE
	0x2		SPI1CFG	SPI1CKR	SPI1DAT	LCD0PWR	LCD0CF	LCD0VBMCN	
	0xF		P3DRV	P4DRV	P5DRV	P0DRV	P1DRV	P2DRV	
0x98	0x0	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
	0x2		LCD0DD	LCD0DE	LCD0DF	LCD0CNTRST	LCD0CN	LCD0BLINK	LCD0TOGR
	0xF					LCD0BUFCN			
0x90	0x0	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H		
	0x2		LCD0D6	LCD0D7	LCD0D8	LCD0D9	LCD0DA	LCD0DB	LCD0DC
	0xF		CRC0DAT	CRC0CN	CRC0IN	CRC0FLIP		CRC0AUTO	CRC0CNT
0x88	0x0	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
	0x2		LCD0D0	LCD0D1	LCD0D2	LCD0D3	LCD0D4	LCD0D5	
	0xF							SFRPGCN	
0x80	0x0	P0	SP	DPL	DPH	PSBANK	SFRNEXT	SFRLAST	PCON
	0x2								
	0xF								

17.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If a high priority interrupt preempts a low priority interrupt, the low priority interrupt will finish execution after the high priority interrupt completes. Each interrupt has an associated interrupt priority bit in in the Interrupt Priority and Extended Interrupt Priority registers used to configure its priority level. Low priority is the default.

If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate. See Table 17.1 on page 239 to determine the fixed priority order used to arbitrate between simultaneously recognized interrupts.

17.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

SFR Definition 17.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PRTC0A	PSMB0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xF6

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	PRTC0A	**SmaRTClock Alarm Interrupt Priority Control.** This bit sets the priority of the SmaRTClock Alarm interrupt. 0: SmaRTClock Alarm interrupt set to low priority level. 1: SmaRTClock Alarm interrupt set to high priority level.
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.

18.5. Flash Write and Erase Guidelines

Any system which contains routines which write or erase flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of VDD, system clock frequency, or temperature. This accidental execution of flash modifying code can result in alteration of flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

To help prevent the accidental modification of flash by firmware, the VDD Monitor must be enabled and enabled as a reset source on C8051F96x devices for the flash to be successfully modified. **If either the VDD Monitor or the VDD Monitor reset source is not enabled, a Flash Error Device Reset will be generated when the firmware attempts to modify the flash.**

The following guidelines are recommended for any system that contains routines which write or erase flash from code.

18.5.1. VDD Maintenance and the VDD Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum VDD rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external VDD brownout circuit to the $\overline{\text{RST}}$ pin of the device that holds the device in reset until VDD reaches the minimum device operating voltage and re-asserts $\overline{\text{RST}}$ if VDD drops below the minimum device operating voltage.
3. Keep the on-chip VDD Monitor enabled and enable the VDD Monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the 'C' compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware," available from the Silicon Laboratories web site.

Notes:
On C8051F96x devices, both the VDD Monitor and the VDD Monitor reset source must be enabled to write or erase flash without generating a Flash Error Device Reset.

On C8051F96x devices, both the VDD Monitor and the VDD Monitor reset source are enabled by hardware after a power-on reset.

4. As an added precaution, explicitly enable the VDD Monitor and enable the VDD Monitor as a reset source inside the functions that write and erase flash memory. The VDD Monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct, but "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

SFR Definition 25.2. PC0PCF: PC0 Mode Pull-Up Configuration

Bit	7	6	5	4	3	2	1	0
Name	PUCAL	CALRES	CALPORT	RES[2:0]			DUTY[1:0]	
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Address = 0xD7; SFR Page = 0x2

Bit	Name	Function
7	PUCAL	**Pull-Up Driver Calibration** 0: Calibration complete or not running. 1: Start calibration of pull up (Self clearing). Calibration determines the lowest usable pull-up strength.
6	CALRES	**Calibration Result** 0: Fail (switch may be closed preventing detection of pull ups). Writes value of 0x11111 to PC0PCF[4:0] 1: Pass (writes calibrated value into PC0PCF[4:0]).
5	CALPORT	**Calibration Port** 0: Calibration on PC0 only. 1: Calibration on PC1 only.
4:2	RES[2:0]	**Pull-Up Resistor Select** Current with force pull-up on bit set (PC0TH.2=1) and VBAT=3.6V. 000: Pull-up disabled. 001: 1 μA.* 010: 4 μA.* 011: 16 μA.* 100: 64 μA.* 101: 256 μA.* 110: 1 mA.* 111: 4 mA.* *The effective average pull-up current depends on selected resistor, pull-up resistor duty-cycle multiplier, and sample rate duty-cycle multiplier.
1:0	DUTY[1:0]	**Pull-Up Resistor Duty Cycle Multiplier** 000: 1/4 (25%)* 001: 3/8 (37.5%)* 010: 1/2 (50%)* 011: 3/4 (75%)* *The final pull-up resistor duty cycle is the sample rate duty-cycle multiplier times the pull-up duty-cycle multiplier.

SFR Definition 25.10. PC0CTR1H: PC0 Counter 1 High (MSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1H[23:16]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDF; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1H[23:16]	**PC0 Counter 1 High Byte** Bits 23:16 of Counter 1.

SFR Definition 25.11. PC0CTR1M: PC0 Counter 1 Middle

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1M[15:8]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDE; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1M[15:8]	**PC0 Counter 1 Middle Byte** Bits 15:8 of Counter 1.

SFR Definition 25.12. PC0CTR1L: PC0 Counter 1 Low (LSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1L[7:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDD; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1L[7:0]	**PC0 Counter 1 Low Byte** Bits 7:0 of Counter 1.

Note: PC0CTR1L must be read before PC0CTR1M and PC0CTR1H to latch the count for reading.

26.5. Setting the LCD Refresh Rate

The clock to the LCD0 module is derived from the SmaRTClock and may be divided down according to the settings in the LCD0CN register. The LCD refresh rate is derived from the LCD0 clock and can be programmed using the LCD0DIVH:LCD0DIVL registers. The LCD mux mode must be taken into account when determining the prescaler value. See the LCD0DIVH/LCD0DIVL register descriptions for more details. For maximum power savings, choose a slow LCD refresh rate and the minimum LCD0 clock frequency. For the least flicker, choose a fast LCD refresh rate.

SFR Definition 26.8. LCD0CLKDIVH: LCD0 Refresh Rate Prescaler High Byte

Bit	7	6	5	4	3	2	1	0
Name							LCD0DIV[9:8]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAA

Bit	Name	Function
7:2	Unused	Read = 000000. Write = Don't Care.
1:0	LCD0DIV[9:8]	**LCD Refresh Rate Prescaler**. Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux\_mode} \times (LCD0DIV + 1)}$

SFR Definition 26.9. LCD0CLKDIVL: LCD Refresh Rate Prescaler Low Byte

Bit	7	6	5	4	3	2	1	0
Name	LCD0DIV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA9

Bit	Name	Function
7:0	LCD0DIV[7:0]	**LCD Refresh Rate Prescaler**. Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux\_mode} \times (LCD0DIV + 1)}$

27.4. Port Match

Port match functionality allows system events to be triggered by a logic value change on P0 or P1. A software controlled value stored in the PnMAT registers specifies the expected or normal logic values of P0 and P1. A Port mismatch event occurs if the logic levels of the Port's input pins no longer match the software controlled value. This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings.

The PnMASK registers can be used to individually select which P0 and P1 pins should be compared against the PnMAT registers. A Port mismatch event is generated if (P0 & P0MASK) does not equal (PnMAT & P0MASK) or if (P1 & P1MASK) does not equal (PnMAT & P1MASK).

A Port mismatch event may be used to generate an interrupt or wake the device from a low power mode. See Section "17. Interrupt Handler" on page 237 and Section "19. Power Management" on page 262 for more details on interrupt and wake-up sources.

SFR Definition 27.4. P0MASK: Port0 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P0MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xC7

Bit	Name	Function
7:0	P0MASK[7:0]	**Port0 Mask Value.** Selects the P0 pins to be compared with the corresponding bits in P0MAT. 0: P0.n pin pad logic value is ignored and cannot cause a Port Mismatch event. 1: P0.n pin pad logic value is compared to P0MAT.n.

SFR Definition 27.5. P0MAT: Port0 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P0MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page= 0x0; SFR Address = 0xD7

Bit	Name	Function
7:0	P0MAT[7:0]	**Port 0 Match Value.** Match comparison value used on Port 0 for bits in P0MASK which are set to 1. 0: P0.n pin logic value is compared with logic LOW. 1: P0.n pin logic value is compared with logic HIGH.

SFR Definition 27.26. P3DRV: Port3 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P3DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA1

Bit	Name	Function
7:0	P3DRV[7:0]	**Drive Strength Configuration Bits for P3.7–P3.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P3.n Output has low output drive strength. 1: Corresponding P3.n Output has high output drive strength.

SFR Definition 27.27. P4: Port4

Bit	7	6	5	4	3	2	1	0
Name	P4[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xD9

Bit	Name	Description	Read	Write
7:0	P4[7:0]	**Port 4 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P4.n Port pin is logic LOW. 1: P4.n Port pin is logic HIGH.

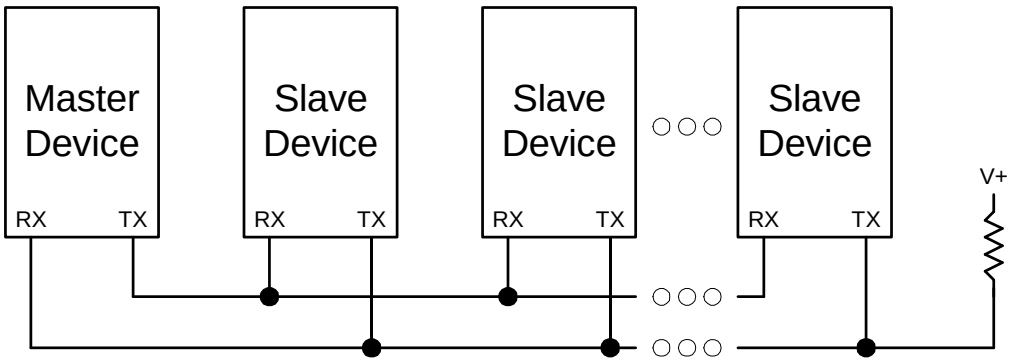


Figure 29.6. UART Multi-Processor Mode Interconnect Diagram

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 356 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 30.1. SPI0CFG: SPI0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Page = 0x0; SFR Address = 0xA1

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI0 Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI0 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.

Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 30.1 for timing parameters.

31.12. SPI Special Function Registers

SPI1 is accessed and controlled through four special function registers in the system controller: SPI1CN Control Register, SPI1DAT Data Register, SPI1CFG Configuration Register, and SPI1CKR Clock Rate Register. The four special function registers related to the operation of the SPI1 Bus are described in the following SFR definitions.

SFR Definition 32.9. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCA

Bit	Name	Function
7:0	TMR2RLL[7:0]	**Timer 2 Reload Register Low Byte.** TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 32.10. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCB

Bit	Name	Function
7:0	TMR2RLH[7:0]	**Timer 2 Reload Register High Byte.** TMR2RLH holds the high byte of the reload value for Timer 2.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:
https://www.silabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.