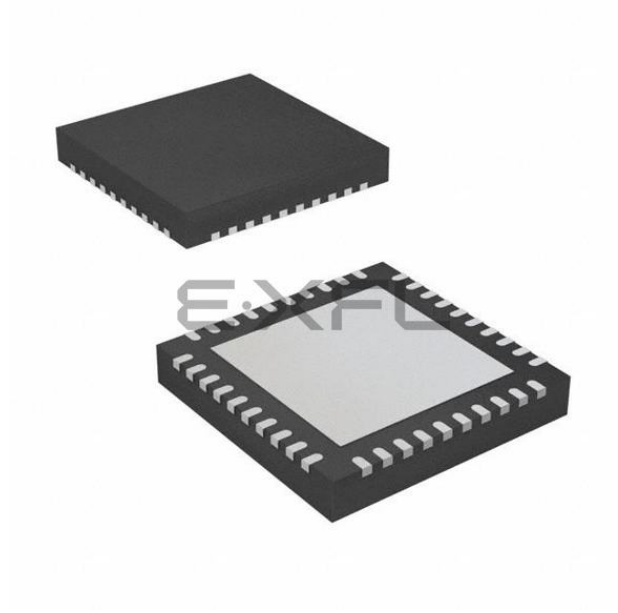

Welcome to E-XFL.COM

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, LCD, POR, PWM, WDT
Number of I/O	34
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	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f965-a-gm

List of Registers

1.2. Port Input/Output

Digital and analog resources are available through 57 I/O pins (C8051F960/2/4/6/8) or 34 I/O pins (C8051F961/3/5/7/9). Port pins are organized as eight byte-wide ports. Port pins can be defined as digital or analog I/O. Digital I/O pins can be assigned to one of the internal digital resources or used as general purpose I/O (GPIO). Analog I/O pins are used by the internal analog resources. P7.0 can be used as GPIO and is shared with the C2 Interface Data signal (C2D). See Section “34. C2 Interface” on page 490 for more details.

The designer has complete control over which digital and analog functions are assigned to individual port pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. See Section “27. Port Input/Output” on page 356 for more information on the Crossbar.

For Port I/Os configured as push-pull outputs, current is sourced from the VIO, VIORF, or VBAT supply pin. Port I/Os used for analog functions can operate up to the supply voltage. See Section “27. Port Input/Output” on page 356 for more information on Port I/O operating modes and the electrical specifications chapter for detailed electrical specifications.

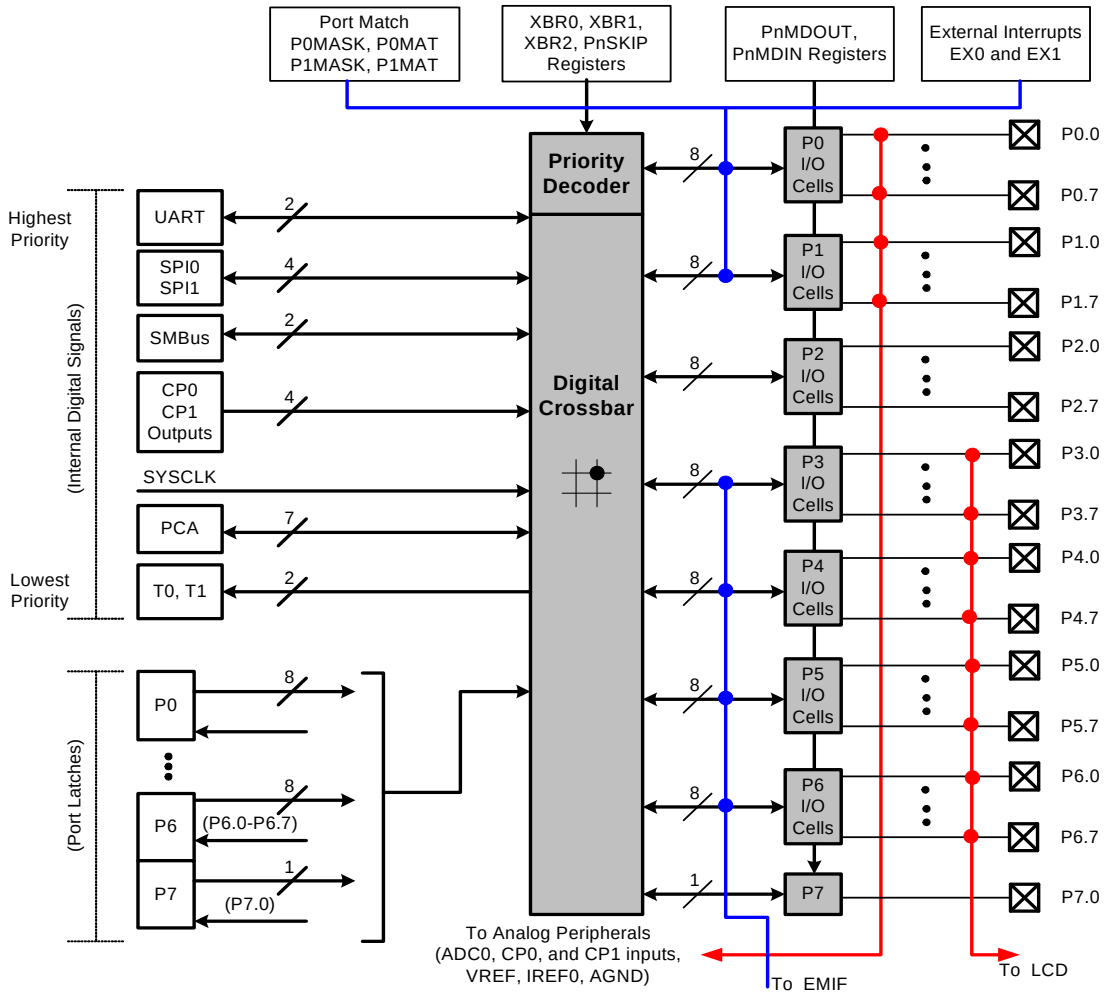

Figure 1.11. Port I/O Functional Block Diagram

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

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—Idle Mode (CPU Inactive, not Fetching Instructions from Flash)					
I_{BAT}[2]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	3.5	—	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	2.6	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	340 360	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	230[5]	—	µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	135	—	µA/MHz

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.

SFR Definition 5.2. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD08BE	AD0TM	AMP0GN
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	0

SFR Page = 0x0; SFR Address = 0xBC

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Divider.** SAR Conversion clock is derived from FCLK by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC[4:0]. SAR Conversion clock requirements are given in Table 4.12. BURSTEN = 0: FCLK is the current system clock. BURSTEN = 1: FCLK is the 20 MHz low power oscillator, independent of the system clock. $AD0SC = \frac{FCLK}{CLK_{SAR}} - 1$ * *Round the result up. or $CLK_{SAR} = \frac{FCLK}{AD0SC + 1}$
2	AD08BE	**ADC0 8-Bit Mode Enable.** 0: ADC0 operates in 10-bit mode (normal operation). 1: ADC0 operates in 8-bit mode.
1	AD0TM	**ADC0 Track Mode.** Selects between Normal or Delayed Tracking Modes. 0: Normal Track Mode: When ADC0 is enabled, conversion begins immediately following the start-of-conversion signal. 1: Delayed Track Mode: When ADC0 is enabled, conversion begins 3 SAR clock cycles following the start-of-conversion signal. The ADC is allowed to track during this time.
0	AMP0GN	**ADC0 Gain Control.** 0: The on-chip PGA gain is 0.5. 1: The on-chip PGA gain is 1.

SFR Definition 5.8. ADC0GTH: ADC0 Greater-Than High Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0GT[15:8]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xC4

Bit	Name	Function
7:0	AD0GT[15:8]	**ADC0 Greater-Than High Byte.** Most Significant Byte of the 16-bit Greater-Than window compare register.

SFR Definition 5.9. ADC0GTL: ADC0 Greater-Than Low Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0GT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xC3

Bit	Name	Function
7:0	AD0GT[7:0]	**ADC0 Greater-Than Low Byte.** Least Significant Byte of the 16-bit Greater-Than window compare register.
Note: In 8-bit mode, this register should be set to 0x00.		

5.10. External Voltage Reference

To use an external voltage reference, REFSL[1:0] should be set to 00. Bypass capacitors should be added as recommended by the manufacturer of the external voltage reference. If the manufacturer does not provide recommendations, a 4.7uF in parallel with a 0.1uF capacitor is recommended.

5.11. Internal Voltage Reference

For applications requiring the maximum number of port I/O pins, or very short VREF turn-on time, the 1.65 V high-speed reference will be the best internal reference option to choose. The high speed internal reference is selected by setting REFSL[1:0] to 11. When selected, the high speed internal reference will be automatically enabled/disabled on an as-needed basis by ADC0.

For applications with a non-varying power supply voltage, using the power supply as the voltage reference can provide ADC0 with added dynamic range at the cost of reduced power supply noise rejection. To use the 1.8 to 3.6 V power supply voltage (V_{DD}) or the 1.8 V regulated digital supply voltage as the reference source, REFSL[1:0] should be set to 01 or 10, respectively.

5.12. Analog Ground Reference

To prevent ground noise generated by switching digital logic from affecting sensitive analog measurements, a separate analog ground reference option is available. When enabled, the ground reference for ADC0 during both the tracking/sampling and the conversion periods is taken from the P0.1/AGND pin. Any external sensors sampled by ADC0 should be referenced to the P0.1/AGND pin. This pin should be connected to the ground terminal of any external sensors sampled by ADC0. If an external voltage reference is used, the P0.1/AGND pin should be connected to the ground of the external reference and its associated decoupling capacitor. The separate analog ground reference option is enabled by setting REFGND to 1. Note that when sampling the internal temperature sensor, the internal chip ground is always used for the sampling operation, regardless of the setting of the REFGND bit. Similarly, whenever the internal 1.65 V high-speed reference is selected, the internal chip ground is always used during the conversion period, regardless of the setting of the REFGND bit.

5.13. Temperature Sensor Enable

The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data. See Section “5.8. Temperature Sensor” on page 97 for details on temperature sensor characteristics when it is enabled.

SFR Definition 11.3. DMA0MINT: DMA0 Mid-Point Interrupt

Bit	7	6	5	4	3	2	1	0
Name		CH6_MINT	CH5_MINT	CH4_MINT	CH3_MINT	CH2_MINT	CH1_MINT	CH0_MINT
Type	R	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 = 0xD4

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_MINT	**Channel 6 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 6. 1: Mid-Point interrupt has not occured on channel 6.
5	CH5_MINT	**Channel 5 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 5. 1: Mid-Point interrupt has not occured on channel 5.
4	CH4_MINT	**Channel 4 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 4. 1: Mid-Point interrupt has not occured on channel 4**.**
3	CH3_MINT	**Channel 3 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 3. 1: Mid-Point interrupt has not occured on channel 3.
2	CH2_MINT	**Channel 2 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 2. 1: Mid-Point interrupt has not occured on channel 2.
1	CH1_MINT	**Channel 1 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 1. 1: Mid-Point interrupt has not occured on channel 1.
0	CH0_MINT	**Channel 0 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 0. 1: Mid-Point interrupt has not occured on channel 0.

Note: Mid-point Interrupt flag is set when the offset address DMA0NAOH/L equals to half of data transfer size DMA0NSZH/L if the transfer size is an even number or half of data transfer size DMA0NSZH/L plus one if the transfer size is an odd number. This flag must be cleared by software or system reset.The mid-point interrupt is enabled by setting bit 6 of DMA0NCF with DMA0SEL configured for the corresponding channel.

SFR Definition 11.6. DMA0NMD: DMA Channel Mode

Bit	7	6	5	4	3	2	1	0
Name								WRAP
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 = 0x2; SFR Address = 0xD6

Bit	Name	Function
7:1	reserved	Read = 0, Write = 0
0	WRAP	**Wrap Enable.** Setting this bit will enable wrapping. The DMA0NSZ register sets the transfer size. Normally the DMA0AO value starts at zero in increases to the DMANSZ minus one. At this point the transfer is complete and the interrupt bit will be set. If the WRAP bit is set, the DMA0NAO will be reset to zero.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 12.1. CRC0CN: CRC0 Control

Bit	7	6	5	4	3	2	1	0
Name				CRC0SEL	CRC0INIT	CRC0VAL	CRC0PNT[1:0]	
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 = 0xF; SFR Address = 0x92

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4	CRC0SEL	**CRC0 Polynomial Select Bit.** This bit selects the CRC0 polynomial and result length (32-bit or 16-bit). 0: CRC0 uses the 32-bit polynomial 0x04C11DB7 for calculating the CRC result. 1: CRC0 uses the 16-bit polynomial 0x1021 for calculating the CRC result.
3	CRC0INIT	**CRC0 Result Initialization Bit.** Writing a 1 to this bit initializes the entire CRC result based on CRC0VAL.
2	CRC0VAL	**CRC0 Set Value Initialization Bit.** This bit selects the set value of the CRC result. 0: CRC result is set to 0x00000000 on write of 1 to CRC0INIT. 1: CRC result is set to 0xFFFFFFFF on write of 1 to CRC0INIT.
1:0	CRC0PNT[1:0]	**CRC0 Result Pointer.** Specifies the byte of the CRC result to be read/written on the next access to CRC0DAT. The value of these bits will auto-increment upon each read or write. For CRC0SEL = 0: 00: CRC0DAT accesses bits 7–0 of the 32-bit CRC result. 01: CRC0DAT accesses bits 15–8 of the 32-bit CRC result. 10: CRC0DAT accesses bits 23–16 of the 32-bit CRC result. 11: CRC0DAT accesses bits 31–24 of the 32-bit CRC result. For CRC0SEL = 1: 00: CRC0DAT accesses bits 7–0 of the 16-bit CRC result. 01: CRC0DAT accesses bits 15–8 of the 16-bit CRC result. 10: CRC0DAT accesses bits 7–0 of the 16-bit CRC result. 11: CRC0DAT accesses bits 15–8 of the 16-bit CRC result.

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.

14.3. AES Block Cipher

The basic AES Block Cipher is the basic encryption/decryption algorithm as defined by the NIST standard. A clock cipher mode is a method of encrypting and decrypting one block of data. The input data and output data are not manipulated, chained, or exclusive ORed with other data. This simple block cipher mode is sometimes called the Electronic Code Book (ECB) mode. The Electronic Codebook Mode is illustrated in Figure 14.3

Each operation represents one block (sixteen bytes) of data. The Plaintext is the plain unencrypted data. The Ciphertext is the encrypted data. The encryption key and decryption keys are symmetric. The decryption key is the inverse key of the decryption key. Note that the Encryption operation is not the same as the decryption operation. The two operations are different and the AES core operates differently depending on whether encryption or decryption is selected.

Note that each encryption or decryption operation is independent of other operations. Also note that the same key is used over and over again for each operation.

15.3. Three-out-of-Six Encoding

Three out of six encoding is similar to Manchester encoding. In Three-out-of-Six encoding a nibble is encoded as a six-bit symbol. Four nibbles are encoded as 24-bits (three bytes).

Two bytes of data to be encoded are written to ENC0M and ENC0L. The MODE bit is set to 1 for Three-out-of-Six encoding. Setting the ENC bit will initiate encoding.

After encoding, the three encoded bytes are in ENC2-0.

Table 15.4. Three-out-of-Six Encoding Nibble

Input			**Encoded Output**			
nibble			**symbol**			
dec	hex	bin	bin	dec	hex	octal
0	0	0000	010110	22	16	26
1	1	0001	001101	13	0D	15
2	2	0010	001110	14	0E	16
3	3	0011	001011	11	0B	13
4	4	0100	011100	28	1C	34
5	5	0101	011001	25	19	31
6	6	0110	011010	26	1A	32
7	7	0111	010011	19	13	23
8	8	1000	101100	44	2C	54
9	9	1001	100101	37	25	45
10	A	1010	100110	38	26	46
11	B	1011	100011	35	23	43
12	C	1100	110100	52	34	64
13	D	1101	110001	49	31	61
14	E	1110	110010	50	32	62
15	F	1111	101001	41	29	51

16.3. SFR Page Stack Example

The following is an example that shows the operation of the SFR Page Stack during interrupts. In this example, the SFR Control register is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to SMBus Address Register (SFR “SMB0ADR”, located at address 0xF4 on SFR Page 0x0). The device is also using the SPI peripheral (SPI0) and the Programmable Counter Array (PCA0) peripheral to generate a PWM output. The PCA is timing a critical control function in its interrupt service routine, and so its associated ISR is set to high priority. At this point, the SFR page is set to access the SMB0ADR SFR (SFRPAGE = 0x0). See Figure 16.2.

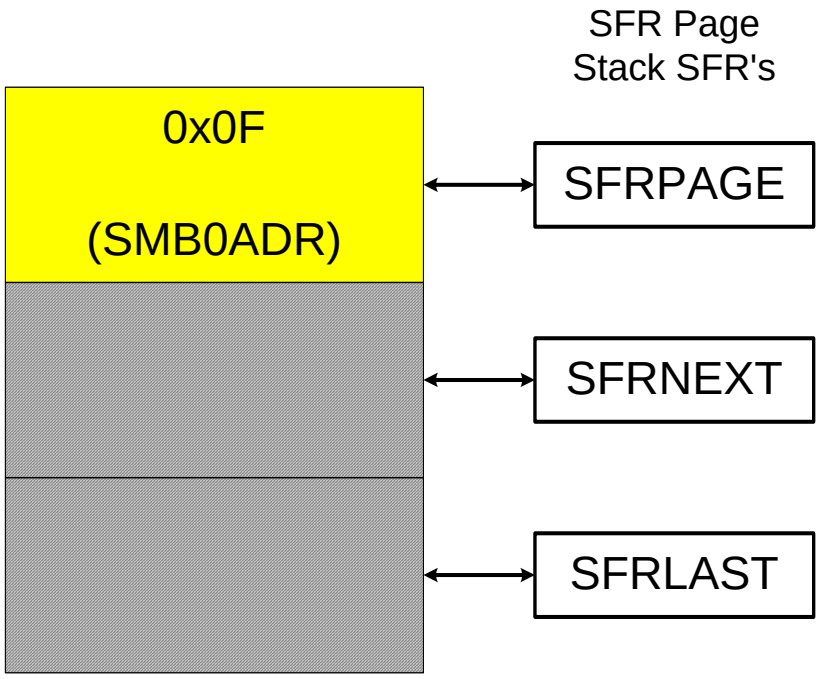

Figure 16.2. SFR Page Stack While Using SFR Page 0x0 To Access SMB0ADR

While CIP-51 executes in-line code (writing a value to SMB0ADR in this example), the SPI0 Interrupt occurs. The CIP-51 vectors to the SPI0 ISR and pushes the current SFR Page value (SFR Page 0x0F) into SFRNEXT in the SFR Page Stack. SFRPAGE is considered the “top” of the SFR Page Stack. Software may switch to any SFR Page by writing a new value to the SFRPAGE register at any time during the SPI0 ISR. See Figure 16.3.

26.3.2. Contrast Control Mode 2 (Minimum Contrast Mode)

In Contrast Control Mode 2, a minimum contrast voltage is maintained, as shown in Figure 26.4. The VLCD supply is powered directly from VBAT as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to acheive the desired minimum contrast voltage on VLCD. Minimum Contrast Mode is selected using the following procedure:

1. Clear Bit 2 of the LCD0MSCN register to 0b (LCD0MSCN &= ~0x04)
2. Set Bit 0 of the LCD0MSCF register to 1b (LCD0MSCF |= 0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

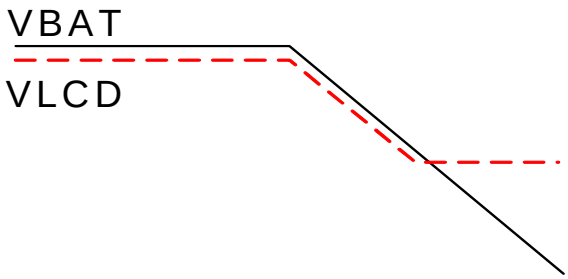

Figure 26.4. Contrast Control Mode 2

26.3.3. Contrast Control Mode 3 (Constant Contrast Mode)

In Contrast Control Mode 3, a constant contrast voltage is maintained. The VLCD supply is regulated to the programmed contrast voltage using a variable resistor between VBAT and VLCD as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to acheive the desired contrast voltage on VLCD. Constant Contrast Mode is selected using the following procedure:

1. Set Bit 2 of the LCD0MSCN register to 1b (LCD0MSCN |= 0x04)
2. Clear Bit 0 of the LCD0MSCF register to 0b (LCD0MSCF &= ~0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

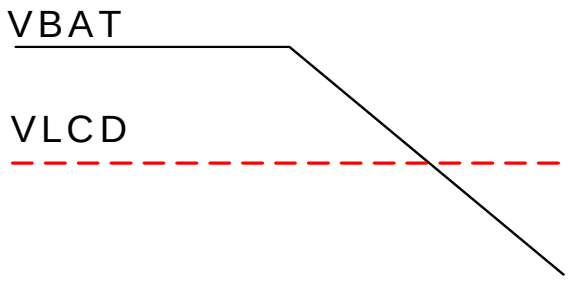

Figure 26.5. Contrast Control Mode 3

28.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 28.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 28.3 for more details.
Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

28.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

28.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 28.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.
Table 28.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 28.5 for SMBus status decoding using the SMB0CN register.

Table 29.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:
1. SCA1−SCA0 and T1M bit definitions can be found in Section 32.1.
2. X = Don't care.

30. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

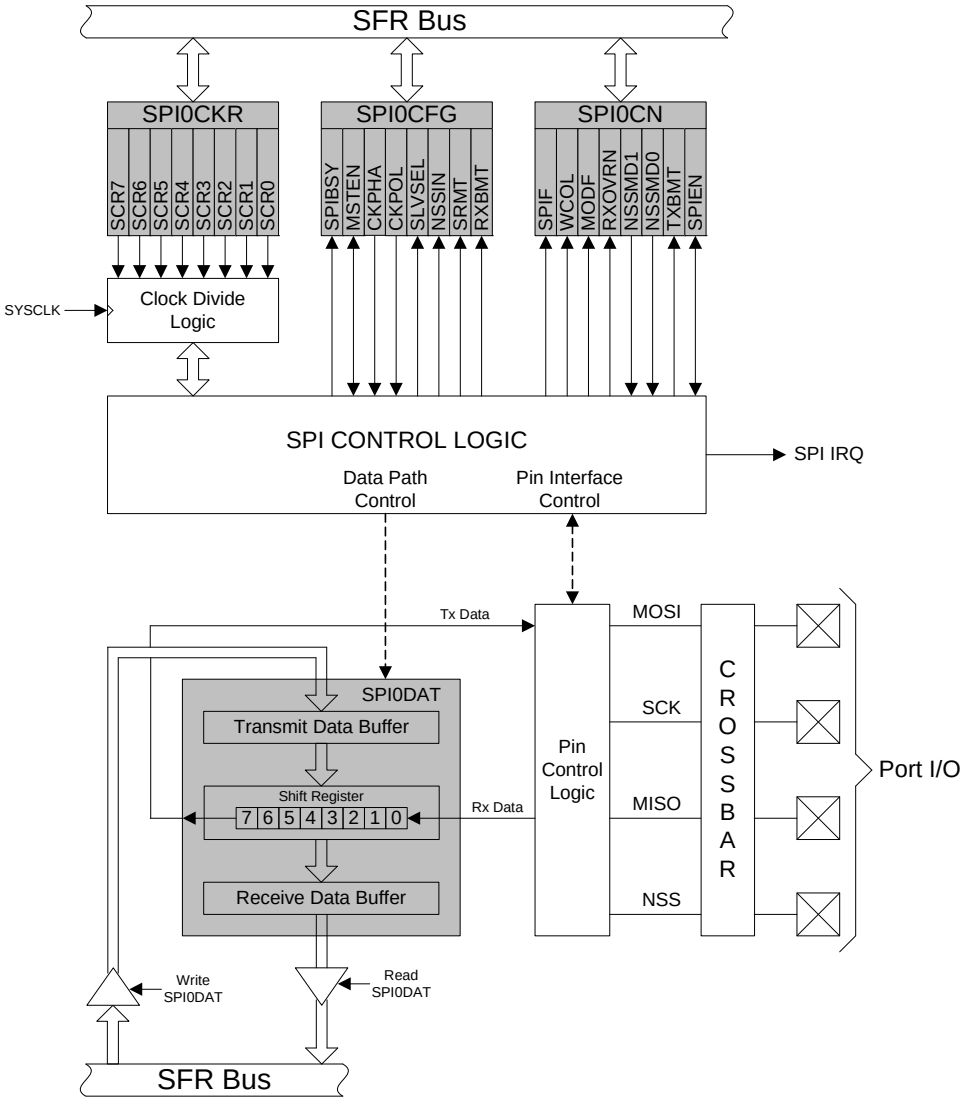

Figure 30.1. SPI Block Diagram

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

Setting TF3CEN to 1 enables the SmaRTClock/External Oscillator Capture Mode for Timer 3. In this mode, T3SPLIT should be set to 0, as the full 16-bit timer is used.

When Capture Mode is enabled, a capture event will be generated either every SmaRTClock rising edge or every 8 external clock cycles, depending on the T3XCLK1 setting. When the capture event occurs, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set (triggering an interrupt if Timer 3 interrupts are enabled). By recording the difference between two successive timer capture values, the SmaRTClock or external clock period can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T3ML = 1b, T3XCLK1 = 0b, and TF3CEN = 1b, Timer 3 will clock every SYSCLK and capture every SmaRTClock rising edge. If SYSCLK is 24.5 MHz and the difference between two successive captures is 350 counts, then the SmaRTClock period is as follows:

350 x (1 / 24.5 MHz) = 14.2 µs.

This mode allows software to determine the exact frequency of the external oscillator in C and RC mode or the time between consecutive SmaRTClock rising edges, which is useful for determining the SmaRTClock frequency.

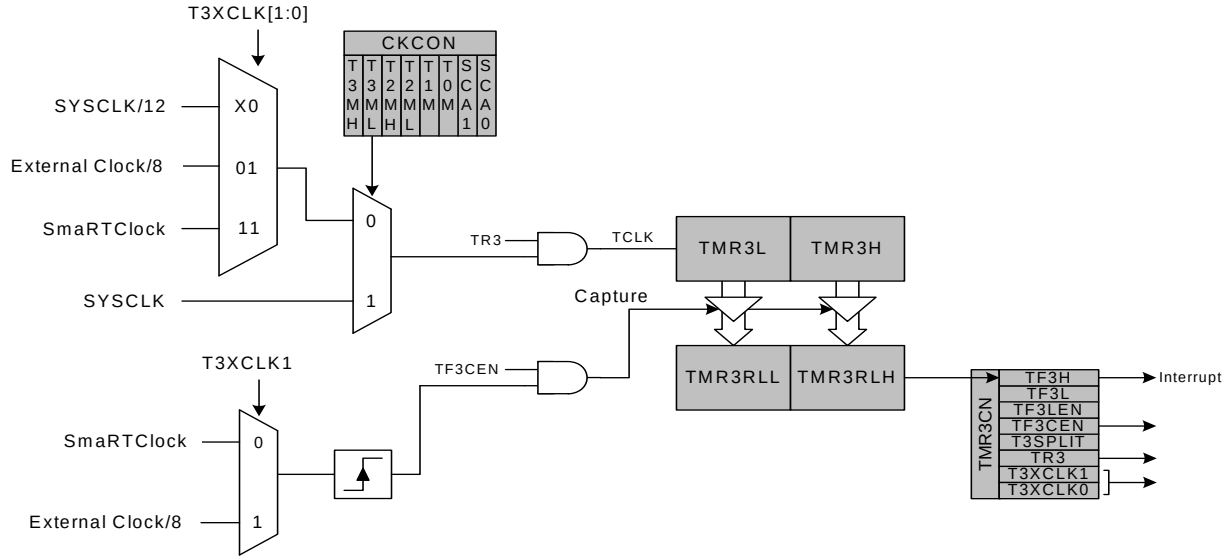

Figure 32.9. Timer 3 Capture Mode Block Diagram

C2 Register Definition 34.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 34.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xB4

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase