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-gqr

List of Registers

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P6.2	B12	35		D I/O or A In	Port 6.2. See Port I/O Section for a complete description.
LCD26				A O	LCD Segment Pin 26
P6.3	B11	33		D I/O or A In	Port 6.3. See Port I/O Section for a complete description.
LCD27				A O	LCD Segment Pin 27
P6.4	B10	29		D I/O or A In	Port 6.4. See Port I/O Section for a complete description.
LCD28				A O	LCD Segment Pin 28
P6.5	B9	27		D I/O or A In	Port 6.5. See Port I/O Section for a complete description.
LCD29				A O	LCD Segment Pin 29
P6.6	B8	25		D I/O or A In	Port 6.6. See Port I/O Section for a complete description.
LCD30				A O	LCD Segment Pin 30
P6.7	D2	18		D I/O or A In	Port 6.7. See Port I/O Section for a complete description.
LCD31				A O	LCD Segment Pin 31

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 (Sleep Mode, SmaRTClock running, 32.768 kHz Crystal, LCD Contrast Mode 3 (2.7 V), charge pump enabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD 1.8 V, T = 25 °C, 2-Mux LCD 1.8 V, T = 25 °C, 3-Mux LCD 1.8 V, T = 25 °C, 4-Mux LCD	— — — —	1.3 1.8 1.8 2.0	— — — —	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.

SFR Definition 11.4. DMA0BUSY: DMA0 Busy

Bit	7	6	5	4	3	2	1	0
Name		CH6_BUSY	CH5_BUSY	CH4_BUSY	CH3_BUSY	CH2_BUSY	CH1_BUSY	CH0_BUSY
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 = 0xD5

Bit	Name	Description	Write	Read
7	Unused		No effect.	Always Reads 0.
6	CH6_BUSY	**Channel 6 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 6.	0: DMA0 channel 6 Idle. 1: DMA0 transfer in progress on channel 6.
5	CH5_BUSY	**Channel 5 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 5.	0: DMA0 channel 5 Idle. 1: DMA0 transfer in progress on channel 5.
4	CH4_BUSY	**Channel 4 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 4.	0: DMA0 channel 4 Idle. 1: DMA0 transfer in progress on channel 4.
3	CH3_BUSY	**Channel 3 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 3.	0: DMA0 channel 3 Idle. 1: DMA0 transfer in progress on channel 3.
2	CH2_BUSY	**Channel 2 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 2.	0: DMA0 channel 2 Idle. 1: DMA0 transfer in progress on channel 2.
1	CH1_BUSY	**Channel 1 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 1.	0: DMA0 channel 1 Idle. 1: DMA0 transfer in progress on channel 1.
0	CH0_BUSY	**Channel 0 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 0.	0: DMA0 channel 0 Idle. 1: DMA0 transfer in progress on channel 0.

14. Advanced Encryption Standard (AES) Peripheral

The C8051F96x includes a hardware implementation of the Advanced Encryption Standard Block Cipher as specified in NIST publication FIPS 197 “Advanced Encryption Standard (AES), November 2001. The Rijndael encryption algorithm was chosen by NIST for the AES block cipher. The AES block cipher can be used to encrypt data for wireless communications. Data can be encrypted before transmission and decrypted upon reception. This provides security for private networks.

The AES block cipher is a Symmetric key encryption algorithm. Symmetric Key encryption relies on secret keys that are known by both the sender and receiver. The decryption key may be obtained using a simple transformation of the encryption key. AES is not a public key encryption algorithm.

The AES block Cipher uses a fixed 16 byte block size. So data less than 16 bytes must be padded with zeros to fill the entire block. Wireless data must be padded and transmitted in 16-byte blocks. The entire 16-byte block must be transmitted to successfully decrypt the information.

The AES engine supports key lengths of 128-bits, 192-bits, or 256-bits. A key size of 128-bits is sufficient to protect the confidentiality of classified secret information. The Advanced Encryption Standard was designed to be secure for at least 20 to 30 years. The 128-bit key provides fastest encryption. The 192-bit and 256-bit key lengths may be used to protect highly sensitive classified top secret information.

Since symmetric key encryption relies on secret keys, the security of the data can only be protected if the key remains secret. If the encryption key is stored in flash memory, then the entire flash should be locked to ensure the encryption key cannot be discovered. (See flash security.)

The basic AES block cipher is implemented in hardware. This hardware accelerator provides performance that may be 1000 times faster than a software implementation. The higher performance translates to a power savings for low-power wireless applications.

The AES block cipher, or block cipher modes based on the AES block cipher, is used in many wireless standards. These include several IEEE standards in the wireless PAN (802.15) and wireless LAN (802.11) working groups.

14.5.2. AES Block Cipher Decryption using SFRs

- First Configure AES Module for AES Block Cipher
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for AES Block Cipher by writing 0x00 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for decryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write ciphertext byte to AES0BIN.
  - Write decryption key byte to AES0KIN.
- Write remaining decryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 plaintext bytes from the AES0YOUT sfr.

If decrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block.

14.6.2. CBC Encryption Initialization Vector Location

The first block to be encrypted uses the initialization vector for the AES0XIN data. Subsequent blocks will use the encrypted ciphertext from the previous block. The DMA is capable of encrypting multiple blocks. If the initialization is located at an arbitrary location in xram, the DMA base address location will need to be changed to the start of the encrypted ciphertext after encrypting the first block. However, if the initialization vector explicitly located in xram immediately before the encrypted ciphertext, the pointer will be advanced to the start of the encrypted ciphertext naturally and multiple blocks can be encrypted autonomously.

14.6.3. CBC Encryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

- Prepare encryption Key, initialization vector, and data to be encrypted in xram.

(The initialization vector should be located immediately before the data to be encrypted to encrypt multiple blocks.)

- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first four DMA channels by clearing bits 0 to 3 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 to DMA0NMD to enable wrapping
  - Write the xram location of encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be encrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0XIN sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move xram to AES0XIN sfr by writing 0x07 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of initialization vector to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- * Configure the fourth DMA channel for the AES0YOUT sfr
  - Select the fourth channel by writing 0x03 to the DMA0SEL sfr
  - Configure the fourth DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write the xram address for encrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first four DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.

SFR Definition 19.1. PCLKACT: Peripheral Active Clock Enable

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

SFR Page = 0xF; SFR Address = 0xF5

Bit	Name	Function
7:4	Unused	Read = 0b; Write = don't care.
3	PCLKACT3	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to the SmaRTClock, Pulse Counter, and PMU0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Active Mode.
2	PCLKACT2	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to Timer 0, Timer 1, Timer 2, and CRC0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Active Mode.
1	PCLKACT1	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to ADC0 and PCA0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to ADC0 and PCA0 in Low Power Active Mode.
0	PCLKACT0	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to UART0, Timer 3, SPI0, and the SMBus revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Active Mode.

SFR Definition 25.6. PC0DCL: PC0 Debounce Configuration Low

Bit	7	6	5	4	3	2	1	0
Name	PC0DCL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	1	0	0

SFR Address = 0xF9; SFR Page = 0x2

Bit	Name	Function
7:0	PC0DCL[7:0]	**Pulse Counter Debounce Low** Number of cumulative good samples seen by the integrator before recognizing the input as low. Setting PC0DCL to 0x00 will disable integrators on both PC0 and PC1. The actual value used is PC0DCL plus one. Sampling a low decrements while sampling a high increments the count. Switch bounce produces a random looking signal. The worst case would be to bounce high at each sample point and not start decrementing the integrator until the switch bounce settled. Therefore, minimum pulse width should account for twice the debounce time. For example, using a sample rate of 1 ms and a PC0DCL value of 0x09 will look for 10 cumulative lows before recognizing the input as low (1 ms x 10 = 10 ms). The minimum pulse width should be 20 ms or greater for this example. If PC0DCL has a value of 0x03 and the sample rate is 500 µs, the integrator would need to see 4 cumulative lows before recognizing the low (500 µs x 4 = 2 ms). The minimum pulse width should be 4 ms for this example.

SFR Definition 26.4. LCD0MSCN: LCD0 Master Control

Bit	7	6	5	4	3	2	1	0
Name		BIASEN	DCBIASOE	CLKOE		LOWDRV	LCDRST	LCDEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAB

Bit	Name	Function
7	Reserved	Read = 0b. Must write 0b.
6	BIASEN	**LCD0 Bias Enable.** LCD0 bias may be disabled when using a static LCD (single backplane), contrast control mode 1 (Bypass Mode) is selected, and the VLCD/VIO Supply Comparator is disabled (LCD0CF.5 = 1). It is required for all other modes. 0: LCD0 Bias is disabled. 1: LCD0 Bias is enabled
5	DCBIASOE	**DCDC Converter Bias Output Enable. (Note 1)** 0: The bias for the DCDC converter is gated off. 1: LCD0 provides the bias for the DCDC converter.
4	CLKOE	**LCD Clock Output Enable.** 0: The clock signal to the LCD0 module is gated off. 1: The SmaRTClock provides the undivided clock to the LCD0 Module.
3	Reserved	Read = 0b. Must write 0b.
2	LOWDRV	**Charge Pump Reduced Drive Mode.** This bit should be set to 1 in Contrast Control Mode 3 and Mode 4 for minimum power consumption. This bit may be set to 0 in these modes to support higher load current requirements. 0: The charge pump operates at full power. 1: The charge pump operates at reduced power.
1	LCDRST	**LCD0 Reset.** Writing a 1 to this bit will clear all the LCD0Dn registers to 0x00. This bit must be cleared by software.
0	LCDEN	**LCD0 Enable.** 0: LCD0 is disabled. 1: LCD0 is enabled.
Note 1: To same bias generator is shared by the DCDC Converter and LCD0.		

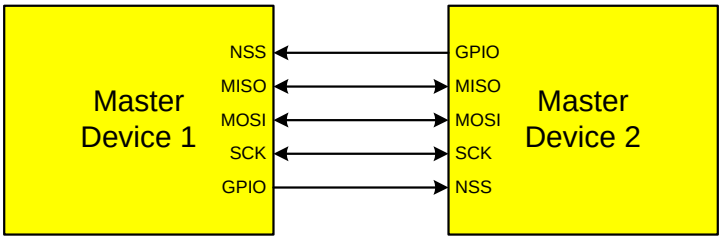

Figure 30.2. Multiple-Master Mode Connection Diagram

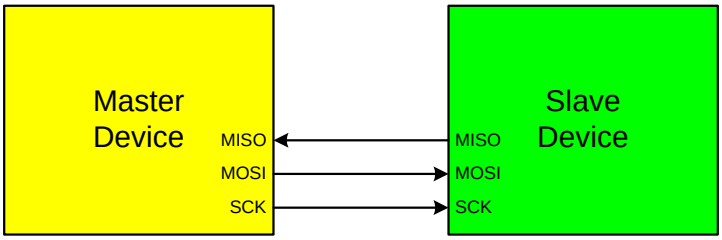

Figure 30.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

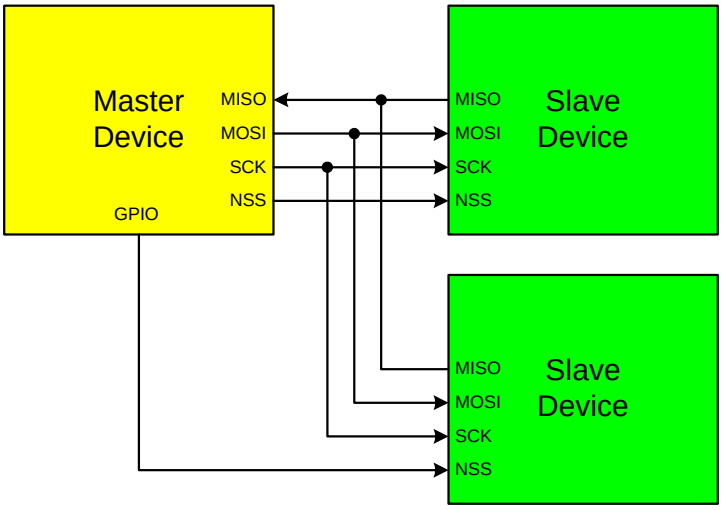

Figure 30.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

30.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data,

31.2. SPI1 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI1 is placed in master mode by setting the Master Enable flag (MSTEN, SPI1CN.6). Writing a byte of data to the SPI1 data register (SPI1DAT) 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 SPI1 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI1CN.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 is set. While the SPI1 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI1DAT.

When configured as a master, SPI1 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI1 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI1CN.6) and SPIEN (SPI1CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI1CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI1 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 31.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 31.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI1CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI1CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 31.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

SFR Definition 32.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8A

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 32.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8B

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 32.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x94

Bit	Name	Function
7:0	TMR3L[7:0]	**Timer 3 Low Byte.** In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

SFR Definition 32.17. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x95

Bit	Name	Function
7:0	TMR3H[7:0]	**Timer 3 High Byte.** In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

33.1. 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 33.1.

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. Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 33.1. 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[1]
1	1	0	SmaRTClock oscillator source divided by 8[2]
1	1	1	Reserved

Notes:
1. External oscillator source divided by 8 is synchronized with the system clock.
2. SmaRTClock oscillator source divided by 8 is synchronized with the system clock.

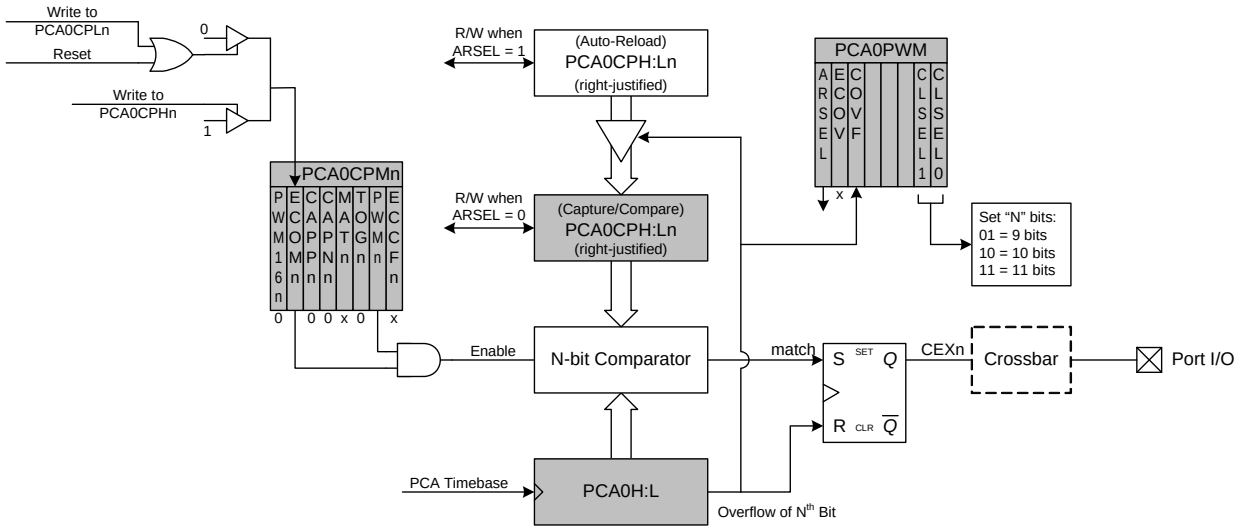

Figure 33.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

33.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 33.4.

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.

$$\text{Duty Cycle} = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 33.4. 16-Bit PWM Duty Cycle

Using Equation 33.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

Document Change List

Revision 0.1 to Revision 0.2

- Added new content to DC0 chapter.
- Reordered chapters.
- Corrections to SFR tables.
- Updated Electrical Specifications.

Revision 0.2 to Revision 0.3

- Added new content to DMA0, CRC1, ENC0, SPI1, and Pulse Counter chapters.
- Added TQFP-80 package variant.
- Added package drawings and landing diagram for TQFP-80 package.
- Added via placement recommendations for DQFN-76 package.
- Updated electrical specifications.
- Corrections to SFR tables.
- Fixed inconsistencies in SFR names.
- Fixed inconsistencies in acronyms and terminology.

Revision 0.3 to Revision 0.5

- Updated maximum IBAT current using precision oscillator in Table 4.4.
- Updated sleep currents in Table 4.4.
- Added Note 1 to Table 4.6.
- Deleted SFR Page Stack Example in Special Function Registers chapter.
- Change description of SFRPGEN bit in SFRPGCN SFR definition.
- Added paragraph to Flash chapter to explain lock byte behavior on 128 kB devices.
- Corrected SFRPAGE in SPI1 SFR definitions 32.1/2/3.

Revision 0.5 to Revision 1.0

- Changed revision in ordering information from A to B.
- Fixed inconsistencies in VIORF pin definitions.
- Added note about IFBANK usage.
- Updated Table 4.4 Digital Supply Current—Sleep Mode (LCD disabled, RTC disabled) 3.6 V, 25 °C maximum to 0.23 µA.
- Fixed inconsistencies in description of reset behavior.
- Added encryption/decryption times to SFR Definition 14.1.
- Fixed inconsistencies in SFR Definition 14.2.
- Fixed inconsistencies in Port P2 through P7 SFR Definitions.
- All TBD specifications have been determined.