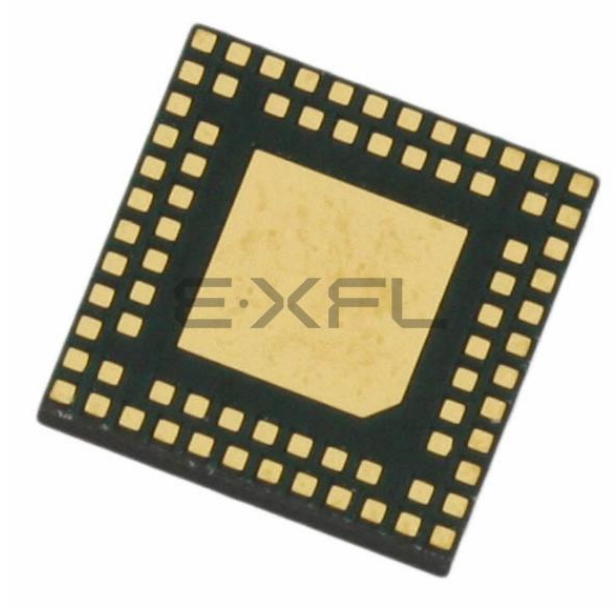


Welcome to E-XFL.COM

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-gm

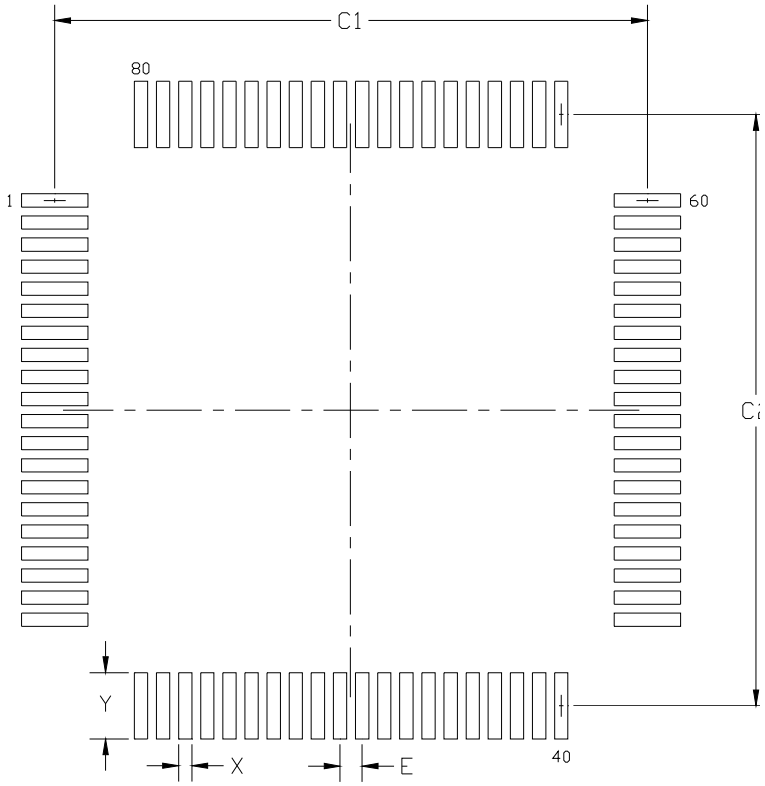


Figure 3.10. TQFP80 Landing Diagram

Table 3.8. TQFP80 Landing Diagram Dimensions

Dimension	Min	Max
C1	13.30	13.40
C2	13.30	13.40
E	0.50 BSC	
X	0.20	0.30
Y	1.40	1.50

Notes:
1. All feature sizes shown are in mm unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

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.

7.3. Comparator Response Time

Comparator response time may be configured in software via the CPTnMD registers described on “CPT0MD: Comparator 0 Mode Selection” on page 109 and “CPT1MD: Comparator 1 Mode Selection” on page 111. Four response time settings are available: Mode 0 (Fastest Response Time), Mode 1, Mode 2, and Mode 3 (Lowest Power). Selecting a longer response time reduces the Comparator active supply current. The Comparators also have low power shutdown state, which is entered any time the comparator is disabled. Comparator rising edge and falling edge response times are typically not equal. See Table 4.16 on page 74 for complete comparator timing and supply current specifications.

7.4. Comparator Hysterisis

The Comparators feature software-programmable hysterisis that can be used to stabilize the comparator output while a transition is occurring on the input. Using the CPTnCN registers, the user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage (i.e., the comparator negative input).

Figure 7.3 shows that when positive hysterisis is enabled, the comparator output does not transition from logic 0 to logic 1 until the comparator positive input voltage has exceeded the threshold voltage by an amount equal to the programmed hysterisis. It also shows that when negative hysterisis is enabled, the comparator output does not transition from logic 1 to logic 0 until the comparator positive input voltage has fallen below the threshold voltage by an amount equal to the programmed hysterisis.

The amount of positive hysterisis is determined by the settings of the CPnHYP bits in the CPTnCN register and the amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits in the same register. Settings of 20 mV, 10 mV, 5 mV, or 0 mV can be programmed for both positive and negative hysterisis. See Section “Table 4.16. Comparator Electrical Characteristics” on page 74 for complete comparator hysterisis specifications.

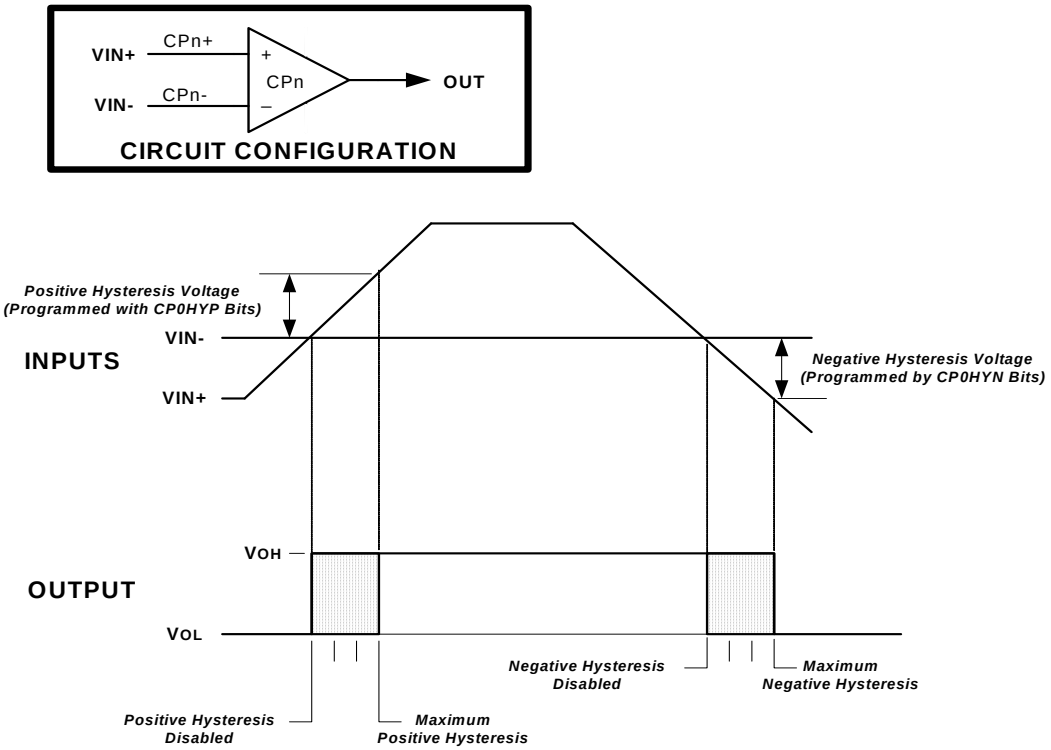


Figure 7.3. Comparator Hysteresis Plot

9. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F96x device family is shown in Figure 9.1

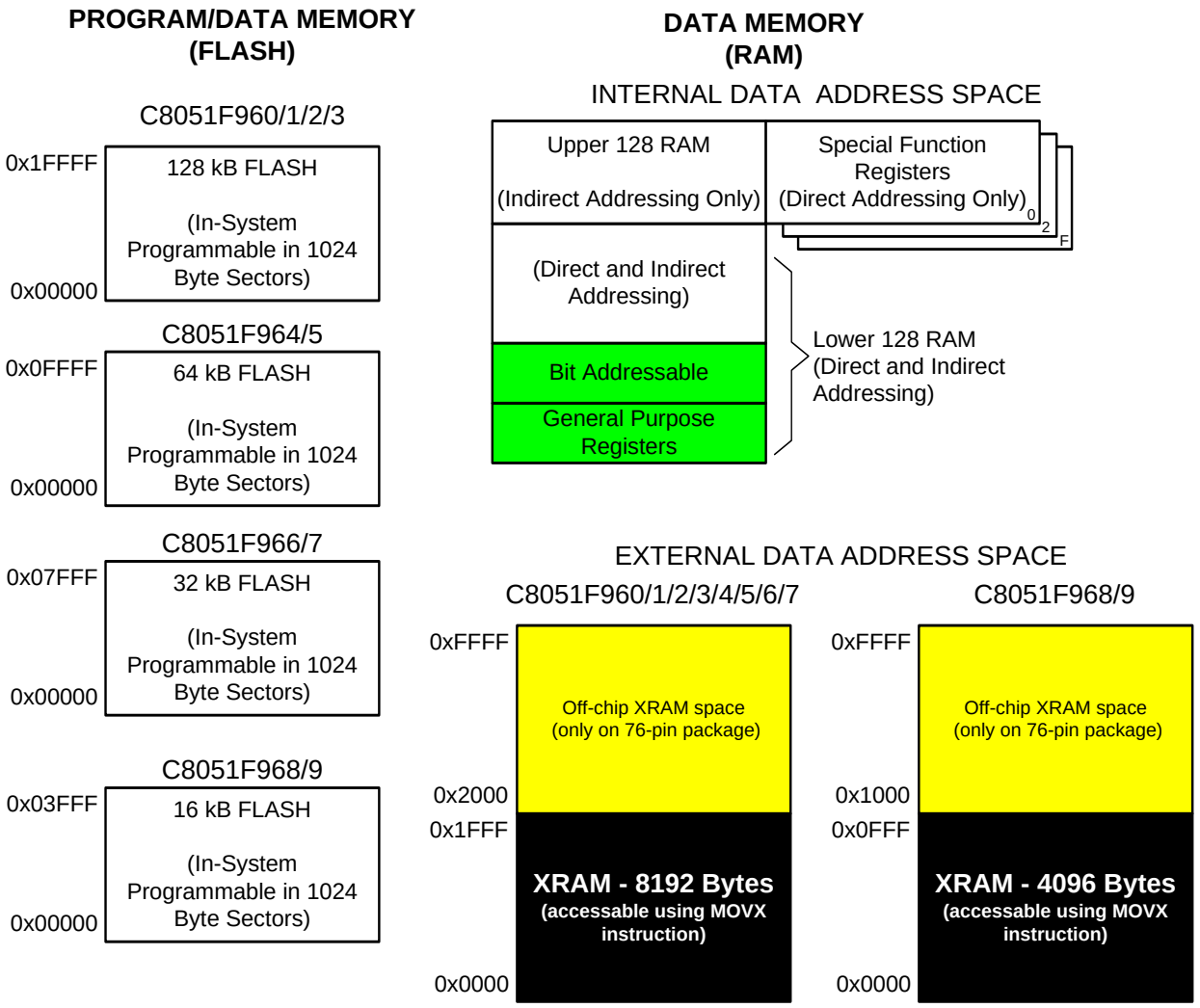


Figure 9.1. C8051F96x Memory Map

9.1. Program Memory

The C8051F960/1/2/3 device flashs have a 128 kB program memory space, C8051F964/5 devices have 64 kB program memory space, C8051F966/7 devices have 32 kB program memory space, and C8051F968/9 devices have a 16 kB program memory space. The devices with 128 kB flash implement this program memory space as in-system re-programmable flash memory in four 32 kB code banks. A common code bank (Bank 0) of 32 kB is always accessible from addresses 0x0000 to 0x7FFF. The upper code banks (Bank 1, Bank 2, and Bank 3) are each mapped to addresses 0x8000 to 0xFFFF, depending on the

12. Cyclic Redundancy Check Unit (CRC0)

C8051F96x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit or 32-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register. CRC0 posts the 16-bit or 32-bit result to an internal register. The internal result register may be accessed indirectly using the CRC0PNT bits and CRC0DAT register, as shown in Figure 12.1. CRC0 also has a bit reverse register for quick data manipulation.

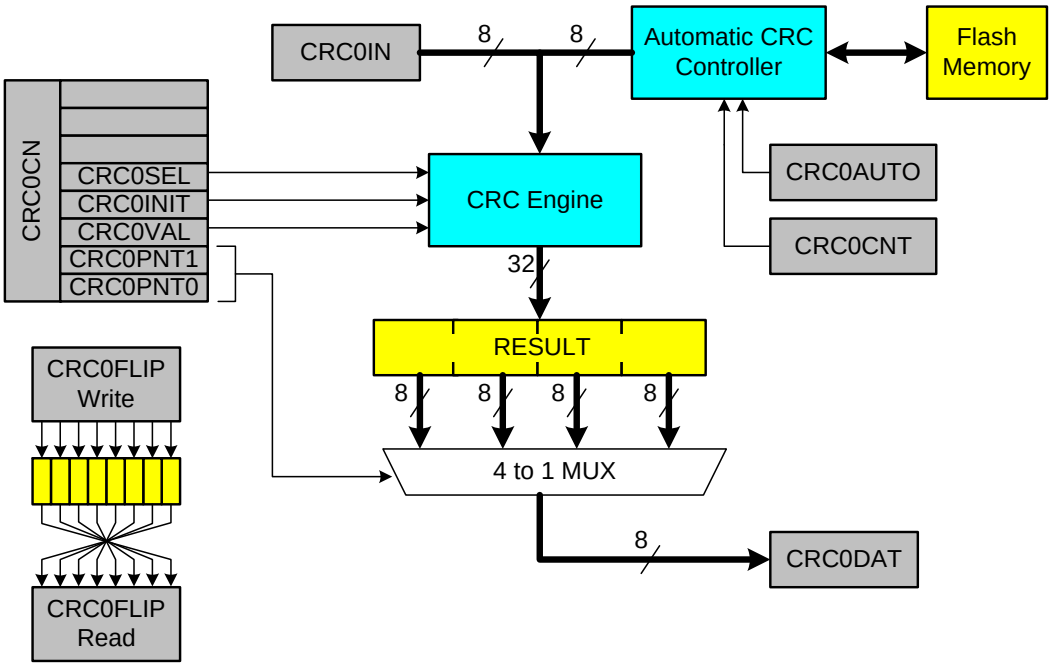


Figure 12.1. CRC0 Block Diagram

12.1. 16-bit CRC Algorithm

The C8051F96x CRC unit calculates the 16-bit CRC MSB-first, using a poly of 0x1021. The following describes the 16-bit CRC algorithm performed by the hardware:

1. XOR the most-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x0000 or 0xFFFF).
   a. If the MSB of the CRC result is set, left-shift the CRC result, and then XOR the CRC result with the polynomial (0x1021).
   b. If the MSB of the CRC result is not set, left-shift the CRC result.
2. Repeat at Step 2a for the number of input bits (8).

14.4.2. AES Block Cipher Encryption 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 encryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write plaintext byte to AES0BIN.
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit encryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 encrypted bytes from the AES0YOUT sfr.

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

14.6.4. CBC Decryption

The AES0 module data flow for CBC decryption is shown in Figure 14.6. The ciphertext is written to the AES0BIN sfr. For the first block, the initialization vector is written to the AES0XIN sfr. For subsequent blocks, the previous block ciphertext is written to the AES0XIN sfr. The AES0DCF sfr is configured to XOR AES0XIN with AES0BIN for the AES core data input. The XOR on the output is not used. The AES core is configured for an encryption operation. The encryption key is written to AES0KIN. The key size is set to the desired key size.

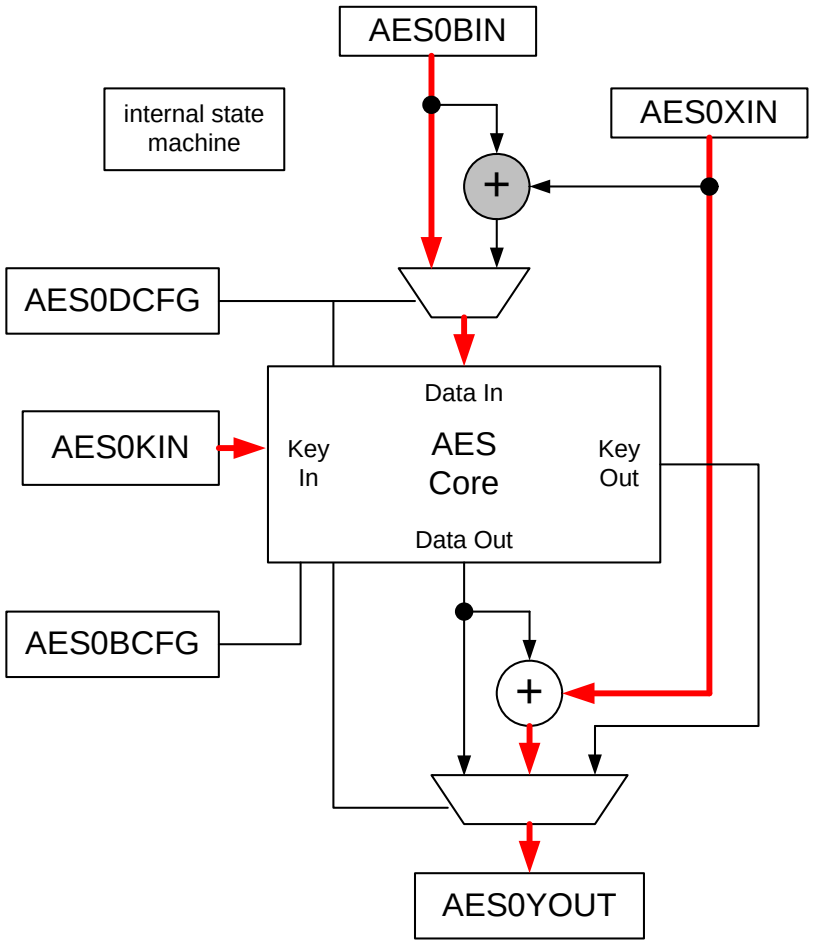


Figure 14.6. CBC Decryption Data Flow

Table 16.1. SFR Map (0xC0–0xFF)

Addr.	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	0x0	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
	0x2	SPI1CN	PC0DCL	PC0DCH	PC0INT0	PC0INT1	DC0RDY		
	0xF		P4MDOUT	P5MDOUT	P6MDOUT	P7MDOUT	CLKMODE	PCLKEN	
0xF0	0x0		P0MDIN	P1MDIN	P2MDIN	SMB0ADR	SMB0ADM	EIP1	EIP2
	0x2		PC0CMP1L	PC0CMP1M	PC0CMP1H	PC0HIST	AES0YOUT		
	0xF		P3MDIN	P4MDIN	P5MDIN	P6MDIN	PCLKACT		
0xE8	0x0	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
	0x2		AES0BCFG	AES0DCFG	AES0BIN	AES0XIN	AES0KIN		
	0xF		DEVICEID	REVID					
0xE0	0x0	ACC	XBR0	XBR1	XBR2	IT01CF		EIE1	EIE2
	0x2		PC0CMP0L	PC0CMP0M	PC0CMP0H	PC0TH			
	0xF		XBR0	XBR1	XBR2	IT01CF			
0xD8	0x0	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0PWM
	0x2		PC0MD	PC0CTR0L	PC0TRML	PC0CTR0H	PC0CTR1L	PC0TRMH	PC0CTR1H
	0xF		P4	P5	P6	P7			
0xD0	0x0	PSW	REF0CN	PCA0CPL5	PCA0CPH5	P0SKIP	P1SKIP	P2SKIP	P0MAT
	0x2		DMA0SEL	DMA0EN	DMA0INT	DMA0MINT	DMA0BUSY	DMA0NMD	PC0PCF
	0xF								
0xC8	0x0	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	PCA0CPM5	P1MAT
	0x2		DMA0NCF	DMA0NBAL	DMA0NBAH	DMA0NAOL	DMA0NAOH	DMA0NSZL	DMA0NSZH
	0xF								
0xC0	0x0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P0MASK
	0x2		PC0STAT	ENC0L	ENC0M	ENC0H	ENC0CN	VREGINSDL	VREGINSDH
	0xF								

SFR Definition 17.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Page = 0x0; SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P1.6 111: Select P1.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P1.6 111: Select P1.7

SFR Definition 19.7. PCON: Power Management Control Register

Bit	7	6	5	4	3	2	1	0
Name	GF[4:0]					PWRSEL	STOP	IDLE
Type	R/W					R/W	W	W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x87

Bit	Name	Description	Write	Read
7:3	GF[5:0]	**General Purpose Flags**	Sets the logic value.	Returns the logic value.
2	PWRSEL	**Power Select**	0: VBAT is selected as the input to VREG0. 1: VDC is selected as the input to VREG0.	
1	STOP	**Stop Mode Select**	Writing 1 places the device in Stop Mode.	N/A
0	IDLE	**Idle Mode Select**	Writing 1 places the device in Idle Mode.	N/A

19.9. Power Management Specifications

See Table 4.7 on page 69 for detailed Power Management Specifications.

24.1. SmaRTClock Interface

The SmaRTClock Interface consists of three registers: RTC0KEY, RTC0ADR, and RTC0DAT. These interface registers are located on the CIP-51's SFR map and provide access to the SmaRTClock internal registers listed in Table 24.1. The SmaRTClock internal registers can only be accessed indirectly through the SmaRTClock Interface.

Table 24.1. SmaRTClock Internal Registers

SmaRTClock Address	SmaRTClock Register	Register Name	Description
0x00–0x03	CAPTUREn	SmaRTClock Capture Registers	Four Registers used for setting the 32-bit SmaRTClock timer or reading its current value.
0x04	RTC0CN	SmaRTClock Control Register	Controls the operation of the SmaRTClock State Machine.
0x05	RTC0XCN	SmaRTClock Oscillator Control Register	Controls the operation of the SmaRTClock Oscillator.
0x06	RTC0XCF	SmaRTClock Oscillator Configuration Register	Controls the value of the progammable oscillator load capacitance and enables/disables AutoStep.
0x07	RTC0CF	SmaRTClock Configuration Register	Contains an alarm enable and flag for each SmaRTClock alarm.
0x08–0x0B	ALARM0Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.
0x0C–0x0F	ALARM1Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.
0x10–0x13	ALARM2Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.

SFR Definition 25.4. PC0STAT: PC0 Status

Bit	7	6	5	4	3	2	1	0
Name	FLUTTER	DIRECTION	STATE[1:0]		PC1PREV	PC0PREV	PC1	PC0
Type	RO	RO	RO		RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0x2

Bit	Name	Function
7	FLUTTER	**Flutter** During quadrature mode, a disparity may occur between the number of negative edges of PC1 and PC0 or the number of positive edges of PC1 and PC0. This could indicate flutter on one reed switch or one reed switch may be faulty. 0: No flutter detected. 1: Flutter detected.
6	DIRECTION	**Direction** Only applicable for quadrature mode. (First letter is PC1; second letter is PC0) 0: Counter clock-wise - (LL-LH-HH-HL) 1: Clock-wise - (LL-HL-HH-LH)
5:4	STATE[1:0]	**PC0 State** Current State of Internal State Machine.
3	PC1PREV	**PC1 Previous** Previous Output of PC1 Integrator.
2	PC0PREV	**PC0 Previous** Previous Output of PC0 Integrator.
1	PC1	**PC1** Current Output of PC1 Integrator.
0	PC0	**PC0** Current Output of PC0 Integrator.

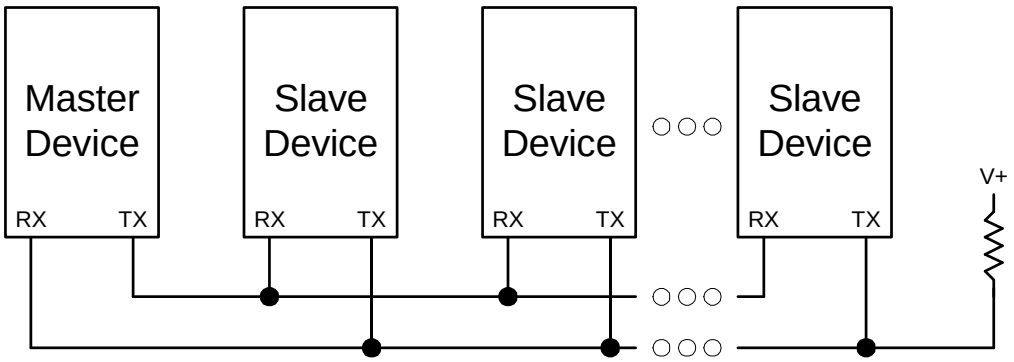


Figure 29.6. UART Multi-Processor Mode Interconnect Diagram

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

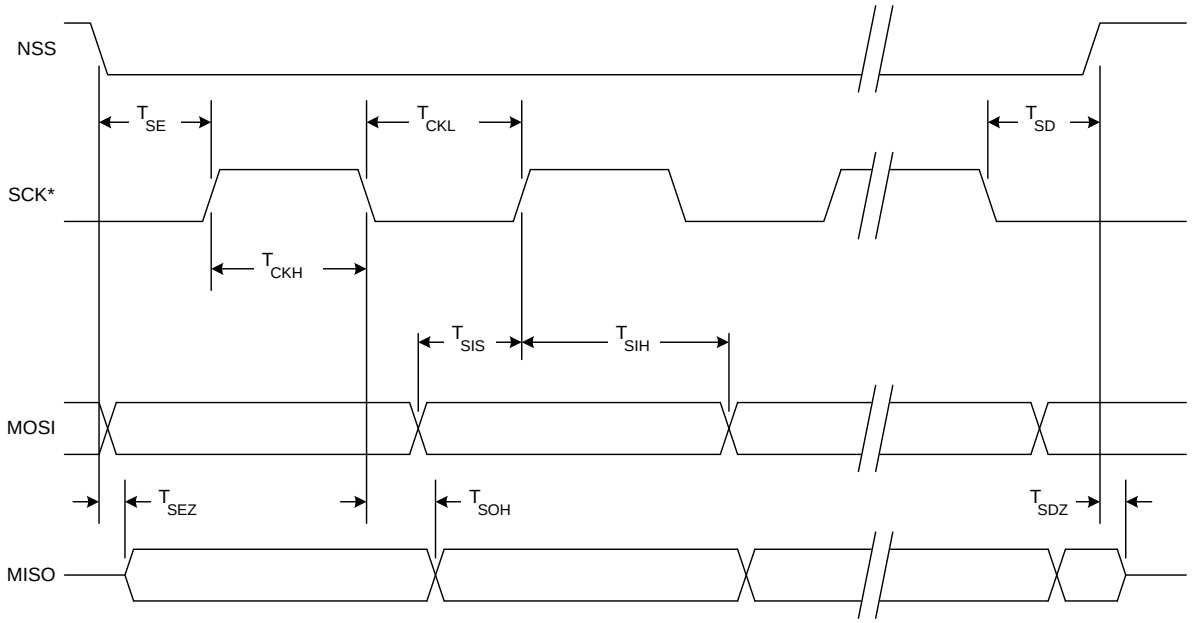
A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte 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 30.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

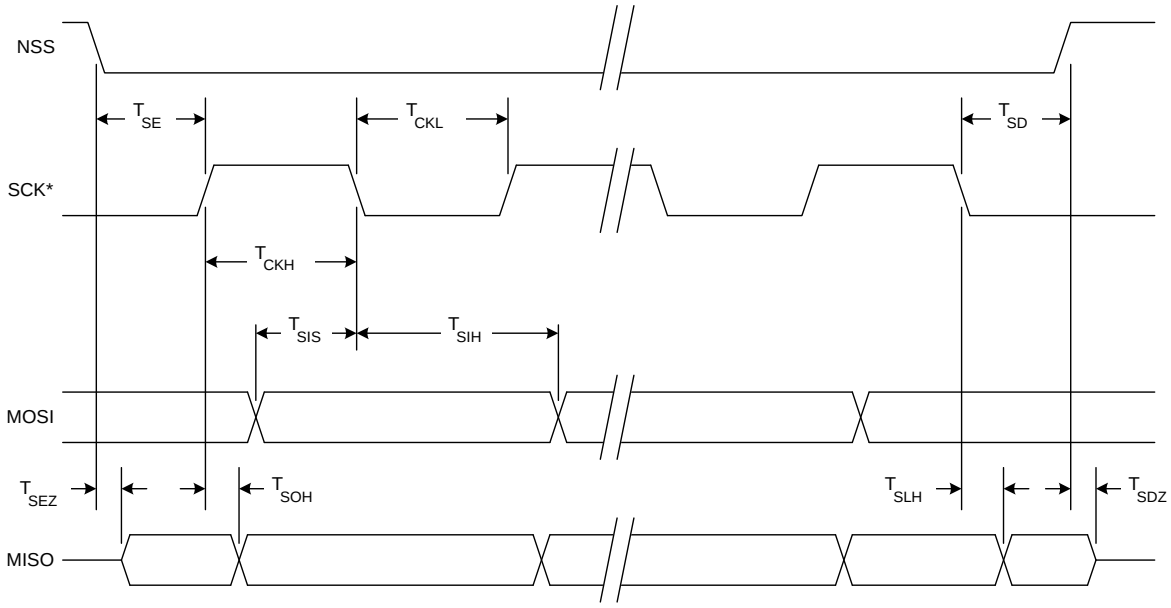
SFR Page = 0x0; SFR Address = 0xF8; Bit-Addressable

Bit	Name	Function
7	SPIF	**SPI0 Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	**Write Collision Flag.** This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	**Mode Fault Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	**Receive Overrun Flag (valid in slave mode only).** This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	**Slave Select Mode.** Selects between the following NSS operation modes: (See Section 30.2 and Section 30.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	**Transmit Buffer Empty.** This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	**SPI0 Enable.** 0: SPI disabled. 1: SPI enabled.



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.11. SPI Slave Timing (CKPHA = 1)

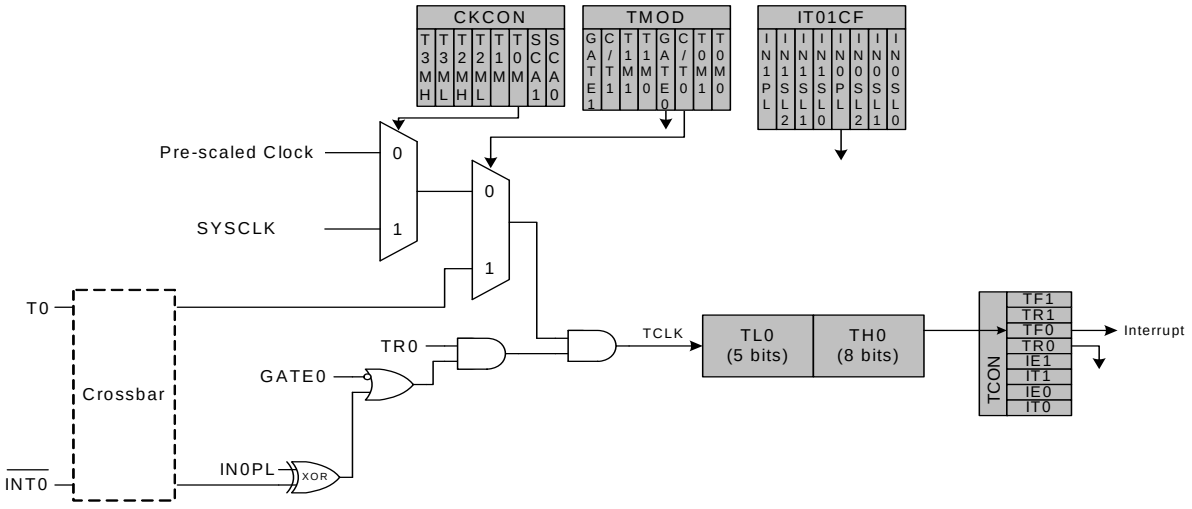


Figure 32.1. T0 Mode 0 Block Diagram

32.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

32.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “17.6. External Interrupts INT0 and INT1” on page 242 for details on the external input signals INT0 and INT1).

When Capture Mode is enabled, a capture event will be generated either every Comparator 0 rising edge or every 8 SmaRTClock clock cycles, depending on the T2XCLK1 setting. When the capture event occurs, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set (triggering an interrupt if Timer 2 interrupts are enabled). By recording the difference between two successive timer capture values, the Comparator 0 or SmaRTClock period can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T2ML = 1b, T2XCLK1 = 0b, and TF2CEN = 1b, Timer 2 will clock every SYSCLK and capture every SmaRTClock clock divided by 8. If the SYSCLK is 24.5 MHz and the difference between two successive captures is 5984, then the SmaRTClock clock is as follows:

24.5 MHz/(5984/8) = 0.032754 MHz or 32.754 kHz.

This mode allows software to determine the exact SmaRTClock frequency in self-oscillate mode and the time between consecutive Comparator 0 rising edges, which is useful for detecting changes in the capacitance of a Touch Sense Switch.

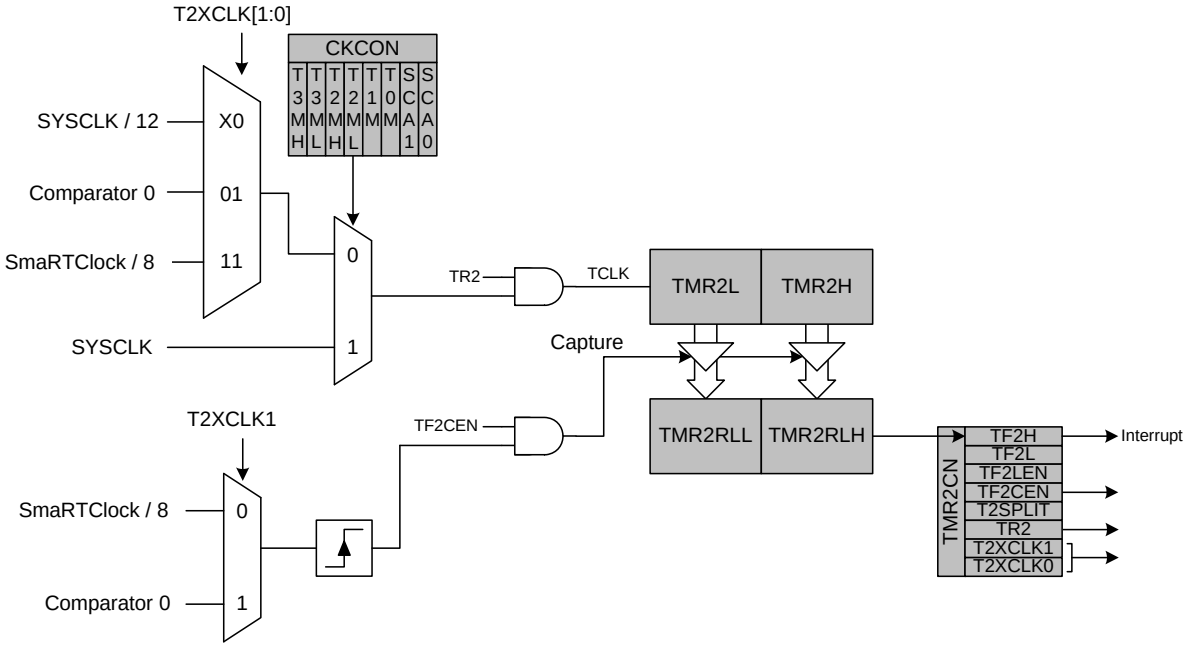


Figure 32.6. Timer 2 Capture Mode Block Diagram

Table 33.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:
1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.