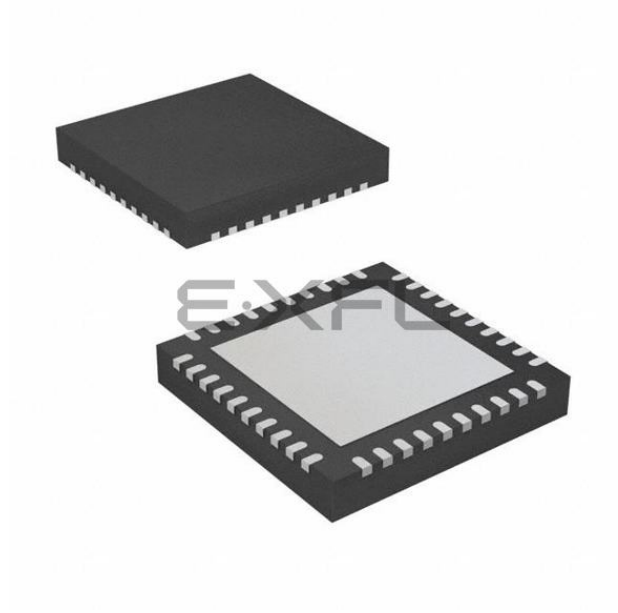

Welcome to E-XFL.COM

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	34
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	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f961-b-gmr

List of Registers

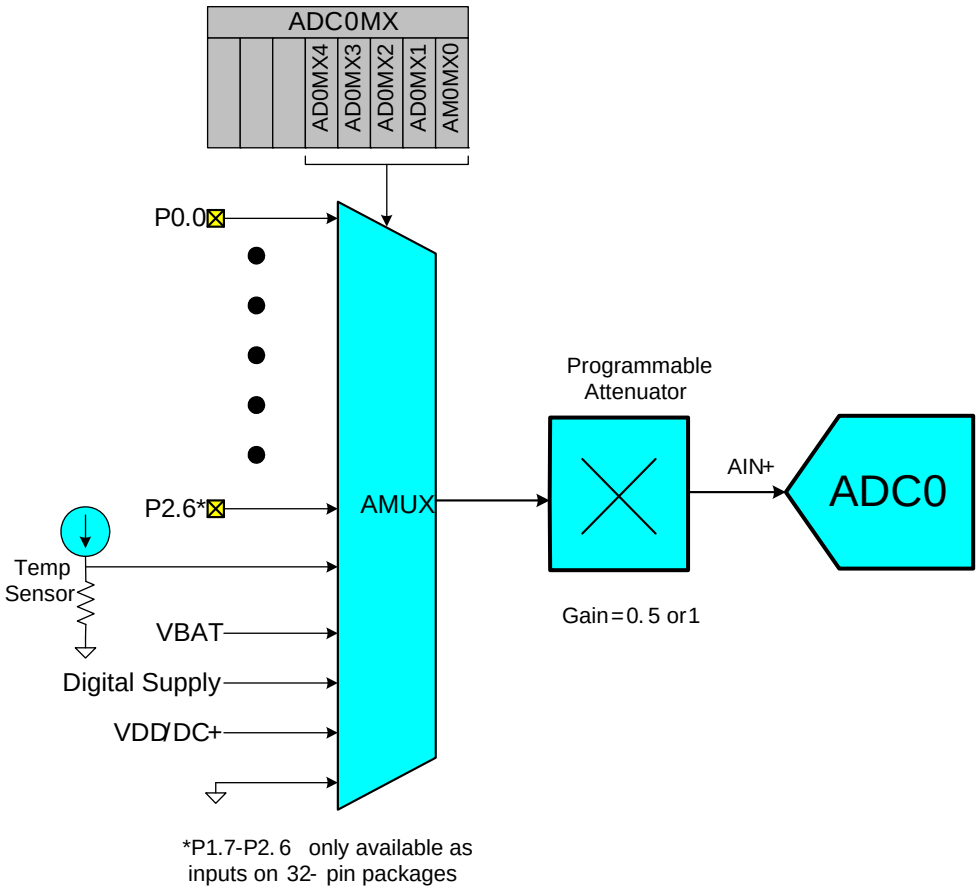

Figure 1.14. ADC0 Multiplexer Block Diagram

1.6. Programmable Current Reference (IREF0)

C8051F96x devices include an on-chip programmable current reference (source or sink) with two output current settings: low power mode and high current mode. The maximum current output in low power mode is 63 µA (1 µA steps) and the maximum current output in high current mode is 504 µA (8 µA steps).

1.7. Comparators

C8051F96x devices include two on-chip programmable voltage comparators: Comparator 0 (CPT0) which is shown in Figure 1.15; Comparator 1 (CPT1) which is shown in Figure 1.16. The two comparators operate identically but may differ in their ability to be used as reset or wake-up sources. See Section “22. Reset Sources” on page 278 and the Section “19. Power Management” on page 257 for details on reset sources and low power mode wake-up sources, respectively.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output when the device is in some low power modes.

The comparator inputs may be connected to Port I/O pins or to other internal signals. Port pins may also be used to directly sense capacitive touch switches. See Application Note AN338 for details on Capacitive Touch Switch sensing.

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P0.6 CNVSTR	A38	76	38	D I/O or A In D In	Port 0.6. See Port I/O Section for a complete description. External Convert Start Input for ADC0. See ADC0 section for a complete description.
P0.7 IREF0	A37	74	37	D I/O or A In A Out	Port 0.7. See Port I/O Section for a complete description. IREF0 Output. See IREF Section for complete description.
P1.0 PC0	A36	72	36	D I/O or A In D I/O	Port 1.0. See Port I/O Section for a complete description. Pulse Counter 0.
P1.1 PC1	A35	70	35	D I/O or A In D I/O	Port 1.1. See Port I/O Section for a complete description. Pulse Counter 1.
P1.2 XTAL3	A34	67	34	D I/O or A In A In	Port 1.2. See Port I/O Section for a complete description. SmaRTClock Oscillator Crystal Input.
P1.3 XTAL4	A33	65	33	D I/O or A In A Out	Port 1.3. See Port I/O Section for a complete description. SmaRTClock Oscillator Crystal Output.
P1.4	A31	60	31	D I/O or A In	Port 1.4. See Port I/O Section for a complete description.
P1.5	A30	57	30	D I/O or A In	Port 1.5. See Port I/O Section for a complete description. VIORF supply.
P1.6	A29	56	29	D I/O or A In	Port 1.6. See Port I/O Section for a complete description. VIORF supply. May also be used as INT0 or INT1.
P1.7	A28	54	28	D I/O or A In	Port 1.7. See Port I/O Section for a complete description. VIORF supply. May also be used as INT0 or INT1.
P2.0	A27	53	27	D I/O or A In	Port 2.0. See Port I/O Section for a complete description. VIORF supply. May also be used as SCK for SPI1.

Table 3.7. TQFP-80 Package Dimensions

Dimension	Min	Nominal	Max
Θ	0°	3.5°	7°
aaa		0.20	
bbb		0.20	
ccc		0.08	
ddd		0.08	
eee		0.05	

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This package outline conforms to JEDEC MS-026, variant ADD.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

8. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 34), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 8.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

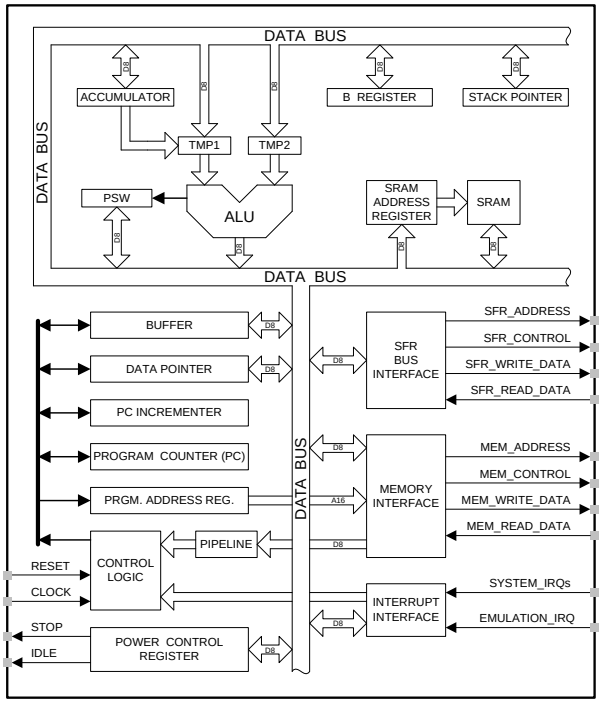

Figure 8.1. CIP-51 Block Diagram

10.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common). The Input Mode of the associated port pins should be set to digital (reset value).
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic 1).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition .

10.3. Port Configuration

The External Memory Interface appears on Ports 3, 4, 5, and 6 when it is used for off-chip memory access. The external memory interface and the LCD cannot be used simultaneously. When using EMIF, all pins on Port 3-6 may only be used for EMIF purposes or as general purpose I/O. The EMIF pinout is shown in Table 10.1 on page 131.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar settings for those pins. See Section “27. Port Input/Output” on page 351 for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to “park” the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

The C8051F960/2/4/6/8 devices support both the multiplexed and non-multiplexed modes. Accessing off-chip memory is not supported by the C8051F961/3/5/7/9 devices.

SFR Definition 12.2. CRC0IN: CRC0 Data Input

Bit	7	6	5	4	3	2	1	0
Name	CRC0IN[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x93

Bit	Name	Function
7:0	CRC0IN[7:0]	**CRC0 Data Input.** Each write to CRC0IN results in the written data being computed into the existing CRC result according to the CRC algorithm described in Section 12.1

SFR Definition 12.3. CRC0DAT: CRC0 Data Output

Bit	7	6	5	4	3	2	1	0
Name	CRC0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x91

Bit	Name	Function
7:0	CRC0DAT[7:0]	**CRC0 Data Output.** Each read or write performed on CRC0DAT targets the CRC result bits pointed to by the CRC0 Result Pointer (CRC0PNT bits in CRC0CN).

14.2.1. Key Inversion 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 the code examples. The steps are listed here for completeness.

Steps to generate the Decryption Key from Encryption Key

- Prepare encryption key and dummy data in xram.
- Reset AES module by clearing bit 3 of AES0BCFG.
- Disable the first three DMA channels by clearing bits 0 to 2 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.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL.
- 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 dummy data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write 0x10 (16) to the DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL
- Configure the third DMA channel for the AES0YOUT sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third 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 the decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH.
  - Clear DMA0NAOH and DMA0NAOL.
- Clear first three DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first three DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for inverse key generation by writing 0x04 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG.
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG.
- Initiate the encryption operation by setting bit 3 of AES0BCFG.
- Wait on the DMA interrupt from DMA channel 2.
- Disable the AES Module by clearing bit 2 of AES0BCFG.
- Disable the DMA by writing 0x00 to DMA0EN.

14.6.4.2. CBC Decryption using SFRs

- First Configure AES Module for CBC Block Cipher Mode Decryption
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for XOR on output data by writing 0x02 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 plaintext byte to AES0BIN.
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Repeat alternating write read sequence 16 times
  - Write initialization vector to AES0XIN
  - Read decrypted data from AES0YOUT

If decrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block. When using Cipher Block Chaining the initialization vector is written to the AES0XIN sfr for the first block only, as described. Additional blocks will chain the ciphertext data from the previous block.

SFR Definition 16.1. SFRPGCN: SFR Page Control

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

;SFR Page = 0xF; SFR Address = 0x8E

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don't Care
0	SFRPGEN	**SFR Automatic Page Control Enable.** Upon interrupt, the C8051 Core will vector to the specified interrupt service routine. This bit controls the automatic preservation and restoration of the SFRPAGE by hardware. 0: SFR Automatic Paging disabled. The C8051 core will neither preserve the SRFPAGE upon entering an interrupt service routine, nor restore the SFRPAGE upon exiting the interrupt service routine. The interrupt service routine should preserve and restore the active SFRPAGE in firmware. 1: SFR Automatic Paging enabled. The C8051 core will preserve the SFRPAGE upon entering an interrupt service routine and restore the SFRPAGE upon exiting the Interrupt service routine. The firmware does not need to preserve and restore the SFRPAGE in the interrupt service routing. However, firmware must set the SFRPAGE within the interrupt service routine before accessing SFRs.

Table 17.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
SmaRTClock Oscillator Fail	0x008B	17	OSCFAIL (RTC0CN.5)[2]	N	N	ERTC0F (EIE2.2)	PFRTC0F (EIP2.2)
SPI1	0x0093	18	SPIF (SPI1CN.7) WCOL (SPI1CN.6) MODF (SPI1CN.5) RXOVRN (SPI1CN.4)	N	N	ESPI1 (EIE2.3)	PSPI1 (EIP2.3)
Pulse Counter	0x009B	19	C0ZF (PC0CN.4) C1ZF (PC0CN.6)	N	N	EPC0 (EIE2.4)	PPC0 (EIP2.4)
DMA0	0x00A3	20	DMAINT0...7 DMAMINT0...7	N	N	EDMA0 (EIE2.5)	PDMA0 (EIP2.5)
Encoder0	0x00AB	21	ENCERR(ENCCN.6)	N	N	EENC0 (EIE2.6)	PENC0 (EIP2.6)
AES	0x00B3	22	AESDONE (AESBCF.5)	N	N	EAES0 (EIE2.7)	PAES0 (EIP2.7)

Notes:
1. Indicates a read-only interrupt pending flag. The interrupt enable may be used to prevent software from vectoring to the associated interrupt service routine.
2. Indicates a register located in an indirect memory space.

17.5. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in the following register descriptions. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 20.3. DC0MD: DC-DC Converter Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved	ILIMIT			FORBYP	AUTOBYP	Reserved	DC0EN
Type	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	0	0

SFR Page = 0x2; SFR Address = 0xB3

Bit	Name	Function
7	Reserved	Read = 0b; Must write 0b.
6:4	ILIMIT	**Peak Current Limit Threshold.** 000: Reserved 001: Peak Inductor current is limited to 200 mA 010: Peak Inductor current is limited to 300 mA 011: Peak Inductor current is limited to 400 mA 100: Peak Inductor current is limited to 500 mA 101: Peak Inductor current is limited to 600 mA 110: Reserved 111: Reserved
3	FORBYP	**Enable Forced Bypass Mode.** 0: Forced bypass mode is disabled. 1: Forced bypass mode is enabled.
2	AUTOBYP	**Enable Automatic Bypass Mode.** 0: Automatic Bypass mode is disabled. 1: Automatic bypass mode is enabled.
1	Reserved	Read = 1b; Must write 1b.
0	DC0EN	**DC-DC Converter Enable.** 0: DC-DC converter is disabled. 1: DC-DC converter is enabled.

Internal Register Definition 24.8. CAPTUREn: SmaRTClock Timer Capture

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE[31:0]							
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

SmaRTClock Addresses: CAPTURE0 = 0x00; CAPTURE1 = 0x01; CAPTURE2 =0x02; CAPTURE3: 0x03.

Bit	Name	Function
7:0	CAPTURE[31:0]	**SmaRTClock Timer Capture.** These 4 registers (CAPTURE3–CAPTURE0) are used to read or set the 32-bit SmaRTClock timer. Data is transferred to or from the SmaRTClock timer when the RTC0SET or RTC0CAP bits are set.
Note: The least significant bit of the timer capture value is CAPTURE0.0.		

Internal Register Definition 24.9. ALARM0Bn: SmaRTClock Alarm 0 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM0[31:0]							
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

SmaRTClock Address: ALARM0B0 = 0x08; ALARM0B1 = 0x09; ALARM0B2 = 0x0A; ALARM0B3 = 0x0B

Bit	Name	Function
7:0	ALARM0[31:0]	**SmaRTClock Alarm 0 Programmed Value.** These 4 registers (ALARM0B3–ALARM0B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM0EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is ALARM0B0.0.		

Internal Register Definition 24.10. ALARM1Bn: SmaRTClock Alarm 1 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM1[31:0]							
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

SmaRTClock Address: ALARM1B0 = 0x0C; ALARM1B1 = 0x0D; ALARM1B2 = 0x0E; ALARM1B3 = 0x0F

Bit	Name	Function
7:0	ALARM1[31:0]	**SmaRTClock Alarm 1 Programmed Value.** These 4 registers (ALARM1B3–ALARM1B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM1EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is iALARM1B0.0.		

Internal Register Definition 24.11. ALARM2Bn: SmaRTClock Alarm 2 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM2[31:0]							
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

SmaRTClock Address: ALARM2B0 = 0x10; ALARM2B1 = 0x11; ALARM2B2 = 0x12; ALARM2B3 = 0x13

Bit	Name	Function
7:0	ALARM2[31:0]	**SmaRTClock Alarm 2 Programmed Value.** These 4 registers (ALARM2B3–ALARM2B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM2EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is ALARM2B0.0.		

25. Low-Power Pulse Counter

The C8051F96x family of microcontrollers contains a low-power Pulse Counter module with advanced features, such as ultra low power input comparators, a wide range of pull up values with a self calibration engine, asymmetrical integrators for low pass filtering and switch debounce, single, dual, and quadrature modes of operation, two 24-bit counters, threshold comparators, and a variety of interrupt and sleep wake up capabilities. This combination of features provides water, gas, and heat metering system designers with an optimal tool for saving power while collecting meter usage data.

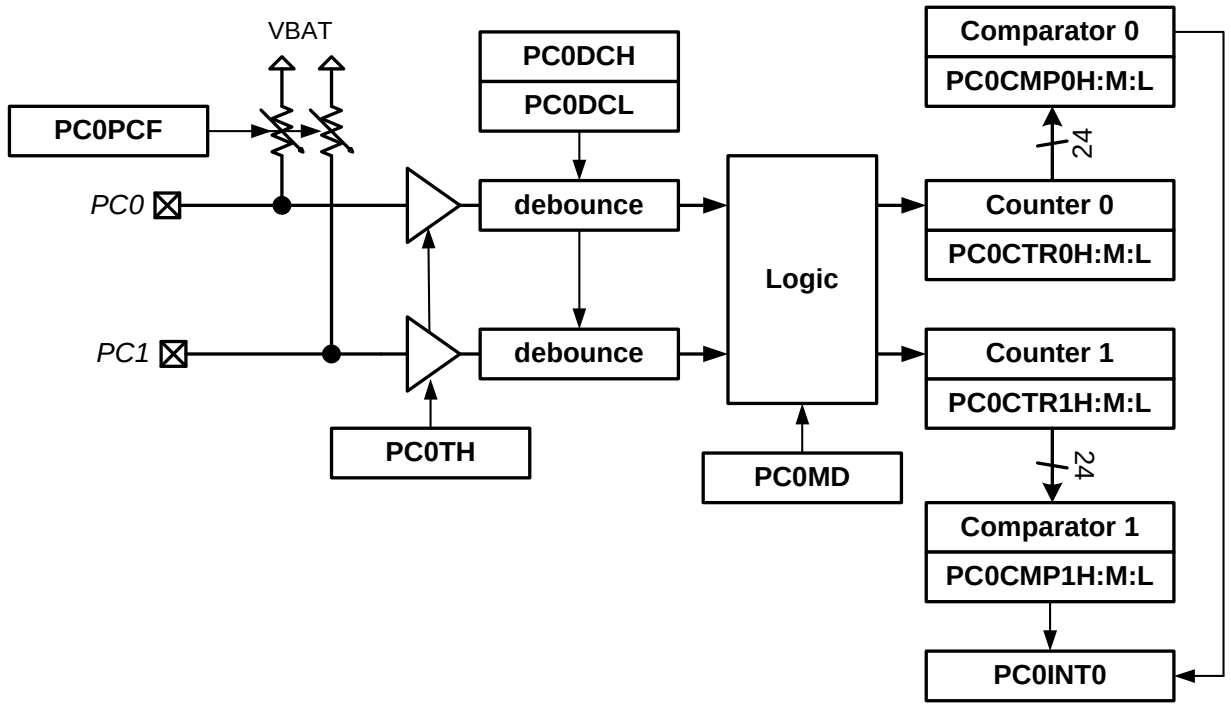

Figure 25.1. Pulse Counter Block Diagram

The low-power Pulse Counter is a low-power sleep-mode peripheral designed primarily to work meters using reed switches, including water and gas meters. The Pulse Counter is very flexible and can count pulses from many different types of sources.

The Pulse Counter operates in sleep mode to enable ultra-low power metering systems. The MCU does not have to wake up on every edge or transition and can remain in sleep mode while the Pulse Counter counts pulses for an extended period of time. The Pulse Counter includes two 24-bit counters. These counters can count up to 16,777,215 (2^{24}-1) transitions in sleep mode before overflowing. The Pulse Counter can wake up the MCU when one of the counters overflows. The Pulse Counter also has two 24-bit comparators. The comparators have the ability to wake up the MCU when the one of the counters reaches a predetermined threshold.

The Pulse Counter uses the RTC clock for sampling, de-bouncing, and managing the low-power pull-up resistors. The RTC must be enabled when counting pulses. The RTC alarms can wake up the MCU periodically to read the pulse counters, instead of using the Pulse Counter comparators. For example, the RTC can wake up the MCU every five minutes. The MCU can then read the Pulse Counter and transmit the information using the UART or a wireless transceiver.

25.3. Programmable Pull-Up Resistors

The Pulse Counter features low-power pull-up resistors with a programmable resistance and duty-cycle. The average pull-up current will depend on the selected resistor, sample rate, and pull-up duty-cycle multiplier. Example code is available that will calculate the values for the Pull-Up configuration SFR (PC0PCF).

Table 25.1through Table 25.3 are used with Equation 25.1 to calculate the average pull-up resistor current. Table 25.4through Table 25.7 give the average current for all combinations.

$$I_{pull\text{-}up} = I_R \times D_{SR} \times D_{PU}$$

Equation 25.1. Average Pull-Up Current

Where:

I_R = Pull-up Resistor current selected by PC0PCF[4:2].

D_{SR} = Sample Rate Duty Cycle Multiplier selected by PC0MD[5:4].

D_{PU} = Pull-Up Duty Cycle Multiplier selected by PC0PCF[4:2].

Table 25.1. Pull-Up Resistor Current

PC0PCF[4:2]	I_R
000	0
001	1 µA
010	4 µA
011	16 µA
100	64 µA
101	256 µA
110	1 mA
111	4 mA

Table 25.2. Sample Rate Duty-Cycle Multiplier

PC0MD[5:4]	D_{SR}
000	1
001	1/2
010	1/4
011	1/8

Table 25.3. Pull-Up Duty-Cycle Multiplier

PC0PCF[4:2]	D_{PU}
000	1/4
001	3/8
010	1/2
011	3/4

SFR Definition 27.1. XBR0: Port I/O Crossbar Register 0

Bit	7	6	5	4	3	2	1	0
Name	CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E
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 = 0x0; SFR Address = 0xE1

Bit	Name	Function
7	CP1AE	**Comparator1 Asynchronous Output Enable.** 0: Asynchronous CP1 output unavailable at Port pin. 1: Asynchronous CP1 output routed to Port pin.
6	CP1E	**Comparator1 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
5	CP0AE	**Comparator0 Asynchronous Output Enable.** 0: Asynchronous CP0 output unavailable at Port pin. 1: Asynchronous CP0 output routed to Port pin.
4	CP0E	**Comparator0 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
3	SYSCKE	**$\overline{\text{SYSCLK}}$ Output Enable.** 0: $\overline{\text{SYSCLK}}$ output unavailable at Port pin. 1: $\overline{\text{SYSCLK}}$ output routed to Port pin.
2	SMB0E	**SMBus I/O Enable.** 0: SMBus I/O unavailable at Port pin. 1: SDA and SCL routed to Port pins.
1	SPI0E	**SPI0 I/O Enable** 0: SPI0 I/O unavailable at Port pin. 1: SCK, MISO, and MOSI (for SPI0) routed to Port pins. NSS (for SPI0) routed to Port pin only if SPI0 is configured to 4-wire mode.
0	URT0E	**UART0 Output Enable.** 0: UART I/O unavailable at Port pin. 1: TX0 and RX0 routed to Port pins P0.4 and P0.5.
Note: SPI0 can be assigned either 3 or 4 Port I/O pins.		

wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

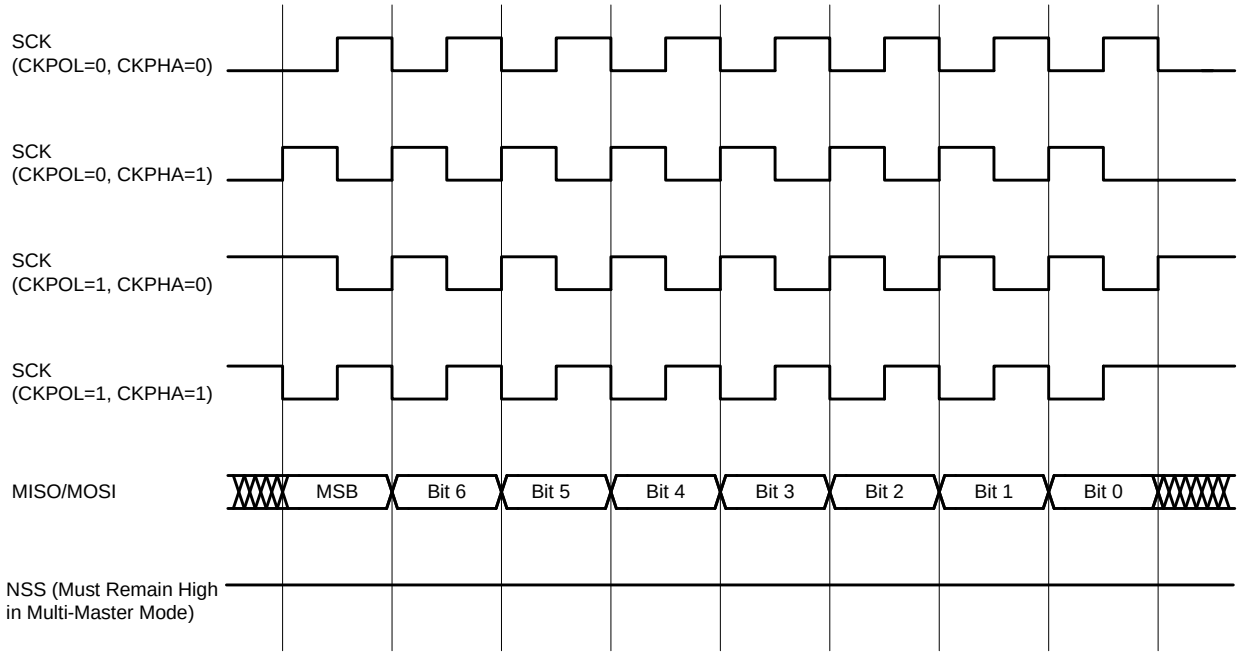

Figure 30.5. Master Mode Data/Clock Timing

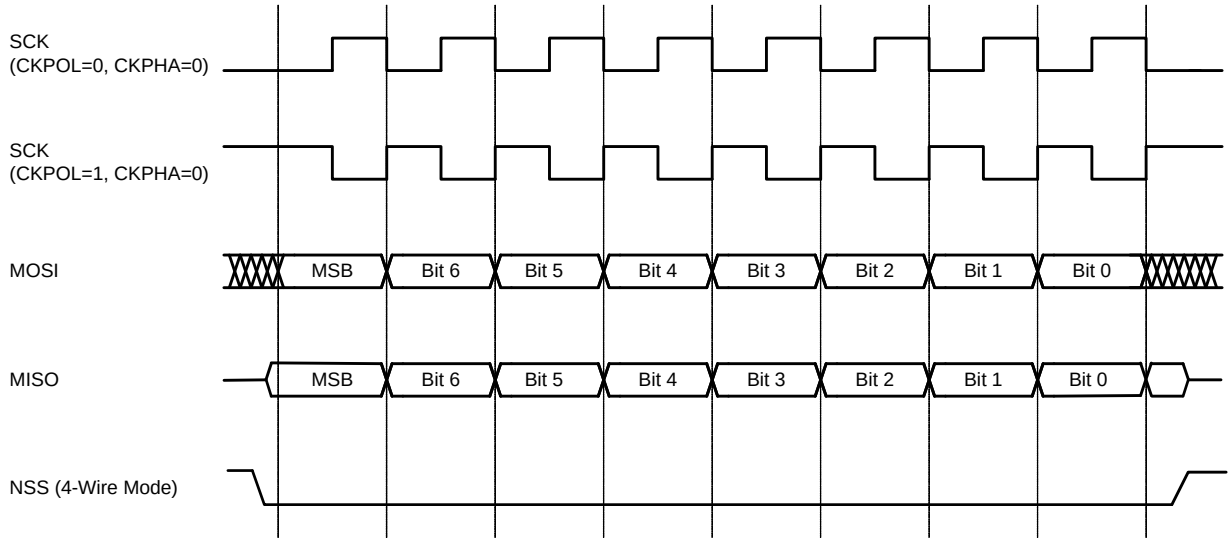

Figure 30.6. Slave Mode Data/Clock Timing (CKPHA = 0)

31. Enhanced Serial Peripheral Interface with DMA Support (SPI1)

The Enhanced Serial Peripheral Interface (SPI1) provides access to a flexible, full-duplex synchronous serial bus. SPI1 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 SPI1 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.

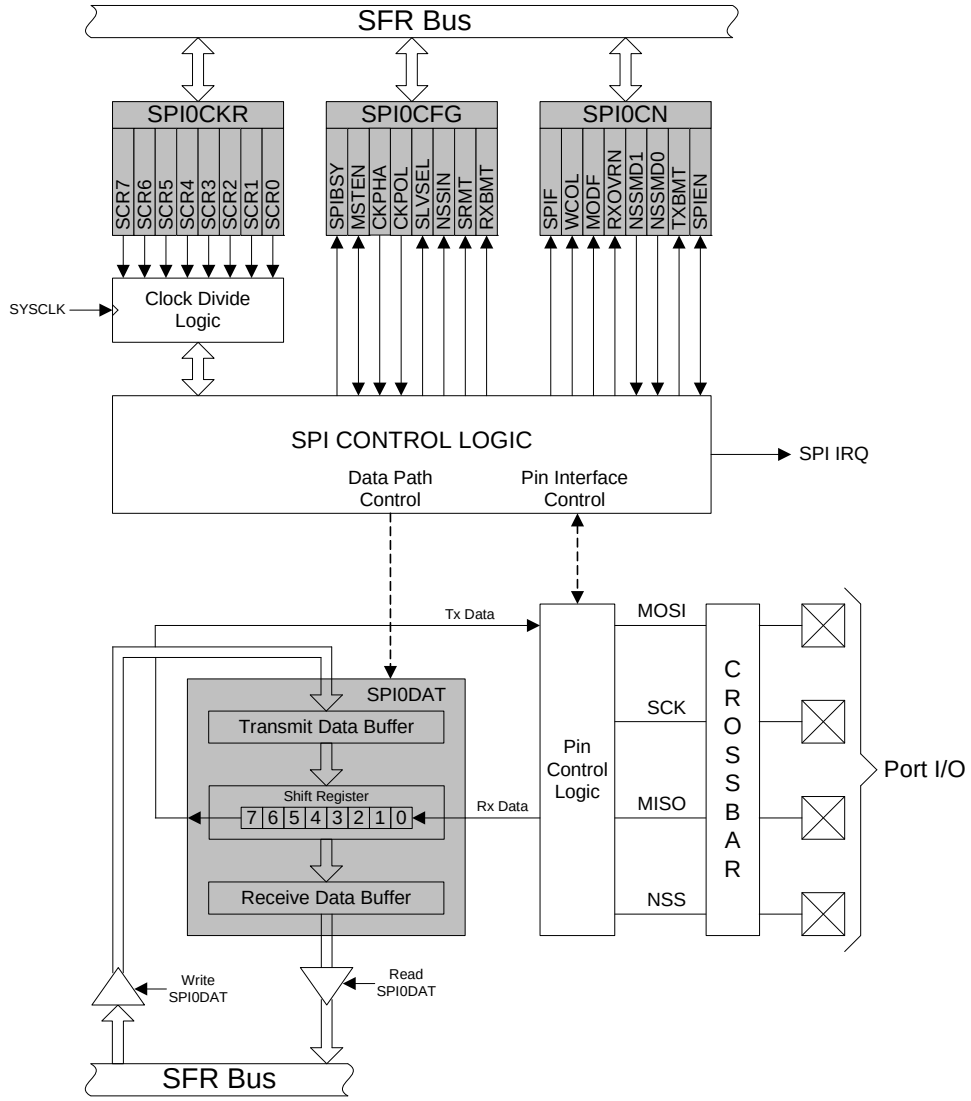

Figure 31.1. SPI Block Diagram