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	57
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K 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/c8051f968-a-gq

Figure 32.2. T0 Mode 2 Block Diagram 452
Figure 32.3. T0 Mode 3 Block Diagram 453
Figure 32.4. Timer 2 16-Bit Mode Block Diagram 458
Figure 32.5. Timer 2 8-Bit Mode Block Diagram 459
Figure 32.6. Timer 2 Capture Mode Block Diagram 460
Figure 32.7. Timer 3 16-Bit Mode Block Diagram 464
Figure 32.8. Timer 3 8-Bit Mode Block Diagram 465
Figure 32.9. Timer 3 Capture Mode Block Diagram 466
Figure 33.1. PCA Block Diagram 470
Figure 33.2. PCA Counter/Timer Block Diagram 472
Figure 33.3. PCA Interrupt Block Diagram 473
Figure 33.4. PCA Capture Mode Diagram 475
Figure 33.5. PCA Software Timer Mode Diagram 476
Figure 33.6. PCA High-Speed Output Mode Diagram 477
Figure 33.7. PCA Frequency Output Mode 478
Figure 33.8. PCA 8-Bit PWM Mode Diagram 479
Figure 33.9. PCA 9, 10 and 11-Bit PWM Mode Diagram 480
Figure 33.10. PCA 16-Bit PWM Mode 481
Figure 33.11. PCA Module 5 with Watchdog Timer Enabled 482
Figure 34.1. Typical C2 Pin Sharing 493

1.3. Serial Ports

The C8051F96x Family includes an SMBus/I^2C interface, a full-duplex UART with enhanced baud rate configuration, and two Enhanced SPI interfaces. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

1.4. Programmable Counter Array

An on-chip Programmable Counter/Timer Array (PCA) is included in addition to the four 16-bit general purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer time base with six programmable capture/compare modules. The PCA clock is derived from one of six sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflows, an External Clock Input (ECI), the system clock, or the external oscillator clock source divided by 8.

Each capture/compare module can be configured to operate in a variety of modes: edge-triggered capture, software timer, high-speed output, pulse width modulator (8, 9, 10, 11, or 16-bit), or frequency output. Additionally, Capture/Compare Module 5 offers watchdog timer (WDT) capabilities. Following a system reset, Module 5 is configured and enabled in WDT mode. The PCA Capture/Compare Module I/O and External Clock Input may be routed to Port I/O via the Digital Crossbar.

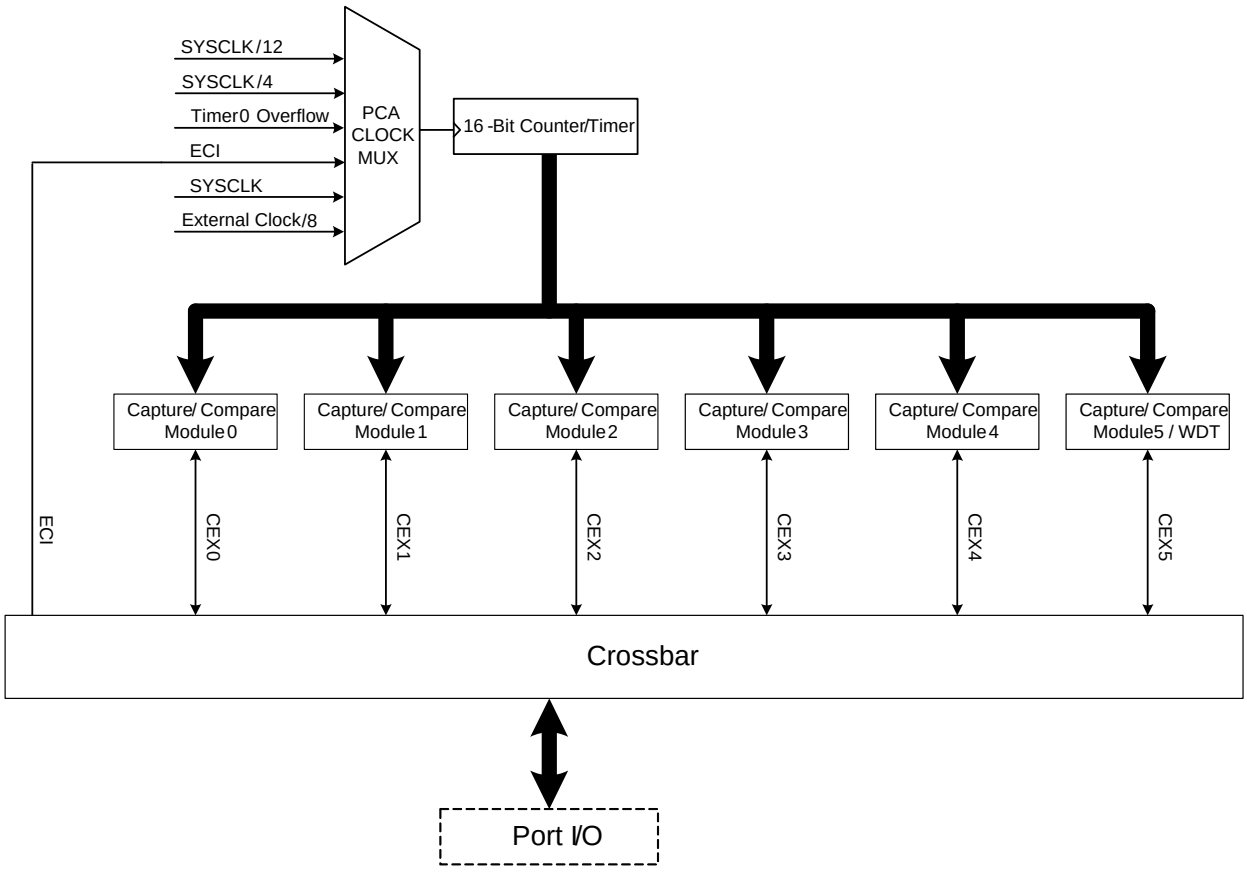

Figure 1.12. PCA Block Diagram

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P4.0	A11	23	11	D I/O or A In	Port 4.0. See Port I/O Section for a complete description.
LCD8				A O	LCD Segment Pin 8
P4.1	B3	7		D I/O or A In	Port 4.1. See Port I/O Section for a complete description.
LCD9				A O	LCD Segment Pin 9
P4.2	B2	5		D I/O or A In	Port 4.2. See Port I/O Section for a complete description.
LCD10				A O	LCD Segment Pin 10
P4.3	B1	3		D I/O or A In	Port 4.3. See Port I/O Section for a complete description.
LCD11				A O	LCD Segment Pin 11
P4.4	D1	80		D I/O or A In	Port 4.4. See Port I/O Section for a complete description.
LCD12				A O	LCD Segment Pin 12
P4.5	B28	77		D I/O or A In	Port 4.5. See Port I/O Section for a complete description.
LCD13				A O	LCD Segment Pin 13
P4.6	B27	75		D I/O or A In	Port 4.6. See Port I/O Section for a complete description.
LCD14				A O	LCD Segment Pin 14
P4.7	B26	73		D I/O or A In	Port 4.7. See Port I/O Section for a complete description.
LCD15				A O	LCD Segment Pin 15
P5.0	B25	71		D I/O or A In	Port 5.0. See Port I/O Section for a complete description.
LCD16				A O	LCD Segment Pin 16
P5.1	B24	69		D I/O or A In	Port 5.1. See Port I/O Section for a complete description.
LCD17				A O	LCD Segment Pin 17

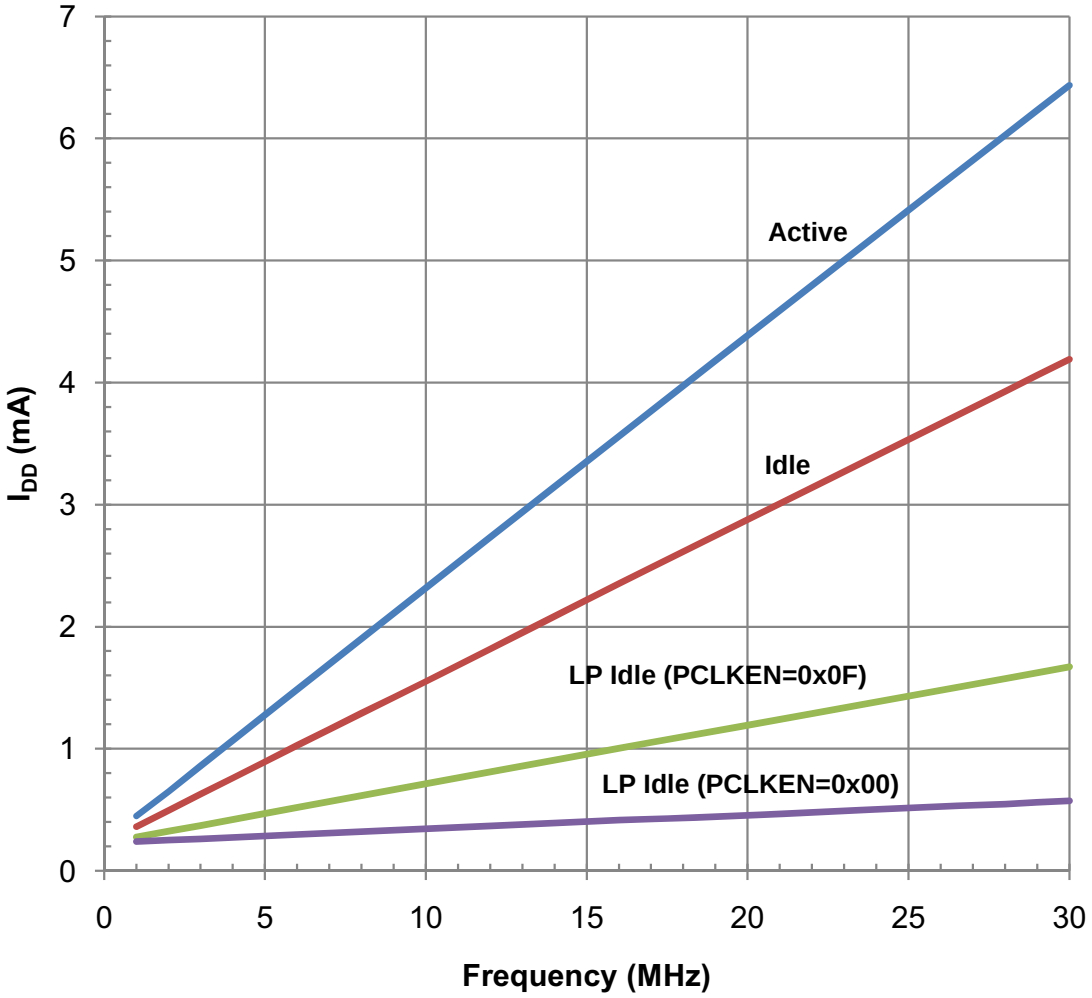

Figure 4.1. Frequency Sensitivity (External CMOS Clock, 25°C)

Table 4.7. Power Management Electrical Specifications

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Idle Mode Wake-up Time		2	—	3	SYSCLKs
Suspend Mode Wake-up Time	CLKDIV = 0x00 Low Power or Precision Osc.	—	400	—	ns
Sleep Mode Wake-up Time		—	2	—	µs

Table 4.8. Flash Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.,

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F960/1/2/3	131072	—	—	bytes
	C8051F964/5	65536	—	—	bytes
	C8051F966/7	32768	—	—	bytes
	C8051F968/9	16384	—	—	bytes
Endurance		20 k	100k	—	Erase/Write Cycles
Erase Cycle Time		28	32	36	ms
Write Cycle Time		57	64	71	µs

Table 4.9. Internal Precision Oscillator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	–40 to +85 °C, V_{BAT} = 1.8–3.8 V	24	24.5	25	MHz
Oscillator Supply Current (from V_{BAT})	25 °C; includes bias current of 50 µA typical	—	300*	—	µA
***Note:** Does not include clock divider or clock tree supply current.					

Table 4.10. Internal Low-Power Oscillator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	–40 to +85 °C, V_{BAT} = 1.8–3.8 V	18	20	22	MHz
Oscillator Supply Current (from V_{BAT})	25 °C No separate bias current required	—	100*	—	µA
***Note:** Does not include clock divider or clock tree supply current.					

Input Voltage	Right-Justified ADC0H:ADC0L (AD0SJST = 000)	Left-Justified ADC0H:ADC0L (AD0SJST = 100)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When the repeat count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, 16, 32, or 64 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0AC register. When a repeat count higher than 1, the ADC output must be right-justified (AD0SJST = 0xx); unused bits in the ADC0H and ADC0L registers are set to 0. The example below shows the right-justified result for various input voltages and repeat counts. Notice that accumulating 2^n samples is equivalent to left-shifting by n bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 16	Repeat Count = 64
V_{REF} x 1023/1024	0x0FFC	0x3FF0	0xFFC0
V_{REF} x 512/1024	0x0800	0x2000	0x8000
V_{REF} x 511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

The AD0SJST bits can be used to format the contents of the 16-bit accumulator. The accumulated result can be shifted right by 1, 2, or 3 bit positions. Based on the principles of oversampling and averaging, the effective ADC resolution increases by 1 bit each time the oversampling rate is increased by a factor of 4. The example below shows how to increase the effective ADC resolution by 1, 2, and 3 bits to obtain an effective ADC resolution of 11-bit, 12-bit, or 13-bit respectively without CPU intervention.

Input Voltage	Repeat Count = 4 Shift Right = 1 11-Bit Result	Repeat Count = 16 Shift Right = 2 12-Bit Result	Repeat Count = 64 Shift Right = 3 13-Bit Result
V_{REF} x 1023/1024	0x07F7	0x0FFC	0x1FF8
V_{REF} x 512/1024	0x0400	0x0800	0x1000
V_{REF} x 511/1024	0x03FE	0x04FC	0x0FF8
0	0x0000	0x0000	0x0000

SFR Definition 8.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xD0; Bit-Addressable

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under software control.
4:3	RS[1:0]	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: ■ An ADD, ADDC, or SUBB instruction causes a sign-change overflow. ■ A MUL instruction results in an overflow (result is greater than 255). ■ A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

Table 10.1. EMIF Pinout (C8051F960/3/6)

Multiplexed Mode		
Signal Name	**Port Pin**	
	8-Bit Mode[1]	**16-Bit Mode[2]**
$\overline{RD}$	P3.6	P3.6
$\overline{WR}$	P3.7	P3.7
ALE	P3.5	P3.5
AD0	P6.0	P6.0
AD1	P6.1	P6.1
AD2	P6.2	P6.2
AD3	P6.3	P6.3
AD4	P6.4	P6.4
AD5	P6.5	P6.5
AD6	P6.6	P6.6
AD7	P6.7	P6.7
A8	—	P5.0
A9	—	P5.1
A10	—	P5.2
A11	—	P5.3
A12	—	P5.4
A13	—	P5.5
A14	—	P5.6
A15	—	P5.7
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
Required I/O:	**11**	**19**

Non Multiplexed Mode		
Signal Name	**Port Pin**	
	8-Bit Mode [1]	**16-Bit Mode[2]**
$\overline{RD}$	P3.6	P3.6
$\overline{WR}$	P3.7	P3.7
D0	P6.0	P6.0
D1	P6.1	P6.1
D2	P6.2	P6.2
D3	P6.3	P6.3
D4	P6.4	P6.4
D5	P6.5	P6.5
D6	P6.6	P6.6
D7	P6.7	P6.7
A0	P5.0	P5.0
A1	P5.1	P5.1
A2	P5.2	P5.2
A3	P5.3	P5.3
A4	P5.4	P5.4
A5	P5.5	P5.5
A6	P5.6	P5.6
A7	P5.7	P5.7
A8	—	P4.0
A9	—	P4.1
A10	—	P4.2
A11	—	P4.3
A12	—	P4.4
A13	—	P4.5
A14	—	P4.6
A15	—	P4.7
Required I/O:	**18**	**26**

Notes:
1. Using 8-bit movx instruction without bank select.
2. Using 16-bit movx instruction.

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.

12.2. 32-bit CRC Algorithm

The C8051F41x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

Step 1. XOR the least-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 (0x00000000 or 0xFFFFFFFF).
Step 2. Right-shift the CRC result.
Step 3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
Step 4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit 'F41x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input)
{
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)

   CRC_acc = CRC_acc ^ CRC_input;

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x00000001) == 0x00000001)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc >> 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc >> 1;
      }
   }
   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 32-bit 'F41x CRC algorithm (an initial value of 0xFFFFFFFF is used):

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.

SFR Definition 16.3. SFRNEXT: SFR Next

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

;SFR Page = All Pages; SFR Address = 0x85

Bit	Name	Function
7:0	SFRNEXT[7:0]	**SFR Page Bits.** This is the value that will go to the SFR Page register upon a return from interrupt. Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the second byte of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to “push” or “pop”. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Table 16.3. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
ADC0AC	0xBA	0x0	ADC0 Accumulator Configuration	88
ADC0CF	0xBC	0x0	ADC0 Configuration	87
ADC0CN	0xE8	All pages	ADC0 Control	86
ADC0GTH	0xC4	0x0	ADC0 Greater-Than Compare High	92
ADC0GTL	0xC3	0x0	ADC0 Greater-Than Compare Low	92
ADC0H	0xBE	0x0	ADC0 High	91
ADC0L	0xBD	0x0	ADC0 Low	91
ADC0LTH	0xC6	0x0	ADC0 Less-Than Compare Word High	93
ADC0LTL	0xC5	0x0	ADC0 Less-Than Compare Word Low	93
ADC0MX	0xBB	0x0	ADC0 MUX	96
ADC0PWR	0xBA	0xF	ADC0 Burst Mode Power-Up Time	89
ADC0TK	0xBB	0xF	ADC0 Tracking Control	90
AES0BCFG	0xE9	0x2	AES0 Block Configuration	202
AES0BIN	0xEB	0x2	AES0 Block Input	204
AES0DCFG	0xEA	0x2	AES0 Data Configuration	203
AES0KIN	0xED	0x2	AES0 Key Input	205
AES0XIN	0xEC	0x2	AES0 XOR Input	205
AES0YOUT	0xF5	0x2	AES Y Out	206
CKCON	0x8E	0x0	Clock Control	449
CLKMODE	0xFD	0xF	Clock Mode	267
CLKSEL	0xA9	0x0 and 0xF	Clock Select	296
CPT0CN	0x9B	0x0	Comparator0 Control	108
CPT0MD	0x9D	0x0	Comparator0 Mode Selection	109
CPT0MX	0x9F	0x0	Comparator0 Mux Selection	113
CPT1CN	0x9A	0x0	Comparator1 Control	110
CPT1MD	0x9C	0x0	Comparator1 Mode Selection	111
CPT1MX	0x9E	0x0	Comparator1 Mux Selection	114
CRC0AUTO	0x96	0xF	CRC0 Automatic Control	166
CRC0CNT	0x97	0xF	CRC0 Automatic Flash Sector Count	166
CRC0CN	0x92	0xF	CRC0 Control	164
CRC0DAT	0x91	0xF	CRC0 Data	165
CRC0FLIP	0x94	0xF	CRC0 Flip	167
CRC0IN	0x93	0xF	CRC0 Input	165

SFR Definition 17.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xE6

Bit	Name	Function
7	ET3	**Enable Timer 3 Interrupt.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6	ECP1	**Enable Comparator1 (CP1) Interrupt.** This bit sets the masking of the CP1 interrupt. 0: Disable CP1 interrupts. 1: Enable interrupt requests generated by the CP1RIF or CP1FIF flags.
5	ECP0	**Enable Comparator0 (CP0) Interrupt.** This bit sets the masking of the CP0 interrupt. 0: Disable CP0 interrupts. 1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.
4	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**Enable ADC0 Conversion Complete Interrupt.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the AD0INT flag.
2	EWADC0	**Enable Window Comparison ADC0 Interrupt.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).
1	ERTC0A	**Enable SmaRTClock Alarm Interrupts.** This bit sets the masking of the SmaRTClock Alarm interrupt. 0: Disable SmaRTClock Alarm interrupts. 1: Enable interrupt requests generated by a SmaRTClock Alarm.
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

18.5.2. PSWE Maintenance

1. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write flash bytes and one routine in code that sets both PSWE and PSEE both to a 1 to erase flash pages.
2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in “AN201: Writing to Flash from Firmware,” available from the Silicon Laboratories web site.
3. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the flash write or erase operation will be serviced in priority order after the flash operation has been completed and interrupts have been re-enabled by software.
4. Make certain that the flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
5. Add address bounds checking to the routines that write or erase flash memory to ensure that a routine called with an illegal address does not result in modification of the flash.

18.5.3. System Clock

1. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
2. If operating from the external oscillator, switch to the internal oscillator during flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the flash operation has completed.

Additional flash recommendations and example code can be found in “AN201: Writing to Flash from Firmware," available from the Silicon Laboratories website.

SFR Definition 19.2. PCLKEN: Peripheral Clock Enable

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

SFR Page = 0xF; SFR Address = 0xFE

Bit	Name	Function
7:4	Unused	Read = 0b; Write = don't care.
3	PCLKEN3	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode. 1: Enable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode.
2	PCLKEN2	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode. 1: Enable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode.
1	PCLKEN1	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disableclocks to ADC0 and PCA0 in Low Power Idle Mode. 1: Enable clocks to ADC0 and PCA0 in Low Power Idle Mode.
0	PCLKEN0	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode. 1: Enable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode.

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a supply monitor reset. See Section “4. Electrical Characteristics” on page 56 for complete electrical characteristics of the active mode supply monitors.
- Software should take care not to inadvertently disable the supply monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to 1 to keep the supply monitor enabled as a reset source.
- The supply monitor must be enabled before selecting it as a reset source**.** Selecting the supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the supply monitor and selecting it as a reset source. See Section “4. Electrical Characteristics” on page 56 for minimum supply monitor turn-on time. **No delay should be introduced in systems where software contains routines that erase or write Flash memory.** The procedure for enabling the V_{DD} supply monitor and selecting it as a reset source is shown below:
  1. Enable the Supply Monitor (VDMEN bit in VDM0CN = 1).
  2. Wait for the Supply Monitor to stabilize (optional).
  3. Select the Supply Monitor as a reset source (PORSF bit in RSTSRC = 1).

27.1. Port I/O Modes of Operation

Port pins P0.0–P6.7 use the Port I/O cell shown in Figure 27.2. The supply pin for P1.4 - P2.3 is VIORF and the supply for all other GPIOs is VIO. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN registers. P7.0 can only be used for digital functtons and is shared with the C2D signal. On reset, all Port I/O cells default to a digital high impedance state with weak pull-ups enabled.

27.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC input, external oscillator input/output, or AGND, VREF, or Current Reference output should be configured for analog I/O (PnMDIN.n = 0). When a pin is configured for analog I/O, its weak pullup and digital receiver are disabled. In most cases, software should also disable the digital output drivers. Port pins configured for analog I/O will always read back a value of 0 regardless of the actual voltage on the pin.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

27.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the supply or GND rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high and low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user must ensure that digital I/O are always internally or externally pulled or driven to a valid logic state. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

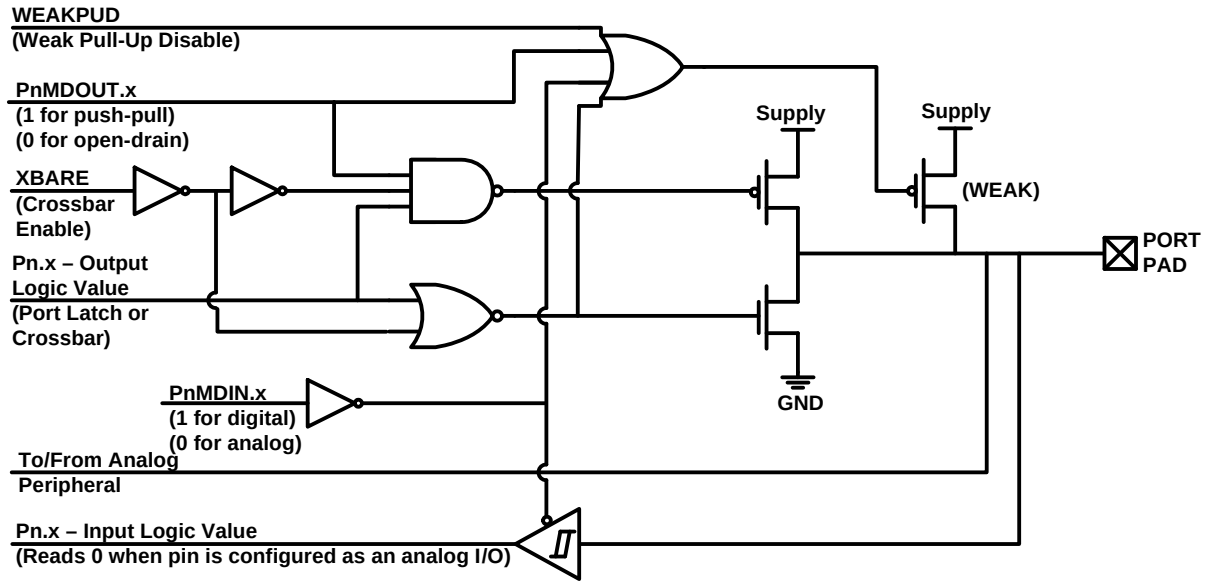

Figure 27.2. Port I/O Cell Block Diagram

SFR Definition 27.18. P2: Port2

Bit	7	6	5	4	3	2	1	0
Name	P2[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0xA0; Bit-Addressable

Bit	Name	Description	Read	Write
7:0	P2[7:0]	**Port 2 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P2.n Port pin is logic LOW. 1: P2.n Port pin is logic HIGH.

SFR Definition 27.19. P2SKIP: Port2 Skip

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD6

Bit	Name	Description	Read	Write
7:0	P2SKIP[7:0]	**Port 1 Crossbar Skip Enable Bits.** These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P2.n pin is not skipped by the Crossbar. 1: Corresponding P2.n pin is skipped by the Crossbar.		

SFR Definition 32.1. CKCON: Clock Control

Bit	7	6	5	4	3	2	1	0
Name	T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA[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 = 0x0; SFR Address = 0x8E

Bit	Name	Function
7	T3MH	**Timer 3 High Byte Clock Select.** Selects the clock supplied to the Timer 3 high byte (split 8-bit timer mode only). 0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 high byte uses the system clock.
6	T3ML	**Timer 3 Low Byte Clock Select.** Selects the clock supplied to Timer 3. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 low byte uses the system clock.
5	T2MH	**Timer 2 High Byte Clock Select.** Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
4	T2ML	**Timer 2 Low Byte Clock Select.** Selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 low byte uses the system clock.
3	T1M	**Timer 1 Clock Select.** Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale bits SCA[1:0]. 1: Timer 1 uses the system clock.
2	T0M	**Timer 0 Clock Select.** Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale bits SCA[1:0]. 1: Counter/Timer 0 uses the system clock.
1:0	SCA[1:0]	**Timer 0/1 Prescale Bits.** These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)