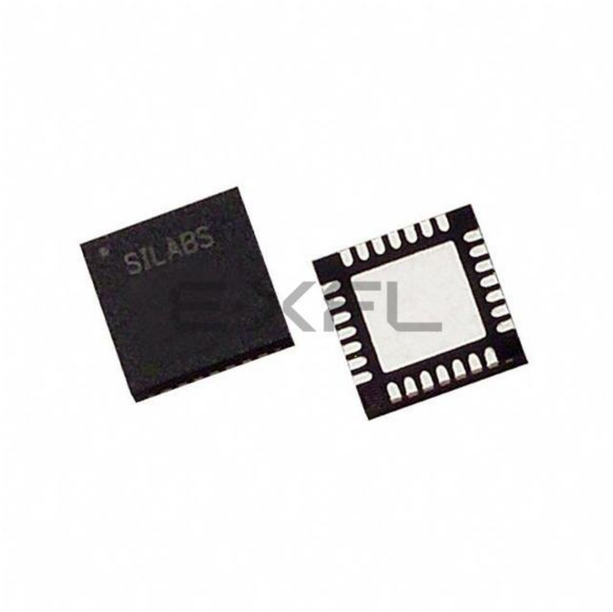

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 17x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f369-c-gmr

3. Global Electrical Characteristics

Table 3.1. Global Electrical Characteristics

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Voltage	SYSCLK = 0 to 50 MHz SYSCLK > 50 MHz	2.7 3.0	3.0 3.3	3.6 3.6	V
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[1,2]	C8051F360/1/2/3/4/5 C8051F366/7/8/9	0 0	— —	100 50	MHz MHz
Specified Operating Temperature Range		–40	—	+85	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[2]	V_{DD} = 3.6 V, F = 100 MHz	—	68	75	mA
	V_{DD} = 3.6 V, F = 25 MHz	—	21	25	mA
	V_{DD} = 3.0 V, F = 100 MHz	—	54	60	mA
	V_{DD} = 3.0 V, F = 25 MHz	—	16	18	mA
	V_{DD} = 3.0 V, F = 1 MHz	—	0.48	—	mA
	V_{DD} = 3.0 V, F = 80 kHz	—	36	—	µA
I_{DD} Supply Sensitivity[3]	F = 25 MHz	—	56	—	%/V
	F = 1 MHz	—	57	—	%/V
I_{DD} Frequency Sensitivity[3,4]	V_{DD} = 3.0 V, F <= 20 MHz, T = 25 °C	—	0.45	—	mA/MHz
	V_{DD} = 3.0 V, F > 20 MHz, T = 25 °C	—	0.38	—	mA/MHz
	V_{DD} = 3.6 V, F <= 20 MHz, T = 25 °C	—	0.61	—	mA/MHz
	V_{DD} = 3.6 V, F > 20 MHz, T = 25 °C	—	0.51	—	mA/MHz

Table 4.1. Pin Definitions for the C8051F36x (Continued)

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
P1.0	45	26	22	D I/O or A In	Port 1.0. See Section 17 for a complete description.
P1.1	44	25	21	D I/O or A In	Port 1.1. See Section 17 for a complete description.
P1.2	41	24	20	D I/O or A In	Port 1.2. See Section 17 for a complete description.
P1.3	40	23	19	D I/O or A In	Port 1.3. See Section 17 for a complete description.
P1.4	39	22	18	D I/O or A In	Port 1.4. See Section 17 for a complete description.
P1.5	38	21	17	D I/O or A In	Port 1.5. See Section 17 for a complete description.
P1.6	37	20	16	D I/O or A In	Port 1.6. See Section 17 for a complete description.
P1.7	36	19	15	D I/O or A In	Port 1.7. See Section 17 for a complete description.
P2.0	35	18	14	D I/O or A In	Port 2.0. See Section 17 for a complete description.
P2.1	34	17	13	D I/O or A In	Port 2.1. See Section 17 for a complete description.
P2.2	33	16	12	D I/O or A In	Port 2.2. See Section 17 for a complete description.
P2.3	32	15	11	D I/O or A In	Port 2.3. See Section 17 for a complete description.
P2.4	29	14	10	D I/O or A In	Port 2.4. See Section 17 for a complete description.
P2.5	28	13	9	D I/O or A In	Port 2.5. See Section 17 for a complete description.
P2.6	27	12	8	D I/O or A In	Port 2.6. See Section 17 for a complete description.
P2.7	26	11	7	D I/O or A In	Port 2.7. See Section 17 for a complete description.
P3.0	25	—	—	D I/O or A In	Port 3.0. See Section 17 for a complete description.

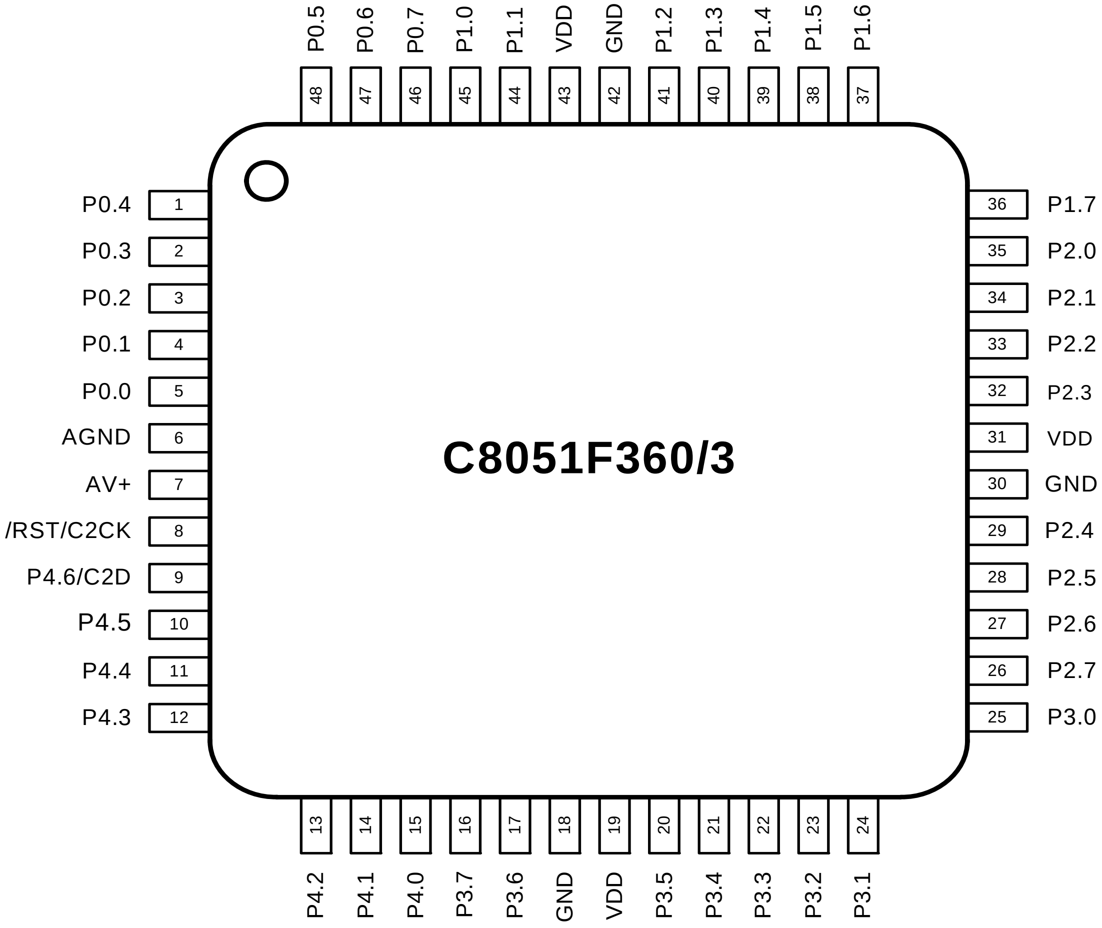

Figure 4.1. TQFP-48 Pinout Diagram (Top View)

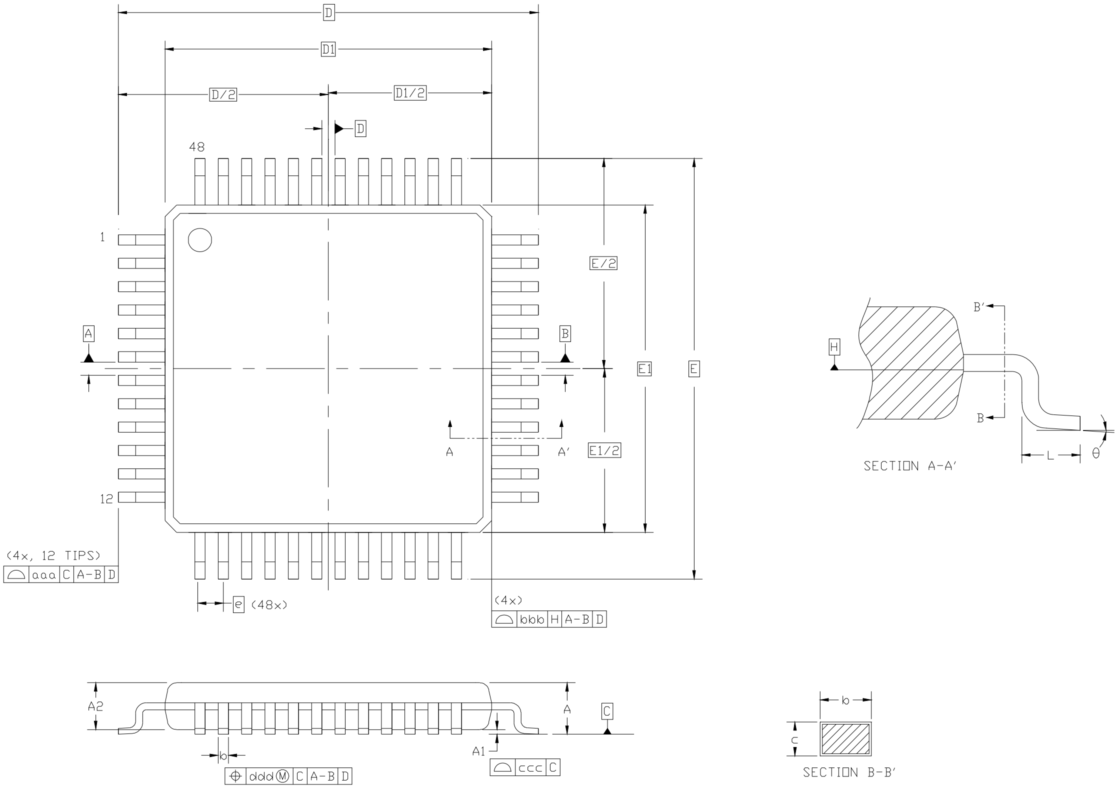

Figure 4.2. TQFP-48 Package Diagram

Table 4.2. TQFP-48 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	—	—	1.20	E	9.00 BSC.		
A1	0.05	—	0.15	E1	7.00 BSC.		
A2	0.95	1.00	1.05	L	0.45	0.60	0.75
b	0.17	0.22	0.27	aaa	0.20		
c	0.09	—	0.20	bbb	0.20		
D	9.00 BSC.			ccc	0.08		
D1	7.00 BSC.			ddd	0.08		
e	0.50 BSC.			θ	0°	3.5°	7°

Notes:

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

9. 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. Included are five 16-bit counter/timers (see description in Section 21), one full-duplex UART (see description in Section 19), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see Section 9.4.6), and up to four byte-wide and one 7-bit-wide I/O Ports (see description in Section 17). The CIP-51 also includes on-chip debug hardware (see description in Section 24), and interfaces directly with the MCU's 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 9.1 for a block diagram).

- Fully Compatible with MCS-51 Instruction Set
- 100 or 50 MIPS Peak Using the On-Chip PLL
- 256 Bytes of Internal RAM
- 8/4 Byte-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic

The CIP-51 includes the following features:

9.1. 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.

With the CIP-51's system clock running at 100 MHz, it has a peak throughput of 100 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

9.4.6.3. SFR Page Stack Example

The following is an example that shows the operation of the SFR Page Stack during interrupts.

In this example, the SFR Page Control is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to OSCICN (SFR “OSCICN”, located at address 0xB6 on SFR Page 0x0F). The device is also using the Programmable Counter Array (PCA) and the 10-bit ADC (ADC0) window comparator to monitor a voltage. The PCA is timing a critical control function in its interrupt service routine (ISR), so its interrupt is enabled and is set to *high* priority. The ADC0 is monitoring a voltage that is less important, but to minimize the software overhead its window comparator is being used with an associated ISR that is set to *low* priority. At this point, the SFR page is set to access the OSCICN SFR (SFRPAGE = 0x0F). See Figure 9.4 below.

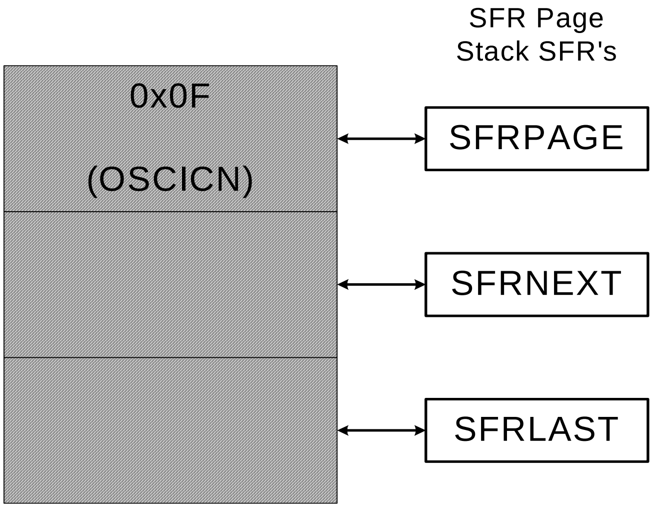

Figure 9.4. SFR Page Stack While Using SFR Page 0x0F To Access OSCICN

While CIP-51 executes in-line code (writing values to OSCICN in this example), ADC0 Window Comparator Interrupt occurs. The CIP-51 vectors to the ADC0 Window Comparator ISR and pushes the current SFR Page value (SFR Page 0x0F) into SFRNEXT in the SFR Page Stack. SFR page 0x00 is then automatically placed in the SFRPAGE register. SFRPAGE is considered the “top” of the SFR Page Stack. Software can now access the ADC0 SFR’s. Software may switch to any SFR Page by writing a new value to the SFRPAGE register at any time during the ADC0 ISR to access SFR’s that are not on SFR Page 0x00. See Figure 9.5 below.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
PCA0CPL2	0xEB	All Pages	PCA Module 2 Capture/Compare Low Byte	page 277
PCA0CPL3	0xED	All Pages	PCA Module 3 Capture/Compare Low Byte	page 277
PCA0CPL4	0xFD	All Pages	PCA Module 4 Capture/Compare Low Byte	page 277
PCA0CPL5	0xF5	All Pages	PCA Module 5 Capture/Compare Low Byte	page 277
PCA0CPM0	0xDA	All Pages	PCA Module 0 Mode	page 276
PCA0CPM1	0xDB	All Pages	PCA Module 1 Mode	page 276
PCA0CPM2	0xDC	All Pages	PCA Module 2 Mode	page 276
PCA0CPM3	0xDD	All Pages	PCA Module 3 Mode	page 276
PCA0CPM4	0xDE	All Pages	PCA Module 4 Mode	page 276
PCA0CPM5	0xDF	All Pages	PCA Module 5 Mode	page 276
PCA0H	0xFA	All Pages	PCA Counter High Byte	page 277
PCA0L	0xF9	All Pages	PCA Counter Low Byte	page 277
PCA0MD	0xD9	All Pages	PCA Mode	page 275
PCON	0x87	All Pages	Power Control	page 106
PLL0CN	0xB3	F	PLL Control	page 179
PLL0DIV	0xA9	F	PLL Divider	page 179
PLL0FLT	0xB2	F	PLL Filter	page 180
PLL0MUL	0xB1	F	PLL Multiplier	page 180
PSCTL	0x8F	0	Flash Write/Erase Control	page 142
PSW	0xD0	All Pages	Program Status Word	page 103
REF0CN	0xD1	All Pages	Voltage Reference Control	page 68[1]
RSTSRC	0xEF	All Pages	Reset Source	page 133
SBUF0	0x99	All Pages	UART 0 Data Buffer	page 224
SCON0	0x98	All Pages	UART 0 Control	page 223
SFR0CN	0xE5	F	SFR Page Control	page 94
SFRLAST	0x86	All Pages	SFR Stack Last Page	page 95
SFRNEXT	0x85	All Pages	SFR Stack Next Page	page 95
SFRPAGE	0xA7	All Pages	SFR Page Select	page 94
SMB0CF	0xC1	All Pages	SMBus Configuration	page 206
SMB0CN	0xC0	All Pages	SMBus Control	page 208
SMB0DAT	0xC2	All Pages	SMBus Data	page 210

Notes:

1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

SFR Definition 10.1. IE: Interrupt Enable

SFR Page: **all pages**
SFR Address: **0xA8**

(bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: EA: Global Interrupt Enable.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit 6: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit 5: ET2: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit 4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit 3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit 2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the /INT1 input.

Bit 1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit 0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the /INT0 input.

Steps 3–8 must be repeated for each byte to be written

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. When writing to addresses located in any of the four code banks, a Flash write block is four bytes long, from addresses ending in 00b to addresses ending in 11b. Writes must be performed sequentially (i.e. addresses ending in 00b, 01b, 10b, and 11b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 11b. The Flash write will be performed following the MOVX write that targets the address ending in 1b. If any bytes in the block do not need to be updated in Flash, they should be written to 0xFF. The recommended procedure for writing Flash in blocks is as follows:

Step 1. Disable interrupts.
Step 2. Set CHBLKW (register CCH0CN) to select block write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write the first data byte to the first block location (ending in 00b).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Write the first key code to FLKEY: 0xA5.
Step 10. Write the second key code to FLKEY: 0xF1.
Step 11. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 12. Clear the PSEE bit (register PSCTL).
Step 13. Using the MOVX instruction, write the second data byte to the second block location (ending in 01b).
Step 14. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 15. Write the first key code to FLKEY: 0xA5.
Step 16. Write the second key code to FLKEY: 0xF1.
Step 17. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 18. Clear the PSEE bit (register PSCTL).
Step 19. Using the MOVX instruction, write the third data byte to the third block location (ending in 10b).
Step 20. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 21. Write the first key code to FLKEY: 0xA5.
Step 22. Write the second key code to FLKEY: 0xF1.
Step 23. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 24. Clear the PSEE bit (register PSCTL).
Step 25. Using the MOVX instruction, write the fourth data byte to the last block location (ending in 11b).
Step 26. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 27. Re-enable interrupts.

Steps 3-26 must be repeated for each block to be written.

13.1.4. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written and erased using the MOVX write instruction (as described in Section 13.1.2 and Section 13.1.3) and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

SFR Definition 14.1. CCH0CN: Cache Control

SFR Page: F
SFR Address: 0x84

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CHWREN	CHRDEN	CHPFEN	CHFLSH	CHRETI	CHISR	CHMOVC	CHBLKW	11100110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CHWREN: Cache Write Enable.
This bit enables the processor to write to the cache memory.
0: Cache contents are not allowed to change, except during Flash writes/erasures or cache locks.
1: Writes to cache memory are allowed.

Bit 6: CHRDEN: Cache Read Enable.
This bit enables the processor to read instructions from the cache memory.
0: All instruction data comes from Flash memory or the prefetch engine.
1: Instruction data is obtained from cache (when available).

Bit 5: CHPFEN: Cache Prefetch Enable.
This bit enables the prefetch engine.
0: Prefetch engine is disabled.
1: Prefetch engine is enabled.

Bit 4: CHFLSH: Cache Flush.
When written to a ‘1’, this bit clears the cache contents. This bit always reads ‘0’.

Bit 3: CHRETI: Cache RETI Destination Enable.
This bit enables the destination of a RETI address to be cached.
0: Destinations of RETI instructions will not be cached.
1: RETI destinations will be cached.

Bit 2: CHISR: Cache ISR Enable.
This bit allows instructions which are part of an Interrupt Service Routine (ISR) to be cached.
0: Instructions in ISRs will not be loaded into cache memory.
1: Instructions in ISRs can be cached.

Bit 1: CHMOVC: Cache MOVC Enable.
This bit allows data requested by a MOVC instruction to be loaded into the cache memory.
0: Data requested by MOVC instructions will not be cached.
1: Data requested by MOVC instructions will be loaded into cache memory.

Bit 0: CHBLKW: Block Write Enable.
This bit allows block writes to Flash memory from software.
0: Each byte of a software Flash write is written individually.
1: Flash bytes are written in groups of four (for code space writes).

17.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
Step 4. Assign Port pins to desired peripherals using the XBRn registers.
Step 5. Enable the Crossbar (XBARE = ‘1’).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a ‘1’ indicates a digital input, and a ‘0’ indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 17.4 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is ‘0’, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a ‘0’ and for pins configured for analog input mode to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to ‘1’ enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. **Port output drivers are disabled while the Crossbar is disabled.**

Table 18.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Slave Receiver	0010	1	0	X	A slave address was received; ACK requested.	Acknowledge received address.	0	0	1
						Do not acknowledge received address.	0	0	0
		1	1	X	Lost arbitration as master; slave address received; ACK requested.	Acknowledge received address.	0	0	1
						Do not acknowledge received address.	0	0	0
						Reschedule failed transfer; do not acknowledge received address.	1	0	0
		0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X
						Reschedule failed transfer.	1	0	X
	0001	1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0
		0	0	X	A STOP was detected while an addressed slave receiver.	No action required (transfer complete).	0	0	X
		0	1	X	Lost arbitration due to a detected STOP.	Abort transfer.	0	0	X
						Reschedule failed transfer.	1	0	X
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1
						Do not acknowledge received byte.	0	0	0
		1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0
						Reschedule failed transfer.	1	0	0

19.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 19.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

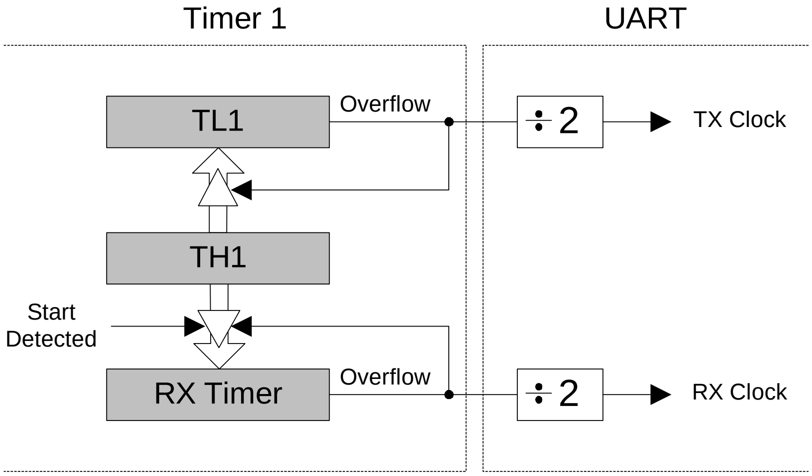

Figure 19.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “21.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 247). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, the external oscillator clock / 8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 19.1-A and Equation 19.1-B.

A) $$\mathrm{UartBaudRate} = \frac{1}{2} \times \mathrm{T1\_Overflow\_Rate}$$

B) $$\mathrm{T1\_Overflow\_Rate} = \frac{\mathrm{T1_{CLK}}}{256 - \mathrm{TH1}}$$

Equation 19.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “21. Timers” on page 245. A quick reference for typical baud rates and system clock frequencies is given in Table 19.1 through Table 19.6. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

Table 19.5. Timer Settings for Standard Baud Rates Using an External 11.0592 MHz Oscillator

	Frequency: 11.0592 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK and Timer Clock from External Osc.	230400	0.00%	48	SYSCLK	XX[2]	1	0xE8
	115200	0.00%	96	SYSCLK	XX	1	0xD0
	57600	0.00%	192	SYSCLK	XX	1	0xA0
	28800	0.00%	384	SYSCLK	XX	1	0x40
	14400	0.00%	768	SYSCLK/12	00	0	0xE0
	9600	0.00%	1152	SYSCLK/12	00	0	0xD0
	2400	0.00%	4608	SYSCLK/12	00	0	0x40
	1200	0.00%	9216	SYSCLK/48	10	0	0xA0
SYSCLK from Internal Osc., Timer Clock from External Osc.	230400	0.00%	48	EXTCLK/8	11	0	0xFD
	115200	0.00%	96	EXTCLK/8	11	0	0xFA
	57600	0.00%	192	EXTCLK/8	11	0	0xF4
	28800	0.00%	384	EXTCLK/8	11	0	0xE8
	14400	0.00%	768	EXTCLK/8	11	0	0xD0
	9600	0.00%	1152	EXTCLK/8	11	0	0xB8

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don’t care.

20.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 20.1. SPI0CFG: SPI0 Configuration

SFR Page: **all pages**
SFR Address: **0xA1**

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic ‘1’ when a SPI transfer is in progress (Master or slave Mode).
Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.
Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*
Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.
Bit 3: SLVSEL: Slave Selected Flag (read only).
This bit is set to logic ‘1’ whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic ‘0’ when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic ‘1’ when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic ‘0’ when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.
Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic ‘1’ when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic ‘0’.
NOTE: RXBMT = 1 when in Master Mode.

***Note:** In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 20.1 for timing parameters.

SFR Definition 21.2. TMOD: Timer Mode

SFR Page: all pages
SFR Address: 0x89

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 10.7).

Bit 6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits 5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit 3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 10.7).

Bit 2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits 1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

SFR Definition 21.8. TMR2CN: Timer 2 Control

SFR Page: **all pages**
SFR Address: **0xC8**

(bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	–	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF2H: Timer 2 High Byte Overflow Flag.
Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. TF2H is not automatically cleared by hardware and must be cleared by software.

Bit 6: TF2L: Timer 2 Low Byte Overflow Flag.
Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF2LEN is set and Timer 2 interrupts are enabled. TF2L will set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.

Bit 5: TF2LEN: Timer 2 Low Byte Interrupt Enable.
This bit enables/disables Timer 2 Low Byte interrupts. If TF2LEN is set and Timer 2 interrupts are enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
0: Timer 2 Low Byte interrupts disabled.
1: Timer 2 Low Byte interrupts enabled.

Bit 4: TF2CEN: Timer 2 Low-Frequency Oscillator Capture Enable.
This bit enables/disables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL. See Section “16. Oscillators” on page 168 for more details.
0: Timer 2 Low-Frequency Oscillator Capture disabled.
1: Timer 2 Low-Frequency Oscillator Capture enabled.

Bit 3: T2SPLIT: Timer 2 Split Mode Enable.
When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.
0: Timer 2 operates in 16-bit auto-reload mode.
1: Timer 2 operates as two 8-bit auto-reload timers.

Bit 2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in this mode.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: T2XCLK: Timer 2 External Clock Select.
This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 2 external clock selection is the system clock divided by 12.
1: Timer 2 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

22.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 22.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic '1'. See Figure 22.3 for details on the PCA interrupt configuration.

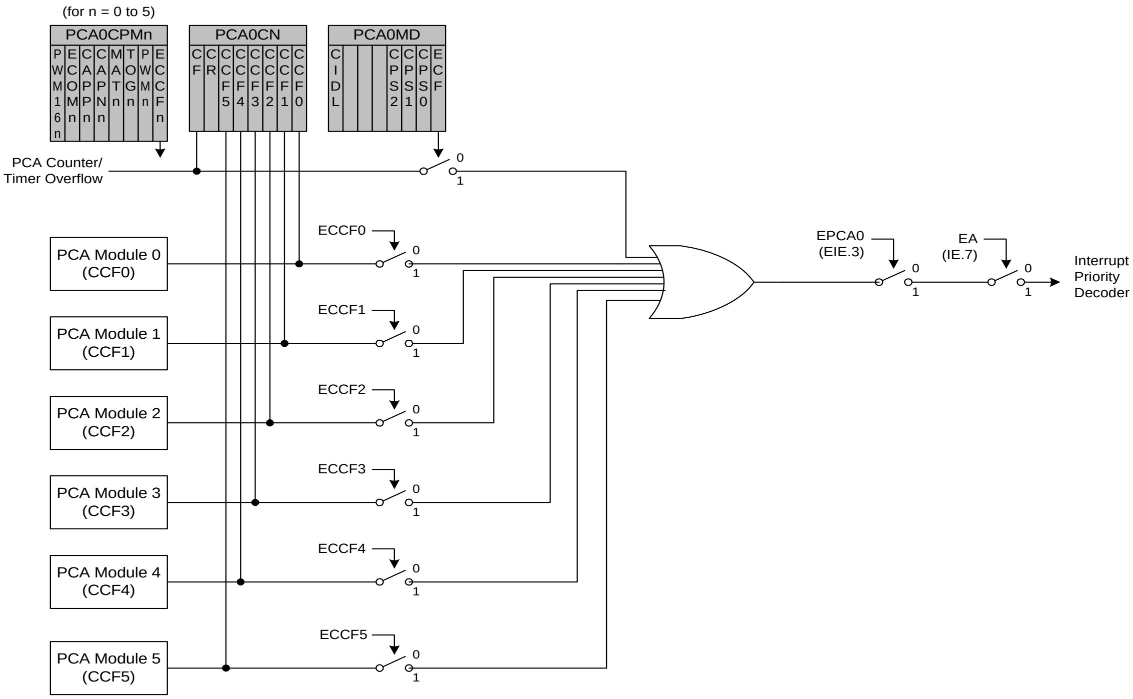

Figure 22.3. PCA Interrupt Block Diagram

C2 Register Definition 24.3. REVID: C2 Revision ID

C2 Address: 0x01

Reset Value
00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit revision ID: 0x00 (Revision A), 0x01 (Revision B), or 0x02 (Revision C).

C2 Register Definition 24.4. FPCTL: C2 Flash Programming Control

C2 Address: 0x02

Reset Value
00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0 FPCTL: Flash Programming Control Register.
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 24.5. FPDAT: C2 Flash Programming Data

C2 Address: 0xB4

Reset Value
00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: FPDAT: C2 Flash Programming Data Register.
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

Code	Command
0x06	Flash Block Read
0x07	Flash Block Write
0x08	Flash Page Erase
0x03	Device Erase