

Welcome to [E-XFL.COM](https://www.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	50MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	17
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f393-a-gmr

C8051F39x/37x

17. Memory Organization	93
17.1. Program Memory	94
17.1.1. MOVX Instruction and Program Memory	94
17.2. Data Memory	94
17.2.1. Internal RAM	94
17.2.1.1. General Purpose Registers	95
17.2.1.2. Bit Addressable Locations	95
17.2.1.3. Stack	95
17.2.2. External RAM	95
18. Device ID Registers	97
19. Special Function Registers	101
19.1. SFR Paging	101
19.2. Interrupts and Automatic SFR Paging	101
19.3. SFR Page Stack Example	103
20. Interrupts	117
20.1. MCU Interrupt Sources and Vectors	118
20.1.1. Interrupt Priorities	118
20.1.2. Interrupt Latency	118
20.2. Interrupt Register Descriptions	120
20.3. External Interrupts INT0 and INT1	128
21. Flash Memory	131
21.1. Programming The Flash Memory	131
21.1.1. Flash Lock and Key Functions	131
21.1.2. Flash Erase Procedure	131
21.1.3. Flash Write Procedure	132
21.2. Non-volatile Data Storage	132
21.3. Security Options	133
21.4. Flash Write and Erase Guidelines	135
21.4.1. V_{DD} Maintenance and the V_{DD} Monitor	135
21.4.2. PSWE Maintenance	135
21.4.3. System Clock	136
22. EEPROM (C8051F37x)	140
22.1. EEPROM Communication Protocol	140
22.1.1. Slave Address Byte	141
22.1.2. Acknowledgement (ACK)	141
22.1.3. Not-Acknowledgement (NACK)	141
22.1.4. Reset	141
22.2. Write Operation	142
22.3. Read Operation	143
22.3.1. Current Address Read	143
22.3.2. Selective Address Read	145
23. Cyclic Redundancy Check Unit (CRC0)	147
23.1. CRC Algorithm	147
23.2. Preparing for a CRC Calculation	149
23.3. Performing a CRC Calculation	149

List of Figures

Figure 1.1. C8051F392/3/6/7/8/9 Block Diagram	18
Figure 1.2. C8051F390/1/4/5 Block Diagram	18
Figure 1.3. C8051F370/1/4/5 Block Diagram	19
Figure 4.1. C8051F392/3/6/7/8/9 QFN-20 Pinout Diagram (Top View)	25
Figure 4.2. C8051F390/1/4/5 Pinout Diagram (Top View)	26
Figure 4.3. C8051F370/1/4/5 Pinout Diagram (Top View)	27
Figure 5.1. QFN-20 Package Drawing	28
Figure 5.2. QFN-20 Recommended PCB Land Pattern	29
Figure 6.1. QFN-24 Package Drawing	30
Figure 6.2. QFN-24 Recommended PCB Land Pattern	31
Figure 7.1. Normal Mode Digital Supply Current vs. Frequency	45
Figure 7.2. Idle Mode Digital Supply Current vs. Frequency	45
Figure 7.3. Precision Temperature Sensor Error vs. Temperature	46
Figure 9.1. ADC0 Functional Block Diagram	50
Figure 9.2. 10-Bit ADC Track and Conversion Example Timing	53
Figure 9.3. ADC0 Equivalent Input Circuits	54
Figure 9.4. ADC Window Compare Example: Right-Justified, Single-Ended Data ..	60
Figure 9.5. ADC Window Compare Example: Left-Justified, Single-Ended Data ...	60
Figure 9.6. ADC0 Multiplexer Block Diagram	61
Figure 10.1. Temperature Sensor Transfer Function	64
Figure 10.2. Temperature Sensor Error with 1-Point Calibration at 0 °C	65
Figure 11.1. IDA0 Functional Block Diagram	66
Figure 11.2. IDA1 Functional Block Diagram	67
Figure 11.3. IDA0 Data Word Mapping	68
Figure 12.1. Voltage Reference Functional Block Diagram	73
Figure 14.1. Comparator0 Functional Block Diagram	76
Figure 14.2. Comparator Hysteresis Plot	77
Figure 14.3. Comparator Input Multiplexer Block Diagram	80
Figure 15.1. CIP-51 Block Diagram	82
Figure 17.1. C8051F39x/37x Memory Map	93
Figure 17.2. Flash Program Memory Map	94
Figure 19.1. SFR Page Stack	102
Figure 19.2. SFR Page Stack While Using SFR Page 0x0F To Access TS0CN ..	103
Figure 19.3. SFR Page Stack After SPI0 Interrupt Occurs	104
Figure 19.4. SFR Page Stack Upon PCA Interrupt Occurring During a SPI0 ISR	105
Figure 19.5. SFR Page Stack Upon Return from PCA0 Interrupt	106
Figure 19.6. SFR Page Stack Upon Return From SPI0 Interrupt	107
Figure 21.1. Security Byte Decoding	133
Figure 22.1. Slave Address Byte Definition	141
Figure 22.2. Write Operation (Single Byte)	142
Figure 22.3. Write Operation (Multiple Bytes)	142
Figure 22.4. Current Address Read Operation (Single Byte)	143
Figure 22.5. Current Address Read Operation (Multiple Bytes)	144

Figure 31.7. Timer 3 16-Bit Mode Block Diagram	259
Figure 31.8. Timer 3 8-Bit Mode Block Diagram	260
Figure 31.9. Timer 3 Low-Frequency Oscillation Capture Mode Block Diagram ...	261
Figure 31.10. Timer 4 16-Bit Mode Block Diagram	265
Figure 31.11. Timer 4 8-Bit Mode Block Diagram	266
Figure 31.12. Timer 5 16-Bit Mode Block Diagram	270
Figure 31.13. Timer 5 8-Bit Mode Block Diagram	271
Figure 32.1. PCA Block Diagram	275
Figure 32.2. PCA Counter/Timer Block Diagram	276
Figure 32.3. PCA Interrupt Block Diagram	277
Figure 32.4. PCA Capture Mode Diagram	279
Figure 32.5. PCA Software Timer Mode Diagram	280
Figure 32.6. PCA High-Speed Output Mode Diagram	281
Figure 32.7. PCA Frequency Output Mode	282
Figure 32.8. PCA 8-Bit PWM Mode Diagram	283
Figure 32.9. PCA 9, 10 and 11-Bit PWM Mode Diagram	284
Figure 32.10. PCA 16-Bit PWM Mode	285
Figure 32.11. PCA Module 2 with Watchdog Timer Enabled	286
Figure 32.12. Comparator Clear Function Diagram	288
Figure 32.13. CEXn with CPCEn = 1, CPCPOL = 0	288
Figure 32.14. CEXn with CPCEn = 1, CPCPOL = 1	289
Figure 32.15. CEXn with CPCEn = 1, CPCPOL = 0	289
Figure 32.16. CEXn with CPCEn = 1, CPCPOL = 1	289
Figure 33.1. Typical C2 Pin Sharing	300

10. Temperature Sensor (C8051F390/2/4/6/8 and C8051F370/4 Only)

A fully C8051F33x-compatible temperature sensor is included on the C8051F390/2/4/6/8 and C8051F370/4 and accessed via the ADC multiplexer in single-ended configuration. For the self-contained precision temperature sensor, refer to Section 8.

To use the ADC to measure the temperature sensor, the ADC mux channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 10.1. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 12.1. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 7.11 for the slope and offset parameters of the temperature sensor.

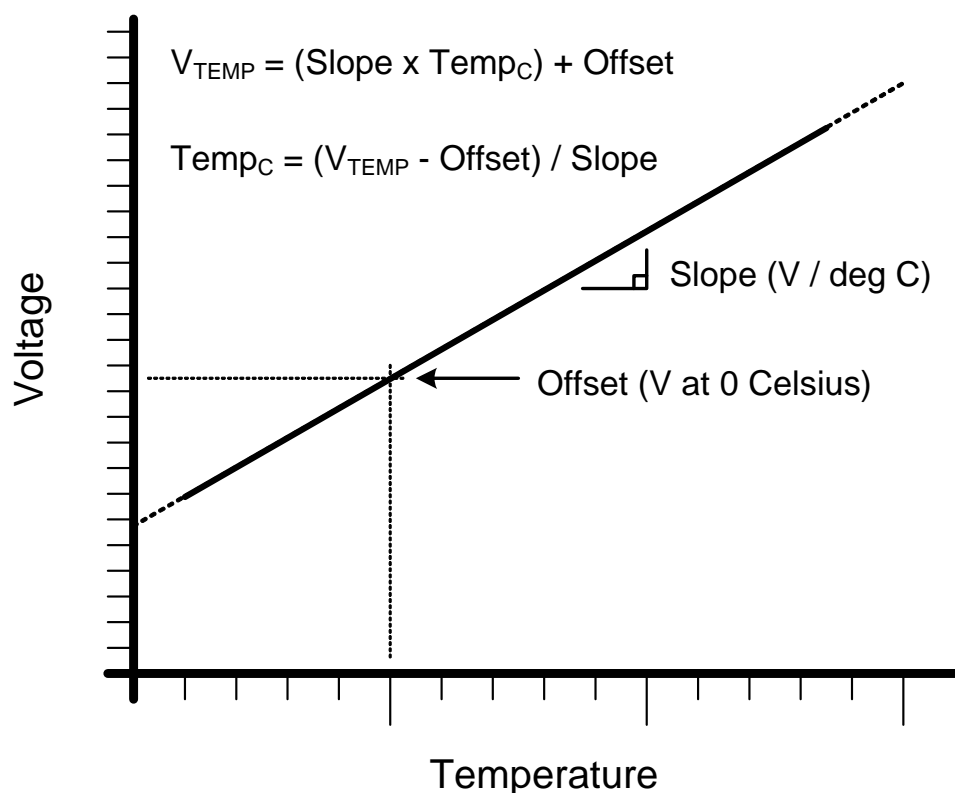


Figure 10.1. Temperature Sensor Transfer Function

14.1. Comparator Multiplexer

C8051F39x/37x devices include an analog input multiplexer to connect Port I/O pins to the comparator inputs. The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 14.3). The CMX0P1–CMX0P0 bits select the Comparator0 positive input; the CMX0N1–CMX0N0 bits select the Comparator0 negative input. **Important Note About Comparator Inputs:** The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “27.6. Special Function Registers for Accessing and Configuring Port I/O” on page 185).

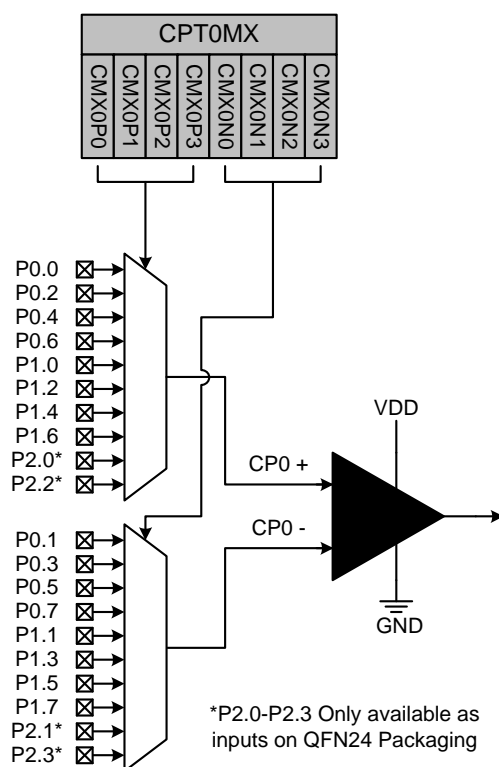


Figure 14.3. Comparator Input Multiplexer Block Diagram

C8051F39x/37x

SFR Definition 18.6. SN0: Serial Number Byte 0

Bit	7	6	5	4	3	2	1	0
Name	SN0							
Type	R							
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xAB; SFR Page = F

Bit	Name	Function
7:0	SN0	Serial Number Byte 0. This read-only register returns the LSB (byte 0) of the serial number.

SFR Definition 19.2. SFRPGCN: SFR Page Control

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

SFR Address = 0xCF; SFR Page = All Pages

Bit	Name	Function
7	Reserved	Must Write 0b
6:4	SFRPGIDX[2:0]	SFR Page Stack Index. This field can be used to access the SFRPAGE values stored in the SFR page stack. It selects which level of the stack is accessible when reading the SFRSTACK register. 000: SFRSTACK contains the value of SFRPAGE, the first/top byte of the SFR page stack 001: SFRSTACK contains the value of the second byte of the SFR page stack 010: SFRSTACK contains the value of the third byte of the SFR page stack 011: SFRSTACK contains the value of the forth byte of the SFR page stack 100: SFRSTACK contains the value of the fifth/bottom byte of the SFR page stack 101: Invalid index 11x: Invalid index
3:1	Reserved	Must Write 000b
0	SFRPGEN	SFR Automatic Page Control Enable. This bit is used to enable automatic page switching on ISR entry/exit. When set to 1, the current SFRPAGE value will be pushed onto the SFR page stack, and SFRPAGE will be set to the page corresponding to the flag which generated the interrupt; upon ISR exit, hardware will pop the value from the SFR page stack and restore SFRPAGE. 0: Disable automatic SFR paging. 1: Enable automatic SFR paging.

20.1. MCU Interrupt Sources and Vectors

The C8051F39x/37x MCUs support 18 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 20.2. Refer to the datasheet 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).

20.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of four priority levels. This differs from the traditional two priority levels on the 8051 core. However, the implementation of the extra levels is backwards-compatible with legacy 8051 code.

An interrupt service routine can be preempted by any interrupt of higher priority. Interrupts at the highest priority level cannot be preempted. Each interrupt has two associated priority bits which are used to configure the priority level. For backwards compatibility, the bits are spread across two different registers. The LSBs of the priority setting are stored in the IP, EIP1 and EIP2 registers, while the MSBs are stored in the IPH, EIP1H and EIP2H registers. Priority levels according to the MSB and LSB are decoded in Table 20.1. The lowest priority setting is the default for all interrupts. If two or more interrupts are recognized simultaneously, the interrupt with the highest priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 20.2. If legacy 8051 operation is desired, the bits of the “High” priority registers (IPH, EIP1H and EIP2H) should all be configured to 0 (this is the reset value of these registers).

Priority MSB (from IPH, EIP1H or EIP2H)	Priority LSB (from IP, EIP1 or EIP2)	Priority Level
0	0	Priority 0 (lowest priority, default)
0	1	Priority 1
1	0	Priority 2
1	1	Priority 3 (highest priority)

Table 20.1. Configurable Interrupt Priority Decoding

20.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction. If more than one interrupt is pending when the CPU exits an ISR, the CPU will service the next highest priority interrupt that is pending.

21.1.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the 512-byte Flash page containing the target location, as described in Section 21.1.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
8. Clear the PSWE bit.

Steps 5–7 must be repeated for each byte to be written. After Flash writes are complete, PSWE should be cleared so that MOVX instructions do not target program memory.

21.2. 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 using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

21.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to '1' before software can modify the Flash memory; both PSWE and PSEE must be set to '1' before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located in Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. See Section "17. Memory Organization" on page 93 for the location of the security byte. The Flash security mechanism allows the user to lock n 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where n is the 1's complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are '1') and locked when any other Flash pages are locked (any bit of the Lock Byte is '0').** An example is shown in Figure 21.1.

Security Lock Byte:	11111101b
1s Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)

Figure 21.1. Security Byte Decoding

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 21.1 summarizes the Flash security features of the C8051F39x/37x devices.

Table 21.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	Flash Error Reset	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	Permitted	N/A

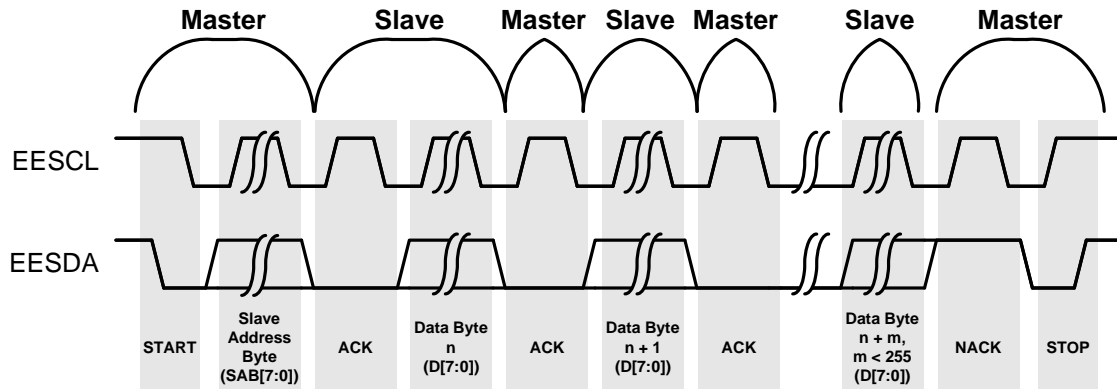


Figure 22.5. Current Address Read Operation (Multiple Bytes)

24.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Section “7. Electrical Characteristics” on page 32 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

24.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 μs , the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read ‘1’, signifying the MCD as the reset source; otherwise, this bit reads ‘0’. Writing a ‘1’ to the MCDRSF bit enables the Missing Clock Detector; writing a ‘0’ disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

24.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a ‘1’ to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read ‘1’ signifying Comparator0 as the reset source; otherwise, this bit reads ‘0’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

24.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section “32.4. Watchdog Timer Mode” on page 286; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

24.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to ‘1’ and a MOVX write operation targets an address above address 0x3DFF.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above address 0x3DFF.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x3DFF.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “21.3. Security Options” on page 133).

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

24.8. Software Reset

Software may force a reset by writing a ‘1’ to the SWRSF bit (RSTSRC.4). The SWRSF bit will read ‘1’ following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

C8051F39x/37x

For example, a tuning-fork crystal of 32 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 26.1, Option 1. With a stray capacitance of 3 pF per pin (6 pF total), the 13 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 26.2.

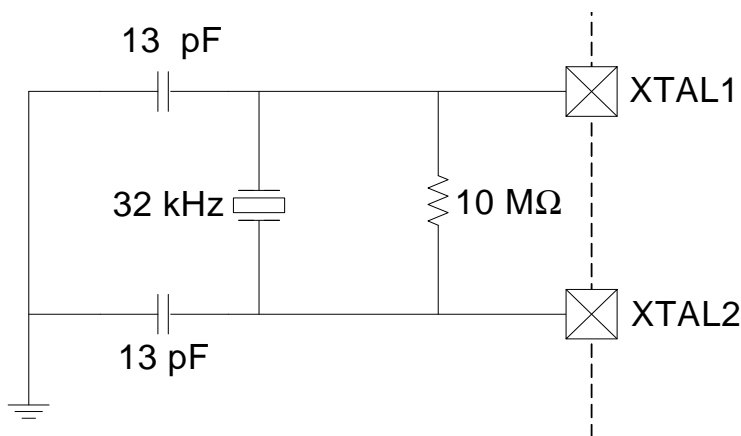


Figure 26.2. External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency (see SFR Definition 26.5).

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock is valid and running. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is:

1. Configure XTAL1 and XTAL2 for analog I/O.
2. Disable the XTAL1 and XTAL2 digital output drivers by writing 1s to the appropriate bits in the Port Latch register.
3. Configure and enable the external oscillator.
4. Wait at least 1 ms.
5. Poll for $XTLVLD \geq 1$.
6. Switch the system clock to the external oscillator.

C8051F39x/37x

SFR Definition 27.11. P1: Port 1

Bit	7	6	5	4	3	2	1	0
Name	P1[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x90; SFR Page = All Pages; Bit Addressable

Bit	Name	Description	Write	Read
7:0	P1[7:0]	Port 1 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: P1.n Port pin is logic LOW. 1: P1.n Port pin is logic HIGH.

SFR Definition 27.12. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF2; SFR Page = All Pages

Bit	Name	Function
7:0	P1MDIN[7:0]	Analog Configuration Bits for P1.7–P1.0 (respectively). Port pins configured for analog mode have their weak pul-lup, digital driver, and digital receiver disabled. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

SFR Definition 28.1. SMB0CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB0	INH0	BUSY0	EXTHOLD0	SMB0TOE	SMB0FTE	SMB0CS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0

Bit	Name	Function
7	ENSMB0	SMBus0 Enable. This bit enables the SMBus0 interface when set to 1. When enabled, the interface constantly monitors the SDA0 and SCL0 pins.
6	INH0	SMBus0 Slave Inhibit. When this bit is set to logic 1, the SMBus0 does not generate an interrupt when slave events occur. This effectively removes the SMBus0 slave from the bus. Master Mode interrupts are not affected.
5	BUSY0	SMBus0 Busy Indicator. This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD0	SMBus0 Setup and Hold Time Extension Enable. This bit controls the SDA0 setup and hold times according to Table 28.2. 0: SDA0 Extended Setup and Hold Times disabled. 1: SDA0 Extended Setup and Hold Times enabled.
3	SMB0TOE	SMBus0 SCL Timeout Detection Enable. This bit enables SCL low timeout detection. If set to logic 1, the SMBus0 forces Timer 3 to reload while SCL0 is high and allows Timer 3 to count when SCL0 goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL0 is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus0 communication.
2	SMB0FTE	SMBus0 Free Timeout Detection Enable. When this bit is set to logic 1, the bus will be considered free if SCL0 and SDA0 remain high for more than 10 SMBus clock source periods.
1:0	SMB0CS[1:0]	SMBus0 Clock Source Selection. These two bits select the SMBus0 clock source, which is used to generate the SMBus0 bit rate. The selected device should be configured according to Equation 28.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

SFR Definition 28.7. SMB0ADM: SMBus0 Slave Address Mask

Bit	7	6	5	4	3	2	1	0
Name	SLVM0[6:0]							EHACK0
Type	R/W							R/W
Reset	1	1	1	1	1	1	1	0

SFR Address = 0xE7; SFR Page = 0

Bit	Name	Function
7:1	SLVM0[6:0]	SMBus0 Slave Address Mask. Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM0[6:0] enables comparisons with the corresponding bit in SLV0[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK0	Hardware Acknowledge Enable. Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

SFR Definition 29.2. SBUF0: Serial (UART0) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x99; SFR Page = All Pages

Bit	Name	Function
7:0	SBUF0[7:0]	Serial Data Buffer Bits 7–0 (MSB–LSB). This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 30.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

30.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
- The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPI0 modes.
- The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
- The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

30.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 30.5. For slave mode, the clock and data relationships are shown in Figure 30.6 and Figure 30.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 30.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-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.

SFR Definition 31.2. CKCON1: Clock Control 1

Bit	7	6	5	4	3	2	1	0
Name					T5MH	T5ML	T4MH	T4ML
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = All Pages

Bit	Name	Function
7:4	Unused	Read = 0000b; Write = don't care
3	T5MH	Timer 5 High Byte Clock Select. Selects the clock supplied to the Timer 5 high byte (split 8-bit timer mode only). 0: Timer 5 high byte uses the clock defined by the T5XCLK bit in TMR5CN. 1: Timer 5 high byte uses the system clock.
2	T5ML	Timer 5 Low Byte Clock Select. Selects the clock supplied to Timer 5. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 5 low byte uses the clock defined by the T5XCLK bit in TMR5CN. 1: Timer 5 low byte uses the system clock.
1	T4MH	Timer 4 High Byte Clock Select. Selects the clock supplied to the Timer 4 high byte (split 8-bit timer mode only). 0: Timer 4 high byte uses the clock defined by the T4XCLK bit in TMR4CN. 1: Timer 4 high byte uses the system clock.
0	T4ML	Timer 4 Low Byte Clock Select. Selects the clock supplied to Timer 4. If Timer 4 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 4 low byte uses the clock defined by the T4XCLK bit in TMR4CN. 1: Timer 4 low byte uses the system clock.

32.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 32.4.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

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

Equation 32.4. 16-Bit PWM Duty Cycle

Using Equation 32.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

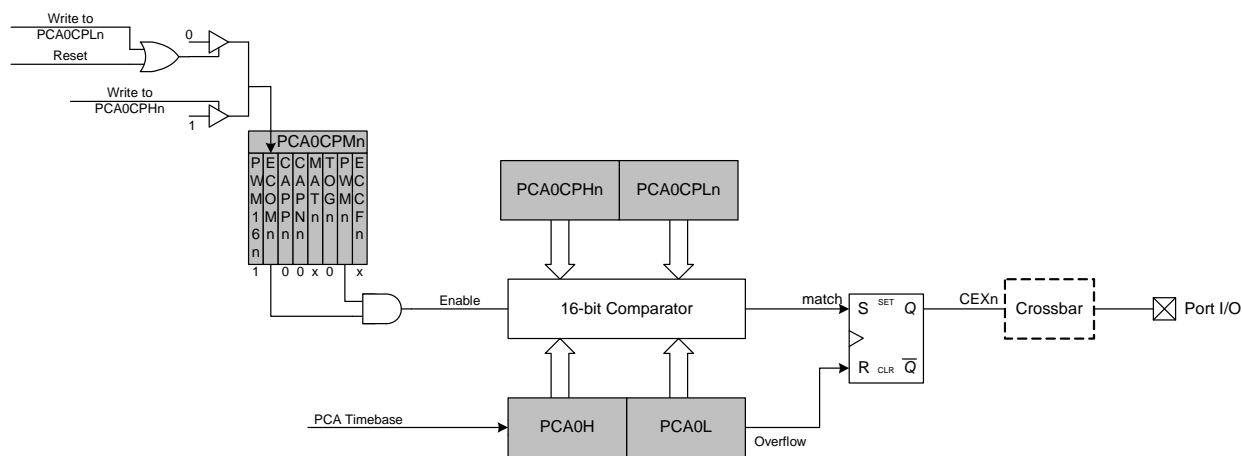


Figure 32.10. PCA 16-Bit PWM Mode