

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	CIP-51 8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, Cap Sense, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f991-c-gm

C8051F99x-C8051F98x

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F99x-C8051F98x family utilizes Silicon Labs' proprietary CIP-51 microcontroller core. 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 CIP-51 core offers all the peripherals included with a standard 8052.

1.1.2. Improved Throughput

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 with a maximum system clock of 12-to-24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with only four instructions taking more than four system clock cycles.

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

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS.

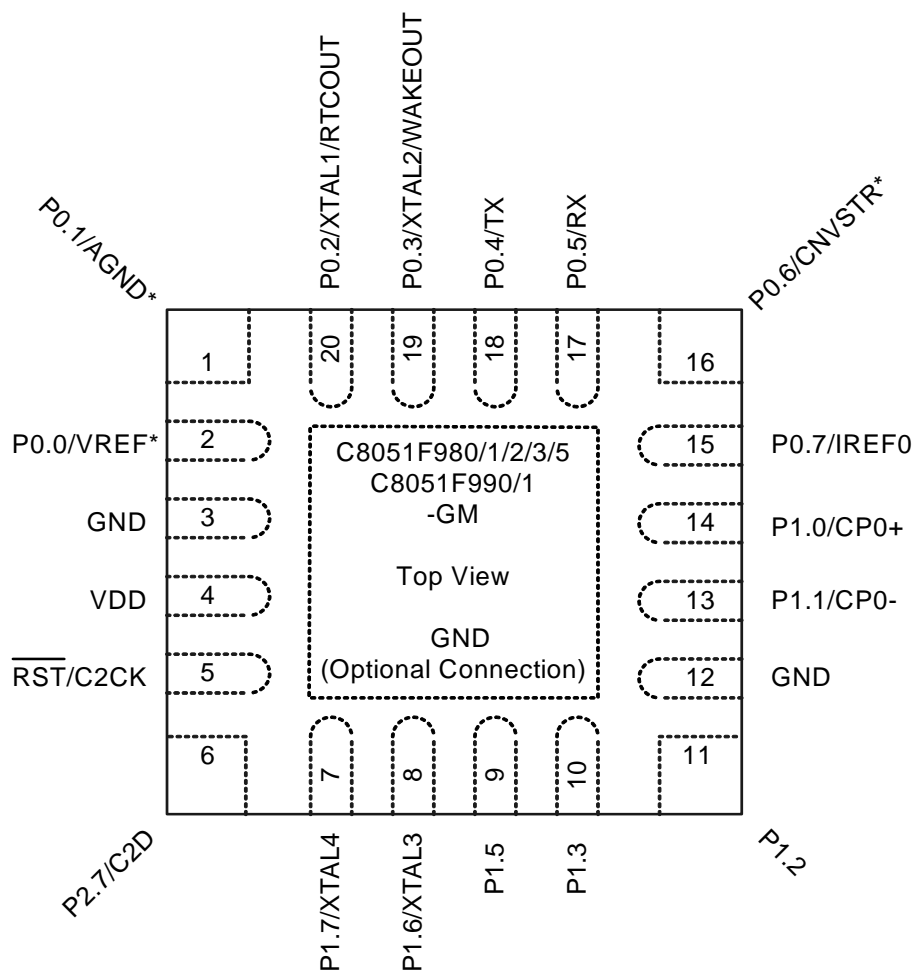
1.1.3. Additional Features

The C8051F99x-C8051F98x SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

The extended interrupt handler provides multiple interrupt sources into the CIP-51 allowing numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} monitor (forces reset when power supply voltage drops below safe levels), a Watchdog Timer, a Missing Clock Detector, SmaRTClock oscillator fail or alarm, a voltage level detection from Comparator0, a forced software reset, an external reset pin, and an illegal Flash access protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash error may be disabled by the user in software. The WDT may be permanently disabled in software after a power-on reset during MCU initialization.

The internal oscillator is factory calibrated to 24.5 MHz and is accurate to $\pm 2\%$ over the full temperature and supply range. The internal oscillator period can also be adjusted by user firmware. An additional 20 MHz low power oscillator is also available which facilitates low-power operation. An external oscillator drive circuit is included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock. If desired, the system clock source may be switched on-the-fly between both internal and external oscillator circuits. An external oscillator can also be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) source, while periodically switching to the fast (up to 25 MHz) internal oscillator as needed.



***Note:** Signal only available on 'F980, 'F982 and 'F990 devices.

Figure 3.1. QFN-20 Pinout Diagram (Top View)

4.2. Electrical Characteristics

Table 4.2. Global Electrical Characteristics

–40 to +85 °C, 25 MHz system clock unless otherwise specified. See "AN358: Optimizing Low Power Operation of the 'F9xx" for details on how to achieve the supply current specifications listed in this table.

Parameter	Conditions	Min	Typ	Max	Units
Supply Voltage (V_{DD})		1.8	2.4	3.6	V
Minimum RAM Data Retention Voltage ¹	not in sleep mode in sleep mode	— —	1.4 0.3	— 0.45	V
SYSCLK (System Clock) ²		0	—	25	MHz
T_{SYSH} (SYSCLK High Time)		18	—	—	ns
T_{SYSL} (SYSCLK Low Time)		18	—	—	ns
Specified Operating Temperature Range		–40	—	+85	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD} ^{3, 4, 5}	$V_{DD} = 1.8\text{--}3.6\text{ V}$, $F = 24.5\text{ MHz}$ (includes precision oscillator current)	—	3.6	4.5	mA
	$V_{DD} = 1.8\text{--}3.6\text{ V}$, $F = 20\text{ MHz}$ (includes low power oscillator current)	—	3.1	—	mA
	$V_{DD} = 1.8\text{ V}$, $F = 1\text{ MHz}$ $V_{DD} = 3.6\text{ V}$, $F = 1\text{ MHz}$ (includes external oscillator/GPIO current)	— —	225 290	— —	μA μA
	$V_{DD} = 1.8\text{--}3.6\text{ V}$, $F = 32.768\text{ kHz}$ (includes SmaRTClock oscillator current)	—	84	—	μA
I_{DD} Frequency Sensitivity ^{1, 3, 5}	$V_{DD} = 1.8\text{--}3.6\text{ V}$, $T = 25\text{ °C}$, $F < 14\text{ MHz}$ (Flash oneshot active, see Section 14.6)	—	174	—	$\mu\text{A/MHz}$
	$V_{DD} = 1.8\text{--}3.6\text{ V}$, $T = 25\text{ °C}$, $F > 14\text{ MHz}$ (Flash oneshot bypassed, see Section 14.6)	—	88	—	$\mu\text{A/MHz}$

C8051F99x-C8051F98x

5.7. ADC0 Analog Multiplexer

ADC0 on C8051F99x-C8051F98x has an analog multiplexer, referred to as AMUX0.

AMUX0 selects the positive inputs to the single-ended ADC0. Any of the following may be selected as the positive input: Port I/O pins, the on-chip temperature sensor, Regulated Digital Supply Voltage (Output of VREG0), VDD Supply, or the positive input may be connected to GND. The ADC0 input channels are selected in the ADC0MX register described in SFR Definition 5.12.

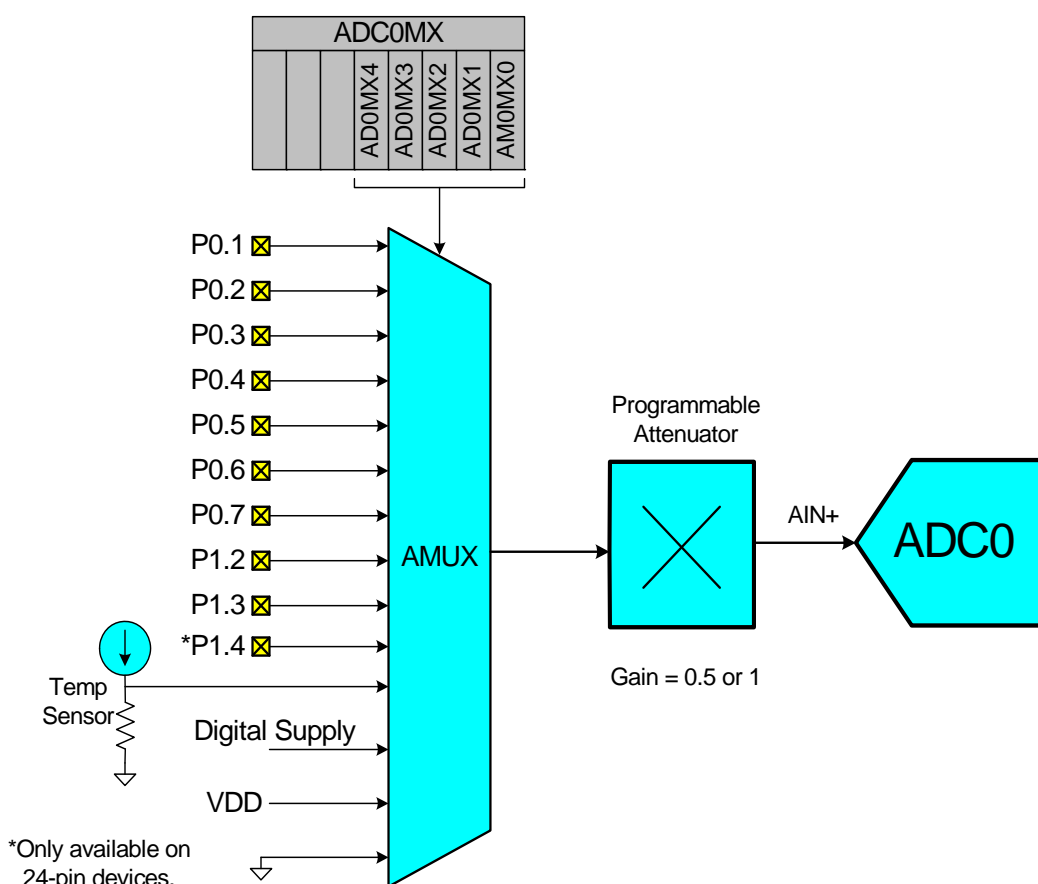


Figure 5.7. ADC0 Multiplexer Block Diagram

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN and disable the digital driver (PnMDOUT = 0 and Port Latch = 1). To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section “21. Port Input/Output” on page 215 for more Port I/O configuration details.

8.12. CS0 Pin Monitor

The CS0 module provides accurate conversions in all operating modes of the CPU, peripherals and I/O ports. Pin monitoring circuits are provided to improve interference immunity from high-current output pin switching. The CS0 Pin Monitor register (CS0PM, SFR Definition 8.14) controls the operation of these pin monitors.

Conversions in the CS0 module are immune to any change on digital inputs and immune to most output switching. Even high-speed serial data transmission will not affect CS0 operation as long as the output load is limited. Output changes that switch large loads such as LEDs and heavily-loaded communications lines can affect conversion accuracy. For this reason, the CS0 module includes pin monitoring circuits that will, if enabled, automatically adjust conversion timing if necessary to eliminate any effect from high-current output pin switching.

The pin monitor enable bit should be set for any output signal that is expected to drive a large load.

Example: The SMBus in a system is heavily loaded with multiple slaves and a long PCB route. Set the SMBus pin monitor enable, SMBPM = 1.

Example: Timer2 controls an LED on Port 1, pin 3 to provide variable dimming. Set the Port SFR write monitor enable, PIOPM = 1.

Example: The SPI bus is used to communicate to a nearby host. The pin monitor is not needed because the output is not heavily loaded, SPIPM remains = 0, the default reset state.

Pin monitors should not be enabled unless they are required. The pin monitor works by repeating any portion of a conversion that may have been corrupted by a change on an output pin. Setting pin monitor enables bits will slow CS0 conversions.

The frequency of CS0 retry operations can be limited by setting the CSPMMD bits. In the default (reset) state, all converter retry requests will be performed. This is the recommended setting for all applications. The number of retries per conversion can be limited to either two or four retries by changing CSPMMD. Limiting the number of retries per conversion ensures that even in circumstances where extremely frequent high-power output switching occurs, conversions will be completed, though there may be some loss of accuracy due to switching noise.

Activity of the pin monitor circuit can be detected by reading the Pin Monitor Event bit, CS0PME, in register CS0CN. This bit will be set if any CS0 converter retries have occurred. It remains set until cleared by software or a device reset.

8.13. Adjusting CS0 For Special Situations

There are several configuration options in the CS0 module designed to modify the operation of the circuit and address special situations. In particular, any circuit with more than 500 ohms of series impedance between the sensor and the device pin may require adjustments for optimal performance. Typical applications which may require adjustments include:

- Touch panel sensors fabricated using a resistive conductor such as indium-tin-oxide (ITO).
- Circuits using a high-value series resistor to isolate the sensor element for high ESD protection.

Most systems will require no fine tuning, and the default settings for CS0DT, CS0DR, CS0IA, CS0RP and CS0LP should be used.

13. Interrupt Handler

The C8051F99x-C8051F98x microcontroller family includes an extended interrupt system supporting multiple interrupt sources and two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Refer to Table 13.1, “Interrupt Summary,” on page 140 for a detailed listing of all interrupt sources supported by the device. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR or an indirect register. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1. If both global interrupts and the specific interrupt source is enabled, a CPU interrupt request is generated when the interrupt-pending flag is set.

As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Some interrupt-pending flags are automatically cleared by hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

13.1. Enabling Interrupt Sources

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1.

13.2. MCU Interrupt Sources and Vectors

The CPU services interrupts by generating an LCALL to a predetermined address (the interrupt vector address) to begin execution of an interrupt service routine (ISR). The interrupt vector addresses associated with each interrupt source are listed in Table 13.1 on page 140. Software should ensure that the interrupt vector for each enabled interrupt source contains a valid interrupt service routine.

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.

SFR Definition 13.7. IT01CF: $\overline{\text{INT0}}/\overline{\text{INT1}}$ Configuration

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

SFR Page = 0x0; SFR Address = 0xE4

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

14.1.2. Flash Erase Procedure

The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire Flash page, perform the following steps:

1. Save current interrupt state and disable interrupts.
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.
6. Using the MOVX instruction, write a data byte to any location within the page to be erased.
7. Clear the PSWE and PSEE bits.
8. Restore previous interrupt state.

Steps 4–6 must be repeated for each 512-byte page to be erased.

Notes:

1. Flash security settings may prevent erasure of some Flash pages, such as the reserved area and the page containing the lock bytes. For a summary of Flash security settings and restrictions affecting Flash erase operations, please see Section “14.3. Security Options” on page 152.
2. 8-bit MOVX instructions cannot be used to erase or write to Flash memory at addresses higher than 0x00FF.

14.1.3. Flash Write Procedure

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.**

The recommended procedure for writing a single byte in Flash is as follows:

1. Save current interrupt state and disable interrupts.
2. Ensure that the Flash byte has been erased (has a value of 0xFF).
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 1024-byte sector.
8. Clear the PSWE bit.
9. Restore previous interrupt state.

Steps 5–7 must be repeated for each byte to be written.

Notes:

1. Flash security settings may prevent writes to some areas of Flash, such as the reserved area. For a summary of Flash security settings and restrictions affecting Flash write operations, please see Section “14.3. Security Options” on page 152.
2. 8-bit MOVX instructions cannot be used to erase or write to Flash memory at addresses higher than 0x00FF.

14.2. 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. MOVX read instructions always target XRAM.

Table 14.1. 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	FEDR	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	FEDR	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	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte—Unlock all pages (if any page is locked)	Only by C2DE	FEDR	FEDR
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR
<ul style="list-style-type: none"> ■ C2DE—C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte) ■ FEDR—Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is 1 after reset) ■ All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset). ■ Locking any Flash page also locks the page containing the Lock Byte. ■ Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase. ■ If user code writes to the Lock Byte, the Lock does not take effect until the next device reset. ■ The scratchpad is locked when all other Flash pages are locked. ■ The scratchpad is erased when a Flash Device Erase command is performed. 			

16. Cyclic Redundancy Check Unit (CRC0)

C8051F99x-C8051F98x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register. CRC0 posts the 16-bit result to an internal register. The internal result register may be accessed indirectly using the CRC0PNT bits and CRC0DAT register, as shown in Figure 16.1. CRC0 also has a bit reverse register for quick data manipulation.

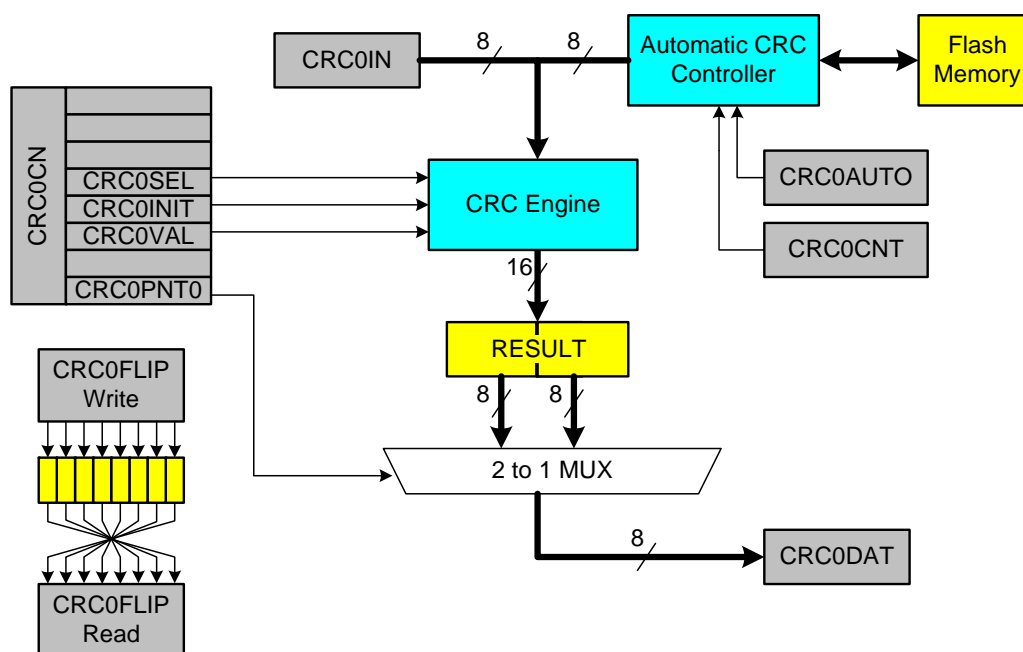


Figure 16.1. CRC0 Block Diagram

16.1. CRC Algorithm

The C8051F99x-C8051F98x CRC unit generates a CRC result equivalent to the following algorithm:

1. XOR the input with the most-significant bits of the current CRC result. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x0000 or 0xFFFF).
- 2a. If the MSB of the CRC result is set, shift the CRC result and XOR the result with the selected polynomial.
- 2b. If the MSB of the CRC result is not set, shift the CRC result.

Repeat Steps 2a/2b for the number of input bits (8). The algorithm is also described in the following example.

SFR Definition 18.1. VDM0CN: VDD Supply Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VDMEN	VDDSTAT	VDDOK		VDDOKIE			
Type	R/W	R	R	R	R/W	R	R	R
Reset	1	Varies	Varies	0	1	0	0	0

SFR Page = 0x0; SFR Address = 0xFF

Bit	Name	Function
7	VDMEN	V_{DD} Supply Monitor Enable. This bit turns the V _{DD} supply monitor circuit on/off. The VDD Supply Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 18.2). 0: V _{DD} Supply Monitor Disabled. 1: V _{DD} Supply Monitor Enabled.
6	VDDSTAT	V_{DD} Supply Status. This bit indicates the current power supply status. 0: V _{DD} is at or below the V _{RST} threshold. 1: V _{DD} is above the V _{RST} threshold.
5	VDDOK	V_{DD} Supply Status (Early Warning). This bit indicates the current V _{DD} power supply status. 0: V _{DD} is at or below the VDD _{WARN} threshold. 1: V _{DD} is above the VDD _{WARN} threshold.
4	Unused	Read = 0b. Write = Don't Care.
3	VDDOKIE	V_{DD} Early Warning Interrupt Enable. Enables the V _{DD} Early Warning Interrupt. 0: V _{DD} Early Warning Interrupt is disabled. 1: V _{DD} Early Warning Interrupt is enabled.
2:0	Unused	Read = 000b. Write = Don't Care.

18.3. 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 Table 4.4 for complete $\overline{\text{RST}}$ pin specifications. The external reset remains functional even when the device is in the low power suspend and sleep modes. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

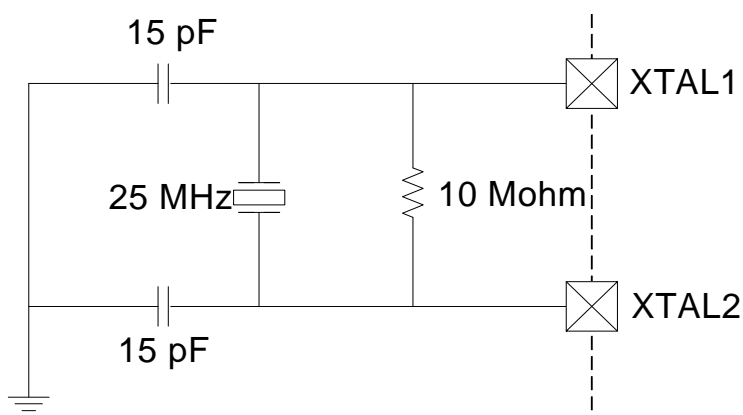


Figure 19.2. 25 MHz External Crystal Example

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. The selection should be based on Table 19.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

Table 19.1. Recommended XFCN Settings for Crystal Mode

XFCN	Crystal Frequency	Bias Current	Typical Supply Current (VDD = 2.4 V)
000	$f \leq 20 \text{ kHz}$	0.5 μA	3.0 μA , $f = 32.768 \text{ kHz}$
001	$20 \text{ kHz} < f \leq 58 \text{ kHz}$	1.5 μA	4.8 μA , $f = 32.768 \text{ kHz}$
010	$58 \text{ kHz} < f \leq 155 \text{ kHz}$	4.8 μA	9.6 μA , $f = 32.768 \text{ kHz}$
011	$155 \text{ kHz} < f \leq 415 \text{ kHz}$	14 μA	28 μA , $f = 400 \text{ kHz}$
100	$415 \text{ kHz} < f \leq 1.1 \text{ MHz}$	40 μA	71 μA , $f = 400 \text{ kHz}$
101	$1.1 \text{ MHz} < f \leq 3.1 \text{ MHz}$	120 μA	193 μA , $f = 400 \text{ kHz}$
110	$3.1 \text{ MHz} < f \leq 8.2 \text{ MHz}$	550 μA	940 μA , $f = 8 \text{ MHz}$
111	$8.2 \text{ MHz} < f \leq 25 \text{ MHz}$	2.6 mA	3.9 mA, $f = 25 \text{ MHz}$

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

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for $\text{XTLVLD} \geq 1$.
4. Switch the system clock to the external oscillator.

C8051F99x-C8051F98x

Internal Register Definition 20.6. RTC0XCF: SmaRTClock Oscillator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AUTOSTP	LOADRDY			LOADCAP			
Type	R/W	R	R	R	R/W			
Reset	0	0	0	0	Varies	Varies	Varies	Varies

SmaRTClock Address = 0x06

Bit	Name	Function
7	AUTOSTP	Automatic Load Capacitance Stepping Enable. Enables/disables automatic load capacitance stepping. 0: Load capacitance stepping disabled. 1: Load capacitance stepping enabled.
6	LOADRDY	Load Capacitance Ready Indicator. Set by hardware when the load capacitance matches the programmed value. 0: Load capacitance is currently stepping. 1: Load capacitance has reached it programmed value.
5:4	Unused	Read = 00b; Write = Don't Care.
3:0	LOADCAP	Load Capacitance Programmed Value. Holds the user's desired value of the load capacitance. See Table 20.2 on page 205.

C8051F99x-C8051F98x

SFR Definition 21.17. P1DRV: Port1 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P1DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x9B

Bit	Name	Function
7:0	P1DRV[7:0]	Drive Strength Configuration Bits for P1.7–P1.0 (respectively). Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P1.n Output has low output drive strength. 1: Corresponding P1.n Output has high output drive strength.

SFR Definition 21.18. P2: Port2

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

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

Bit	Name	Description	Read	Write
7	P2	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: P2.7 Port pin is logic LOW. 1: P2.7 Port pin is logic HIGH.
6:0	Unused	Read = 0000000b; Write = Don't Care.		

22. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compatible with the System Management Bus Specification, version 1.1, and compatible with the I²C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The SMBus peripheral can be fully driven by software (i.e. software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 22.1.

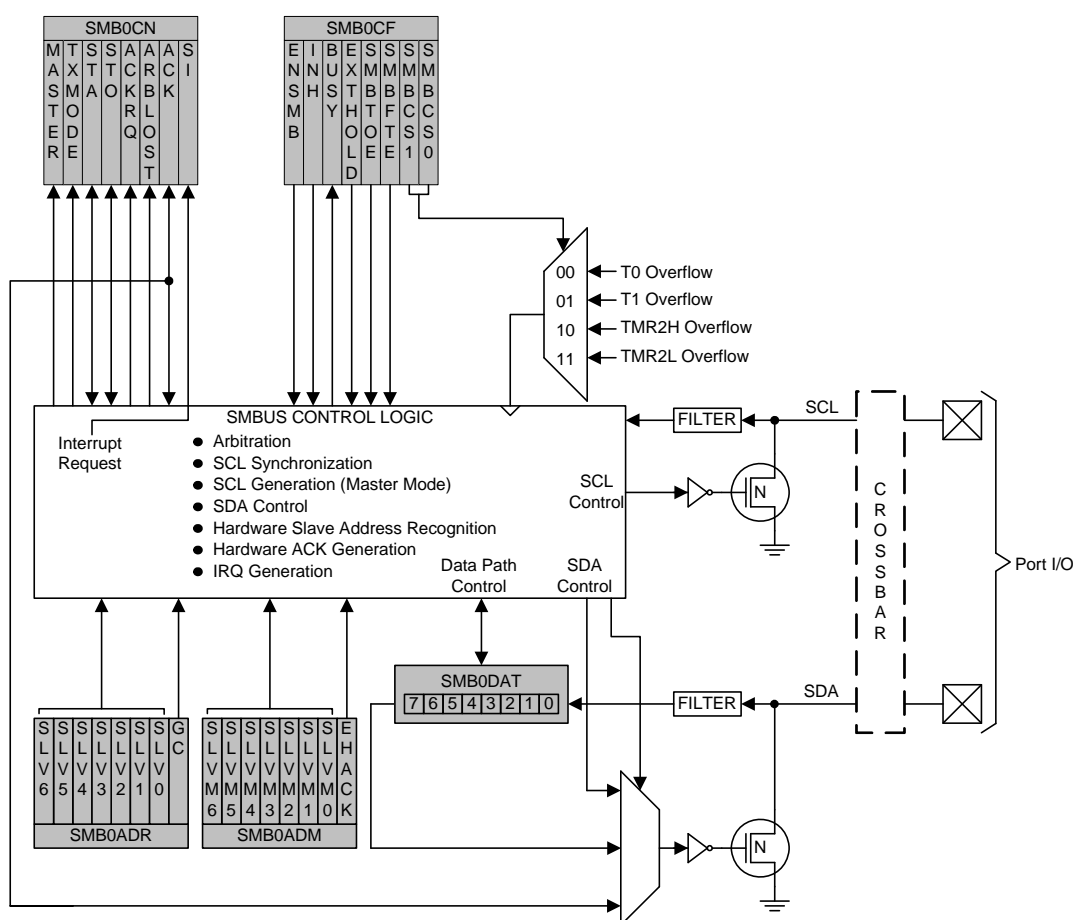


Figure 22.1. SMBus Block Diagram

25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

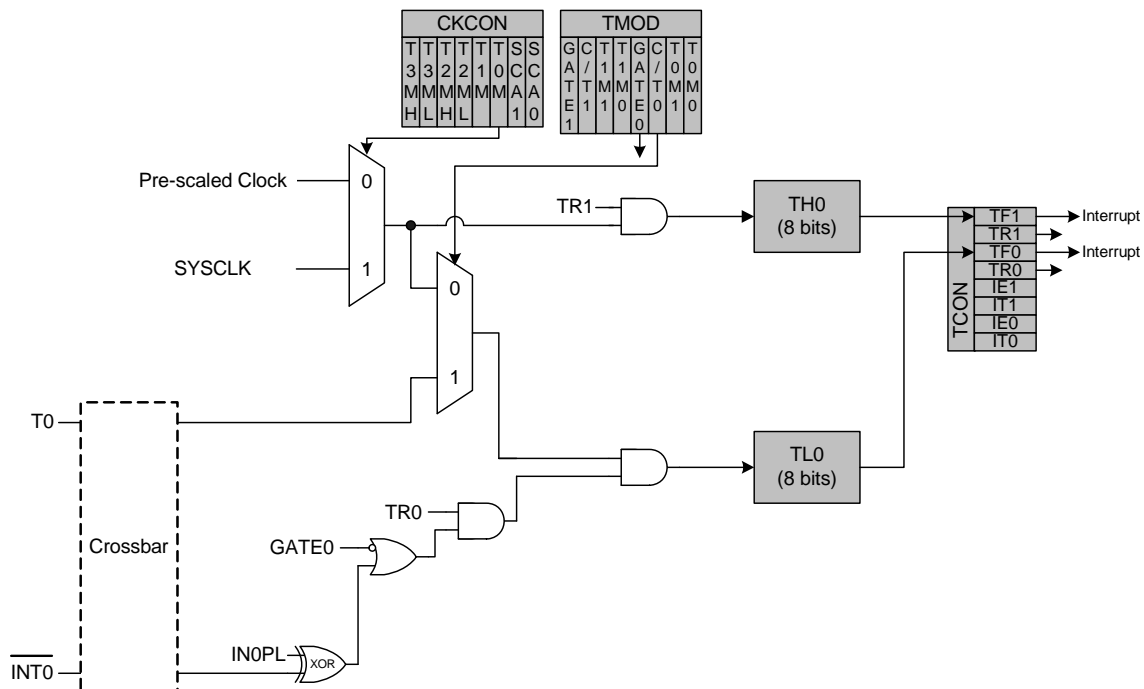


Figure 25.3. T0 Mode 3 Block Diagram

25.2.3. Comparator 0/SmaRTClock Capture Mode

The Capture Mode in Timer 2 allows either Comparator 0 or the SmaRTClock period to be measured against the system clock or the system clock divided by 12. Comparator 0 and the SmaRTClock period can also be compared against each other. Timer 2 Capture Mode is enabled by setting TF2CEN to 1. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

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

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

$$24.5 \text{ MHz} / (5984 / 8) = 0.032754 \text{ MHz or } 32.754 \text{ kHz.}$$

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

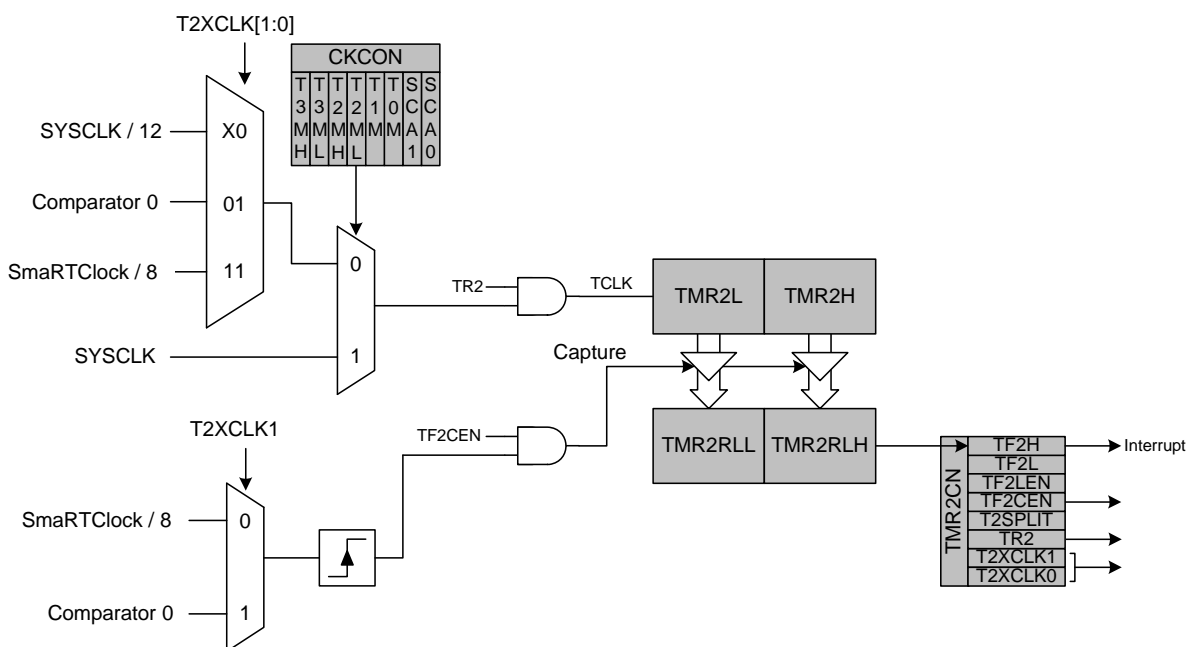


Figure 25.6. Timer 2 Capture Mode Block Diagram

C8051F99x-C8051F98x

SFR Definition 25.9. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCA

Bit	Name	Function
7:0	TMR2RLL[7:0]	Timer 2 Reload Register Low Byte. TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 25.10. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCB

Bit	Name	Function
7:0	TMR2RLH[7:0]	Timer 2 Reload Register High Byte. TMR2RLH holds the high byte of the reload value for Timer 2.

C8051F99x-C8051F98x

The 8-bit offset held in PCA0CPH2 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 26.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL2) + (256 - PCA0L)$$

Equation 26.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH2 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF2 flag (PCA0CN.2) while the WDT is enabled.

26.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

1. Disable the WDT by writing a 0 to the WDTE bit.
2. Select the desired PCA clock source (with the CPS2–CPS0 bits).
3. Load PCA0CPL2 with the desired WDT update offset value.
4. Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
5. Enable the WDT by setting the WDTE bit to 1.
6. Reset the WDT timer by writing to PCA0CPH2.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL2 defaults to 0x00. Using Equation 26.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 26.3 lists some example timeout intervals for typical system clocks.

Table 26.3. Watchdog Timer Timeout Intervals¹

System Clock (Hz)	PCA0CPL2	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500 ²	255	257
3,062,500 ²	128	129.5
3,062,500 ²	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168
Notes:		
1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.		
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.		

oscillator while in Sleep mode.

- Added a note to “15.5. Sleep Mode” and SFR Definition “15.3. Stop Mode” regarding not disabling the POR Supply Monitor while operating above 2.4 V.
- Adjusted QFN20 c, D2, and E2 package specifications in Table 3.2.