



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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	24
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	0.9V ~ 3.6V
Data Converters	A/D 23x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f920-gq

Table of Contents

1. System Overview	18
1.1. CIP-51™ Microcontroller Core	21
1.1.1. Fully 8051 Compatible	21
1.1.2. Improved Throughput	21
1.1.3. Additional Features	21
1.2. Port Input/Output	21
1.3. Serial Ports	22
1.4. Programmable Counter Array	23
1.5. 10-Bit SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode	23
1.6. Programmable Current Reference (IREF0)	25
1.7. Comparators	25
2. Ordering Information	27
3. Pinout and Package Definitions	28
4. Electrical Characteristics	43
4.1. Absolute Maximum Specifications	43
4.2. Electrical Characteristics	44
5. 10-Bit SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode	65
5.1. Output Code Formatting	66
5.2. Modes of Operation	67
5.2.1. Starting a Conversion	67
5.2.2. Tracking Modes	68
5.2.3. Burst Mode	69
5.2.4. Settling Time Requirements	70
5.2.5. Gain Setting	71
5.3. 8-Bit Mode	71
5.4. Programmable Window Detector	78
5.4.1. Window Detector In Single-Ended Mode	80
5.4.2. ADC0 Specifications	80
5.5. ADC0 Analog Multiplexer	81
5.6. Temperature Sensor	83
5.6.1. Calibration	84
5.7. Voltage and Ground Reference Options	86
5.8. External Voltage References	87
5.9. Internal Voltage References	87
5.10. Analog Ground Reference	87
5.11. Temperature Sensor Enable	87
5.12. Voltage Reference Electrical Specifications	88
6. Programmable Current Reference (IREF0)	89
6.1. IREF0 Specifications	89
7. Comparators	90
7.1. Comparator Inputs	90

22. SMBus	233
22.1.Supporting Documents	234
22.2.SMBus Configuration.....	234
22.3.SMBus Operation	235
22.3.1.Transmitter Vs. Receiver.....	235
22.3.2.Arbitration.....	235
22.3.3.Clock Low Extension.....	236
22.3.4.SCL Low Timeout.....	236
22.3.5.SCL High (SMBus Free) Timeout	236
22.4.Using the SMBus.....	237
22.4.1.SMBus Configuration Register.....	238
22.4.2.SMB0CN Control Register	241
22.4.3.Hardware Slave Address Recognition	244
22.4.4.Data Register	246
22.5.SMBus Transfer Modes.....	247
22.5.1.Write Sequence (Master)	247
22.5.2.Read Sequence (Master)	248
22.5.3.Write Sequence (Slave)	249
22.5.4.Read Sequence (Slave)	250
22.6.SMBus Status Decoding.....	250
23. UART0.....	255
23.1.Enhanced Baud Rate Generation.....	256
23.2.Operational Modes	257
23.2.1.8-Bit UART	257
23.2.2.9-Bit UART	258
23.3.Multiprocessor Communications	258
24. Enhanced Serial Peripheral Interface (SPI0 and SPI1).....	263
24.1.Signal Descriptions.....	264
24.1.1.Master Out, Slave In (MOSI).....	264
24.1.2.Master In, Slave Out (MISO).....	264
24.1.3.Serial Clock (SCK)	264
24.1.4.Slave Select (NSS)	264
24.2.SPI Master Mode Operation	265
24.3.SPI Slave Mode Operation	267
24.4.SPI Interrupt Sources	267
24.5.Serial Clock Phase and Polarity	268
24.6.SPI Special Function Registers	270
25. Timers.....	278
25.1.Timer 0 and Timer 1	280
25.1.1.Mode 0: 13-bit Counter/Timer	280
25.1.2.Mode 1: 16-bit Counter/Timer	281
25.1.3.Mode 2: 8-bit Counter/Timer with Auto-Reload.....	282
25.1.4.Mode 3: Two 8-bit Counter/Timers (Timer 0 Only).....	283
25.2.Timer 2	288
25.2.1.16-bit Timer with Auto-Reload.....	288

Table 4.9. ADC0 Electrical Characteristics

$V_{DD} = 1.8$ to 3.6 V, $V_{REF} = 1.65$ V (REFSL[1:0] = 11), -40 to $+85$ °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution		10			bits
Integral Nonlinearity		—	± 0.5	± 1	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	± 0.5	± 1	LSB
Offset Error		—	$\pm < 1$	± 2	LSB
Full Scale Error		—	± 1	± 2.5	LSB
Dynamic performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, 300 kps)					
Signal-to-Noise Plus Distortion		54	58	—	dB
Signal-to-Distortion		—	73	—	dB
Spurious-Free Dynamic Range		—	75	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	7.33	MHz
Conversion Time in SAR Clocks	10-bit Mode 8-bit Mode	13 11	— —	— —	clocks
Track/Hold Acquisition Time		1.5	—	—	μ s
Throughput Rate		—	—	300	kps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND)	0	—	V_{REF}	V
Absolute Pin Voltage with respect to GND	Single Ended	0	—	V_{DD}	V
Sampling Capacitance	1x Gain 0.5x Gain	—	30 28	—	pF
Input Multiplexer Impedance		—	5	—	k Ω
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Conversion Mode (300 kps) Tracking Mode (0 kps)	— —	800 680	— —	μ A
Power Supply Rejection	Internal High Speed V_{REF} External V_{REF}	— —	67 74	— —	dB

C8051F93x-C8051F92x

SFR Definition 5.15. REF0CN: Voltage Reference Control

Bit	7	6	5	4	3	2	1	0
Name			REFGND	REFSL		TEMPE		REFOE
Type	R	R	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	1	1	0	0	0

SFR Page = 0x0; SFR Address = 0xD1

Bit	Name	Function
7:6	Unused	Unused. Read = 00b; Write = Don't Care.
5	REFGND	Analog Ground Reference. Selects the ADC0 ground reference. 0: The ADC0 ground reference is the GND pin. 1: The ADC0 ground reference is the P0.1/AGND pin.
4:3	REFSL	Voltage Reference Select. Selects the ADC0 voltage reference. 00: The ADC0 voltage reference is the P0.0/VREF pin. 01: The ADC0 voltage reference is the VDD/DC+ pin. 10: The ADC0 voltage reference is the internal 1.8 V digital supply voltage. 11: The ADC0 voltage reference is the internal 1.65 V high speed voltage reference.
2	TEMPE	Temperature Sensor Enable. Enables/Disables the internal temperature sensor. 0: Temperature Sensor Disabled. 1: Temperature Sensor Enabled.
1	Unused	Unused. Read = 0b; Write = Don't Care.
0	REFOE	Internal Voltage Reference Output Enable. Connects/Disconnects the internal voltage reference to the P0.0/VREF pin. 0: Internal 1.68 V Precision Voltage Reference disabled and not connected to P0.0/VREF. 1: Internal 1.68 V Precision Voltage Reference enabled and connected to P0.0/VREF.

5.12. Voltage Reference Electrical Specifications

See Table 4.11 on page 60 for detailed Voltage Reference Electrical Specifications.

C8051F93x-C8051F92x

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
Program Branching			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

Notes on Registers, Operands and Addressing Modes:

Rn—Register R0–R7 of the currently selected register bank.

@Ri—Data RAM location addressed indirectly through R0 or R1.

rel—8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

#data—8-bit constant

#data16—16-bit constant

bit—Direct-accessed bit in Data RAM or SFR

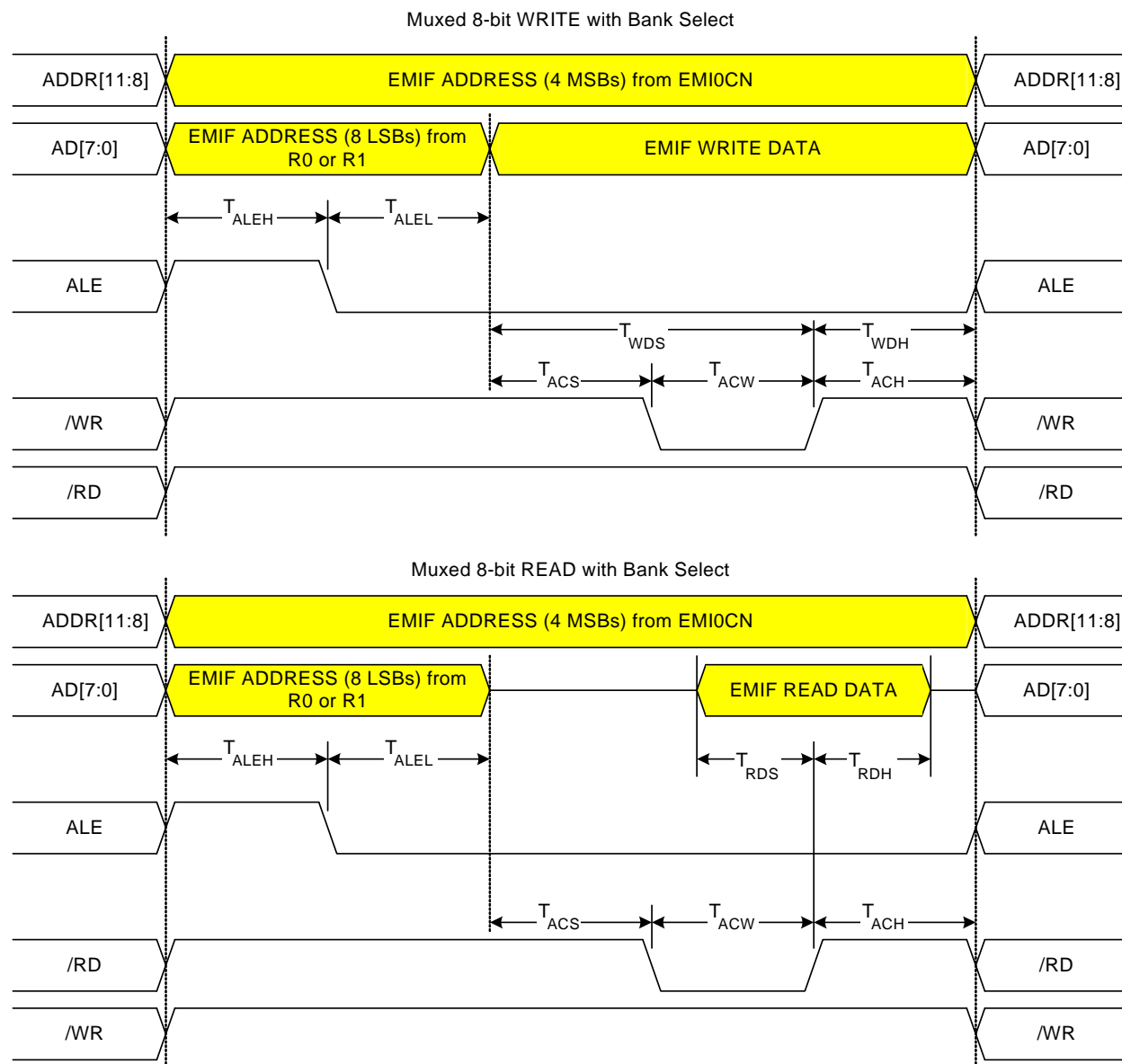
addr11—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

addr16—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.
All mnemonics copyrighted © Intel Corporation 1980.

C8051F93x-C8051F92x

10.8.2.1. Multiplexed 8-bit MOVX with Bank Select: EMI0CF[3:2] = 10.



Note: See the Port Input/Output chapter to determine which port pins are mapped to the ADDR[11:8], AD[7:0], ALE, /RD, and /WR signals.

Figure 10.6. Multiplexed 8-bit MOVX with Bank Select Timing

13.5.2. PSWE Maintenance

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

13.5.3. System Clock

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

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

15. Cyclic Redundancy Check Unit (CRC0)

C8051F93x-C8051F92x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit or 32-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register. CRC0 posts the 16-bit or 32-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 15.1. CRC0 also has a bit reverse register for quick data manipulation.

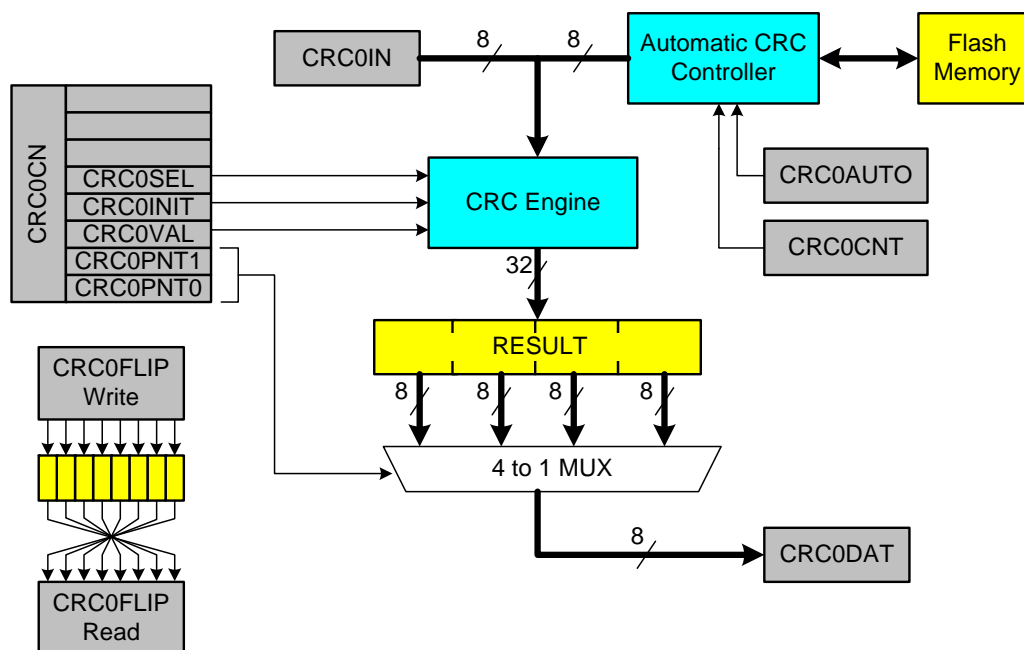


Figure 15.1. CRC0 Block Diagram

15.1. CRC Algorithm

The C8051F93x-C8051F92x 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 (0x00000000 or 0xFFFFFFFF).
- 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.

C8051F93x-C8051F92x

16.1. Startup Behavior

On initial power-on, the dc-dc converter outputs a constant 50% duty cycle until there is sufficient voltage on the output capacitor to maintain regulation. The size of the output capacitor and the amount of load current present during startup will determine the length of time it takes to charge the output capacitor.

During initial power-on reset, the maximum peak inductor current threshold, which triggers the overcurrent protection circuit, is set to approximately 125 mA. This generates a “soft-start” to limit the output voltage slew rate and prevent excessive in-rush current at the output capacitor. In order to ensure reliable startup of the dc-dc converter, the following restrictions have been imposed:

- The maximum dc load current allowed during startup is given in Table 4.15 on page 62. If the dc-dc converter is powering external sensors or devices through the VDD/DC+ pin or through GPIO pins, then the current supplied to these sensors or devices is counted towards this limit. The in-rush current into capacitors does not count towards this limit.
- The maximum total output capacitance is given in Table 4.15 on page 62. This value includes the required 1 µF ceramic output capacitor and any additional capacitance connected to the VDD/DC+ pin.

Once initial power-on is complete, the peak inductor current limit can be increased by software as shown in Table 16.1. Limiting the peak inductor current can allow the device to start up near the battery’s end of life.

Table 16.1. IPeak Inductor Current Limit Settings

SWSEL	ILIMIT	Peak Current (mA)
1	0	100
0	0	125
1	1	250
0	1	500

The peak inductor current is dependent on several factors including the dc load current and can be estimated using following equation:

$$I_{PK} = \frac{2 I_{LOAD}(VDD/DC+ - VBAT)}{\text{efficiency} \times \text{inductance} \times \text{frequency}}$$

efficiency = 0.80

inductance = 0.68 µH

frequency = 2.4 MHz

When the RC oscillator is first enabled, the external oscillator valid detector allows software to determine when oscillation has stabilized. The recommended procedure for starting the RC oscillator is:

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

19.3.3. External Capacitor Mode

If a capacitor is used as the external oscillator, the circuit should be configured as shown in Figure 19.1, Option 3. The capacitor should be added to XTAL2, and XTAL2 should be configured for analog I/O with the digital output drivers disabled. XTAL1 is not affected in RC mode.

The capacitor should be no greater than 100 pF; however, for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. The oscillation frequency and the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register can be determined by the following equation:

$$f = \frac{KF}{C \times V_{DD}}$$

where

f = frequency of clock in MHz

R = pull-up resistor value in k Ω

V_{DD} = power supply voltage in Volts

C = capacitor value on the XTAL2 pin in pF

Below is an example of selecting the capacitor and finding the frequency of oscillation Assume $V_{DD} = 3.0$ V and f = 150 kHz:

$$f = \frac{KF}{C \times V_{DD}}$$

$$0.150 \text{ MHz} = \frac{KF}{C \times 3.0}$$

Since a frequency of roughly 150 kHz is desired, select the K Factor from Table 19.2 as KF = 22:

$$0.150 \text{ MHz} = \frac{22}{C \times 3.0 \text{ V}}$$

$$C = \frac{22}{0.150 \text{ MHz} \times 3.0 \text{ V}}$$

$$C = 48.8 \text{ pF}$$

Therefore, the XFCN value to use in this example is 011 and C is approximately 50 pF.

The recommended startup procedure for C mode is the same as RC mode.

C8051F93x-C8051F92x

SFR Definition 19.2. OSCICN: Internal Oscillator Control

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

SFR Page = 0x0; SFR Address = 0xB2

Bit	Name	Function
7	IOSCEN	Internal Oscillator Enable. 0: Internal oscillator disabled. 1: Internal oscillator enabled.
6	IFRDY	Internal Oscillator Frequency Ready Flag. 0: Internal oscillator is not running at its programmed frequency. 1: Internal oscillator is running at its programmed frequency.
5:0	Reserved	Reserved. Read = 001111b. Must Write 001111b.

Note: It is recommended to use read-modify-write operations such as ORL and ANL to set or clear the enable bit of this register.

SFR Definition 19.3. OSCICL: Internal Oscillator Calibration

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

SFR Page = 0x0; SFR Address = 0xB3

Bit	Name	Function
7	SSE	Spread Spectrum Enable. 0: Spread Spectrum clock dithering disabled. 1: Spread Spectrum clock dithering enabled.
6:0	OSCICL	Internal Oscillator Calibration. Factory calibrated to obtain a frequency of 24.5 MHz. Incrementing this register decreases the oscillator frequency and decrementing this register increases the oscillator frequency. The step size is approximately 1% of the calibrated frequency. The recommended calibration frequency range is between 16 and 24.5 MHz.

20. SmaRTClock (Real Time Clock)

C8051F93x-C8051F92x devices include an ultra low power 32-bit SmaRTClock Peripheral (Real Time Clock) with alarm. The SmaRTClock has a dedicated 32 kHz oscillator that can be configured for use with or without a crystal. No external resistor or loading capacitors are required. The on-chip loading capacitors are programmable to 16 discrete levels allowing compatibility with a wide range of crystals. The SmaRTClock can operate directly from a 0.9–3.6 V battery voltage and remains operational even when the device goes into its lowest power down mode.

The SmaRTClock allows a maximum of 36 hour 32-bit independent time-keeping when used with a 32.768 kHz Watch Crystal. The SmaRTClock provides an Alarm and Missing SmaRTClock events, which could be used as reset or wakeup sources. See Section “18. Reset Sources” on page 180 and Section “14. Power Management” on page 156 for details on reset sources and low power mode wake-up sources, respectively.

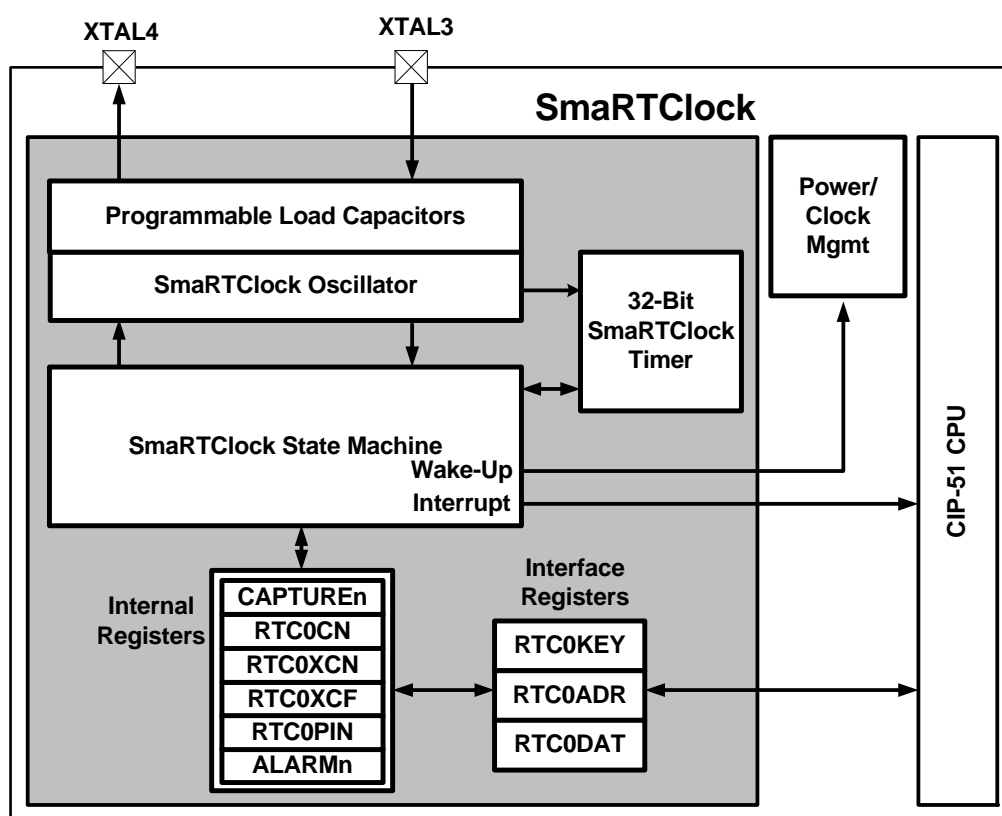


Figure 20.1. SmaRTClock Block Diagram

23.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

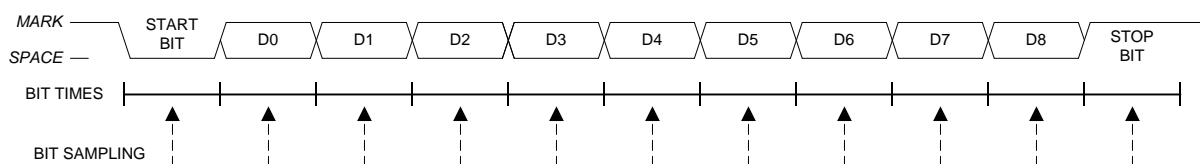


Figure 23.5. 9-Bit UART Timing Diagram

23.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

C8051F93x-C8051F92x

SFR Definition 23.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	S0MODE		MCE0	REN0	TB80	RB80	TI0	RI0
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x98; Bit-Addressable

Bit	Name	Function
7	S0MODE	Serial Port 0 Operation Mode. Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate. 1: 9-bit UART with Variable Baud Rate.
6	Unused	Unused. Read = 1b. Write = Don't Care.
5	MCE0	Multiprocessor Communication Enable. For Mode 0 (8-bit UART): Checks for valid stop bit. 0: Logic level of stop bit is ignored. 1: RI0 will only be activated if stop bit is logic level 1. For Mode 1 (9-bit UART): Multiprocessor Communications Enable. 0: Logic level of ninth bit is ignored. 1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.
4	REN0	Receive Enable. 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB80	Ninth Transmission Bit. The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB80	Ninth Receive Bit. RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI0	Transmit Interrupt Flag. Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	Receive Interrupt Flag. Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

25.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “12.5. Interrupt Register Descriptions” on page 136); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section “12.5. Interrupt Register Descriptions” on page 136). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

25.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “21.3. Priority Crossbar Decoder” on page 216 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 25.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 12.7). Setting GATE0 to 1 allows the timer to be controlled by the external input signal INT0 (see Section “12.5. Interrupt Register Descriptions” on page 136), facilitating pulse width measurements

Table 25.1. Timer 0 Running Modes

TR0	GATE0	INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal INT1 is used with Timer 1; the INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 12.7).

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.

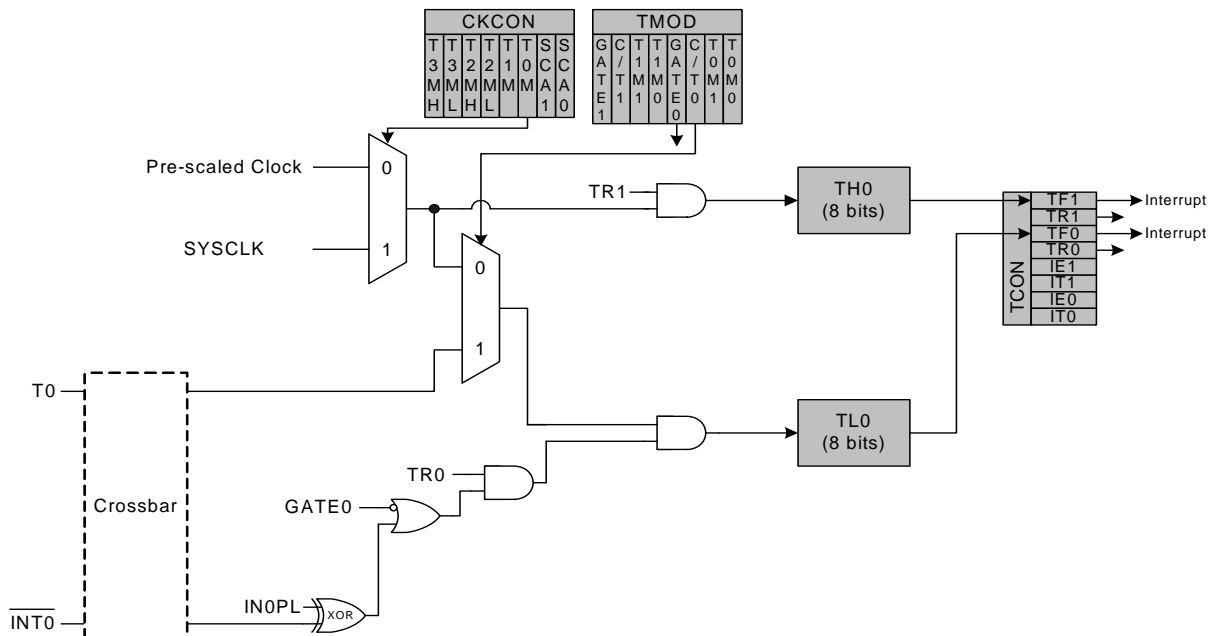


Figure 25.3. T0 Mode 3 Block Diagram

SFR Definition 25.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8C

Bit	Name	Function
7:0	TH0[7:0]	Timer 0 High Byte. The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 25.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8D

Bit	Name	Function
7:0	TH1[7:0]	Timer 1 High Byte. The TH1 register is the high byte of the 16-bit Timer 1.

25.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.8. TMR3RLH holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, the external oscillator clock source divided by 8, or Comparator 1. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bits (T3XCLK[1:0] in TMR3CN), as follows:

T3MH	T3XCLK[1:0]	TMR3H Clock Source
0	00	SYSCLK / 12
0	01	Comparator 1
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK[1:0]	TMR3L Clock Source
0	00	SYSCLK / 12
0	01	Comparator 1
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

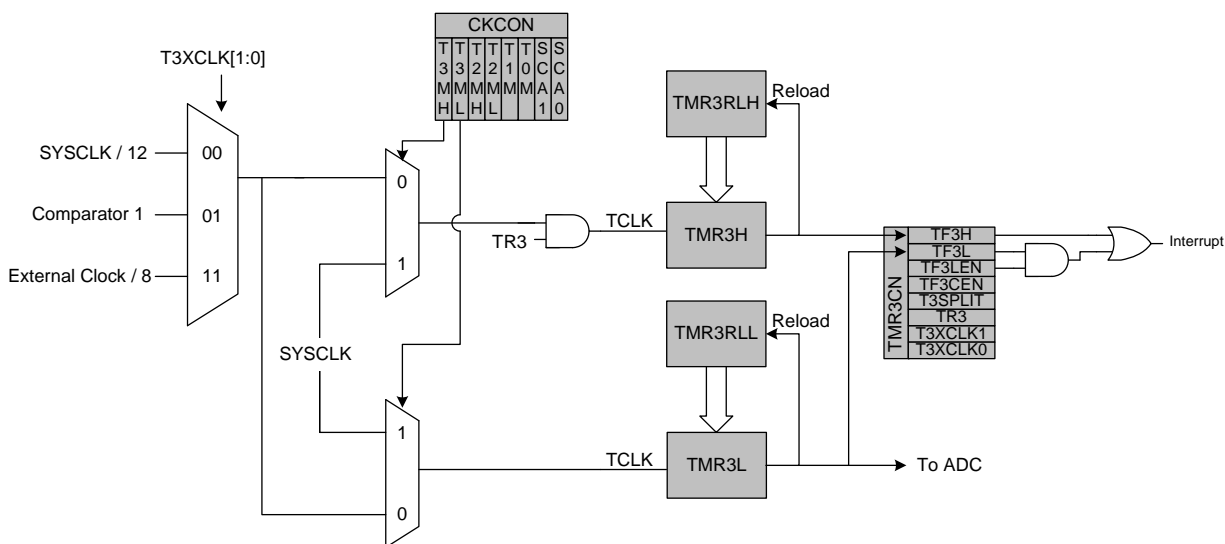


Figure 25.8. Timer 3 8-Bit Mode Block Diagram.

C8051F93x-C8051F92x

SFR Definition 26.7. PCA0CPLn: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB,
PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xD2

SFR Pages: PCA0CPL0 = 0x0, PCA0CPL1 = 0x0, PCA0CPL2 = 0x0,
PCA0CPL3 = 0x0, PCA0CPL4 = 0x0, PCA0CPL5 = 0x0

Bit	Name	Function
7:0	PCA0CPn[7:0]	PCA Capture Module Low Byte. The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOMn bit to a 0.		

SFR Definition 26.8. PCA0CPHn: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPH0 = 0xFC, PCA0CPH1 = 0xEA, PCA0CPH2 = 0xEC,
PCA0CPH3 = 0xEE, PCA0CPH4 = 0xFE, PCA0CPH5 = 0xD3

SFR Pages: PCA0CPH0 = 0x0, PCA0CPH1 = 0x0, PCA0CPH2 = 0x0,
PCA0CPH3 = 0x0, PCA0CPH4 = 0x0, PCA0CPH5 = 0x0

Bit	Name	Function
7:0	PCA0CPn[15:8]	PCA Capture Module High Byte. The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOMn bit to a 1.		