



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 <sup>2</sup> C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (4K 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	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f983-c-gmr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f983-c-gmr</a>

# C8051F99x-C8051F98x

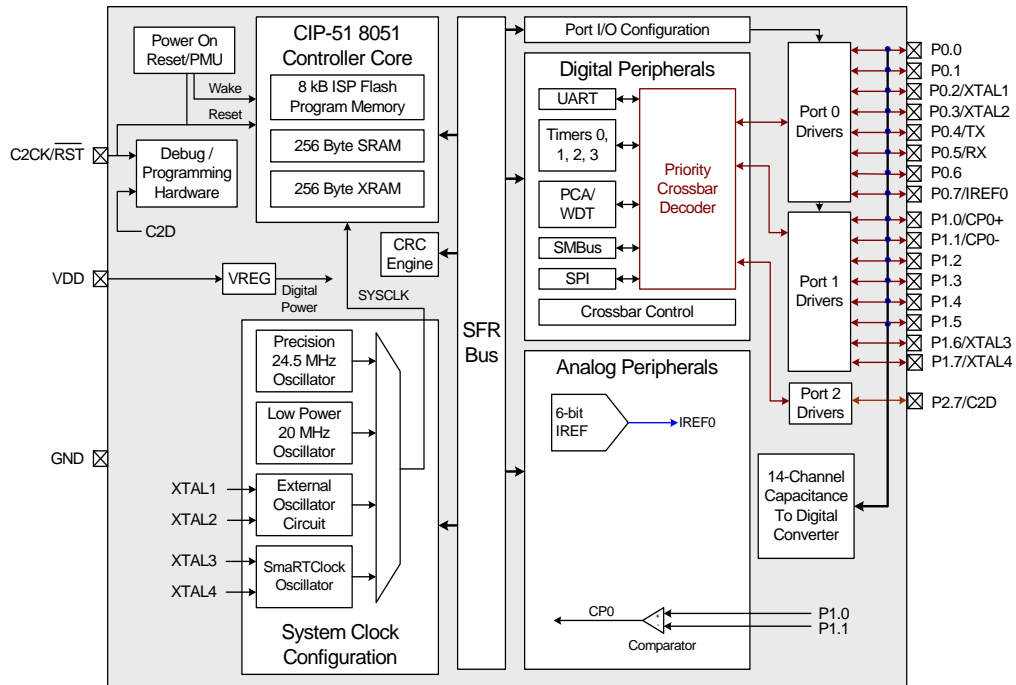
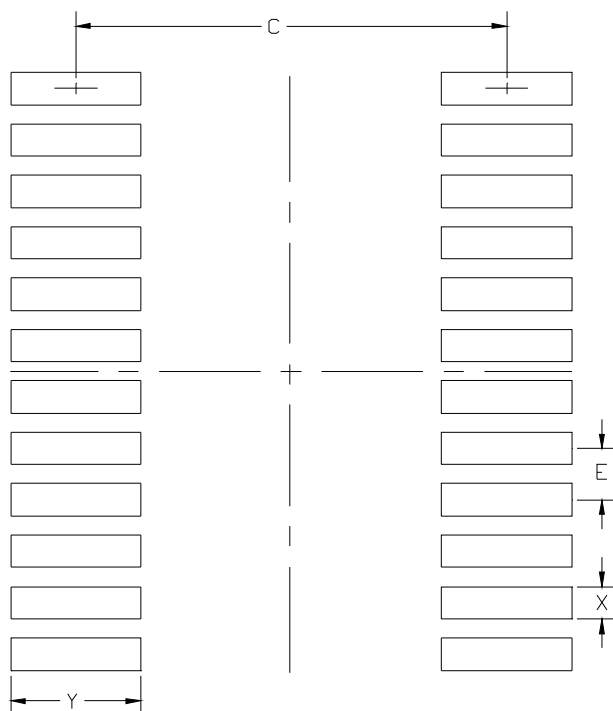


Figure 1.13. C8051F997 Block Diagram



**Figure 3.12. QSOP-24 Landing Diagram**

**Table 3.7. PCB Land Pattern**

Dimension	MIN	MAX
C	5.20	5.30
E	0.635 BSC	
X	0.30	0.40
Y	1.50	1.60

**Notes:**

**General**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

**Solder Mask Design**

1. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60  $\mu\text{m}$  minimum, all the way around the pad.

**Stencil Design**

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.

**Card Assembly**

1. A No-Clean, Type-3 solder paste is recommended.
2. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

# C8051F99x-C8051F98x

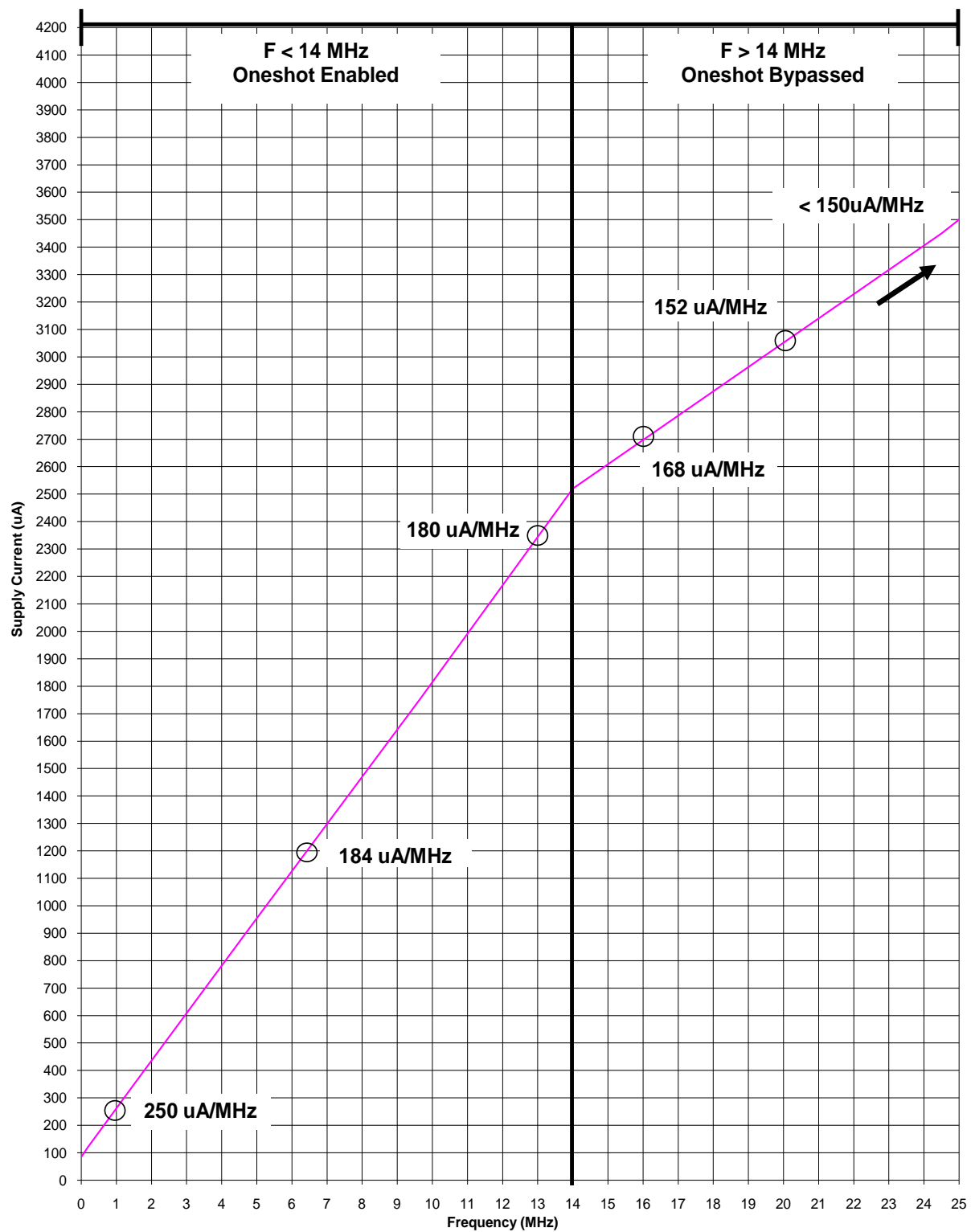
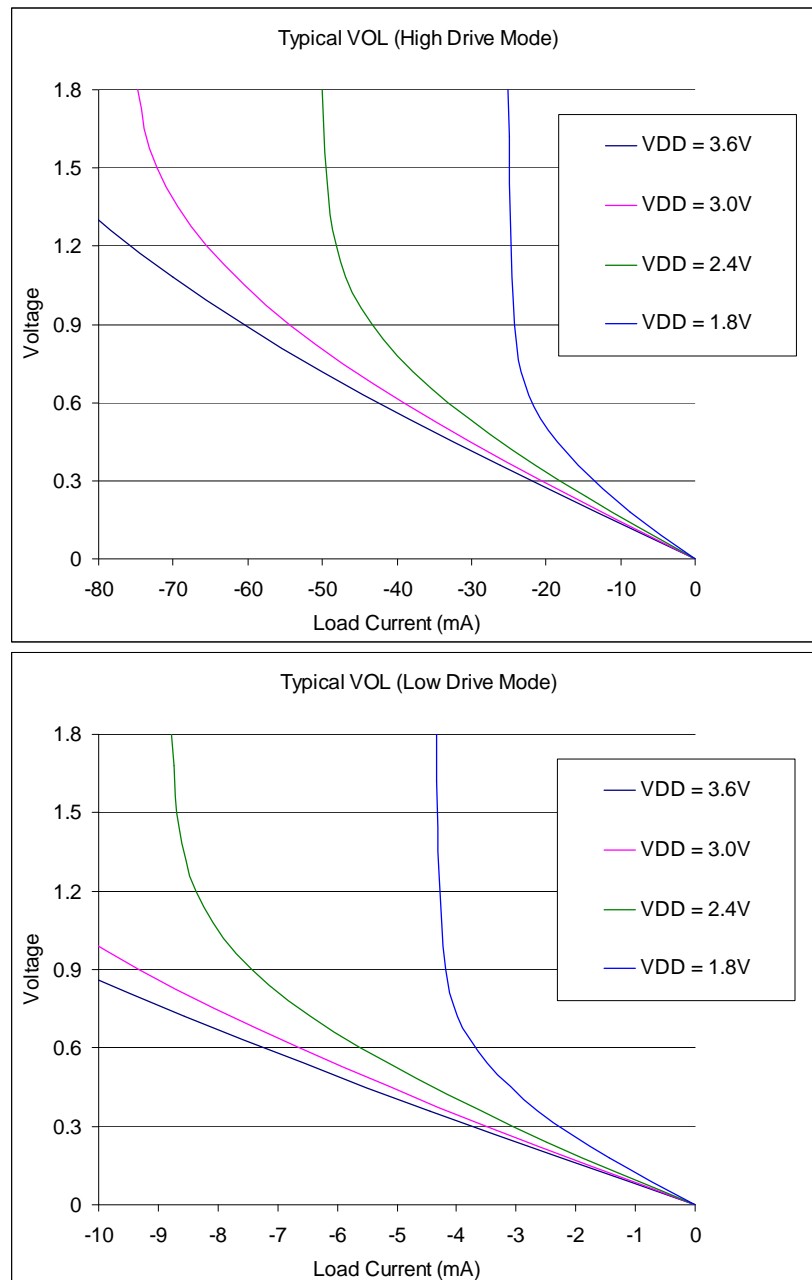


Figure 4.1. Active Mode Current (External CMOS Clock)



**Figure 4.4. Typical VOL Curves, 1.8–3.6 V**

## 5.2. Modes of Operation

ADC0 has a maximum conversion speed of 300 ksp/s in 10-bit mode. The ADC0 conversion clock (SARCLK) is a divided version of the system clock when burst mode is disabled (BURSTEN = 0), or a divided version of the low power oscillator when burst mode is enabled (BURSEN = 1). The clock divide value is determined by the AD0SC bits in the ADC0CF register.

### 5.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a 1 to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 3 overflow
5. A rising edge on the CNVSTR input signal (pin P0.6)

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. When Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See "25. Timers" on page 278 for timer configuration.

**Important Note About Using CNVSTR:** The CNVSTR input pin also functions as Port pin P0.6. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.6 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.6, set to 1 Bit 6 in register P0SKIP. See "21. Port Input/Output" on page 215 for details on Port I/O configuration.

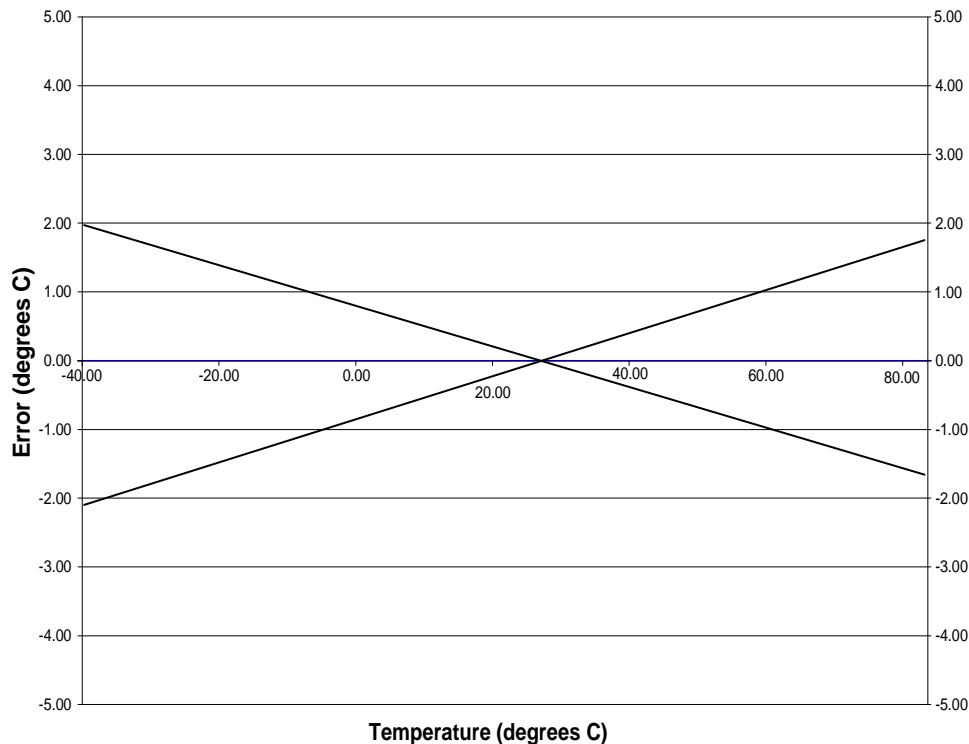
## 5.8.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 4.11 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.9 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

A single-point offset measurement of the temperature sensor is performed on each device during production test. The measurement is performed at 25 °C  $\pm$  5 °C, using the ADC with the internal high speed reference buffer selected as the Voltage Reference. The direct ADC result of the measurement is stored in the SFR registers TOFFH and TOFFL, shown in SFR Definition 5.13 and SFR Definition 5.14.



**Figure 5.9. Temperature Sensor Error with 1-Point Calibration ( $V_{REF} = 1.65$  V)**

## 8.9. Automatic Scanning (Method 2—CS0SMEN = 1)

When CS0SMEN is enabled, CS0 uses an alternate autoscanning method that uses the contents of CS0SCAN0 and CS0SCAN1 to determine which channels to include in the scan. This maximizes flexibility for application development and can result in more power efficient scanning. The following procedure can be used to configure the device for Automatic Scanning with CS0SMEN = 1.

1. Set the CS0SMEN bit to 1.
2. Select the start of conversion mode (CS0CM[2:0]) if not already configured. Mode 101b is the mode of choice for most systems.
3. Configure the CS0SCAN0 and CS0SCAN1 registers to enable channels in the scan.
4. Configure the CS0THH:CS0THL digital comparator threshold and polarity.
5. Enable wake from suspend on end of scan (CS0WOI = 1) if this functionality is desired.
6. Set CS0SS to point to the first channel in the scan. Note: CS0SS uses the same bit mapping as the CS0MX register.
7. Issue a start of conversion (BUSY = 1).
8. Enable the CS0 Wakeup Source and place the device in Suspend mode (optional).

If using Mode 101b, scanning will stop once a “touch” has been detected using the digital comparator. The CS0MX register will contain the channel mux value of the channel that caused the interrupt. Setting the busy bit when servicing the interrupt will cause the scan to continue where it left off. Scanning will also stop after all channels have been sampled and no “touches” have been detected. If the CS0WOI bit is set, a wake from suspend event will be generated. Note: When automatic scanning is enabled, the contents of the CS0MX register are only valid when the digital comparator interrupt is set and BUSY = 0.

## 8.10. CS0 Comparator

The CS0 comparator compares the latest capacitive sense conversion result with the value stored in CS0THH:CS0THL. If the result is less than or equal to the stored value, the CS0CMPF bit(CS0CN:0) is set to 0. If the result is greater than the stored value, CS0CMPF is set to 1.

If the CS0 conversion accumulator is configured to accumulate multiple conversions, a comparison will not be made until the last conversion has been accumulated.

An interrupt will be generated if CS0 greater-than comparator interrupts are enabled by setting the ECSDC bit (EIE2.5) when the comparator sets CS0CMPF to 1.

If auto-scan is running when the comparator sets the CS0CMPF bit, no further auto-scan initiated conversions will start until firmware sets CS0BUSY to 1.

A CS0 greater-than comparator event can wake a device from suspend mode. This feature is useful in systems configured to continuously sample one or more capacitive sense channels. The device will remain in the low-power suspend state until the captured value of one of the scanned channels causes a CS0 greater-than comparator event to occur. It is not necessary to have CS0 comparator interrupts enabled in order to wake a device from suspend with a greater-than event.

For a summary of behavior with different CS0 comparator, auto-scan, and auto accumulator settings, please see Table 8.1.



## SFR Definition 8.7. CS0SS: Capacitive Sense Auto-Scan Start Channel

Bit	7	6	5	4	3	2	1	0
Name				CS0SS[4:0]				
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xDD

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't care
4:0	CS0SS[4:0]	<p><b>Starting Channel for Auto-Scan.</b></p> <p>Sets the first CS0 channel to be selected by the mux for Capacitive Sense conversion when auto-scan is enabled and active. All channels detailed in CS0MX SFR Definition 8.15 are possible choices for this register.</p> <p>When auto-scan is enabled, a write to CS0SS will also update CS0MX.</p>

## SFR Definition 8.8. CS0SE: Capacitive Sense Auto-Scan End Channel

Bit	7	6	5	4	3	2	1	0
Name				CS0SE[4:0]				
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xDE

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't care
4:0	CS0SE[4:0]	<p><b>Ending Channel for Auto-Scan.</b></p> <p>Sets the last CS0 channel to be selected by the mux for Capacitive Sense conversion when auto-scan is enabled and active. All channels detailed in CS0MX SFR Definition 8.15 are possible choices for this register.</p>

# C8051F99x-C8051F98x

## 9. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 27) and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 9.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

### 9.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

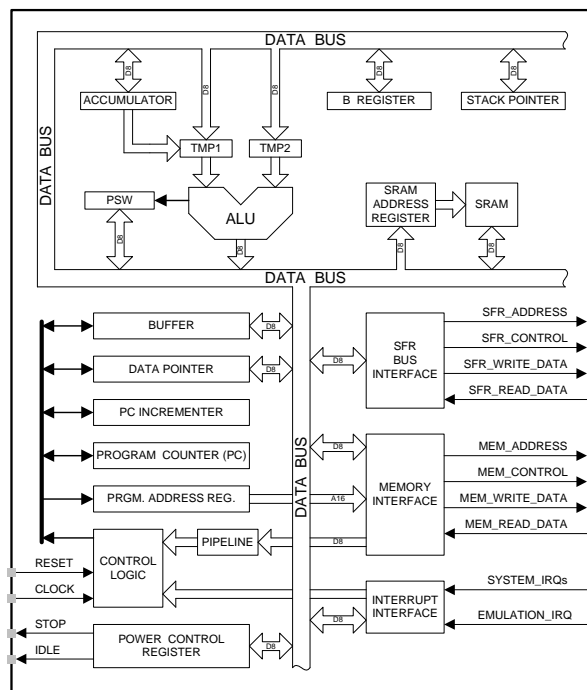


Figure 9.1. CIP-51 Block Diagram

# C8051F99x-C8051F98x

## SFR Definition 12.1. SFR Page: SFR Page

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

SFR Page = All; SFR Address = 0xA7

Bit	Name	Function
7:0	SFRPAGE[7:0]	<b>SFR Page.</b> Specifies the SFR Page used when reading, writing, or modifying special function registers.

**Table 12.3. Special Function Registers**

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

Register	Address	SFR Page	Description	Page
ACC	0xE0	All	Accumulator	126
ADC0AC	0xBA	0x0	ADC0 Accumulator Configuration	76
ADC0CF	0x97	0x0	ADC0 Configuration	75
ADC0CN	0xE8	0x0	ADC0 Control	74
ADC0GTH	0xC4	0x0	ADC0 Greater-Than Compare High	80
ADC0GTL	0xC3	0x0	ADC0 Greater-Than Compare Low	80
ADC0H	0xBE	0x0	ADC0 High	79
ADC0L	0xBD	0x0	ADC0 Low	79
ADC0LTH	0xC6	0x0	ADC0 Less-Than Compare Word High	81
ADC0LTL	0xC5	0x0	ADC0 Less-Than Compare Word Low	81
ADC0MX	0x96	0x0	AMUX0 Channel Select	84
ADC0PWR	0xBB	All	ADC0 Burst Mode Power-Up Time	77
ADC0TK	0xBC	All	ADC0 Tracking Control	78
B	0xF0	All	B Register	126
CKCON	0x8E	0x0	Clock Control	279
CLKSEL	0xA9	All	Clock Select	193
CPT0CN	0x9B	0x0	Comparator0 Control	96
CPT0MD	0x9D	0x0	Comparator0 Mode Selection	97
CPT0MX	0x9F	0x0	Comparator0 Mux Selection	99
CRC0AUTO	0x9E	All	CRC0 Automatic Control	177
CRC0CN	0x84	All	CRC0 Control	175
CRC0CNT	0x9A	All	CRC0 Automatic Flash Sector Count	178
CRC0DAT	0x86	All	CRC0 Data	176
CRC0FLIP	0x9C	All	CRC0 Flip	179
CRC0IN	0x85	All	CRC0 Input	176
CS0CF	0xAA	0x0	CS0 Configuration	108
CS0CN	0xB0	0x0	CS0 Control	107

# C8051F99x-C8051F98x

## SFR Definition 15.1. PMU0CF: Power Management Unit Configuration<sup>1,2,3</sup>

Bit	7	6	5	4	3	2	1	0
Name	SLEEP	SUSPEND	CLEAR	RSTWK	RTCFWK	RTCAWK	PMATWK	CPT0WK
Type	W	W	W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xB5

Bit	Name	Description	Write	Read
7	SLEEP	<b>Sleep Mode Select</b>	Writing 1 places the device in Sleep Mode.	N/A
6	SUSPEND	<b>Suspend Mode Select</b>	Writing 1 places the device in Suspend Mode.	N/A
5	CLEAR	<b>Wake-up Flag Clear</b>	Writing 1 clears all wake-up flags.	N/A
4	RSTWK	<b>Reset Pin Wake-up Flag</b>	N/A	Set to 1 if a glitch has been detected on $\overline{RST}$ .
3	RTCFWK	<b>SmaRTClock Oscillator Fail Wake-up Source Enable and Flag</b>	0: Disable wake-up on SmaRTClock Osc. Fail. 1: Enable wake-up on SmaRTClock Osc. Fail.	Set to 1 if the SmaRTClock Oscillator has failed.
2	RTCAWK	<b>SmaRTClock Alarm Wake-up Source Enable and Flag</b>	0: Disable wake-up on SmaRTClock Alarm. 1: Enable wake-up on SmaRTClock Alarm.	Set to 1 if a SmaRTClock Alarm has occurred.
1	PMATWK	<b>Port Match Wake-up Source Enable and Flag</b>	0: Disable wake-up on Port Match Event. 1: Enable wake-up on Port Match Event.	Set to 1 if a Port Match Event has occurred.
0	CPT0WK	<b>Comparator0 Wake-up Source Enable and Flag</b>	0: Disable wake-up on Comparator0 rising edge. 1: Enable wake-up on Comparator0 rising edge.	Set to 1 if Comparator0 rising edge caused the last wake-up.

### Notes:

1. Read-modify-write operations (ORL, ANL, etc.) should not be used on this register. Wake-up sources must be re-enabled each time the SLEEP or SUSPEND bits are written to 1.
2. The Low Power Internal Oscillator cannot be disabled and the MCU cannot be placed in Suspend or Sleep Mode if any wake-up flags are set to 1. Software should clear all wake-up sources after each reset and after each wake-up from Suspend or Sleep Modes.
3. PMU0 requires two system clocks to update the wake-up source flags after waking from Suspend mode. The wake-up source flags will read '0' during the first two system clocks following the wake from Suspend mode.

## Internal Register Definition 20.5. RTC0XCN: SmaRTClock Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	AGCEN	XMODE	BIASX2	CLKVLD	LFOEN			
Type	R/W	R/W	R/W	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address = 0x05

Bit	Name	Function
7	AGCEN	<b>SmaRTClock Oscillator Automatic Gain Control (AGC) Enable.</b> 0: AGC disabled. 1: AGC enabled.
6	XMODE	<b>SmaRTClock Oscillator Mode.</b> Selects Crystal or Self Oscillate Mode. 0: Self-Oscillate Mode selected. 1: Crystal Mode selected.
5	BIASX2	<b>SmaRTClock Oscillator Bias Double Enable.</b> <b>Enables/disables the Bias Double feature.</b> 0: Bias Double disabled. 1: Bias Double enabled.
4	CLKVLD	<b>SmaRTClock Oscillator Crystal Valid Indicator.</b> Indicates if oscillation amplitude is sufficient for maintaining oscillation. 0: Oscillation has not started or oscillation amplitude is too low to maintain oscillation. 1: Sufficient oscillation amplitude detected.
3	LFOEN	<b>Low Frequency Oscillator Enable and Select.</b> Overrides XMODE and selects the internal low frequency oscillator (LFO) as the SmaRTClock oscillator source. 0: XMODE determines SmaRTClock oscillator source. 1: LFO enabled and selected as SmaRTClock oscillator source.
2:0	Unused	Read = 000b; Write = Don't Care.

## SFR Definition 21.3. XBR2: Port I/O Crossbar Register 2

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

SFR Page = 0x0; SFR Address = 0xE3

Bit	Name	Function
7	WEAKPUD	<b>Port I/O Weak Pullup Disable.</b> 0: Weak Pullups enabled (except for Port I/O pins configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	<b>Crossbar Enable.</b> 0: Crossbar disabled. 1: Crossbar enabled.
5:0	Unused	Read = 000000b; Write = Don't Care.
<b>Note:</b> The Crossbar must be enabled (XBARE = 1) to use any Port pin as a digital output.		

# C8051F99x-C8051F98x

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 22.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

**Table 22.2. Minimum SDA Setup and Hold Times**

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	$T_{low} - 4$ system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks
<b>*Note:</b> Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “22.3.4. SCL Low Timeout” on page 238). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 22.4).

**Table 22.6. SMBus Status Decoding With Hardware ACK Generation Enabled  
(EHACK = 1)**

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000
	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100



**Table 22.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1) (Continued)**

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
						Reschedule failed transfer	1	0	X	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		0	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
						Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

## SFR Definition 25.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x94

Bit	Name	Function
7:0	TMR3L[7:0]	<b>Timer 3 Low Byte.</b> In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

## SFR Definition 25.17. TMR3H Timer 3 High Byte

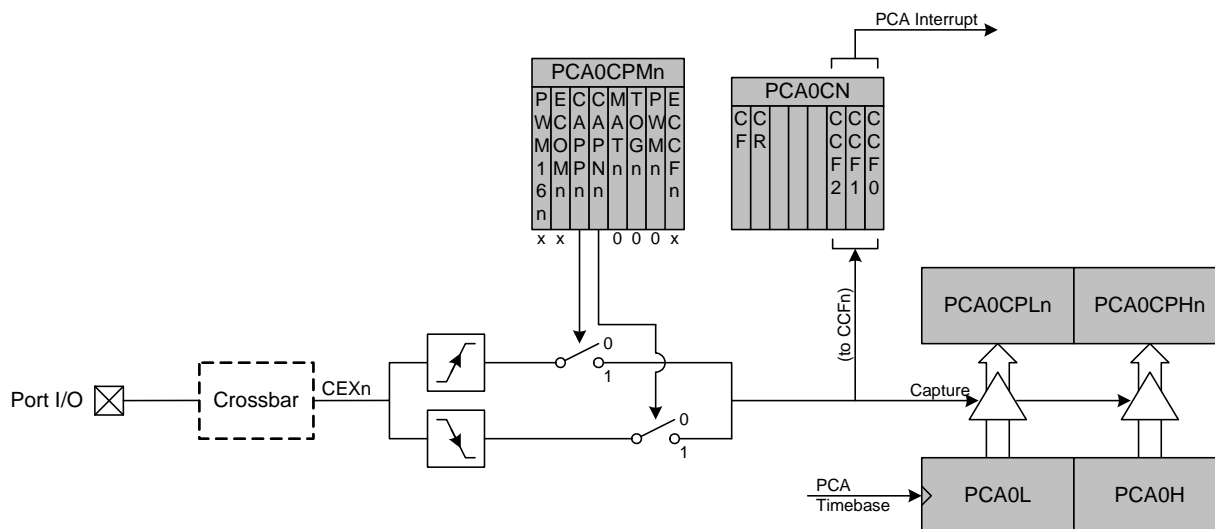
Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x95

Bit	Name	Function
7:0	TMR3H[7:0]	<b>Timer 3 High Byte.</b> In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

## 26.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEX<sub>n</sub> pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPL<sub>n</sub> and PCA0CPH<sub>n</sub>). The CAPP<sub>n</sub> and CAPN<sub>n</sub> bits in the PCA0CPM<sub>n</sub> register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCF<sub>n</sub>) in PCA0CN is set to logic 1. An interrupt request is generated if the CCF<sub>n</sub> interrupt for that module is enabled. The CCF<sub>n</sub> bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPP<sub>n</sub> and CAPN<sub>n</sub> bits are set to logic 1, then the state of the Port pin associated with CEX<sub>n</sub> can be read directly to determine whether a rising-edge or falling-edge caused the capture.



**Figure 26.4. PCA Capture Mode Diagram**

**Note:** The CEX<sub>n</sub> input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

## 26.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 2. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 2 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA counter high byte; the Module 2 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

### 26.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2–CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 26.11).

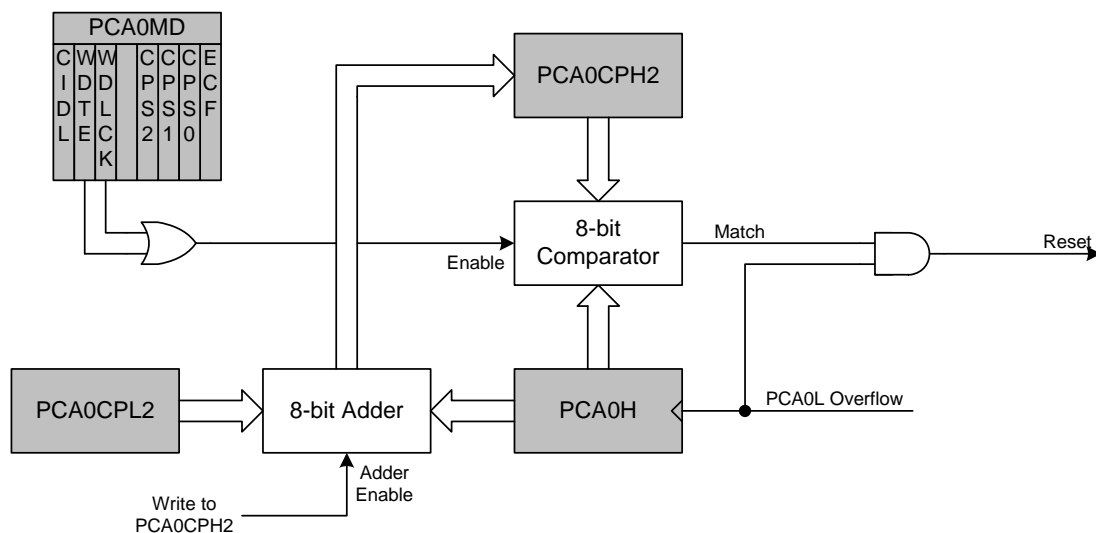


Figure 26.11. PCA Module 2 with Watchdog Timer Enabled

# C8051F99x-C8051F98x

## SFR Definition 26.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS2	CPS1	CPS0	ECF
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD9

Bit	Name	Function
7	CIDL	<b>PCA Counter/Timer Idle Control.</b> Specifies PCA behavior when CPU is in Idle Mode. 0: PCA continues to function normally while the system controller is in Idle Mode. 1: PCA operation is suspended while the system controller is in Idle Mode.
6	WDTE	<b>Watchdog Timer Enable.</b> If this bit is set, PCA Module 2 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 2 enabled as Watchdog Timer.
5	WDLCK	<b>Watchdog Timer Lock.</b> This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = don't care.
3:1	CPS[2:0]	<b>PCA Counter/Timer Pulse Select.</b> These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 110: SmarTClock divided by 8 (synchronized with the system clock) 111: Reserved
0	ECF	<b>PCA Counter/Timer Overflow Interrupt Enable.</b> This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.
<b>Note:</b> When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.		