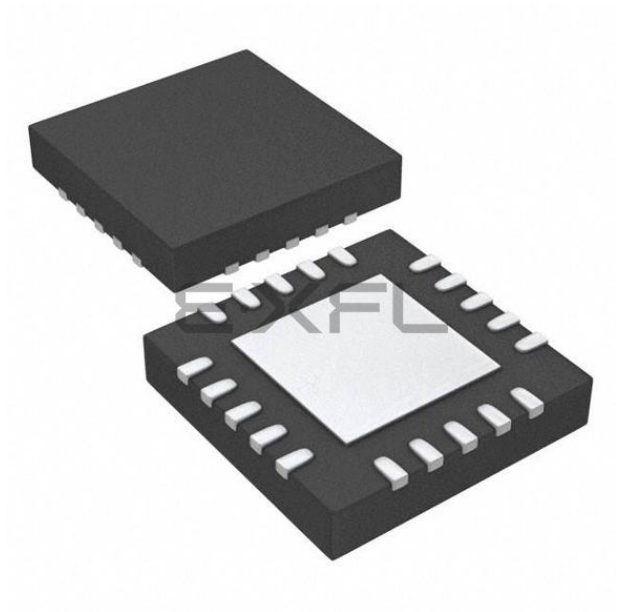

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 16x10b; D/A 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f396-a-gm

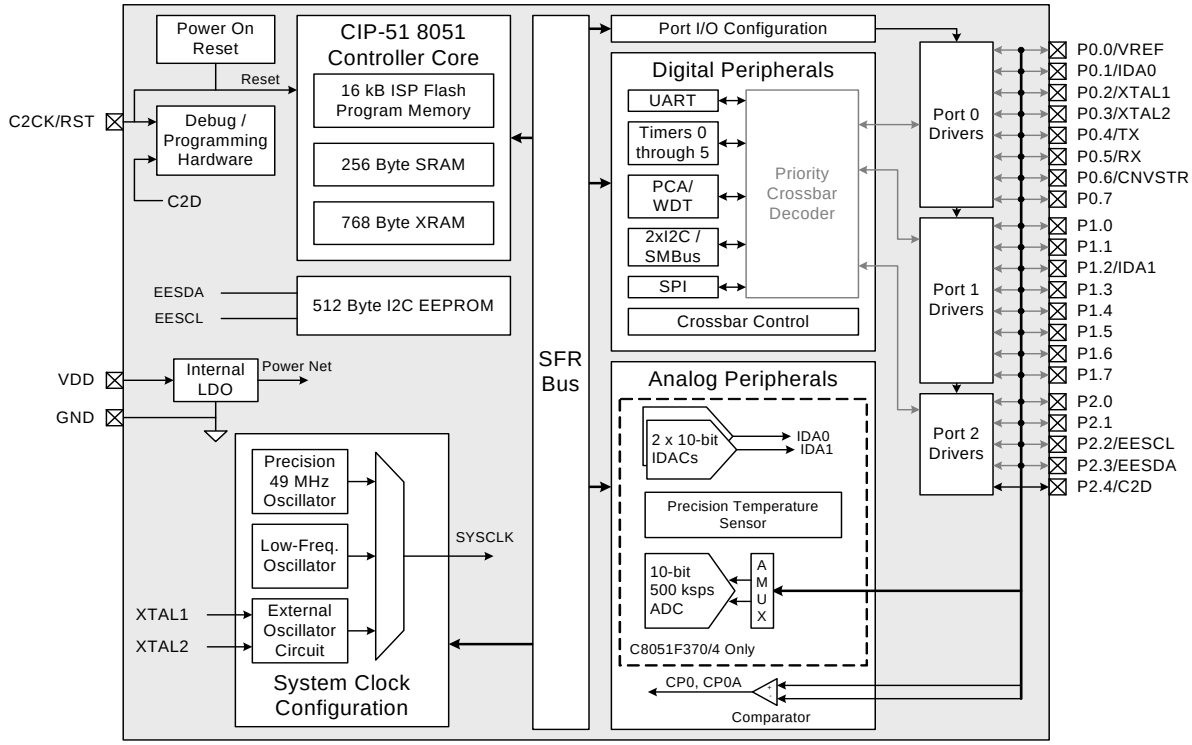

Figure 1.3. C8051F370/1/4/5 Block Diagram

Table 4.1. Pin Definitions for the C8051F39x/37x (Continued)

Name	Pin ‘F392/3/6/7/8/9	Pin ’F390/1/4/5	Pin ’F370/1/4/5	Type	Description
P2.2	—	8	—	D I/O or A In	Port 2.2.
P2.2 EESCL	-	—	8	D I/O or A In D I/O	Port 2.2. EEPROM SCL Connection.
P2.3	—	7	—	D I/O or A In	Port 2.3.
P2.3 EESDA	-	—	7	D I/O or A In D I/O	Port 2.3. EEPROM SDA Connection.
P2.4	—	6	6	D I/O	Port 2.4. (Also C2D on 24-pin Packaging)

7.2. Electrical Characteristics

Table 7.2. Global Electrical Characteristics

–40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), 50 MHz system clock, unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Supply Voltage (V_{DD})	Normal Operation	V_{RST} [1]	3.0	3.6	V
	Writing or Erasing Flash Memory	1.8	3.0	3.6	V
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock) [2]		0	—	50	MHz
T_{SYSH} (SYSCLK High Time)		9.5	—	—	ns
T_{SYSL} (SYSCLK Low Time)		9.5	—	—	ns
Specified Operating Temperature Range	C8051F39x	–40	—	+105	°C
	C8051F37x	–40	—	+85	°C
Digital Supply Current—CPU Active (Normal Mode, Fetching Instructions from Flash)					
I_{DD} [3, 4]	V_{DD} = 3.6 V, F = 50 MHz	—	7.1	7.8	mA
	V_{DD} = 3.0 V, F = 50 MHz	—	7.0	7.7	mA
	V_{DD} = 3.6 V, F = 25 MHz	—	4.6	5.2	mA
	V_{DD} = 3.0 V, F = 25 MHz	—	4.5	5.1	mA
	V_{DD} = 3.6 V, F = 1 MHz	—	0.35	—	mA
	V_{DD} = 3.0 V, F = 1 MHz	—	0.35	—	mA
	V_{DD} = 3.0 V, F = 80 kHz	—	0.25	—	mA

Notes:

1. Given in Table 7.4 on page 36.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization data; Not production tested.
4. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing an “sjmp $” loop, which is the compiled form of a while(1) loop in C. One iteration requires 3 CPU clock cycles, and the Flash memory is read on each cycle. The supply current will vary slightly based on the physical location of the sjmp instruction and the number of Flash address lines that toggle as a result. In the worst case, current can increase by up to 30% if the sjmp loop straddles a 64-byte Flash address boundary (e.g., 0x007F to 0x0080). Real-world code with larger loops and longer linear sequences will have few transitions across the 64-byte boundary.

Table 7.16. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Response Time Mode 0, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	370	—	ns
	CP0+ – CP0– = –100 mV	—	135	—	ns
Response Time Mode 3, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	1575	—	ns
	CP0+ – CP0– = –100 mV	—	3705	—	ns
Common-Mode Rejection Ratio		—	0.6	5	mV/V
Positive Hysteresis Mode 0 (CPMD = 00)	CP0HYP1–0 = 00	—	0.5	—	mV
	CP0HYP1–0 = 01	—	8	—	mV
	CP0HYP1–0 = 10	—	16	—	mV
	CP0HYP1–0 = 11	—	32	—	mV
Negative Hysteresis Mode 0 (CPMD = 00)	CP0HYN1–0 = 00	—	0.5	—	mV
	CP0HYN1–0 = 01	—	–8	—	mV
	CP0HYN1–0 = 10	—	–16	—	mV
	CP0HYN1–0 = 11	—	–32	—	mV
Positive Hysteresis Mode 1 (CPMD = 01)	CP0HYP1–0 = 00	—	0.5	—	mV
	CP0HYP1–0 = 01	—	6	—	mV
	CP0HYP1–0 = 10	—	12	—	mV
	CP0HYP1–0 = 11	—	24.5	—	mV
Negative Hysteresis Mode 1 (CPMD = 01)	CP0HYN1–0 = 00	—	0.5	—	mV
	CP0HYN1–0 = 01	—	–6	—	mV
	CP0HYN1–0 = 10	—	–12	—	mV
	CP0HYN1–0 = 11	—	–24.5	—	mV
Positive Hysteresis Mode 2 (CPMD = 10)	CP0HYP1–0 = 00	—	0.7	—	mV
	CP0HYP1–0 = 01	—	4.5	—	mV
	CP0HYP1–0 = 10	—	10	—	mV
	CP0HYP1–0 = 11	—	19	—	mV
Negative Hysteresis Mode 2 (CPMD = 10)	CP0HYN1–0 = 00	—	0.7	—	mV
	CP0HYN1–0 = 01	—	–4.5	—	mV
	CP0HYN1–0 = 10	—	–10	—	mV
	CP0HYN1–0 = 11	—	–19	—	mV
Positive Hysteresis Mode 3 (CPMD = 11)	CP0HYP1–0 = 00	—	1.6	2.3	mV
	CP0HYP1–0 = 01	2	4	6	mV
	CP0HYP1–0 = 10	4.8	8	11	mV
	CP0HYP1–0 = 11	10	15.5	21	mV

7.3. Typical Performance Curves

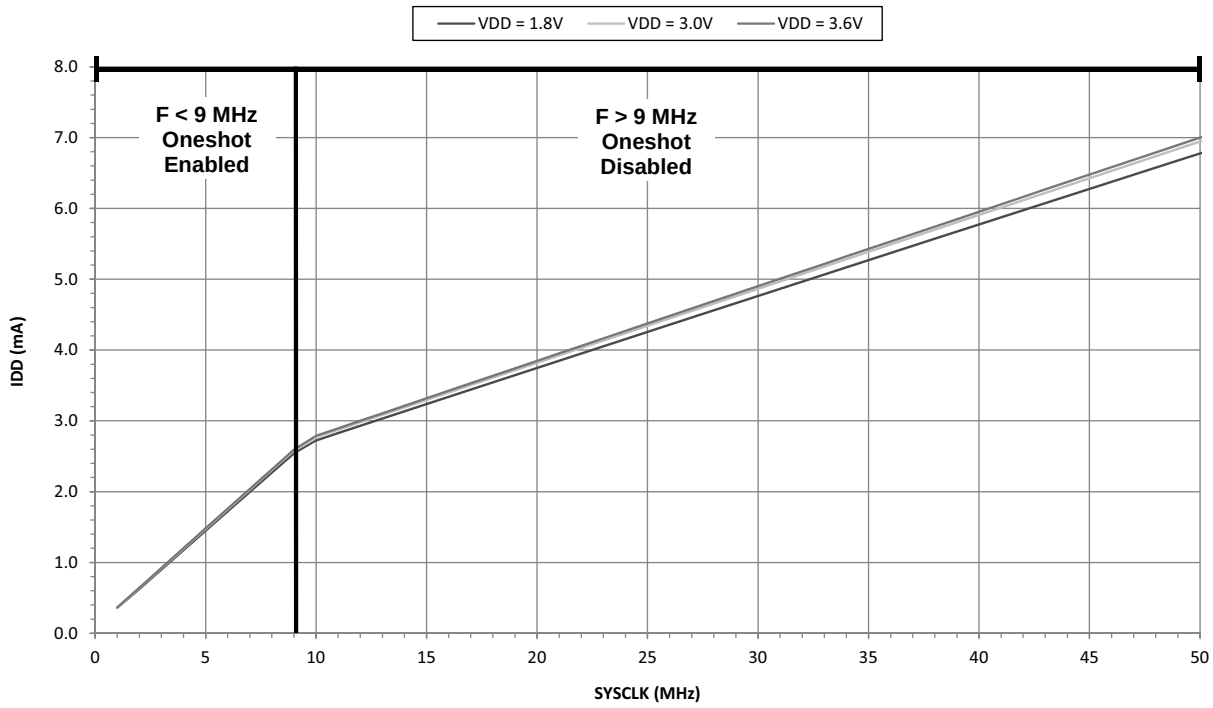

Figure 7.1. Normal Mode Digital Supply Current vs. Frequency

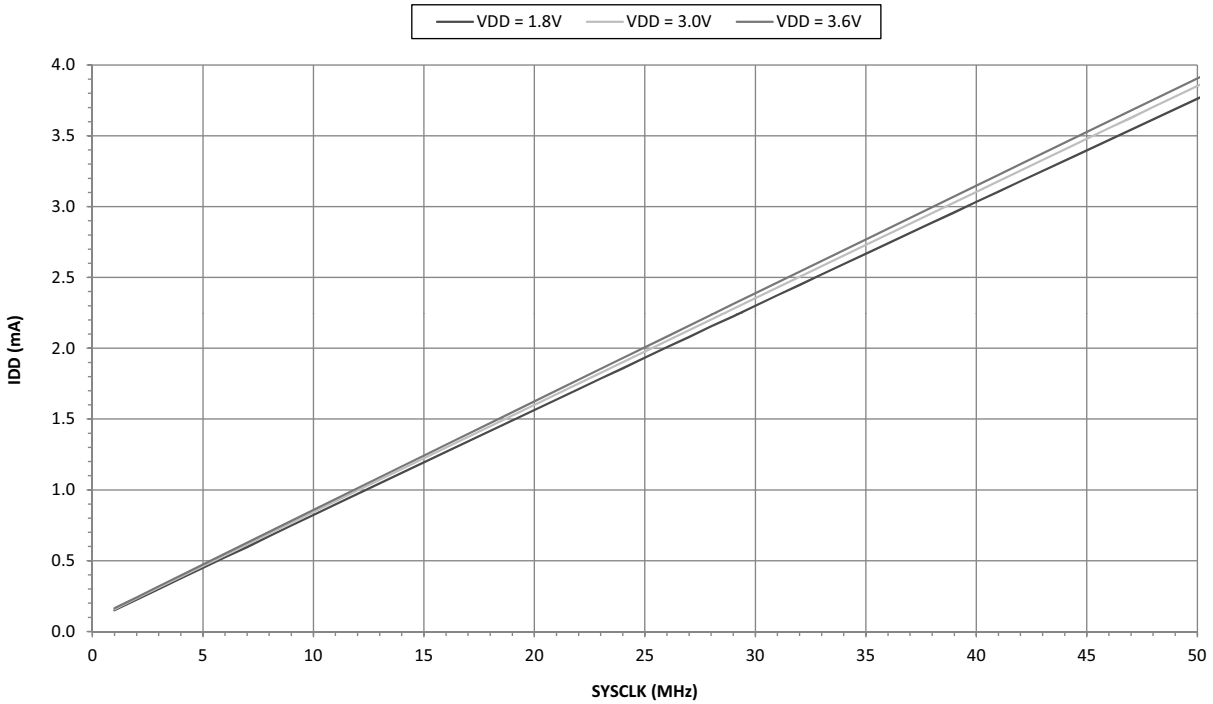

Figure 7.2. Idle Mode Digital Supply Current vs. Frequency

SFR Definition 9.2. ADC0H: ADC0 Data Word MSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBE; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0H[7:0]	**ADC0 Data Word High-Order Bits.** For AD0LJST = 0: Bits 7–2 will read 000000b. Bits 1–0 are the upper 2 bits of the 10-bit ADC0 Data Word. For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word.

SFR Definition 9.3. ADC0L: ADC0 Data Word LSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBD; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0L[7:0]	**ADC0 Data Word Low-Order Bits.** For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit Data Word. For AD0LJST = 1: Bits 7–6 are the lower 2 bits of the 10-bit Data Word. Bits 5–0 will read 000000b.

SFR Definition 11.2. IDA0H: IDA0 Data Word MSB

Bit	7	6	5	4	3	2	1	0
Name	IDA0[9:2]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x97; SFR Page = 0

Bit	Name	Function
7:0	IDA0[9:2]	**IDA0 Data Word High-Order Bits.** Upper 8 bits of the 10-bit IDA0 Data Word.

SFR Definition 11.3. IDA0L: IDA0 Data Word LSB

Bit	7	6	5	4	3	2	1	0
Name	IDA0[1:0]							
Type	R/W		R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x96; SFR Page = 0

Bit	Name	Function
7:6	IDA0[1:0]	**IDA0 Data Word Low-Order Bits.** Lower 2 bits of the 10-bit IDA0 Data Word.
5:0	Unused	Unused. Read = 000000b. Write = Don't care.

13. Voltage Regulator

C8051F39x/37x devices include an internal regulator that regulates the internal core supply from a V_{DD} supply of 1.8 to 3.6 V. The regulator has two power-saving modes built in to help reduce current consumption in low-power applications. These modes are accessed through the REG0CN register.

13.1. Power Modes

Under default conditions, the internal regulator will remain on when the device enters STOP mode. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the $\overline{RST}$ pin and a full power cycle of the device are the only methods of generating a reset.

SFR Definition 13.1. REG0CN: Voltage Regulator Control

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

SFR Address = 0xC9; SFR Page = All Pages

Bit	Name	Function
7:4	Reserved	Must Write 0000b.
3	STOPCF	**Stop Mode Configuration.** This bit configures the regulator's behavior when the device enters STOP mode. 0: Regulator is still active in STOP mode. Any enabled reset source will reset the device. 1: Regulator is shut down in STOP mode. Only the $\overline{RST}$ pin or power cycle can reset the device.
2:0	Reserved	Must Write 000b.

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 14.2). Selecting a longer response time reduces the Comparator supply current.

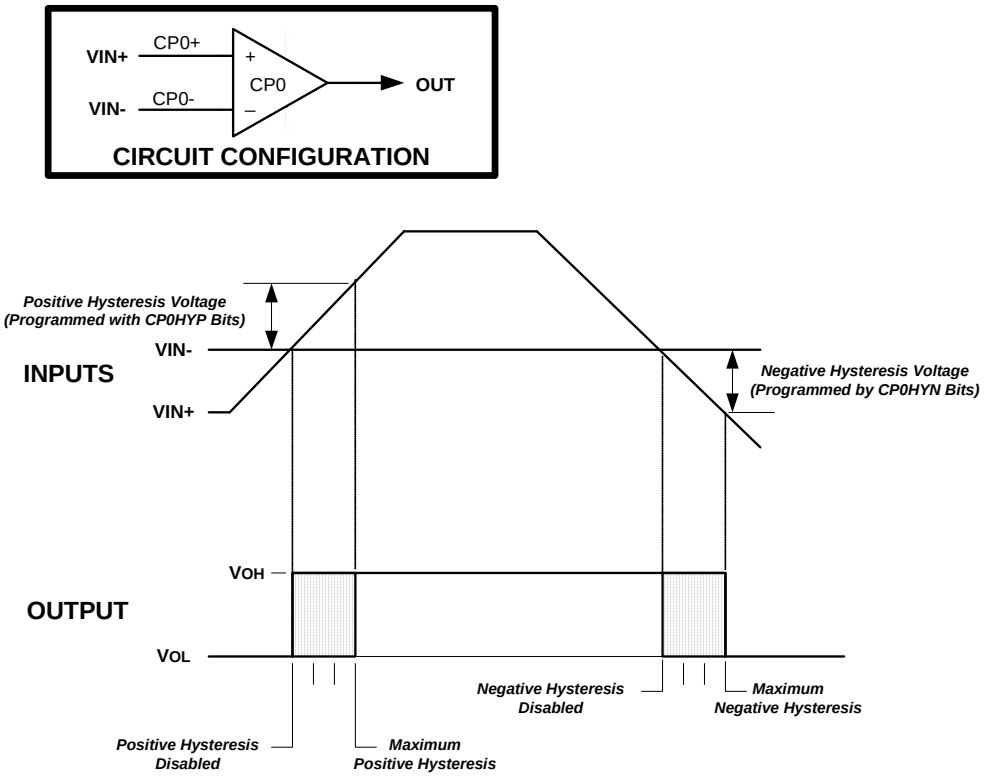

Figure 14.2. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 14.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 14.2, settings of 20, 10, or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “20.1. MCU Interrupt Sources and Vectors” on page 118). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CP0RIF flag is set to logic 1 upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE to a logic 1. The Comparator0 falling-edge interrupt mask is enabled by setting CP0FIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed.

With the CIP-51's maximum system clock at 48 MHz, it has a peak throughput of 48 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

Programming and Debugging Support
In-system programming of the EPROM program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2).

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources. C2 details can be found in Section “33. C2 Interface” on page 297.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

15.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

15.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 15.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

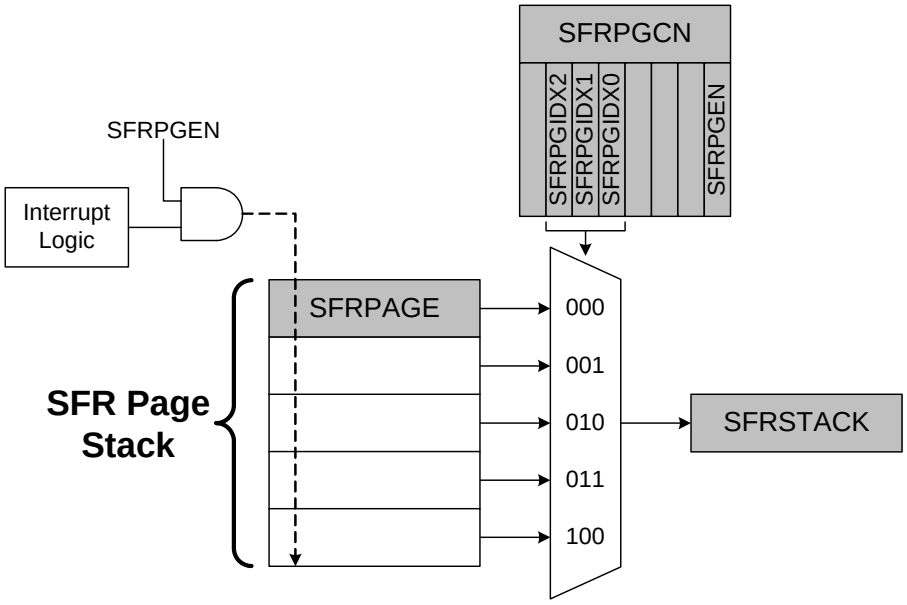

Figure 19.1. SFR Page Stack

Upon an interrupt, hardware performs the five following operations:

1. The value (if any) in the SFRPGIDX = 011b location is pushed to the SFRPAGE = 100b location.
2. The value (if any) in the SFRPGIDX = 010b location is pushed to the SFRPAGE = 011b location.
3. The value (if any) in the SFRPGIDX = 001b location is pushed to the SFRPAGE = 010b location.
4. The current SFRPAGE value is pushed to the SFRPGIDX = 001b location in the stack.
5. SFRPAGE is set to the page corresponding to the flag which generated the interrupt.

On a return from interrupt, hardware performs the four following operations:

1. The SFR page stack is popped resulting in the value in the SFRPGIDX = 001b location returning to the SFRPAGE register, thereby restoring the SFR page context without software intervention.
2. The value in the SFRPGIDX = 010b location of the stack is placed in the SFRPGIDX = 001b location.
3. The value in the SFRPGIDX = 011b location of the stack is placed in the SFRPGIDX = 010b location.
4. The value in the SFRPGIDX = 100b location of the stack is placed in the SFRPGIDX = 011b location.

Automatic switching of the SFR page by hardware upon interrupt entries and exits may be enabled or disabled using the SFR Automatic Page Control Enable Bit (SFRPGEN) located in SFRPGCN. The automatic SFR page switching is enabled after a reset until disabled by firmware.

On the execution of the RETI instruction in the SPI0 ISR, the value in SFRPAGE register is overwritten with the contents at the SFRPGIDX = 001b location. The CIP-51 may now access the TS0CN register as it did prior to the interrupts occurring. See Figure 19.6.

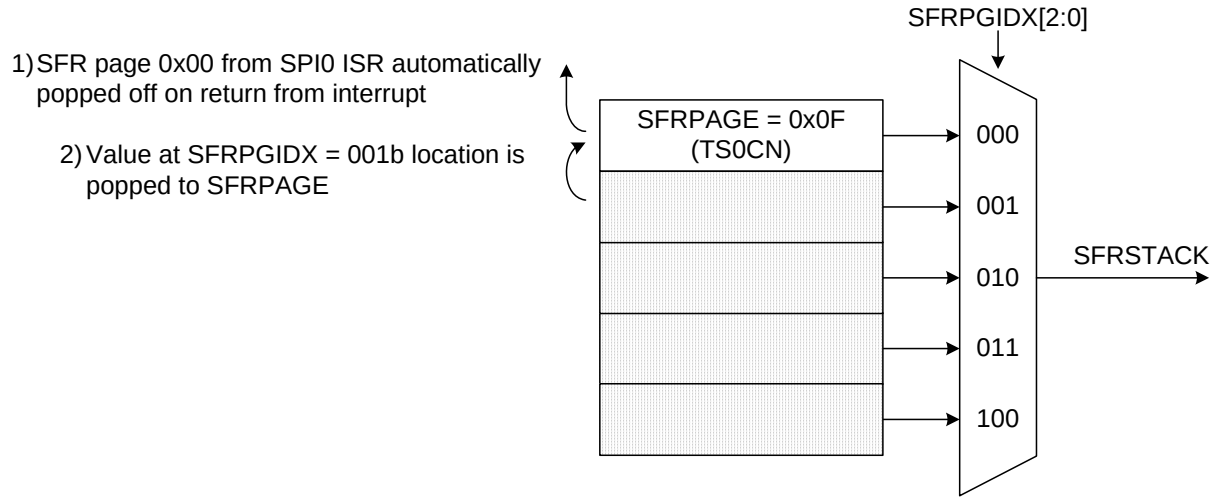

Figure 19.6. SFR Page Stack Upon Return From SPI0 Interrupt

Push operations on the SFR page stack only occur on interrupt service, and pop operations only occur on interrupt exit (execution on the RETI instruction). The automatic switching of the SFRPAGE and operation of the SFR page stack as described above can be disabled in software by clearing the SFR Automatic Page Enable Bit (SFRPGEN) in the SFR Page Control Register (SFRPGCN).

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: /INT1 input is active low. 1: /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 /INT1. Note that this pin assignment is independent of the Crossbar; /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

27.3. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 27.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.3 and/or P0.2 if the external oscillator circuit is enabled, P0.6 if the ADC or IDAC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or Comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

Figure 27.3 shows all of the potential peripheral-to-pin assignments available to the crossbar. Note that this does not mean any peripheral can always be assigned to the highlighted pins. The actual pin assignments are determined by the priority of the enabled peripherals.

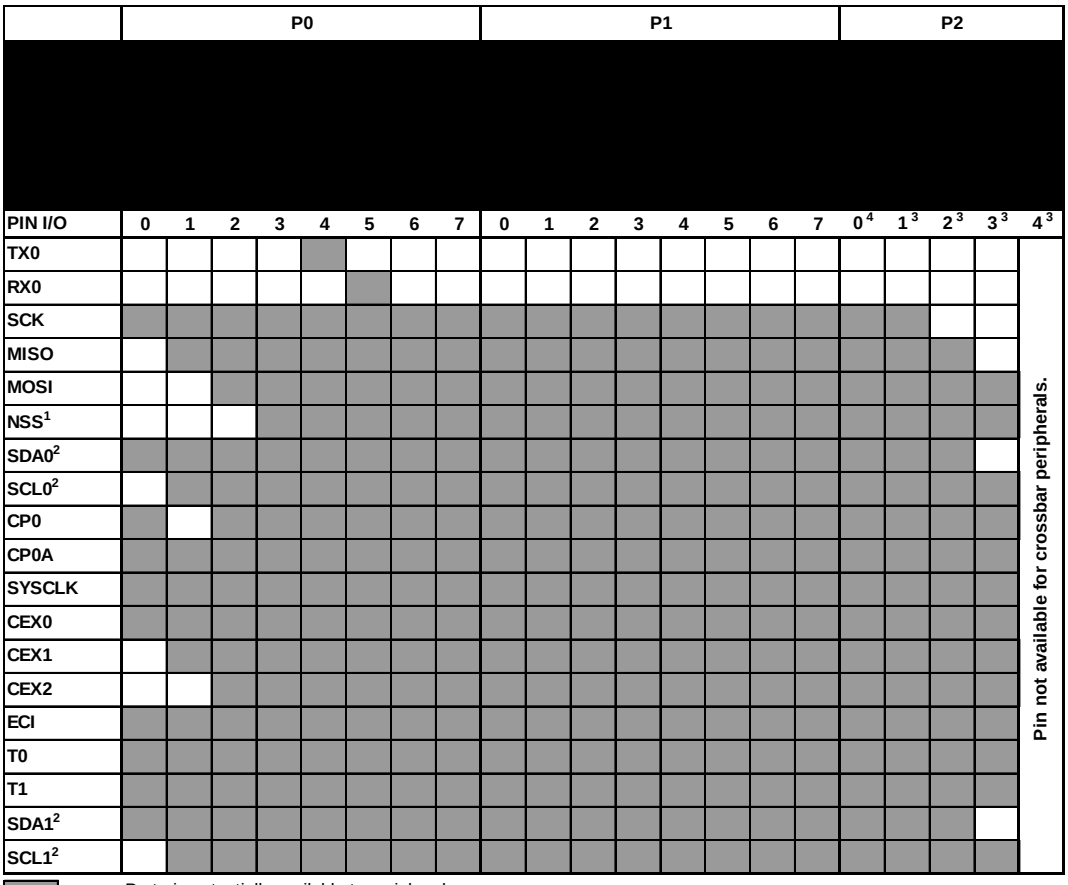

Figure 27.3. Crossbar Priority Decoder - Possible Pin Assignments

Table 28.3. Sources for Hardware Changes to SMBnCN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTERn	■ A START is generated.	■ A STOP is generated. ■ Arbitration is lost.
TXMODEn	■ START is generated. ■ SMBnDAT is written before the start of an SMBus frame.	■ A START is detected. ■ Arbitration is lost. ■ SMBnDAT is not written before the start of an SMBus frame.
STAn	■ A START followed by an address byte is received.	■ Must be cleared by software.
STOn	■ A STOP is detected while addressed as a slave. ■ Arbitration is lost due to a detected STOP.	■ A pending STOP is generated.
ACKRQn	■ A byte has been received and an ACK response value is needed (only when hardware ACK is not enabled).	■ After each ACK cycle.
ARBLOSTn	■ A repeated START is detected as a MASTER when STAn is low (unwanted repeated START). ■ SCLn is sensed low while attempting to generate a STOP or repeated START condition. ■ SDAn is sensed low while transmitting a 1 (excluding ACK bits).	■ Each time SIn is cleared.
ACKn	■ The incoming ACK value is low (ACKNOWLEDGE).	■ The incoming ACK value is high (NOT ACKNOWLEDGE).
SIn	■ A START has been generated. ■ Lost arbitration. ■ A byte has been transmitted and an ACK/NACK received. ■ A byte has been received. ■ A START or repeated START followed by a slave address + R/W has been received. ■ A STOP has been received.	■ Must be cleared by software.

28.4.5. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 28.4.4.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register and the SMBus Slave Address Mask register. A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in bit positions of the slave address mask SLVM[6:0] enable a comparison between the received slave address and the hardware's slave address SLV[6:0] for those bits. A 0 in a bit of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on

SFR Definition 28.7. SMB0ADM: SMBus0 Slave Address Mask

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

SFR Address = 0xE7; SFR Page = 0

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

SFR Definition 28.8. SMB1ADR: SMBus1 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV1[6:0]							GC1
Type	R/W							R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD7; SFR Page = F

Bit	Name	Function
7:1	SLV1[6:0]	**SMBus1 Hardware Slave Address.** Defines the SMBus1 Slave Address(es) for automatic hardware acknowledgment. Only address bits which have a 1 in the corresponding bit position in SLVM1[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC1	**General Call Address Enable.** When hardware address recognition is enabled (EHACK1 = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

SFR Definition 31.9. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; SFR Page = 0; Bit-Addressable

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Low-Frequency Oscillator Capture Enable.** When set to 1, this bit enables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Unused. Read = 0b; Write = Don't Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

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 32.5, where PCA0L is the value of the PCA0L register at the time of the update.

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

Equation 32.5. 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.

32.4.2. 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 32.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 32.3 lists some example timeout intervals for typical system clocks.

Table 32.3. Watchdog Timer Timeout Intervals[1]

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.

SFR Definition 32.2. 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 Address = 0xD9; SFR Page = All Pages

Bit	Name	Function
7	CIDL	**PCA Counter/Timer Idle Control.** 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	**Watchdog Timer Enable.** 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	**Watchdog Timer Lock.** 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	Unused. Read = 0b, Write = Don't care.
3:1	CPS[2:0]	**PCA Counter/Timer Pulse Select.** 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: Low frequency oscillator divided by 8 111: Reserved
0	ECF	**PCA Counter/Timer Overflow Interrupt Enable.** 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.
Note: 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.		