



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

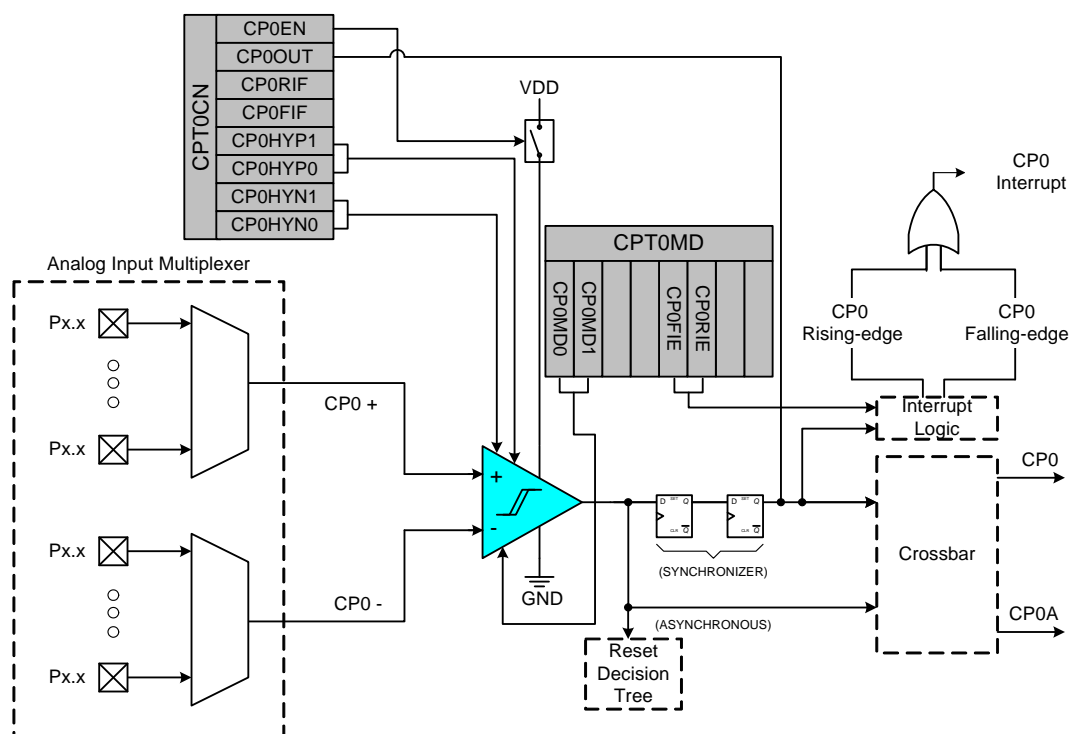
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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](#)"

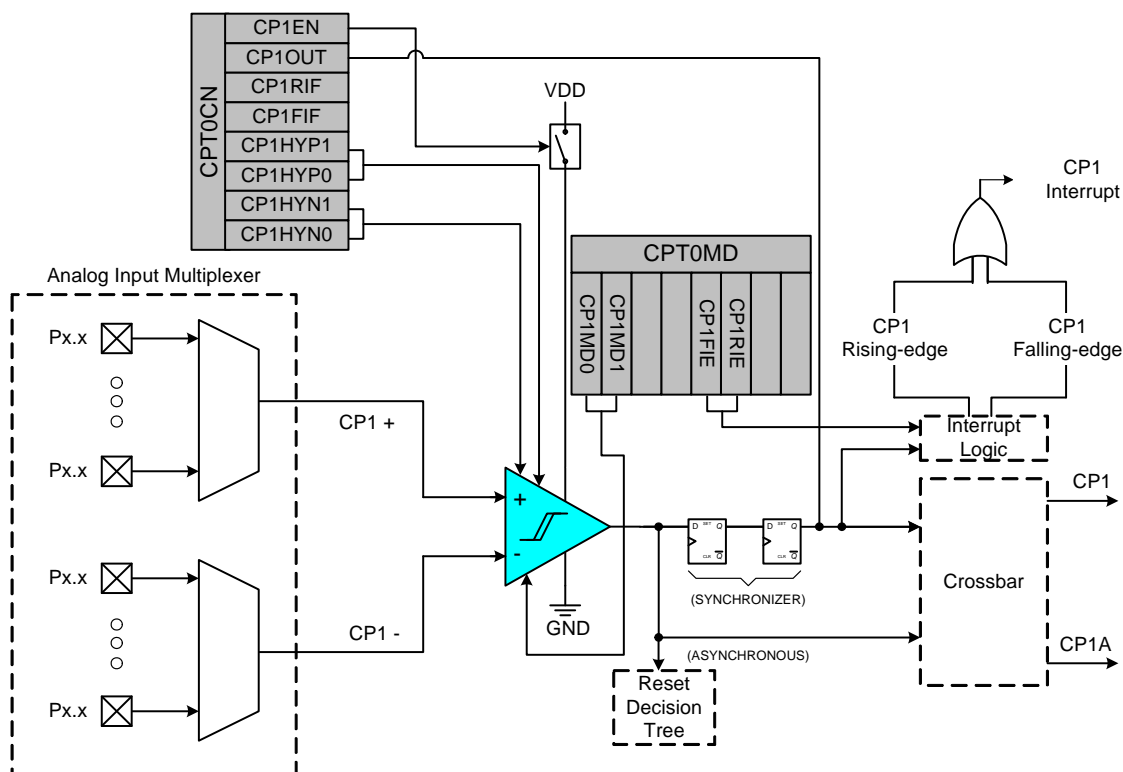
#### Details

Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	0.9V ~ 3.6V
Data Converters	A/D 15x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f911-gm">https://www.e-xfl.com/product-detail/silicon-labs/c8051f911-gm</a>

## C8051F91x-C8051F90x



**Figure 1.9. Comparator 0 Functional Block Diagram**



**Figure 1.10. Comparator 1 Functional Block Diagram**

# C8051F91x-C8051F90x

## 2. Ordering Information

**Table 2.1. Product Selection Guide**

Ordering Part Number	MIPS (Peak)	Flash Memory (kB)	RAM (bytes)	SmaRTClock Real Time Clock	SMBus/I <sup>2</sup> C	UART	Enhanced SPI	Timers (16-bit)	Programmable Counter Array	Digital Port I/Os	10-bit 300ksp ADC	Programmable Current Reference	Internal Voltage Reference	Temperature Sensor	Analog Comparators	Lead-free (RoHS Compliant)	C8051F9xx Plus Features*	Package
C8051F912-GM	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	QFN-24
C8051F912-GU	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	QSOP-24
C8051F912-GD	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	Tested Die
C8051F911-GM	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		QFN-24
C8051F911-GU	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		QSOP-24
C8051F911-GD	25	16	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		Tested Die
C8051F902-GM	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	QFN-24
C8051F902-GU	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	QSOP-24
C8051F902-GD	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓	✓	Tested Die
C8051F901-GM	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		QFN-24
C8051F901-GU	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		QSOP-24
C8051F901-GD	25	8	768	✓	1	1	2	4	✓	16	✓	✓	✓	✓	2	✓		Tested Die

\*The 'F9xx Plus features are a set of enhancements that allow greater power efficiency and increased functionality. They include 12-bit ADC mode, PWM Enhanced IREF, ultra-low power SmaRTClock LFO, VBAT input voltage from 0.9 to 3.6 V, and VBAT battery low indicator. The 'F9xx Plus features are described in detail in "AN431: F93x-F90x Software Porting Guide."

**Table 4.3. Port I/O DC Electrical Characteristics**

$V_{DD} = 1.8$  to  $3.6$  V,  $-40$  to  $+85$  °C unless otherwise specified.

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	High Drive Strength, PnDRV.n = 1				V
	IOH = $-3$ mA, Port I/O push-pull	$V_{DD} - 0.7$	—	—	
	IOH = $-10$ $\mu$ A, Port I/O push-pull	$V_{DD} - 0.1$	—	—	
	IOH = $-10$ mA, Port I/O push-pull		See Chart		
	Low Drive Strength, PnDRV.n = 0				
	IOH = $-1$ mA, Port I/O push-pull	$V_{DD} - 0.7$	—	—	
	IOH = $-10$ $\mu$ A, Port I/O push-pull	$V_{DD} - 0.1$	—	—	
	IOH = $-3$ mA, Port I/O push-pull	—	See Chart	—	
Output Low Voltage	High Drive Strength, PnDRV.n = 1				V
	I <sub>OL</sub> = $8.5$ mA	—	—	$0.6$	
	I <sub>OL</sub> = $10$ $\mu$ A	—	—	$0.1$	
	I <sub>OL</sub> = $25$ mA	—	See Chart	—	
	Low Drive Strength, PnDRV.n = 0				
	I <sub>OL</sub> = $1.4$ mA	—	—	$0.6$	
	I <sub>OL</sub> = $10$ $\mu$ A	—	—	$0.1$	
	I <sub>OL</sub> = $4$ mA	—	See Chart	—	
Input High Voltage	$V_{DD} = 2.0$ to $3.6$ V	$V_{DD} - 0.6$	—	—	V
	$V_{DD} = 0.9$ to $2.0$ V	$0.7 \times V_{DD}$	—	—	V
Input Low Voltage	$V_{DD} = 2.0$ to $3.6$ V	—	—	$0.6$	V
	$V_{DD} = 0.9$ to $2.0$ V	—	—	$0.3 \times V_{DD}$	V
Input Leakage Current	Weak Pullup Off	—	—	$\pm 1$	$\mu$ A
	Weak Pullup On, $V_{IN} = 0$ V, $V_{DD} = 1.8$ V	—	$4$	—	
	Weak Pullup On, $V_{IN} = 0$ V, $V_{DD} = 3.6$ V	—	$20$	$35$	

## 7. Comparators

C8051F91x-C8051F90x devices include two on-chip programmable voltage comparators: Comparator 0 (CPT0) is shown in Figure 7.1; Comparator 1 (CPT1) is shown in Figure 7.2. The two comparators operate identically, but may differ in their ability to be used as reset or wake-up sources. See the Reset Sources chapter and the Power Management chapter for details on reset sources and low power mode wake-up sources, respectively.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a digital synchronous “latched” output (CP0, CP1), or a digital asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output when the device is in some low power modes.

### 7.1. Comparator Inputs

Each Comparator performs an analog comparison of the voltage levels at its positive (CP0+ or CP1+) and negative (CP0- or CP1-) input. Both comparators support multiple port pin inputs multiplexed to their positive and negative comparator inputs using analog input multiplexers. The analog input multiplexers are completely under software control and configured using SFR registers. See Section “7.6. Comparator0 and Comparator1 Analog Multiplexers” on page 95 for details on how to select and configure Comparator inputs.

**Important Note About Comparator Inputs:** The Port pins selected as Comparator inputs should be configured as analog inputs and skipped by the Crossbar. See the Port I/O chapter for more details on how to configure Port I/O pins as Analog Inputs. The Comparator may also be used to compare the logic level of digital signals, however, Port I/O pins configured as digital inputs must be driven to a valid logic state (HIGH or LOW) to avoid increased power consumption.

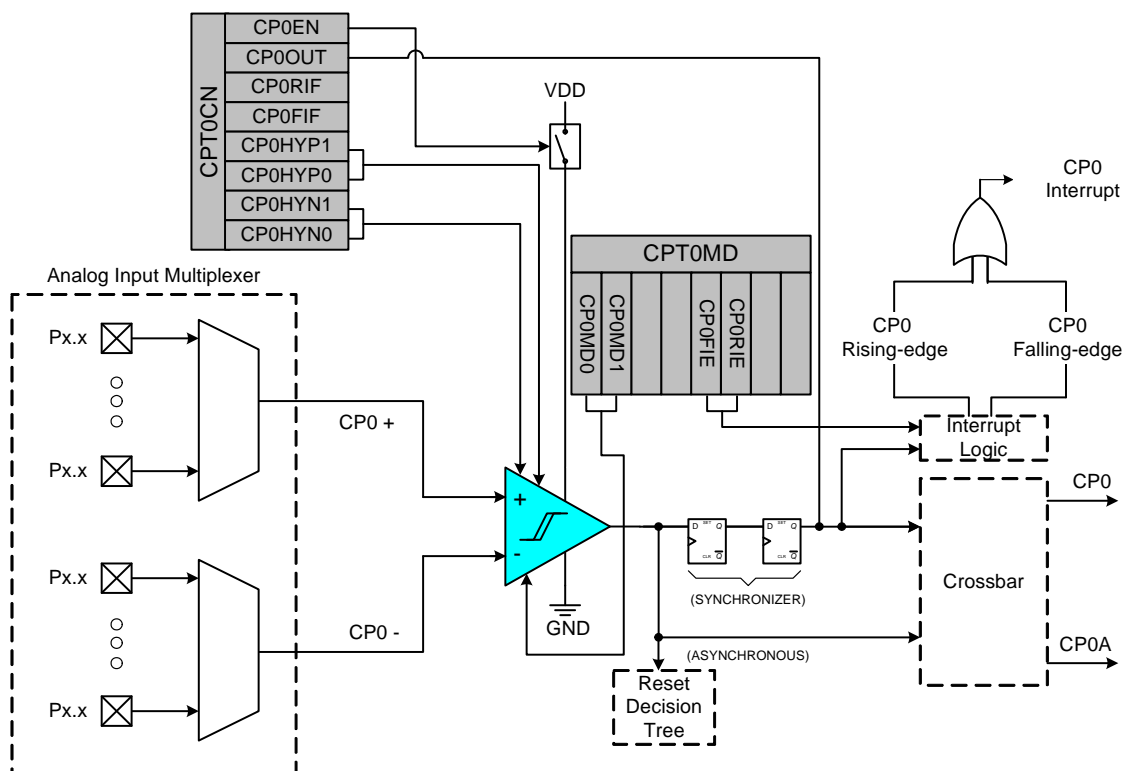


Figure 7.1. Comparator 0 Functional Block Diagram

# C8051F91x-C8051F90x

**Table 8.1. CIP-51 Instruction Set Summary**

Mnemonic	Description	Bytes	Clock Cycles
<b>Arithmetic Operations</b>			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
<b>Logical Operations</b>			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3

## 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.

# C8051F91x-C8051F90x

**Table 12.1. Interrupt Summary**

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 ( $\overline{\text{INT0}}$ )	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 ( $\overline{\text{INT1}}$ )	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON.0) TI0 (SCON.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
SmaRTClock Alarm	0x0043	8	ALRM (RTC0CN.2) <sup>2</sup>	N	N	EARTC0 (EIE1.1)	PARTC0 (EIP1.1)
ADC0 Window Comparator	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 End of Conversion	0x0053	10	AD0INT (ADC0STA.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
Comparator1	0x006B	13	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
Supply Monitor Early Warning	0x007B	15	VDDOK (VDM0CN.5) <sup>1</sup> <b>VBATOK (VDM0CN.4)</b> <sup>1, 3</sup>			EWARN (EIE2.0)	PWARN (EIP2.0)
Port Match	0x0083	16	None			EMAT (EIE2.1)	PMAT (EIP2.1)
SmaRTClock Oscillator Fail	0x008B	17	OSCFail (RTC0CN.5) <sup>2</sup>	N	N	ERTC0F (EIE2.2)	PFRTC0F (EIP2.2)
SPI1	0x0093	18	SPIF (SPI1CN.7) WCOL (SPI1CN.6) MODF (SPI1CN.5) RXOVRN (SPI1CN.4)	N	N	ESPI1 (EIE2.3)	PSPI1 (EIP2.3)

**Notes:**

1. Indicates a read-only interrupt pending flag. The interrupt enable may be used to prevent software from vectoring to the associated interrupt service routine.
2. Indicates a register located in an indirect memory space.
3. **8Blue** text Indicates a bit only available on 'F912 and 'F902 devices.



## 13. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 4.6 for complete Flash memory electrical characteristics.

### 13.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “27. C2 Interface” on page 312.

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before programming Flash memory using MOVX, Flash programming operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software. For detailed guidelines on programming Flash from firmware, please see Section “13.5. Flash Write and Erase Guidelines” on page 137.

To ensure the integrity of the Flash contents, the on-chip VDD Monitor must be enabled and enabled as a reset source in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Any attempt to write or erase Flash memory while the VDD Monitor is disabled, or not enabled as a reset source, will cause a Flash Error device reset.

#### 13.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 13.2.

## SFR Definition 15.4. CRC0AUTO: CRC0 Automatic Control

Bit	7	6	5	4	3	2	1	0
Name	AUTOEN	CRCDONE	CRC0ST[5:0]					
Type	R/W							R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x96

Bit	Name	Function
7	AUTOEN	<b>Automatic CRC Calculation Enable.</b> When AUTOEN is set to 1, any write to CRC0CN will initiate an automatic CRC starting at Flash sector CRC0ST and continuing for CRC0CNT sectors.
6	CRCDONE	<b>CRCDONE Automatic CRC Calculation Complete.</b> Set to 0 when a CRC calculation is in progress. Code execution is stopped during a CRC calculation; therefore, reads from firmware will always return 1.
5:0	CRC0ST[5:0]	<b>Automatic CRC Calculation Starting Flash Sector.</b> These bits specify the Flash sector to start the automatic CRC calculation. The starting address of the first Flash sector included in the automatic CRC calculation is CRC0ST x Page Size. <b>Note:</b> 'F91x and 'F90x devices have a page size of 512 bytes.

## 15.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 15.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

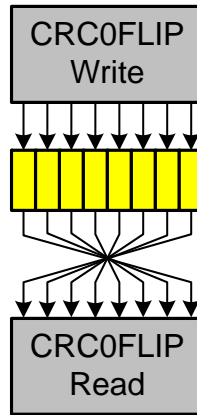


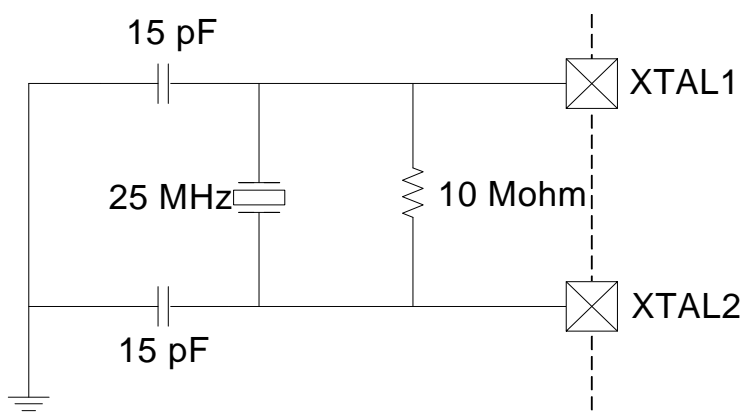
Figure 15.2. Bit Reverse Register

### SFR Definition 15.6. CRC0FLIP: CRC0 Bit Flip

Bit	7	6	5	4	3	2	1	0
Name	CRC0FLIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x95

Bit	Name	Function
7:0	CRC0FLIP[7:0]	<b>CRC0 Bit Flip.</b> Any byte written to CRC0FLIP is read back in a bit-reversed order, i.e. the written LSB becomes the MSB. For example: If 0xC0 is written to CRC0FLIP, the data read back will be 0x03. If 0x05 is written to CRC0FLIP, the data read back will be 0xA0.



**Figure 19.2. 25 MHz External Crystal Example**

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

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency. The selection should be based on Table 19.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

**Table 19.1. Recommended XFCN Settings for Crystal Mode**

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

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

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD => 1.
4. Switch the system clock to the external oscillator.

## 20.1. SmARTClock Interface

The SmARTClock Interface consists of three registers: RTC0KEY, RTC0ADR, and RTC0DAT. These interface registers are located on the CIP-51's SFR map and provide access to the SmARTClock internal registers listed in Table 20.1. The SmARTClock internal registers can only be accessed indirectly through the SmARTClock Interface.

**Table 20.1. SmARTClock Internal Registers**

SmaRTClock Address	SmaRTClock Register	Register Name	Description
0x00–0x03	CAPTUREn	SmaRTClock Capture Registers	Four Registers used for setting the 32-bit SmARTClock timer or reading its current value.
0x04	RTC0CN	SmaRTClock Control Register	Controls the operation of the SmARTClock State Machine.
0x05	RTC0XCN	SmaRTClock Oscillator Control Register	Controls the operation of the SmARTClock Oscillator. <b>Note:</b> Some bits in this register are only available on 'F912 and 'F902 devices.
0x06	RTC0XCF	SmaRTClock Oscillator Configuration Register	Controls the value of the programmable oscillator load capacitance and enables/disables AutoStep.
0x07	RTC0PIN	SmaRTClock Pin Configuration Register	Forces XTAL3 and XTAL4 to be internally shorted. <b>Note:</b> This register also contains other reserved bits which should not be modified.
0x08–0x0B	ALARMn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmARTClock alarm value.

### 20.1.1. SmARTClock Lock and Key Functions

The SmARTClock Interface is protected with a lock and key function. The SmARTClock Lock and Key Register (RTC0KEY) must be written with the correct key codes, in sequence, before writes and reads to RTC0ADR and RTC0DAT may be performed. The key codes are: 0xA5, 0xF1. There are no timing restrictions, but the key codes must be written in order. If the key codes are written out of order, the wrong codes are written, or an indirect register read or write is attempted while the interface is locked, the SmARTClock interface will be disabled, and the RTC0ADR and RTC0DAT registers will become inaccessible until the next system reset. Once the SmARTClock interface is unlocked, software may perform any number of accesses to the SmARTClock registers until the interface is re-locked or the device is reset. Any write to RTC0KEY while the SmARTClock interface is unlocked will re-lock the interface.

Reading the RTC0KEY register at any time will provide the SmARTClock Interface status and will not interfere with the sequence that is being written. The RTC0KEY register description in SFR Definition 20.1 lists the definition of each status code.

# C8051F91x-C8051F90x

## SFR Definition 21.8. P0: Port0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0x80; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	<b>Port 0 Data.</b> Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.

## SFR Definition 21.9. P0SKIP: Port0 Skip

Bit	7	6	5	4	3	2	1	0
Name	P0SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xD4

Bit	Name	Function
7:0	P0SKIP[7:0]	<b>Port 0 Crossbar Skip Enable Bits.</b> These bits select Port 0 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P0.n pin is not skipped by the Crossbar. 1: Corresponding P0.n pin is skipped by the Crossbar.

# C8051F91x-C8051F90x

## SFR Definition 21.15. P1MDIN: Port1 Input Mode

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

SFR Page = 0x0; SFR Address = 0xF2

Bit	Name	Function
7	Unused	<b>Unused.</b> Read =0b; Write = Don't Care.
6:0	P1MDIN[6:0]	<b>Analog Configuration Bits for P1.6–P1.0 (respectively).</b> Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

## SFR Definition 21.16. P1MDOUT: Port1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDOUT[6:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA5

Bit	Name	Function
7	Unused	<b>Unused.</b> Read =0b; Write = Don't Care.
6:0	P1MDOUT[6:0]	<b>Output Configuration Bits for P1.6–P1.0 (respectively).</b> These bits control the digital driver even when the corresponding bit in register P1MDIN is logic 0. 0: Corresponding P1.n Output is open-drain. 1: Corresponding P1.n Output is push-pull.

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 228). 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).



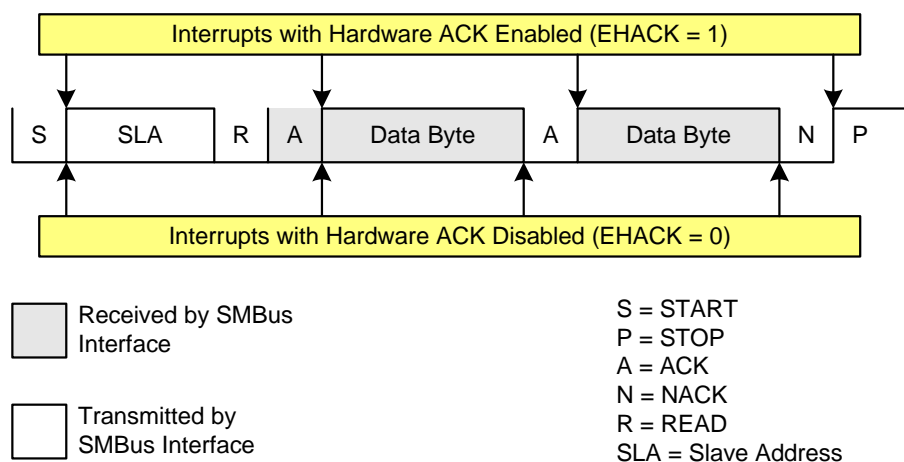
## 22.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 22.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.



**Figure 22.6. Typical Master Read Sequence**

## SFR Definition 23.2. SBUF0: Serial (UART0) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0[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 Page = 0x0; SFR Address = 0x99

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

**SFR Definition 25.1. CKCON: Clock Control**

Bit	7	6	5	4	3	2	1	0
Name	T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

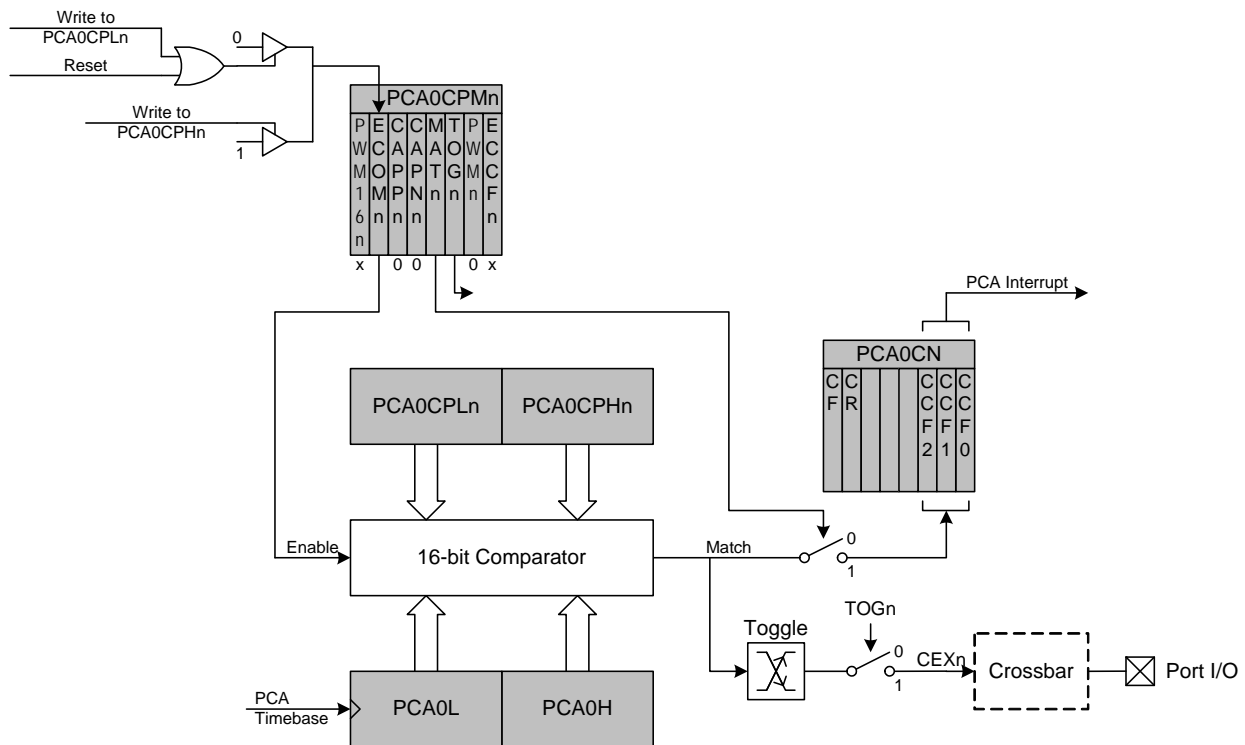
SFR Page = 0x0; SFR Address = 0x8E

Bit	Name	Function
7	T3MH	<b>Timer 3 High Byte Clock Select.</b> Selects the clock supplied to the Timer 3 high byte (split 8-bit timer mode only). 0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 high byte uses the system clock.
6	T3ML	<b>Timer 3 Low Byte Clock Select.</b> Selects the clock supplied to Timer 3. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 low byte uses the system clock.
5	T2MH	<b>Timer 2 High Byte Clock Select.</b> Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
4	T2ML	<b>Timer 2 Low Byte Clock Select.</b> Selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 low byte uses the system clock.
3	T1M	<b>Timer 1 Clock Select.</b> Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale bits SCA[1:0]. 1: Timer 1 uses the system clock.
2	T0M	<b>Timer 0 Clock Select.</b> Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale bits SCA[1:0]. 1: Counter/Timer 0 uses the system clock.
1:0	SCA[1:0]	<b>Timer 0/1 Prescale Bits.</b> These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)

## 26.3.3. High-Speed Output Mode

In High-speed output mode, a module's associated CEX<sub>n</sub> pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPH<sub>n</sub> and PCA0CPL<sub>n</sub>). When a match 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. Setting the TOG<sub>n</sub>, MAT<sub>n</sub>, and ECOM<sub>n</sub> bits in the PCA0CPM<sub>n</sub> register enables the High-Speed Output mode. If ECOM<sub>n</sub> is cleared, the associated pin will retain its state, and not toggle on the next match event.

**Important Note About Capture/Compare Registers:** When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPL<sub>n</sub> clears the ECOM<sub>n</sub> bit to 0; writing to PCA0CPH<sub>n</sub> sets ECOM<sub>n</sub> to 1.



**Figure 26.6. PCA High-Speed Output Mode Diagram**

The 8-bit offset held in PCA0CPH5 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 26.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$\text{Offset} = (256 \times \text{PCA0CPL5}) + (256 - \text{PCA0L})$$

## Equation 26.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH5 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF5 flag (PCA0CN.5) while the WDT is enabled.

### 26.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a 0 to the WDTE bit.
- Select the desired PCA clock source (with the CPS2–CPS0 bits).
- Load PCA0CPL5 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to 1.
- Reset the WDT timer by writing to PCA0CPH5.

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 PCA0CPL5 defaults to 0x00. Using Equation 26.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 26.3 lists some example timeout intervals for typical system clocks.

**Table 26.3. Watchdog Timer Timeout Intervals**

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500*	255	257
3,062,500*	128	129.5
3,062,500*	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168
*Note: Internal SYSCLK reset frequency = Internal Oscillator divided by 8.		