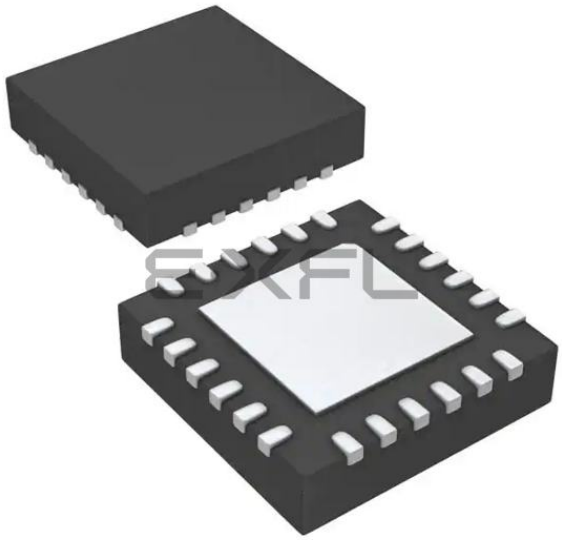


Welcome to E-XFL.COM

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	CIP-51 8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, 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	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 10x12b
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	https://www.e-xfl.com/product-detail/silicon-labs/c8051f986-c-gm

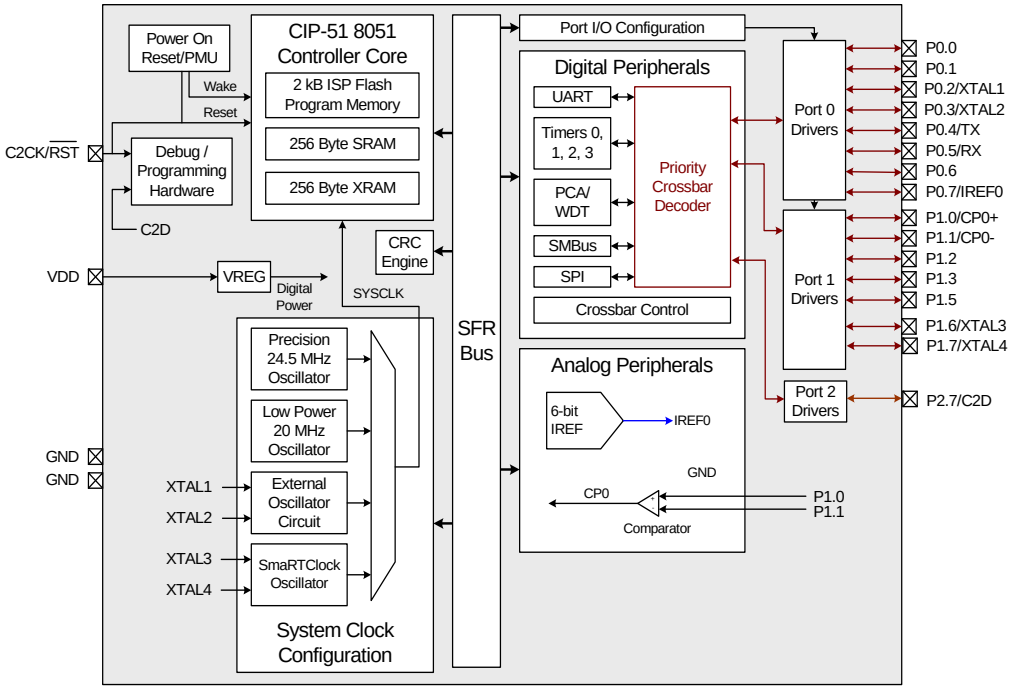


Figure 1.5. C8051F985 Block Diagram

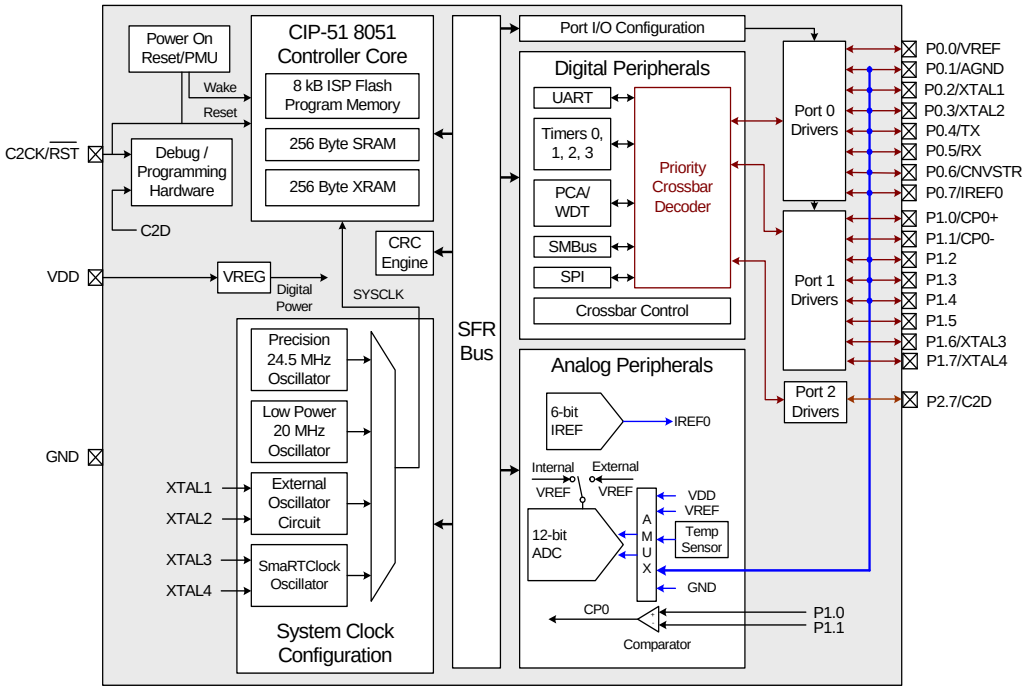
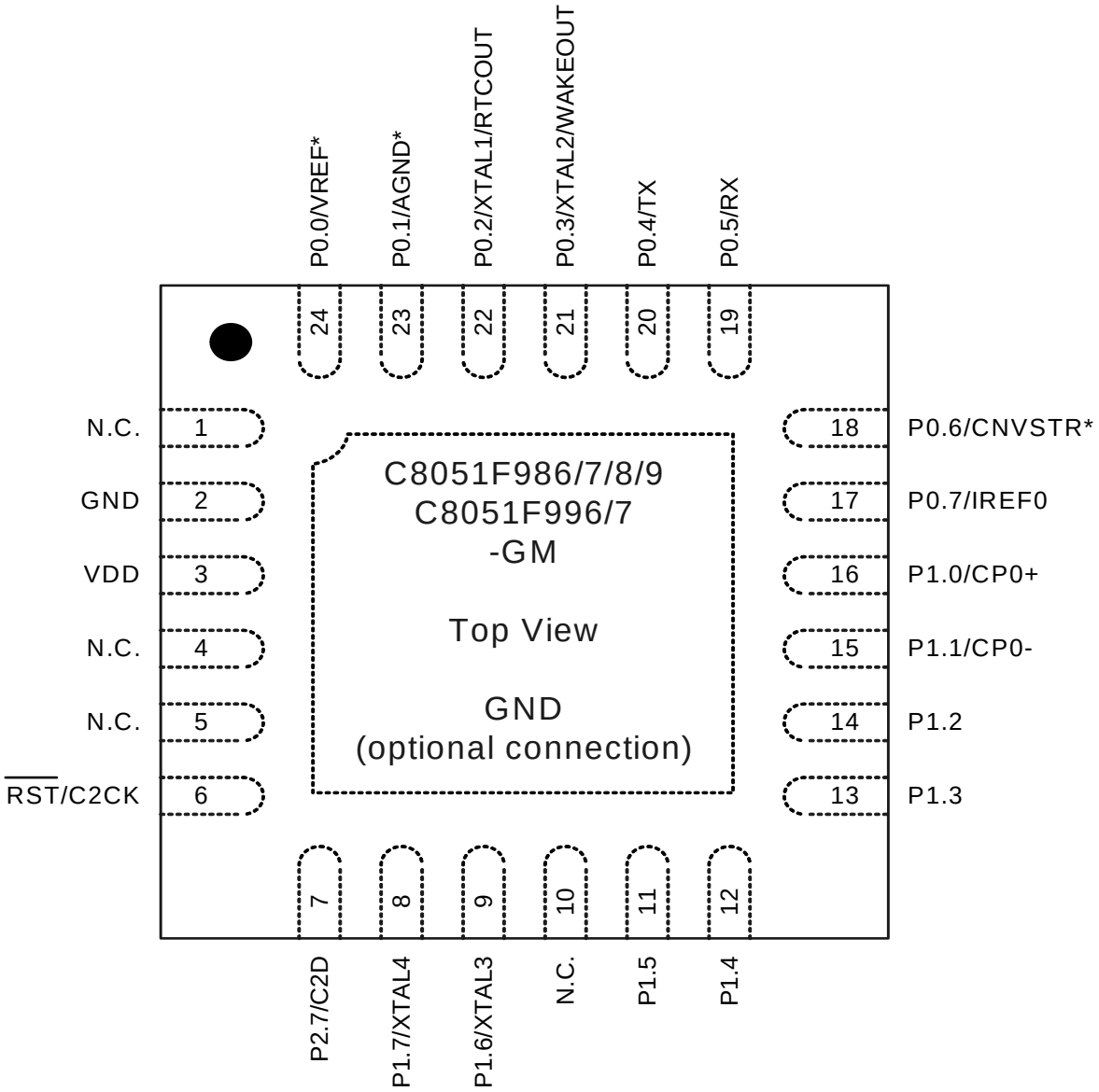


Figure 1.6. C8051F986 Block Diagram



***Note:** Signal only available on ‘F986, ‘F988, and ‘F996 devices.

Figure 3.2. QFN-24 Pinout Diagram (Top View)

4. Electrical Characteristics

Throughout the Electrical Characteristics chapter, “VDD” refers to the Supply Voltage.

4.1. Absolute Maximum Specifications

Table 4.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient Temperature under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any Port I/O Pin or $\overline{\text{RST}}$ with Respect to GND		–0.3	—	V_{DD} + 0.3	V
Voltage on V_{DD} with Respect to GND		–0.3	—	4.0	V
Maximum Total Current through V_{DD} or GND		—	—	500	mA
Maximum Current through $\overline{\text{RST}}$ or any Port Pin		—	—	100	mA
Maximum Total Current through all Port Pins		—	—	200	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

5.2. Modes of Operation

ADC0 has a maximum conversion speed of 300 ksps 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.

SFR Definition 5.5. ADC0TK: ADC0 Burst Mode Track Time

Bit	7	6	5	4	3	2	1	0
Name	Reserved		AD0TK[5:0]					
Type	R	R	R/W					
Reset	0	0	0	1	1	1	1	0

SFR Page = All; SFR Address = 0xBC

Bit	Name	Function
7	Reserved	Read = 0b; Write = Must Write 0b.
6	Unused	Read = 0b; Write = Don't Care.
5:0	AD0TK[5:0]	**ADC0 Burst Mode Track Time.** Sets the time delay between consecutive conversions performed in Burst Mode. The ADC0 Burst Mode Track time is programmed according to the following equation: $AD0TK = 63 - \left(\frac{Ttrack}{50ns} - 1\right)$ or $Ttrack = (64 - AD0TK)50ns$

Notes:

1. If AD0TM is set to 1, an additional 3 SAR clock cycles of Track time will be inserted prior to starting the conversion.
2. The Burst Mode Track delay is not inserted prior to the first conversion. The required tracking time for the first conversion should be met by the Burst Mode Power-Up Time.

6. Programmable Current Reference (IREF0)

C8051F99x-C8051F98x devices include an on-chip programmable current reference (source or sink) with two output current settings: Low Power Mode and High Current Mode. The maximum current output in Low Power Mode is 63 µA (1 µA steps) and the maximum current output in High Current Mode is 504 µA (8 µA steps).

The current source/sink is controlled though the IREF0CN special function register. It is enabled by setting the desired output current to a non-zero value. It is disabled by writing 0x00 to IREF0CN. The port I/O pin associated with ISRC0 should be configured as an analog input and skipped in the Crossbar. See Section "21. Port Input/Output" on page 215 for more details.

SFR Definition 6.1. IREF0CN: Current Reference Control

Bit	7	6	5	4	3	2	1	0
Name	SINK	MODE	IREF0DAT					
Type	R/W	R/W	R/W					
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD6

Bit	Name	Function
7	SINK	**IREF0 Current Sink Enable.** Selects if IREF0 is a current source or a current sink. 0: IREF0 is a current source. 1: IREF0 is a current sink.
6	MDSEL	**IREF0 Output Mode Select.** Selects Low Power or High Current Mode. 0: Low Power Mode is selected (step size = 1 µA). 1: High Current Mode is selected (step size = 8 µA).
5:0	IREF0DAT[5:0]	**IREF0 Data Word.** Specifies the number of steps required to achieve the desired output current. Output current = direction x step size x IREF0DAT. IREF0 is in a low power state when IREF0DAT is set to 0x00.

6.1. PWM Enhanced Mode

The precision of the current reference can be increased by fine tuning the IREF0 output using a PWM signal generated by the PCA. This mode allows the IREF0DAT bits to perform a course adjustment on the IREF0 output. Any available PCA channel can perform a fine adjustment on the IREF0 output. When enabled (PWMEN = 1), the CEX signal selected using the PWMSS bit field is internally routed to IREF0 to control the on time of a current source having the weight of 2 LSBs. With the two least significant bits of IREF0DAT set to 00b, applying a 100% duty cycle on the CEX signal will be equivalent to setting the two LSBs of IREF0DAT to 10b. PWM enhanced mode is enabled and setup using the IREF0CF register.

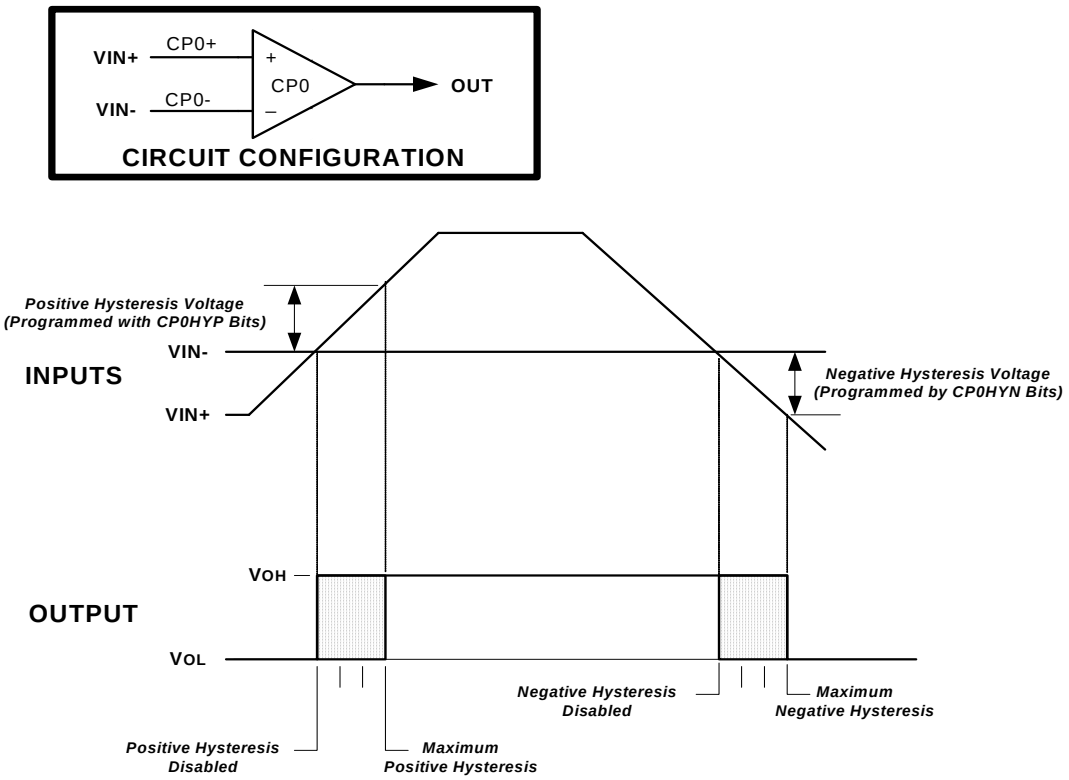


Figure 7.2. Comparator Hysteresis Plot

7.5. Comparator Register Descriptions

The SFRs used to enable and configure the comparator are described in the following register descriptions. The comparator must be enabled by setting the CP0EN bit to logic 1 before it can be used. From an enabled state, a comparator can be disabled and placed in a low power state by clearing the CP0EN bit to logic 0.

Important Note About Comparator Settings: False rising and falling edges can be detected by the Comparator while powering 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. The Comparator Power Up Time is specified in Section “Table 4.14. Comparator Electrical Characteristics” on page 63.

SFR Definition 7.3. CPT0MX: Comparator0 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Page = 0x0; SFR Address = 0x9F

Bit	Name	Function
7:4	CMX0N	**Comparator0 Negative Input Selection.** Selects the negative input channel for Comparator0. 0000: Reserved 0001: Reserved 0010: Reserved 0011: Reserved 0100: P1.1 0101: Reserved 0110: Reserved 0111: Reserved 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Capacitive Touch Sense Compare 1101: VDD divided by 2 1110: Digital Supply Voltage 1111: Ground
3:0	CMX0P	**Comparator0 Positive Input Selection.** Selects the positive input channel for Comparator0. 0000: Reserved 0001: Reserved 0010: Reserved 0011: Reserved 0100: P1.0 0101: Reserved 0110: Reserved 0111: Reserved 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Capacitive Touch Sense Compare 1101: VDD divided by 2 1110: VDD Supply Voltage 1111: VDD Supply Voltage

Table 13.1. 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 ($\overline{INT0}$)	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 ($\overline{INT1}$)	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 (SCON0.0) TI0 (SCON0.1)	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)[2]	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 (ADC0CN.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)
Reserved	0x006B	13					
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)[1]	N	N	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)[2]	N	N	ERTC0F (EIE2.2)	PFRTC0F (EIP2.2)
Reserved	0x0093	18					
CS0 Conversion Complete	0x009B	19	CS0INT (CS0CN.5)	Y	N	ECSCPT (EIE2.4)	PCSCPT (EIP2.4)
CS0 Digital Comparator	0x00A3	20	CS0CMPF (CS0CN.0)	Y	N	ECSDC (EIE2.5)	PCSDC (EIP2.5)
CS0 End of Scan	0x00AB	21	CS0EOS (CS0CN.6)	Y	N	ECSEOS (EIE2.6)	PCSEOS (EIP2.6)

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.

SFR Definition 13.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3		ECP0	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All; SFR Address = 0xE6

Bit	Name	Function
7	ET3	**Enable Timer 3 Interrupt.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6	Unused	Read = 0b. Write = Don't care.
5	ECP0	**Enable Comparator0 (CP0) Interrupt.** This bit sets the masking of the CP0 interrupt. 0: Disable CP0 interrupts. 1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.
4	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**Enable ADC0 Conversion Complete Interrupt.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the AD0INT flag.
2	EWADC0	**Enable Window Comparison ADC0 Interrupt.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).
1	ERTC0A	**Enable SmaRTClock Alarm Interrupts.** This bit sets the masking of the SmaRTClock Alarm interrupt. 0: Disable SmaRTClock Alarm interrupts. 1: Enable interrupt requests generated by a SmaRTClock Alarm.
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

The 16-bit C8051F99x-C8051F98x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

Table 16.1 lists several input values and the associated outputs using the 16-bit C8051F99x-C8051F98x CRC algorithm:

Table 16.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

16.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRC0VAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRC0INIT).

16.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks. The following steps can be used to automatically perform a CRC on Flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRC0CNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute any additional code until the CRC operation completes. **See the note in SFR Definition 16.1. CRC0CN: CRC0 Control for more information on how to properly initiate a CRC calculation.**
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result using the procedure below.

16.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

SFR Definition 19.4. OSCXCN: External Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	XCLKVLD	XOSCMD[2:0]			Reserved	XFCN[2:0]		
Type	R	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 = 0xB1

Bit	Name	Function
7	XCLKVLD	**External Oscillator Valid Flag.** Provides External Oscillator status and is valid at all times for all modes of operation except External CMOS Clock Mode and External CMOS Clock Mode with divide by 2. In these modes, XCLKVLD always returns 0. 0: External Oscillator is unused or not yet stable. 1: External Oscillator is running and stable.
6:4	XOSCMD	**External Oscillator Mode Bits.** Configures the external oscillator circuit to the selected mode. 00x: External Oscillator circuit disabled. 010: External CMOS Clock Mode. 011: External CMOS Clock Mode with divide by 2 stage. 100: RC Oscillator Mode. 101: Capacitor Oscillator Mode. 110: Crystal Oscillator Mode. 111: Crystal Oscillator Mode with divide by 2 stage.
3	Reserved	Read = 0b. Must Write 0b.
2:0	XFCN	**External Oscillator Frequency Control Bits.** Controls the external oscillator bias current. 000-111: See Table 19.1 on page 190 (Crystal Mode) or Table 19.2 on page 191 (RC or C Mode) for recommended settings.

22. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compatible with the System Management Bus Specification, version 1.1, and compatible with the I^2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

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

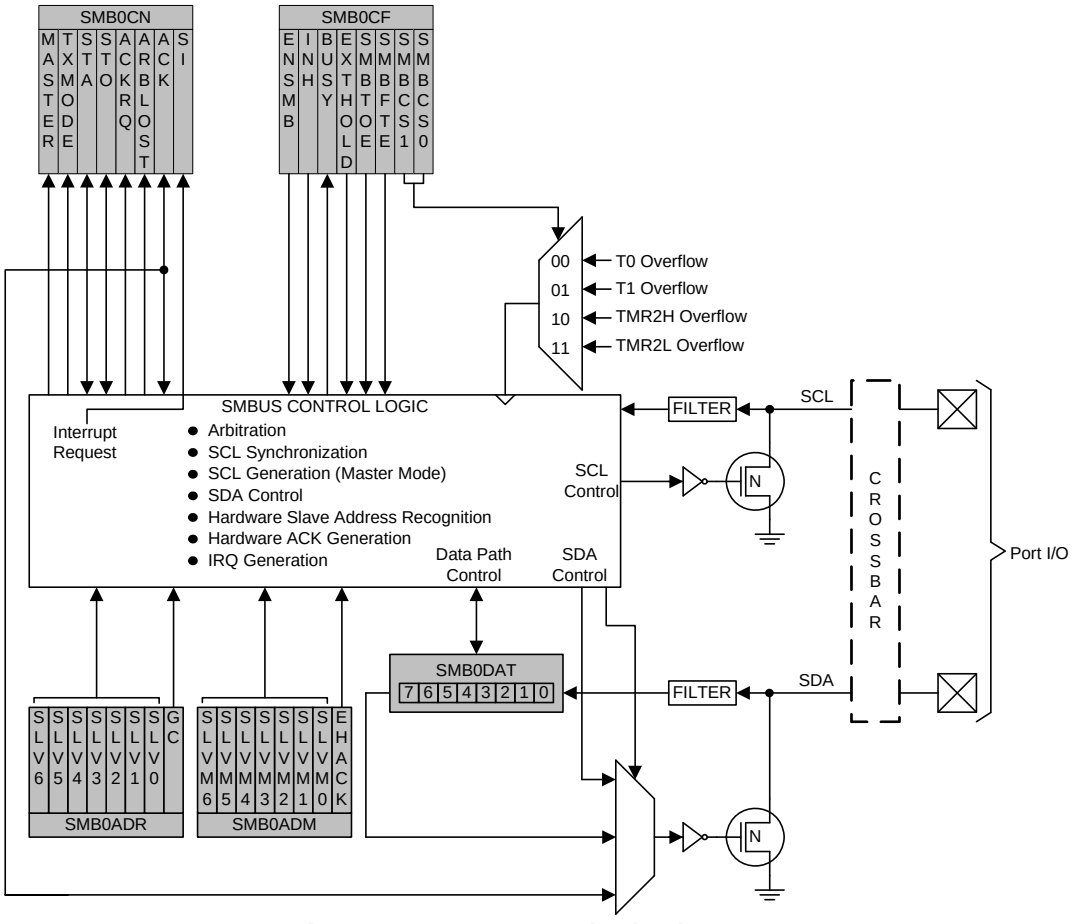


Figure 22.1. SMBus Block Diagram

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

***Note:** 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).

SFR Definition 25.2. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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 = 0x88; Bit-Addressable

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured $\overline{\text{INT1}}$ interrupt will be edge or level sensitive. $\overline{\text{INT1}}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 13.7). 0: $\overline{\text{INT1}}$ is level triggered. 1: $\overline{\text{INT1}}$ is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 13.7). 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

SFR Definition 25.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	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 = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 13.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 13.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

26. Programmable Counter Array

The programmable counter array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and three 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled. The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, SmaRTClock divided by 8, Timer 0 overflows, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8 to 11-Bit PWM, or 16-Bit PWM (each mode is described in Section “26.3. Capture/Compare Modules” on page 303). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 26.1

Important Note: The PCA Module 2 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See Section 26.4 for details.

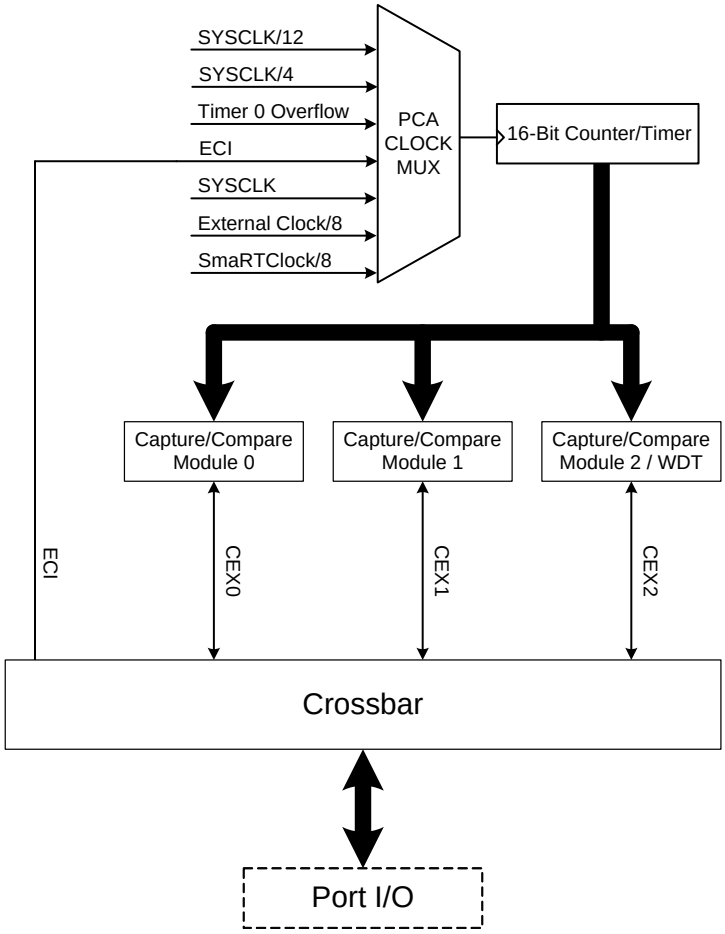


Figure 26.1. PCA Block Diagram

26.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 26.4.

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 PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 26.4. 16-Bit PWM Duty Cycle

Using Equation 26.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

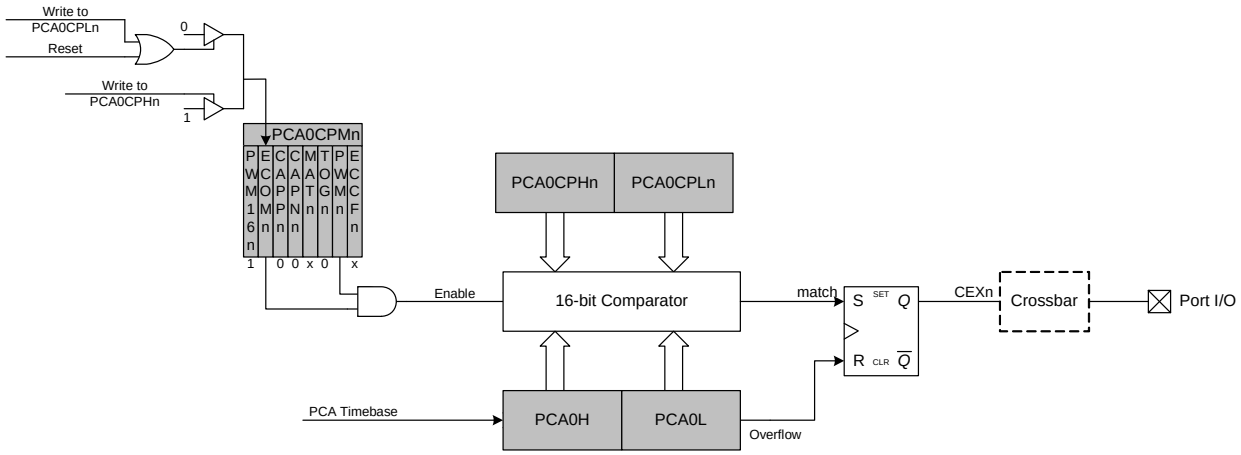


Figure 26.10. PCA 16-Bit PWM Mode

26.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 26.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR				CCF2	CCF1	CCF0
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 = 0xD8; Bit-Addressable

Bit	Name	Function
7	CF	**PCA Counter/Timer Overflow Flag.** Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA Counter/Timer Run Control.** This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled. 1: PCA Counter/Timer enabled.
5:3	Unused	Read = 000b, Write = don't care.
2:0	CCF[2:0]	**PCA Module n Capture/Compare Flag.** These bits are set by hardware when a match or capture occurs in the associated PCA Module n. When the CCFn interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.