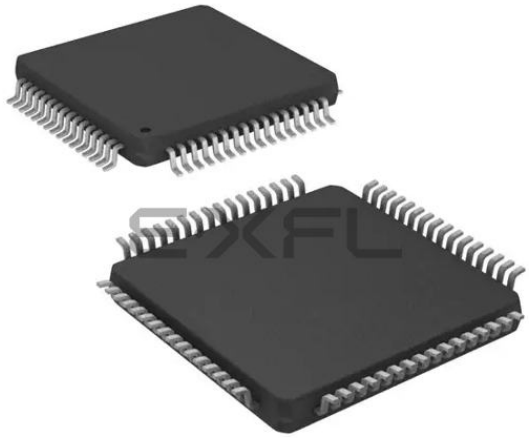


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	100MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f133-gqr

1.1.3. Additional Features

Several key enhancements are implemented in the CIP-51 core and peripherals to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 20 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

There are up to seven reset sources for the MCU: an on-board V_{DD} monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR0 input pin, and the $\overline{RST}$ pin. The $\overline{RST}$ pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the $\overline{RST}$ pin. Each reset source except for the V_{DD} monitor and Reset Input pin may be disabled by the user in software; the V_{DD} monitor is enabled/disabled via the MONEN pin. The Watchdog Timer may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand alone clock generator which is used by default as the system clock after any reset. If desired, the clock source may be switched on the fly to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the 24.5 MHz internal oscillator as needed. Additionally, an on-chip PLL is provided to achieve higher system clock speeds for increased throughput.

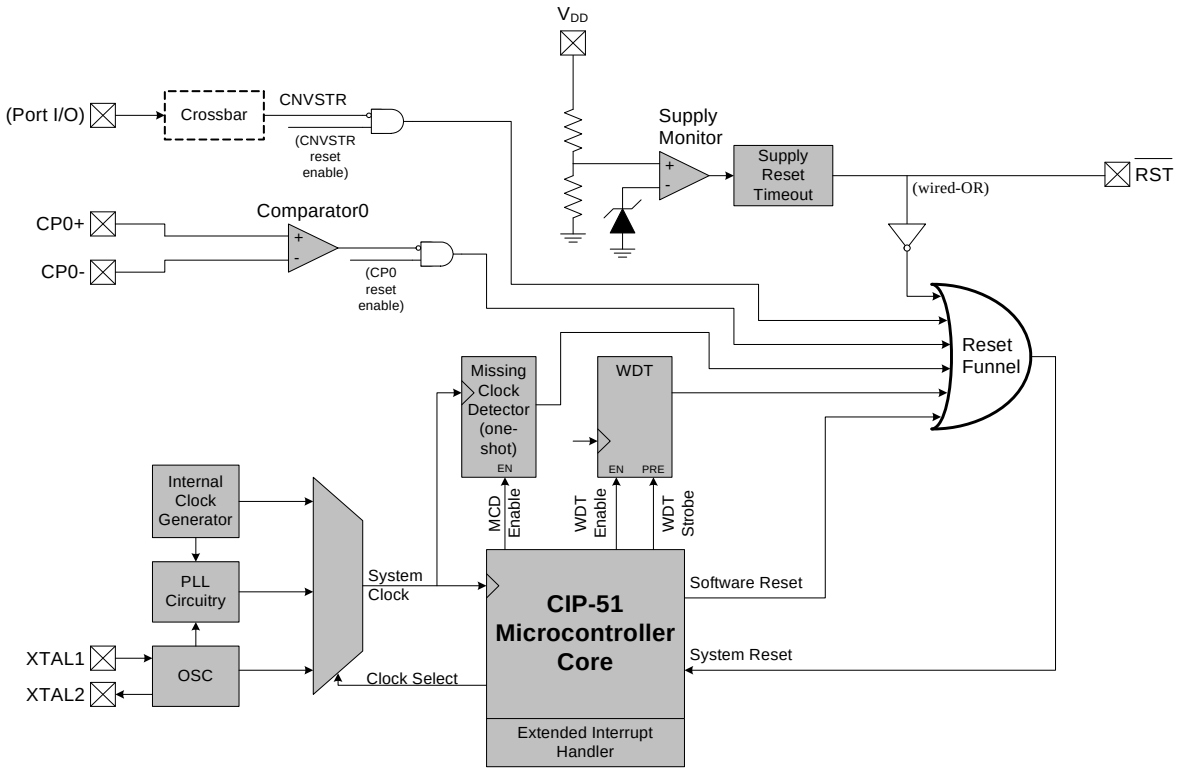


Figure 1.7. On-Board Clock and Reset

1.10. 12-bit Digital to Analog Converters

The C8051F12x devices have two integrated 12-bit Digital to Analog Converters (DACs). The MCU data and control interface to each DAC is via the Special Function Registers. The MCU can place either or both of the DACs in a low power shutdown mode.

The DACs are voltage output mode and include a flexible output scheduling mechanism. This scheduling mechanism allows DAC output updates to be forced by a software write or scheduled on a Timer 2, 3, or 4 overflow. The DAC voltage reference is supplied from the dedicated VREFD input pin on the 100-pin TQFP devices or via the internal Voltage reference on the 64-pin TQFP devices. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADCs.

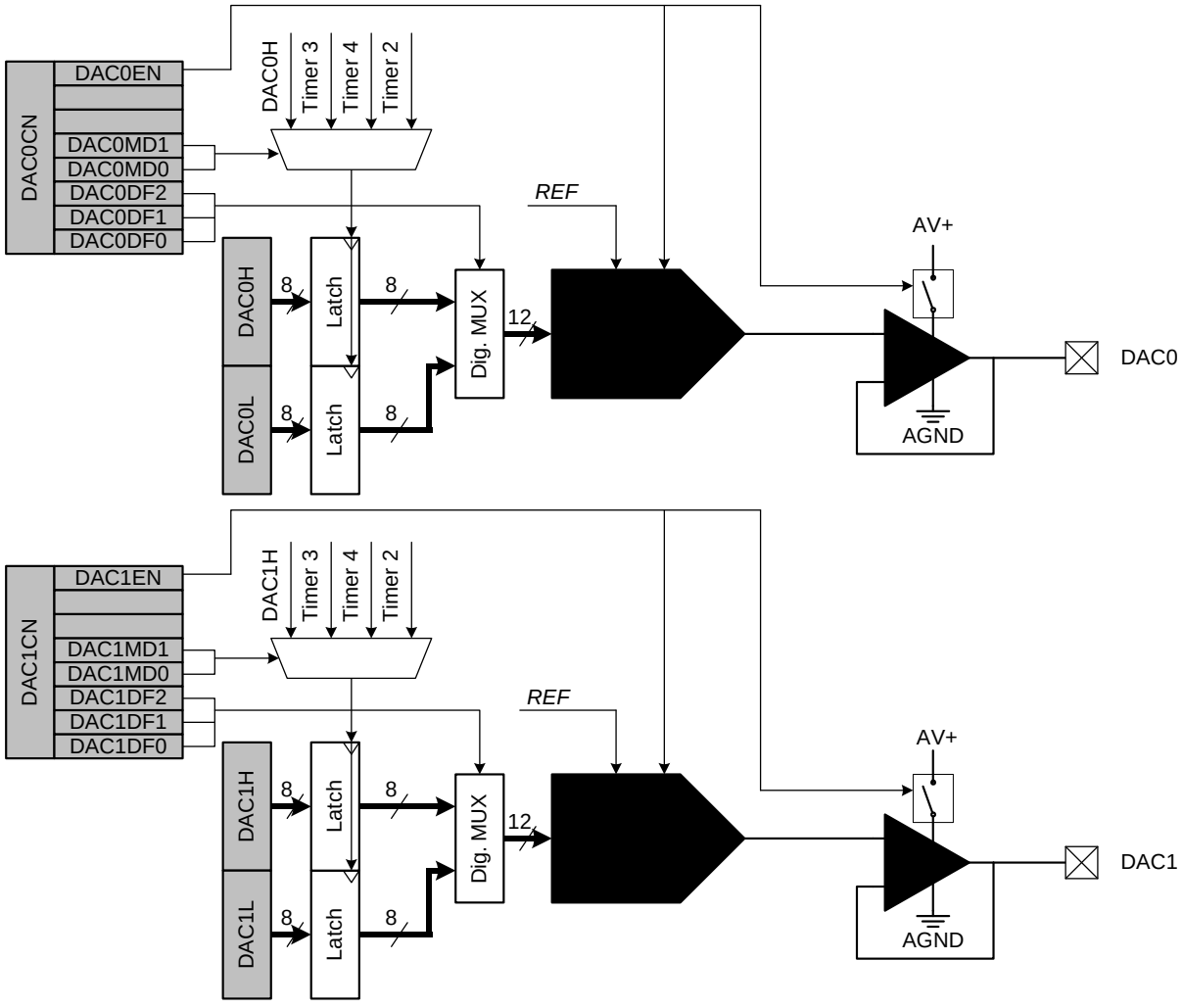


Figure 1.15. DAC System Block Diagram

SFR Definition 6.1. AMX0CF: AMUX0 Configuration

SFR Page: 0
SFR Address: 0xBA

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bit3: AIN67IC: AIN0.6, AIN0.7 Input Pair Configuration Bit.
0: AIN0.6 and AIN0.7 are independent single-ended inputs.
1: AIN0.6, AIN0.7 are (respectively) +, - differential input pair.
Bit2: AIN45IC: AIN0.4, AIN0.5 Input Pair Configuration Bit.
0: AIN0.4 and AIN0.5 are independent single-ended inputs.
1: AIN0.4, AIN0.5 are (respectively) +, - differential input pair.
Bit1: AIN23IC: AIN0.2, AIN0.3 Input Pair Configuration Bit.
0: AIN0.2 and AIN0.3 are independent single-ended inputs.
1: AIN0.2, AIN0.3 are (respectively) +, - differential input pair.
Bit0: AIN01IC: AIN0.0, AIN0.1 Input Pair Configuration Bit.
0: AIN0.0 and AIN0.1 are independent single-ended inputs.
1: AIN0.0, AIN0.1 are (respectively) +, - differential input pair.

Note: The ADC0 Data Word is in 2's complement format for channels configured as differential.

SFR Definition 7.5. ADC2: ADC2 Data Word

SFR Page: 2
SFR Address: 0xBE

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–0: ADC2 Data Word.

Single-Ended Example:
8-bit ADC Data Word appears in the ADC2 Data Word Register as follows:
Example: ADC2 Data Word Conversion Map, Single-Ended AIN2.0 Input
(AMX2CF = 0x00; AMX2SL = 0x00)

AIN2.0–AGND (Volts)	ADC2
VREF * (255/256)	0xFF
VREF * (128/256)	0x80
VREF * (64/256)	0x40
0	0x00

$$Code = Vin \times \frac{Gain}{VREF} \times 256$$

Differential Example:
8-bit ADC Data Word appears in the ADC2 Data Word Register as follows:
Example: ADC2 Data Word Conversion Map, Differential AIN2.0-AIN2.1 Input
(AMX2CF = 0x01; AMX2SL = 0x00)

AIN2.0–AIN2.1 (Volts)	ADC2
VREF * (127/128)	0x7F
VREF * (64/128)	0x40
0	0x00
–VREF * (64/128)	0xC0 (-64d)
–VREF * (128/128)	0x80 (-128d)

$$Code = Vin \times \frac{Gain}{2 \times VREF} \times 256$$

Figure 7.4. ADC2 Data Word Example

SFR Definition 8.3. DAC0CN: DAC0 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
DAC0EN	-	-	DAC0MD1	DAC0MD0	DAC0DF2	DAC0DF1	DAC0DF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD4
SFR Page: 0

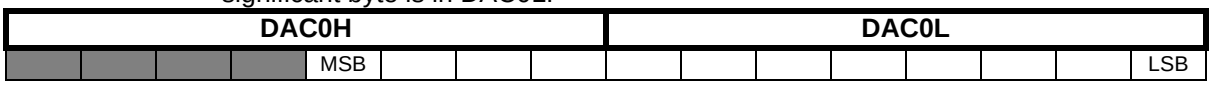
Bit7: DAC0EN: DAC0 Enable Bit.
0: DAC0 Disabled. DAC0 Output pin is disabled; DAC0 is in low-power shutdown mode.
1: DAC0 Enabled. DAC0 Output pin is active; DAC0 is operational.

Bits6–5: UNUSED. Read = 00b; Write = don't care.

Bits4–3: DAC0MD1–0: DAC0 Mode Bits.
00: DAC output updates occur on a write to DAC0H.
01: DAC output updates occur on Timer 3 overflow.
10: DAC output updates occur on Timer 4 overflow.
11: DAC output updates occur on Timer 2 overflow.

Bits2–0: DAC0DF2–0: DAC0 Data Format Bits:

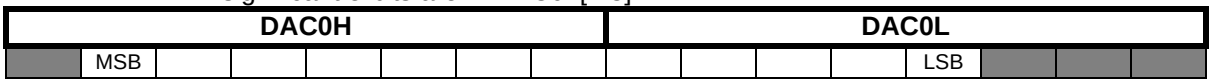
000: The most significant nibble of the DAC0 Data Word is in DAC0H[3:0], while the least significant byte is in DAC0L.



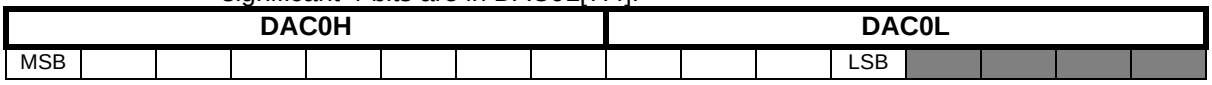
001: The most significant 5-bits of the DAC0 Data Word is in DAC0H[4:0], while the least significant 7-bits are in DAC0L[7:1].

010: The most significant 6-bits of the DAC0 Data Word is in DAC0H[5:0], while the least significant 6-bits are in DAC0L[7:2].

011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits are in DAC0L[7:3].



1xx: The most significant 8-bits of the DAC0 Data Word is in DAC0H[7:0], while the least significant 4-bits are in DAC0L[7:4].



10. Comparators

Two on-chip programmable voltage comparators are included, as shown in Figure 10.1. The inputs of each comparator are available at dedicated pins. The output of each comparator is optionally available at the package pins via the I/O crossbar. When assigned to package pins, each comparator output can be programmed to operate in open drain or push-pull modes. See **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** for Crossbar and port initialization details.

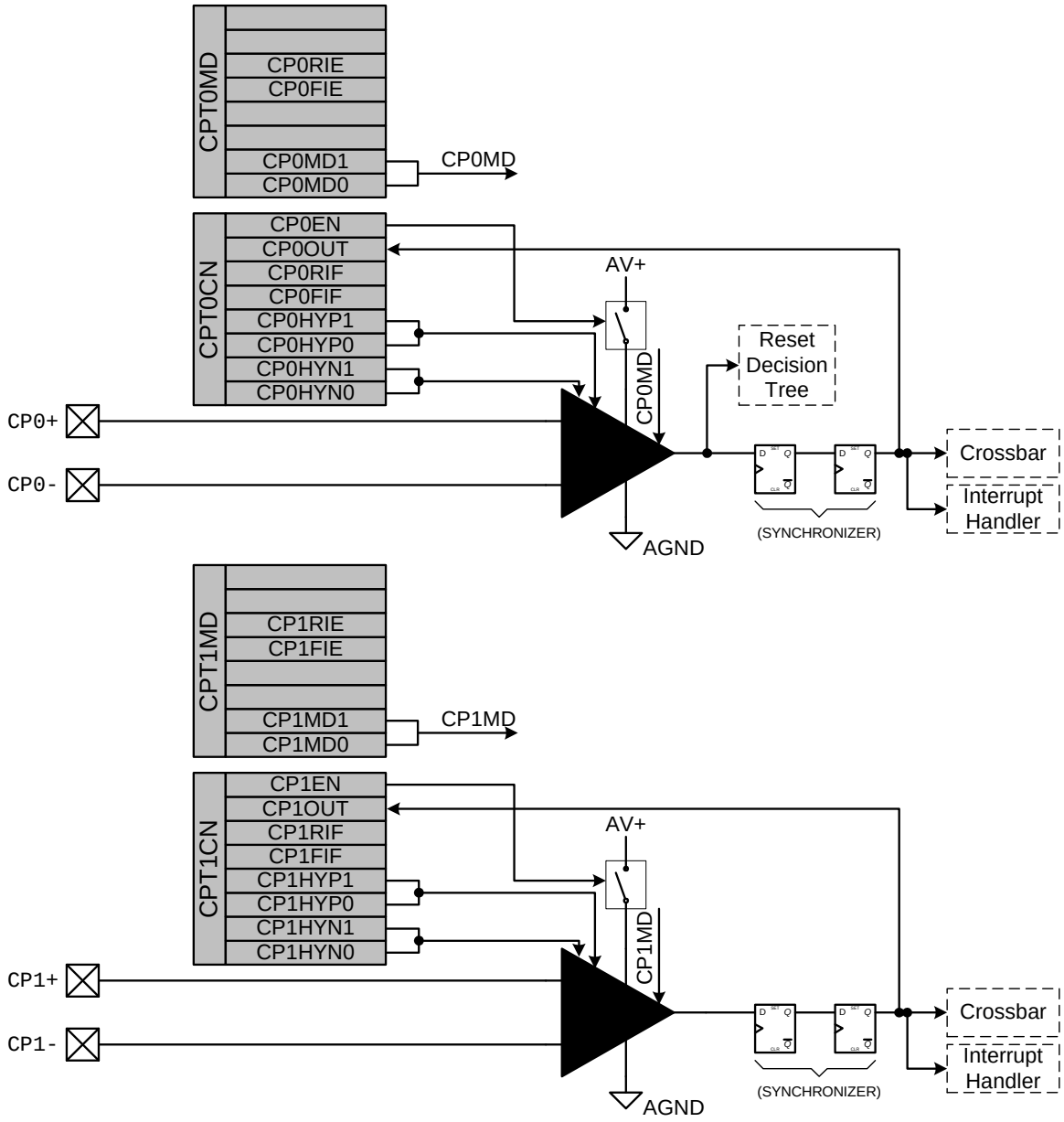


Figure 10.1. Comparator Functional Block Diagram

Comparator interrupts can be generated on rising-edge and/or falling-edge output transitions. (For interrupt enable and priority control, see **Section "11.3. Interrupt Handler" on page 154**). The CP0FIF flag is set upon a Comparator0 falling-edge interrupt, and the CP0RIF flag is set upon the Comparator0 rising-edge interrupt. Once set, these bits remain set until cleared by software. The Output State of Comparator0 can be obtained at any time by reading the CP0OUT bit. Comparator0 is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0. Comparator0 can also be programmed as a reset source; for details, see Section "**13.5. Comparator0 Reset**" on page **179**.

Note that after being enabled, there is a Power-Up time (listed in Table 10.1) during which the comparator outputs stabilize. The states of the Rising-Edge and Falling-Edge flags are indeterminant after comparator Power-Up and should be explicitly cleared before the comparator interrupts are enabled or the comparators are configured as a reset source.

Comparator0 response time may be configured in software via the CP0MD1-0 bits in register CPT0MD (see SFR Definition 10.2). Selecting a longer response time reduces the amount of current consumed by Comparator0. See Table 10.1 for complete timing and current consumption specifications.

The hysteresis of each comparator is software-programmable via its respective Comparator control register (CPT0CN and CPT1CN for Comparator0 and Comparator1, respectively). 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 output of the comparator can be polled in software, or can be used as an interrupt source. Each comparator can be individually enabled or disabled (shutdown). When disabled, the comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, its interrupt capability is suspended and its supply current falls to less than 100 nA. Comparator inputs can be externally driven from –0.25 V to (AV+) + 0.25 V without damage or upset.

Comparator0 hysteresis is programmed using bits 3-0 in the Comparator0 Control Register CPT0CN (shown in SFR Definition 10.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in SFR Definition 10.1, the negative hysteresis can be programmed to three different settings, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

The operation of Comparator1 is identical to that of Comparator0, though Comparator1 may not be configured as a reset source. Comparator1 is controlled by the CPT1CN Register (SFR Definition 10.3) and the CPT1MD Register (SFR Definition 10.4). The complete electrical specifications for the Comparators are given in Table 10.1.

13.3. External Reset

The external RST pin provides a means for external circuitry to force the MCU into a reset state. Asserting the RST pin low will cause the MCU to enter the reset state. It may be desirable to provide an external pull-up and/or decoupling of the RST pin to avoid erroneous noise-induced resets. The MCU will remain in reset until at least 12 clock cycles after the active-low RST signal is removed. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

13.4. Missing Clock Detector Reset

The Missing Clock Detector is essentially a one-shot circuit that is triggered by the MCU system clock. If the system clock goes away for more than 100 µs, the one-shot will time out and generate a reset. After a Missing Clock Detector reset, the MCDRSF flag (RSTSRC.2) will be set, signifying the MSD as the reset source; otherwise, this bit reads '0'. The state of the RST pin is unaffected by this reset. Setting the MCDRSF bit, RSTSRC.2 (see Section "**14. Oscillators**" on page **185**) enables the Missing Clock Detector.

13.5. Comparator0 Reset

Comparator0 can be configured as a reset input by writing a '1' to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled using CPT0CN.7 (see Section "**10. Comparators**" on page **119**) prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (CP0+ pin) is less than the inverting input voltage (CP0- pin), the MCU is put into the reset state. After a Comparator0 Reset, the C0RSEF flag (RSTSRC.5) will read '1' signifying Comparator0 as the reset source; otherwise, this bit reads '0'. The state of the RST pin is unaffected by this reset.

13.6. External CNVSTR0 Pin Reset

The external CNVSTR0 signal can be configured as a reset input by writing a '1' to the CNVRSEF flag (RSTSRC.6). The CNVSTR0 signal can appear on any of the P0, P1, P2 or P3 I/O pins as described in Section "**18.1. Ports 0 through 3 and the Priority Crossbar Decoder**" on page **238**. Note that the Crossbar must be configured for the CNVSTR0 signal to be routed to the appropriate Port I/O. The Crossbar should be configured and enabled before the CNVRSEF is set. When configured as a reset, CNVSTR0 is active-low and level sensitive. CNVSTR0 cannot be used to start ADC0 conversions when it is configured as a reset source. After a CNVSTR0 reset, the CNVRSEF flag (RSTSRC.6) will read '1' signifying CNVSTR0 as the reset source; otherwise, this bit reads '0'. The state of the /RST pin is unaffected by this reset.

13.7. Watchdog Timer Reset

The MCU includes a programmable Watchdog Timer (WDT) running off the system clock. A WDT overflow will force the MCU into the reset state. To prevent the reset, the WDT must be restarted by application software before overflow. If the system experiences a software or hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset. This should prevent the system from running out of control.

Following a reset the WDT is automatically enabled and running with the default maximum time interval. If desired the WDT can be disabled by system software or locked on to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the RST pin is unaffected by this reset.

The WDT consists of a 21-bit timer running from the programmed system clock. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT

reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. Watchdog features are controlled via the Watchdog Timer Control Register (WDTCN) shown in SFR Definition 13.1.

13.7.1. Enable/Reset WDT

The watchdog timer is both enabled and reset by writing 0xA5 to the WDTCN register. The user's application software should include periodic writes of 0xA5 to WDTCN as needed to prevent a watchdog timer overflow. The WDT is enabled and reset as a result of any system reset.

13.7.2. Disable WDT

Writing 0xDE followed by 0xAD to the WDTCN register disables the WDT. The following code segment illustrates disabling the WDT:

```
CLR    EA            ; disable all interrupts
MOV    WDTCN,#0DEh   ; disable software watchdog timer
MOV    WDTCN,#0ADh
SETB   EA            ; re-enable interrupts
```

The writes of 0xDE and 0xAD must occur within 4 clock cycles of each other, or the disable operation is ignored. This means that the prefetch engine should be enabled and interrupts should be disabled during this procedure to avoid any delay between the two writes.

13.7.3. Disable WDT Lockout

Writing 0xFF to WDTCN locks out the disable feature. Once locked out, the disable operation is ignored until the next system reset. Writing 0xFF does not enable or reset the watchdog timer. Applications always intending to use the watchdog should write 0xFF to WDTCN in the initialization code.

13.7.4. Setting WDT Interval

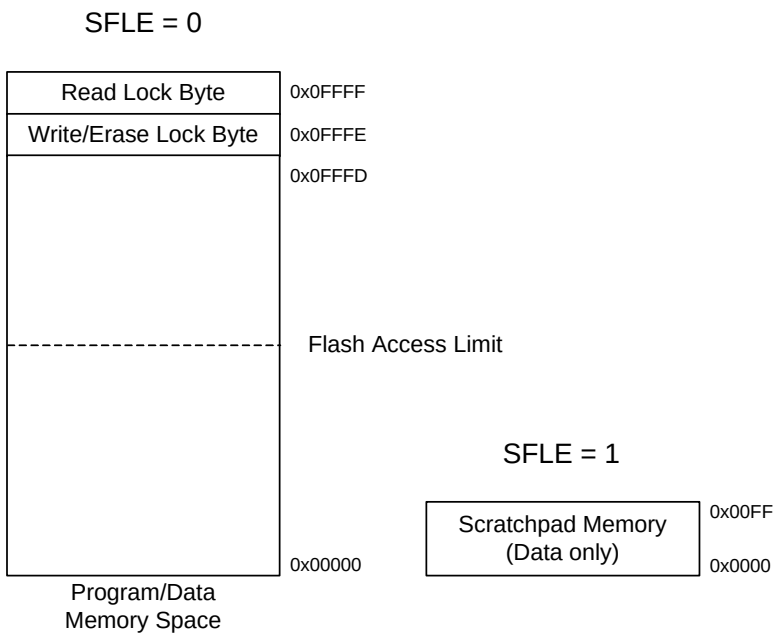
WDTCN.[2:0] control the watchdog timeout interval. The interval is given by the following equation:

$4^{3 + WDTCN[2-0]} \times T_{sysclk}$; where T_{sysclk} is the system clock period.

For a 3 MHz system clock, this provides an interval range of 0.021 ms to 349.5 ms. WDTCN.7 must be logic 0 when setting this interval. Reading WDTCN returns the programmed interval. WDTCN.[2:0] reads 111b after a system reset.

Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

Bit	Memory Block
7	N/A
6	N/A
5	N/A
4	N/A
3	0x0C000 - 0x0FFFF
2	0x08000 - 0x0BFFF
1	0x04000 - 0x07FFF
0	0x00000 - 0x03FFF



Flash Read Lock Byte

Bits7–0: Each bit locks a corresponding block of memory. (Bit7 is MSB).

0: Read operations are locked (disabled) for corresponding block across the JTAG interface.

1: Read operations are unlocked (enabled) for corresponding block across the JTAG interface.

Flash Write/Erase Lock Byte

Bits7–0: Each bit locks a corresponding block of memory.

0: Write/Erase operations are locked (disabled) for corresponding block across the JTAG interface.

1: Write/Erase operations are unlocked (enabled) for corresponding block across the JTAG interface.

NOTE: When the highest block is locked, the security bytes may be written but not erased.

Flash access Limit Register (FLACL)

The Flash Access Limit is defined by the setting of the FLACL register, as described in SFR Definition 15.1. Firmware running at or above this address is prohibited from using the MOVX and MOVC instructions to read, write, or erase Flash locations below this address.

Figure 15.3. 64 kB Flash Memory Map and Security Bytes

SFR Definition 16.2. CCH0TN: Cache Tuning

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CHMSCTL				CHALGM	CHFIXM	CHMSTH		00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA2
SFR Page: F

Bits 7–4: CHMSCTL: Cache Miss Penalty Accumulator (Bits 4–1).
These are bits 4-1 of the Cache Miss Penalty Accumulator. To read these bits, they must first be latched by reading the CHMSCTH bits in the CCH0MA Register (See SFR Definition 16.4).

Bit 3: CHALGM: Cache Algorithm Select.
This bit selects the cache replacement algorithm.
0: Cache uses Rebound algorithm.
1: Cache uses Pseudo-random algorithm.

Bit 2: CHFIXM: Cache Fix MOVC Enable.
This bit forces MOVC writes to the cache memory to use slot 0.
0: MOVC data is written according to the current algorithm selected by the CHALGM bit.
1: MOVC data is always written to cache slot 0.

Bits 1–0: CHMSTH: Cache Miss Penalty Threshold.
These bits determine when missed instruction data will be cached.
If data takes longer than CHMSTH clocks to obtain, it will be cached.

SFR Definition 16.3. CCH0LC: Cache Lock Control

R/W	R/W	R	R	R	R	R	R	Reset Value
CHPUSH	CHPOP	CHSLOT						00111110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA3
SFR Page: F

Bit 7: CHPUSH: Cache Push Enable.
This bit enables cache push operations, which will lock information in cache slots using MOVC instructions.
0: Cache push operations are disabled.
1: Cache push operations are enabled. When a MOVC read is executed, the requested 4-byte segment containing the data is locked into the cache at the location indicated by CHSLOT, and CHSLOT is decremented.
Note that no more than 61 cache slots should be locked at one time, since the entire cache will be unlocked when CHSLOT is equal to 0.

Bit 6: CHPOP: Cache Pop.
Writing a ‘1’ to this bit will increment CHSLOT and then unlock that location. This bit always reads ‘0’. Note that Cache Pop operations should not be performed while CHSLOT = 111110b. “Pop”ing more Cache slots than have been “Push”ed will have indeterminate results on the Cache performance.

Bits 5–0: CHSLOT: Cache Slot Pointer.
These read-only bits are the pointer into the cache lock stack. Locations above CHSLOT are locked, and will not be changed by the processor, except when CHSLOT equals 0.

17.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

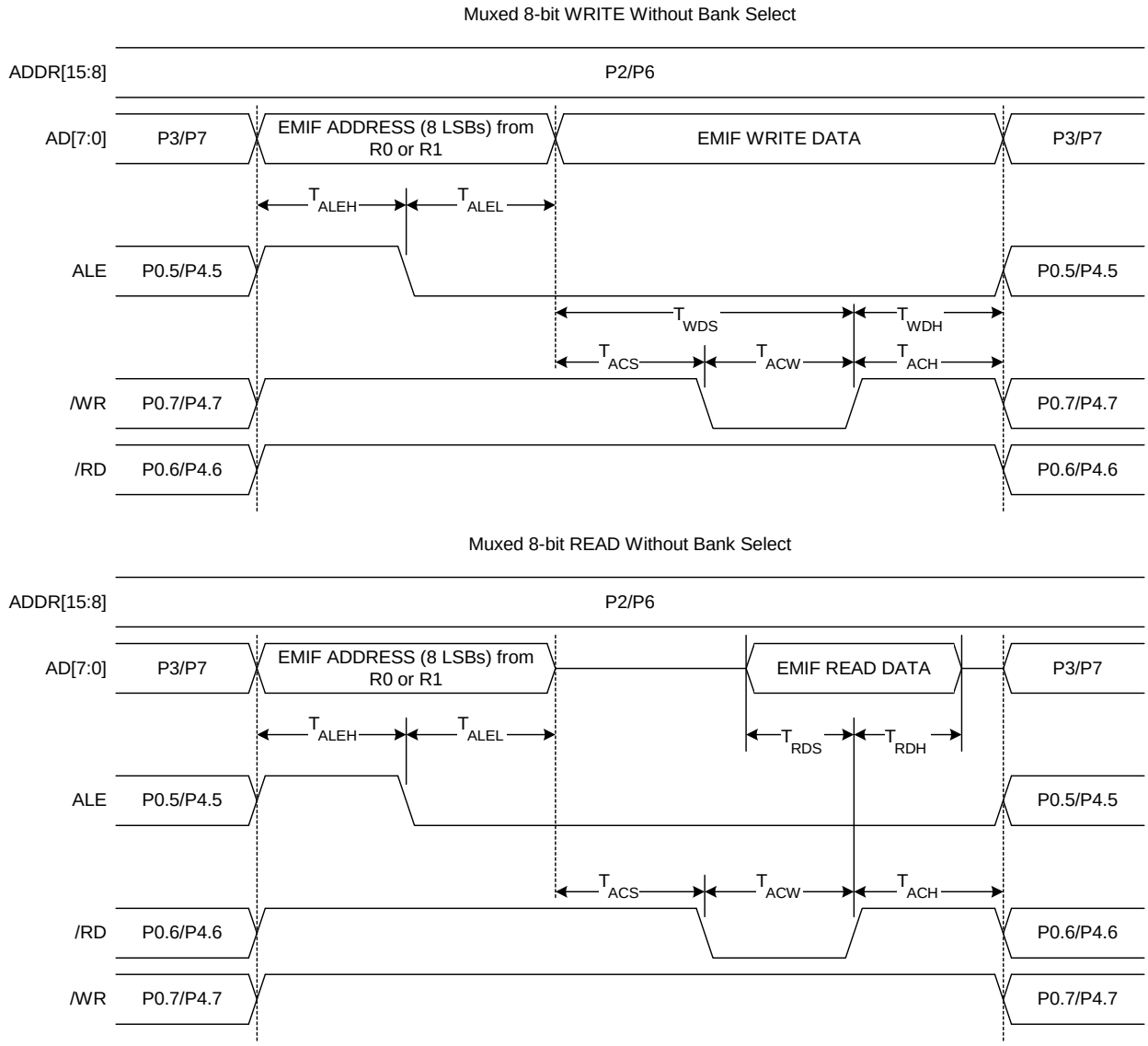


Figure 17.8. Multiplexed 8-bit MOVX without Bank Select Timing

SFR Definition 18.2. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE2
SFR Page: F

Bit7: SYSCKE: /SYSCLK Output Enable Bit.
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK (divided by 1, 2, 4, or 8) routed to Port pin. divide factor is determined by the CLKDIV1–0 bits in register CLKSEL (See **Section “14. Oscillators” on page 185**).

Bit6: T2EXE: T2EX Input Enable Bit.
0: T2EX unavailable at Port pin.
1: T2EX routed to Port pin.

Bit5: T2E: T2 Input Enable Bit.
0: T2 unavailable at Port pin.
1: T2 routed to Port pin.

Bit4: INT1E: /INT1 Input Enable Bit.
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port pin.

Bit3: T1E: T1 Input Enable Bit.
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit2: INT0E: /INT0 Input Enable Bit.
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port pin.

Bit1: T0E: T0 Input Enable Bit.
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

20.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

20.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

20.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

20.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

20.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 20.2, Figure 20.3, and Figure 20.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section "**18. Port Input/Output**" on page **235** for general purpose port I/O and crossbar information.

22.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB81 (SCON1.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB81 (SCON1.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: (1) RI1 must be logic 0, and (2) if MCE1 is logic 1, the 9th bit must be logic 1 (when MCE1 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF1, the ninth bit is stored in RB81, and the RI1 flag is set to '1'. If the above conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set to '1'. A UART1 interrupt will occur if enabled when either TI1 or RI1 is set to '1'.

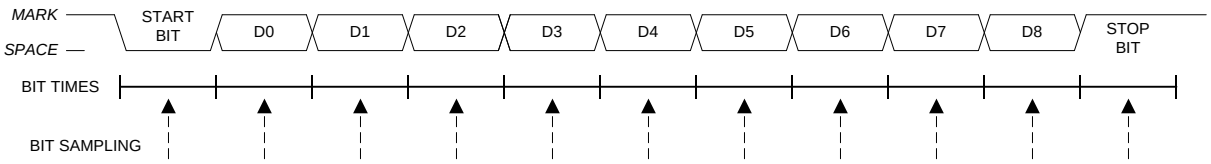


Figure 22.5. 9-Bit UART Timing Diagram

SFR Definition 23.3. CKCON: Clock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	T1M	T0M	-	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8E
SFR Page: 0

Bits7–5: UNUSED. Read = 000b, Write = don't care.
Bit4: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.
Bit3: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic 1.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Counter/Timer 0 uses the system clock.
Bit2: UNUSED. Read = 0b, Write = don't care.
Bits1–0: SCA1–SCA0: Timer 0/1 Prescale Bits
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8*

***Note:** External clock divided by 8 is synchronized with the system clock, and external clock must be less than or equal to the system clock frequency to operate the timer in this mode.

SFR Definition 23.4. TL0: Timer 0 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8A
SFR Page: 0

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

24.2.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 24.1.

Equation 24.1. Square Wave Frequency Output

$$F_{sqr} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA0 counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register.

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'.

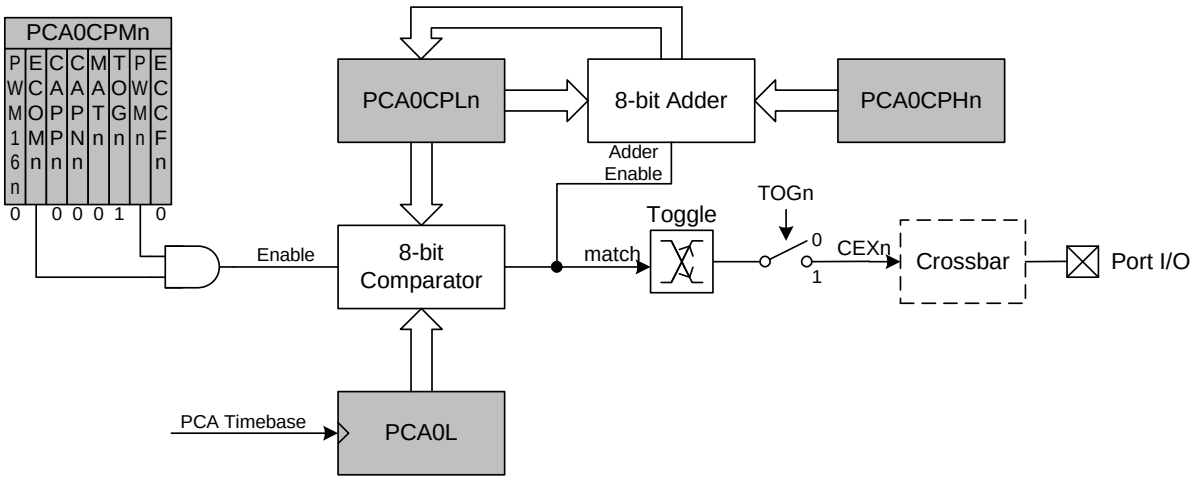


Figure 24.7. PCA Frequency Output Mode

NOTES: