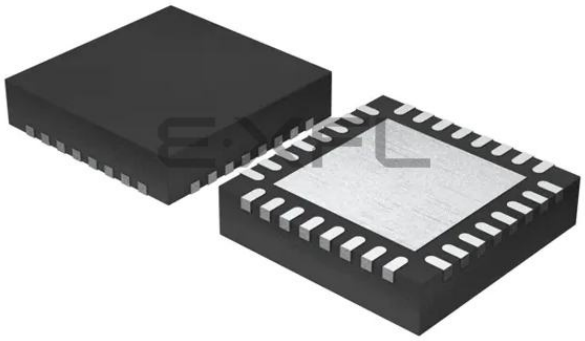
Silicon Labs - C8051F930-GMR Datasheet



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	Obsolete
Core Processor	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	24
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	0.9V ~ 3.6V
Data Converters	A/D 23x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f930-gmr

4.2. Electrical Characteristics

Table 4.2. Global Electrical Characteristics

–40 to +85 °C, 25 MHz system clock unless otherwise specified. See "AN358: Optimizing Low Power Operation of the ‘F9xx" for details on how to achieve the supply current specifications listed in this table.

Parameter	Conditions	Min	Typ	Max	Units
Battery Supply Voltage (VBAT)	One-Cell Mode Two-Cell Mode	0.9 1.8	1.2 2.4	1.8 3.6	V
Supply Voltage (VDD/DC+)	One-Cell Mode Two-Cell Mode	1.8 1.8	1.9 2.4	3.6 3.6	V
Minimum RAM Data Retention Voltage[1]	VDD (not in Sleep Mode) VBAT (in Sleep Mode)	— —	1.4 0.3	— 0.5	V
SYSCLK (System Clock)[2]		0	—	25	MHz
T_{SYSH} (SYSCLK High Time)		18	—	—	ns
T_{SYSL} (SYSCLK Low Time)		18	—	—	ns
Specified Operating Temperature Range		–40	—	+85	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD} [3, 4, 5, 6]	V_{DD} = 1.8–3.6 V, F = 24.5 MHz (includes precision oscillator current)	—	4.1	5.0	mA
	V_{DD} = 1.8–3.6 V, F = 20 MHz (includes low power oscillator current)	—	3.5	—	mA
	V_{DD} = 1.8 V, F = 1 MHz V_{DD} = 3.6 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	295 365	— —	µA µA
	V_{DD} = 1.8–3.6 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	90	—	µA
I_{DD} Frequency Sensitivity[3, 5, 6]	V_{DD} = 1.8–3.6 V, T = 25 °C, F < 10 MHz (Flash oneshot active, see Section 13.6)	—	226	—	µA/MHz
	V_{DD} = 1.8–3.6 V, T = 25 °C, F > 10 MHz (Flash oneshot bypassed, see Section 13.6)	—	120	—	µA/MHz
Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)					
I_{DD}[4, 6, 7]	V_{DD} = 1.8–3.6 V, F = 24.5 MHz (includes precision oscillator current)	—	2.5	3.0	mA
	V_{DD} = 1.8–3.6 V, F = 20 MHz (includes low power oscillator current)	—	1.8	—	mA
	V_{DD} = 1.8 V, F = 1 MHz V_{DD} = 3.6 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	165 235	— —	µA µA
	V_{DD} = 1.8–3.6 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	84	—	µA
I_{DD} Frequency Sensitivity[1,6,7]	V_{DD} = 1.8–3.6 V, T = 25 °C	—	95	—	µA/MHz

SFR Definition 7.6. CPT1MX: Comparator1 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX1N[3:0]				CMX1P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9E

Bit	Name	Function
7:4	CMX1N	**Comparator1 Negative Input Selection.** Selects the negative input channel for Comparator1. 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 (C8051F920/30 Only) 1000: P2.1 (C8051F920/30 Only) 1001: P2.3 (C8051F920/30 Only) 1010: P2.5 (C8051F920/30 Only) 1011: Reserved 1100: Capacitive Touch Sense Compare 1101: VDD/DC+ divided by 2 1110: Digital Supply Voltage 1111: Ground
3:0	CMX1P	**Comparator1 Positive Input Selection.** Selects the positive input channel for Comparator1. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 (C8051F920/30 Only) 1001: P2.2 (C8051F920/30 Only) 1010: P2.4 (C8051F920/30 Only) 1011: P2.6 (C8051F920/30 Only) 1100: Capacitive Touch Sense Compare 1101: VDD/DC+ divided by 2 1110: VBAT Supply Voltage 1111: VDD/DC+ Supply Voltage

9.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F93x-C8051F92x implements 64 kB (C8051F930/1) or 32 kB (C8051F920/1) of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0xFBFF (C8051F930/1) or 0x7FFF (C8051F920/1). The address 0xFBFF (C8051F930/1) or 0x7FFF (C8051F920/1) serves as the security lock byte for the device. Any addresses above the lock byte are reserved.

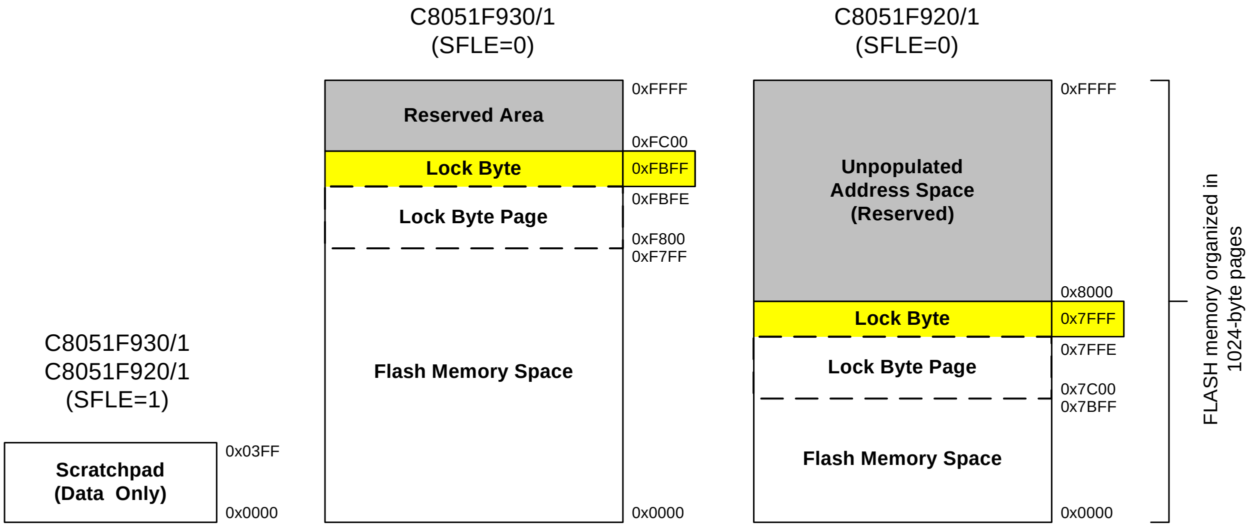


Figure 9.2. Flash Program Memory Map

9.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F93x-C8051F92x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F93x-C8051F92x to update program code and use the program memory space for non-volatile data storage. Refer to Section “13. Flash Memory” on page 145 for further details.

10.5. External Memory Interface Operating Modes

The external data memory space can be configured in one of four operating modes, shown in Figure 10.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 10.2). These modes are summarized below. Timing diagrams for the different modes can be found in Section “10.6. External Memory Interface Timing” on page 118.

10.5.1. Internal XRAM Only

When EMI0CF.[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap, and will always target on-chip XRAM. As an example, if the entire address space is consecutively written and the data pointer is incremented after each write, the write pointer will always point to the first byte of on-chip XRAM after the last byte of on-chip XRAM has been written.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

10.5.2. Split Mode without Bank Select

When EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the on-chip XRAM boundary will access on-chip XRAM space.
- Effective addresses above the on-chip XRAM boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, **an 8-bit MOVX operation will not drive the upper 4-bits A[11:8] of the Address Bus during an off-chip access.** This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 12-bits of the Address Bus A[11:0] are driven during the off-chip transaction.

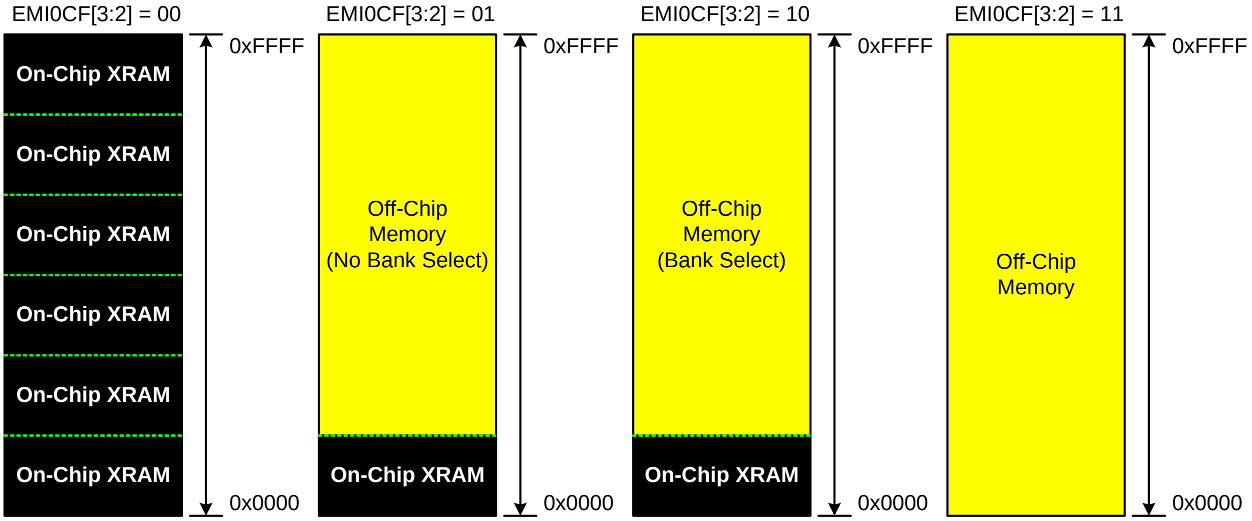


Figure 10.3. EMIF Operating Modes

Table 11.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page
CPT1MD	0x9C	0x0	Comparator1 Mode Selection	96
CPT1MX	0x9E	0x0	Comparator1 Mux Selection	99
CRC0AUTO	0x96	0xF	CRC0 Automatic Control	169
CRC0CN	0x92	0xF	CRC0 Control	167
CRC0CNT	0x97	0xF	CRC0 Automatic Flash Sector Count	169
CRC0DAT	0x91	0xF	CRC0 Data	168
CRC0FLIP	0x95	0xF	CRC0 Flip	170
CRC0IN	0x93	0xF	CRC0 Input	168
DC0CF	0x96	0x0	DC0 (DC-DC Converter) Configuration	178
DC0CN	0x97	0x0	DC0 (DC-DC Converter) Control	177
DPH	0x83	All	Data Pointer High	106
DPL	0x82	All	Data Pointer Low	106
EIE1	0xE6	All	Extended Interrupt Enable 1	139
EIE2	0xE7	All	Extended Interrupt Enable 2	141
EIP1	0xF6	0x0	Extended Interrupt Priority 1	140
EIP2	0xF7	0x0	Extended Interrupt Priority 2	142
EMI0CF	0xAB	0x0	EMIF Configuration	120
EMI0CN	0xAA	0x0	EMIF Control	119
EMI0TC	0xAF	0x0	EMIF Timing Control	121
FLKEY	0xB7	0x0	Flash Lock And Key	154
FLSCL	0xB6	0x0	Flash Scale	154
IE	0xA8	All	Interrupt Enable	137
IP	0xB8	0x0	Interrupt Priority	138
IREF0CN	0xB9	0x0	Current Reference IREF Control	89
IT01CF	0xE4	0x0	$\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration	144
OSCICL	0xB3	0x0	Internal Oscillator Calibration	194
OSCICN	0xB2	0x0	Internal Oscillator Control	194
OSCXCN	0xB1	0x0	External Oscillator Control	195
P0	0x80	All	Port 0 Latch	225
P0DRV	0xA4	0xF	Port 0 Drive Strength	227
P0MASK	0xC7	0x0	Port 0 Mask	222
P0MAT	0xD7	0x0	Port 0 Match	222
P0MDIN	0xF1	0x0	Port 0 Input Mode Configuration	226
P0MDOUT	0xA4	0x0	Port 0 Output Mode Configuration	226

12. Interrupt Handler

The C8051F93x-C8051F92x microcontroller family includes an extended interrupt system supporting multiple interrupt sources and two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Refer to Table 12.1, “Interrupt Summary,” on page 135 for a detailed listing of all interrupt sources supported by the device. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR or an indirect register. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1. If both global interrupts and the specific interrupt source is enabled, a CPU interrupt request is generated when the interrupt-pending flag is set.

As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Some interrupt-pending flags are automatically cleared by hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

12.1. Enabling Interrupt Sources

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1.

12.2. MCU Interrupt Sources and Vectors

The CPU services interrupts by generating an LCALL to a predetermined address (the interrupt vector address) to begin execution of an interrupt service routine (ISR). The interrupt vector addresses associated with each interrupt source are listed in Table 12.1 on page 135. Software should ensure that the interrupt vector for each enabled interrupt source contains a valid interrupt service routine.

Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag.

SFR Definition 12.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
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 = All Pages; SFR Address = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	**Enable All Interrupts.** Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**Enable Serial Peripheral Interface (SPI0) Interrupt.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Enable Timer 2 Interrupt.** This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	**Enable UART0 Interrupt.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Enable Timer 1 Interrupt.** This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	**Enable External Interrupt 1.** This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	**Enable Timer 0 Interrupt.** This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	**Enable External Interrupt 0.** This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

12.6. External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “25.1. Timer 0 and Timer 1” on page 280) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 12.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT0}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “21.3. Priority Crossbar Decoder” on page 216 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

13.4. Determining the Device Part Number at Run Time

In many applications, user software may need to determine the MCU part number at run time in order to determine the hardware capabilities. The part number can be determined by reading the value of the Flash byte at address 0xFFFE.

The value of the Flash byte at address 0xFFFE can be decoded as follows:

0x56—C8051F930
0x5E—C8051F931
0xB1—C8051F920
0xB3—C8051F921

14.2. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Note: To ensure the MCU enters a low power state upon entry into Idle Mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6) to logic 0. See the note in SFR Definition 13.3. FLSCL: Flash Scale for more information on how to properly clear the BYPASS bit.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “18.6. PCA Watchdog Timer Reset” on page 184 for more information on the use and configuration of the WDT.

14.3. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the precision internal oscillator and CPU are stopped; the state of the low power oscillator and the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout of 100 µs.

Stop Mode is a legacy 8051 power mode; it will not result in optimal power savings. Sleep or Suspend mode will provide more power savings if the MCU needs to be inactive for a long period of time.

On C8051F930, C8051F931, C8051F920, and C8051F921 devices, the Precision Oscillator Bias is not automatically disabled and should be disabled by software to achieve the lowest possible Stop mode current.

Note: To ensure the MCU enters a low power state upon entry into Stop Mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6) to logic 0. See the note in SFR Definition 13.3. FLSCL: Flash Scale for more information on how to properly clear the BYPASS bit.

15.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should select the desired polynomial and set the initial value of the result. Two polynomials are available: 0x1021 (16-bit) and 0x04C11DB7 (32-bit). The CRC0 result may be initialized to one of two values: 0x00000000 or 0xFFFFFFFF. The following steps can be used to initialize CRC0.

1. Select a polynomial (Set CRC0SEL to 0 for 32-bit or 1 for 16-bit).
2. Select the initial result value (Set CRC0VAL to 0 for 0x00000000 or 1 for 0xFFFFFFFF).
3. Set the result to its initial value (Write 1 to CRC0INIT).

15.3. 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 Flash sectors. 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 Flash sectors to perform in the CRC calculation to CRC0CNT.
   Note: Each Flash sector is 1024 bytes.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes.
   See the note in SFR Definition 15.1. 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.

15.4. Accessing the CRC0 Result

The internal CRC0 result is 32-bits (CRC0SEL = 0b) or 16-bits (CRC0SEL = 1b). 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.

16.2. High Power Applications

The dc-dc converter is designed to provide the system with 65 mW of output power, however, it can safely provide up to 100 mW of output power without any risk of damage to the device. For high power applications, the system should be carefully designed to prevent unwanted VBAT and VDD/DC+ Supply Monitor resets, which are more likely to occur when the dc-dc converter output power exceeds 65mW. In addition, output power above 65 mW causes the dc-dc converter to have relaxed output regulation, high output ripple and more analog noise. At high output power, an inductor with low DC resistance should be chosen in order to minimize power loss and maximize efficiency.

The combination of high output power and low input voltage will result in very high peak and average inductor currents. If the power supply has a high internal resistance, the transient voltage on the VBAT terminal could drop below 0.9 V and trigger a VBAT Supply Monitor Reset, even if the open-circuit voltage is well above the 0.9 V threshold. While this problem is most often associated with operation from very small batteries or batteries that are near the end of their useful life, it can also occur when using bench power supplies that have a slow transient response; the supply’s display may indicate a voltage above 0.9 V, but the minimum voltage on the VBAT pin may be lower. A similar problem can occur at the output of the dc-dc converter: using the default low current limit setting (125 mA) can trigger V_{DD} Supply Monitor resets if there is a high transient load current, particularly if the programmed output voltage is at or near 1.8 V.

16.3. Pulse Skipping Mode

The dc-dc converter allows the user to set the minimum pulse width such that if the duty cycle needs to decrease below a certain width in order to maintain regulation, an entire "clock pulse" will be skipped.

Pulse skipping can provide substantial power savings, particularly at low values of load current. The converter will continue to maintain a minimum output voltage at its programmed value when pulse skipping is employed, though the output voltage ripple can be higher. Another consideration is that the dc-dc will operate with pulse-frequency modulation rather than pulse-width modulation, which makes the switching frequency spectrum less predictable; this could be an issue if the dc-dc converter is used to power a radio. Figure 4.5 and Figure 4.6 on page 47 and 48 show the effect of pulse skipping on power consumption.

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a VDD/DC+ supply monitor reset. See Section “4. Electrical Characteristics” on page 43 for complete electrical characteristics of the VDD/DC+ monitor.
- Software should take care not to inadvertently disable the V_{DD} Monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to '1' to keep the V_{DD} Monitor enabled as a reset source.
- The VDD/DC+ supply monitor must be enabled before selecting it as a reset source**.** Selecting the VDD/DC+ supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the VDD/DC+ supply monitor and selecting it as a reset source. See Section “4. Electrical Characteristics” on page 43 for minimum VDD/DC+ Supply Monitor turn-on time. **No delay should be introduced in systems where software contains routines that erase or write Flash memory.** The procedure for enabling the VDD/DC+ supply monitor and selecting it as a reset source is shown below:

1. Enable the VDD/DC+ Supply Monitor (VDMEN bit in VDM0CN = 1).
2. Wait for the VDD/DC+ Supply Monitor to stabilize (optional).
3. Select the VDD/DC+ Supply Monitor as a reset source (PORSF bit in RSTSRC = 1).

SFR Definition 18.1. VDM0CN: VDD/DC+ Supply Monitor Control

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	VDMEN	VDDSTAT	VDDOK	Reserved	Reserved	Reserved		
Type	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Reset	1	Varies	Varies	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xFF

Bit	Name	Function
7	VDMEN	**VDD/DC+ Supply Monitor Enable.** This bit turns the VDD/DC+ supply monitor circuit on/off. The VDD/DC+ Supply Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 18.2). 0: VDD/DC+ Supply Monitor Disabled. 1: VDD/DC+ Supply Monitor Enabled.
6	VDDSTAT	**VDD/DC+ Supply Status.** This bit indicates the current power supply status. 0: VDD/DC+ is at or below the V_{RST} threshold. 1: VDD/DC+ is above the V_{RST} threshold.
5	VDDOK	**VDD/DC+ Supply Status (Early Warning).** This bit indicates the current power supply status. 0: VDD/DC+ is at or below the V_{WARN} threshold. 1: VDD/DC+ is above the V_{WARN} monitor threshold.
4:2	Reserved	**Reserved.** Read = 000b. Must Write 000b.
1:0	Unused	**Unused.** Read = 00b. Write = Don’t Care.

19. Clocking Sources

C8051F93x-C8051F92x devices include a programmable precision internal oscillator, an external oscillator drive circuit, a low power internal oscillator, and a SmaRTClock real time clock oscillator. The precision internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 19.1. The external oscillator can be configured using the OSCXCN register. The low power internal oscillator is automatically enabled and disabled when selected and deselected as a clock source. SmaRTClock operation is described in the SmaRTClock oscillator chapter.

The system clock (SYSCLK) can be derived from the precision internal oscillator, external oscillator, low power internal oscillator, or SmaRTClock oscillator. The global clock divider can generate a system clock that is 1, 2, 4, 8, 16, 32, 64, or 128 times slower that the selected input clock source. Oscillator electrical specifications can be found in the Electrical Specifications Chapter.

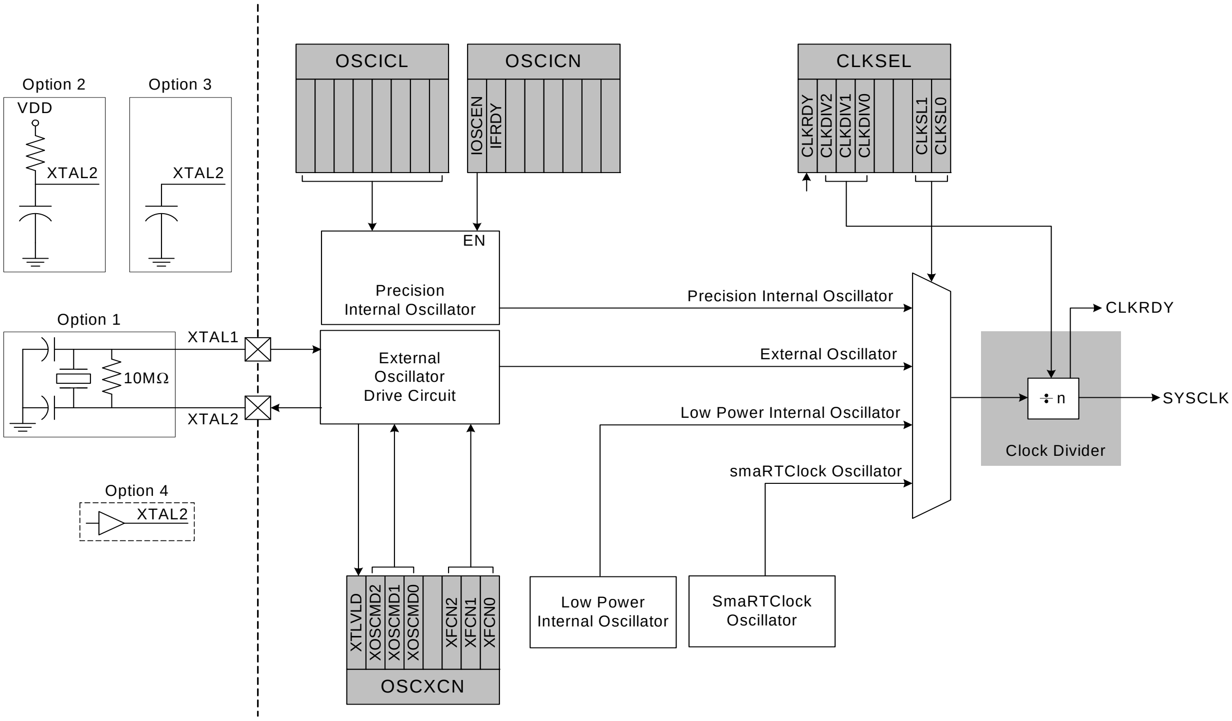


Figure 19.1. Clocking Sources Block Diagram

The proper way of changing the system clock when both the clock source and the clock divide value are being changed is as follows:

If switching from a fast “undivided” clock to a slower “undivided” clock:

a. Change the clock divide value.

b. Poll for CLKRDY > 1.

c. Change the clock source.

If switching from a slow “undivided” clock to a faster “undivided” clock:

a. Change the clock source.

b. Change the clock divide value.

c. Poll for CLKRDY > 1.

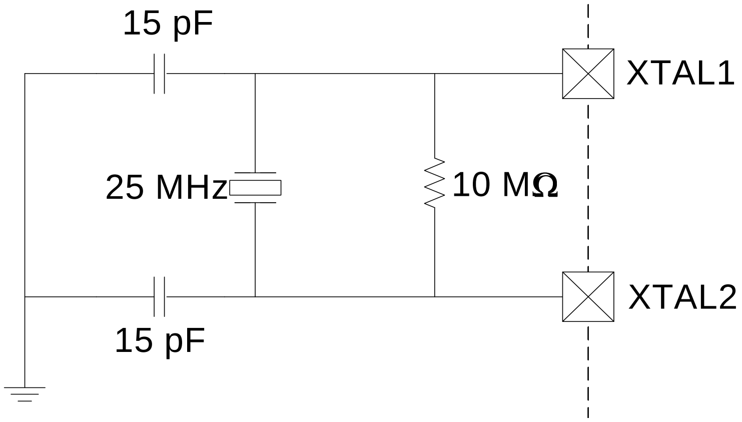


Figure 19.2. 25 MHz External Crystal Example

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 ≤ 20 kHz	0.5 µA	3.0 µA, f = 32.768 kHz
001	20 kHz < f ≤ 58 kHz	1.5 µA	4.8 µA, f = 32.768 kHz
010	58 kHz < f ≤ 155 kHz	4.8 µA	9.6 µA, f = 32.768 kHz
011	155 kHz < f ≤ 415 kHz	14 µA	28 µA, f = 400 kHz
100	415 kHz < f ≤ 1.1 MHz	40 µA	71 µA, f = 400 kHz
101	1.1 MHz < f ≤ 3.1 MHz	120 µA	193 µA, f = 400 kHz
110	3.1 MHz < f ≤ 8.2 MHz	550 µA	940 µA, f = 8 MHz
111	8.2 MHz < f ≤ 25 MHz	2.6 mA	3.9 mA, f = 25 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:

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.

The SmaRTClock alarm event can be configured to reset the MCU, wake it up from a low power mode, or generate an interrupt. See Section “12. Interrupt Handler” on page 133, Section “14. Power Management” on page 156, and Section “18. Reset Sources” on page 180 for more information.

The following steps can be used to set up a SmaRTClock Alarm:

1. Disable SmaRTClock Alarm Events (RTC0AEN = 0).
2. Set the ALARMn registers to the desired value.
3. Enable SmaRTClock Alarm Events (RTC0AEN = 1).

Notes:

- The ALRM bit, which is used as the SmaRTClock Alarm Event flag, is cleared by disabling SmaRTClock Alarm Events (RTC0AEN = 0).
- If AutoReset is disabled, disabling (RTC0AEN = 0) then Re-enabling Alarm Events (RTC0AEN = 1) after a SmaRTClock Alarm without modifying ALARMn registers will automatically schedule the next alarm after 2^32 SmaRTClock cycles (approximately 36 hours using a 32.768 kHz crystal).
- The SmaRTClock Alarm Event flag will remain asserted for a maximum of one SmaRTClock cycle. See Section “14. Power Management” on page 156 for information on how to capture a SmaRTClock Alarm event using a flag which is not automatically cleared by hardware.

20.3.3. Software Considerations for using the SmaRTClock Timer and Alarm

The SmaRTClock timer and alarm have two operating modes to suit varying applications. The two modes are described below:

Mode 1:
The first mode uses the SmaRTClock timer as a perpetual timebase which is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. The alarm interval is software managed and is added to the ALRMn registers by software after each alarm. This allows the alarm match value to always stay ahead of the timer by one software managed interval. If software uses 32-bit unsigned addition to increment the alarm match value, then it does not need to handle overflows since both the timer and the alarm match value will overflow in the same manner.

This mode is ideal for applications which have a long alarm interval (e.g. 24 or 36 hours) and/or have a need for a perpetual timebase. An example of an application that needs a perpetual timebase is one whose wake-up interval is constantly changing. For these applications, software can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

Mode 2:
The second mode uses the SmaRTClock timer as a general purpose up counter which is auto reset to zero by hardware after each alarm. The alarm interval is managed by hardware and stored in the ALRMn registers. Software only needs to set the alarm interval once during device initialization. After each alarm, software should keep a count of the number of alarms that have occurred in order to keep track of time.

This mode is ideal for applications that require minimal software intervention and/or have a fixed alarm interval. This mode is the most power efficient since it requires less CPU time per alarm.

Table 23.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	**Baud Rate % Error**	**Oscilla-tor Divide Factor**	**Timer Clock Source**	**SCA1–SCA0 (pre-scale select)^1**	**T1M^1**	**Timer 1 Reload Value (hex)**
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX^2	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in Section 25.1.
2. X = Don’t care.

Table 23.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	**Baud Rate % Error**	**Oscilla-tor Divide Factor**	**Timer Clock Source**	**SCA1–SCA0 (pre-scale select)^1**	**T1M^1**	**Timer 1 Reload Value (hex)**
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX^2	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in Section 25.1.
2. X = Don’t care.

SFR Definition 25.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCC

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 25.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCD

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

26.3.5. 8-Bit, 9-Bit, 10-Bit and 11-Bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

26.3.5.1. 8-Bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 26.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module’s capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 26.2.

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.

$$\text{Duty Cycle } = \frac{(256 - PCA0CPHn)}{256}$$

Equation 26.2. 8-Bit PWM Duty Cycle

Using Equation 26.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

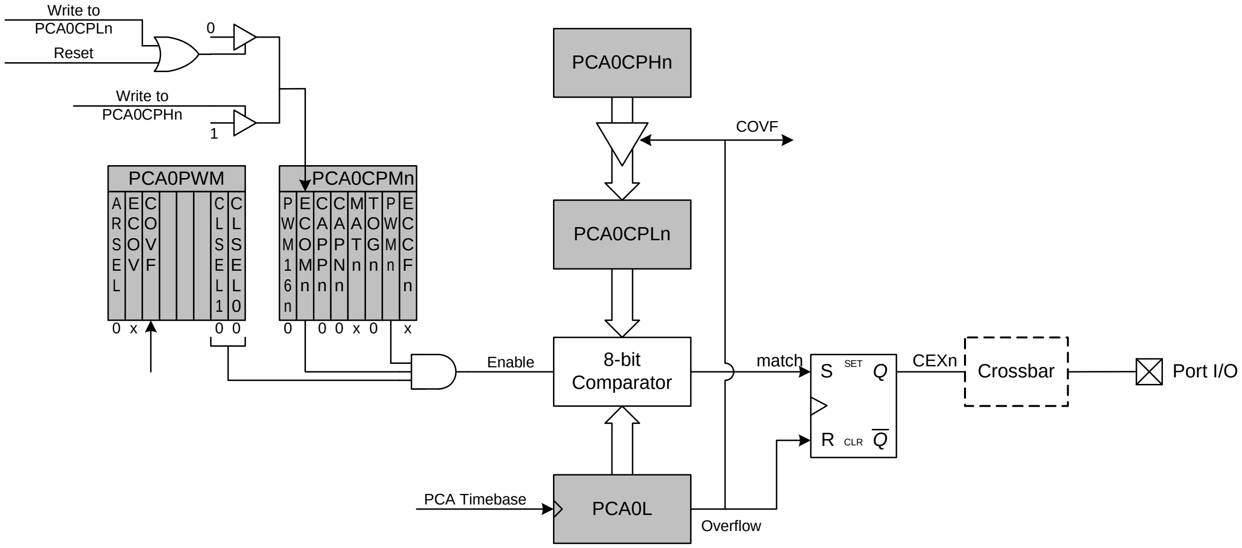


Figure 26.8. PCA 8-Bit PWM Mode Diagram