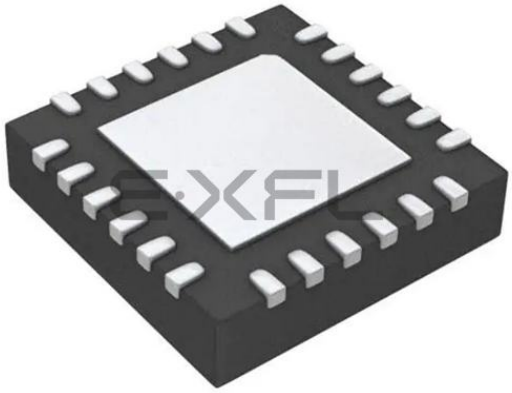

Welcome to E-XFL.COM

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-gmr

Table 3.1. Pin Definitions for the C8051F99x-C8051F98x (Continued)

Name	Pin Numbers			Type	Description
	‘F980/1/2 ‘F983/5 ‘F990/1 -GM	‘F986/7 ‘F988/9 ‘F996/7 -GM	‘F986/7 ‘F988/9 ‘F996/7 -GU		
P0.1/	1	23	2	D I/O or A In	Port 0.1. See Port I/O Section for a complete description.
AGND*				G	Optional Analog Ground. See Section “5.9. Voltage and Ground Reference Options” on page 88.
P0.2/	20	22	1	D I/O or A In	Port 0.2. See Port I/O Section for a complete description.
XTAL1/				A In	External Clock Input. This pin is the external oscillator return for a crystal or resonator. See Section “19. Clocking Sources” on page 188.
RTCOUT				D Out	Buffered SmaRTClock oscillator output.
P0.3/	19	21	24	D I/O or A In	Port 0.3. See Section “21. Port Input/Output” on page 215 for a complete description.
XTAL2/				A Out	External Clock Output. This pin is the excitation driver for an external crystal or resonator.
				D In	External Clock Input. This pin is the external clock input in external CMOS clock mode.
				A In	External Clock Input. This pin is the external clock input in capacitor or RC oscillator configurations. See Section “19. Clocking Sources” on page 188 for complete details.
WAKEOUT				D Out	Wake-up request signal to wake up external devices.
P0.4/	18	20	23	D I/O or A In	Port 0.4. See Section “21. Port Input/Output” on page 215 for a complete description.
TX				D Out	UART TX Pin. See Section “21. Port Input/Output” on page 215.
P0.5/	17	19	22	D I/O or A In	Port 0.5. See Section “21. Port Input/Output” on page 215 for a complete description.
RX				D In	UART RX Pin. See Section “21. Port Input/Output” on page 215.

***Note:** Available only on the C8051F980/2/6/8 and C8051F990/6 devices.

Table 4.2. Global Electrical Characteristics (Continued)

–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

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Digital Supply Current depends upon the particular code being executed. The values in this table are obtained with the CPU executing an “sjmp $” loop, which is the compiled form of a while(1) loop in C. One iteration requires 3 CPU clock cycles, and the Flash memory is read on each cycle. The supply current will vary slightly based on the physical location of the sjmp instruction and the number of Flash address lines that toggle as a result. In the worst case, current can increase by up to 30% if the sjmp loop straddles a 64-byte Flash address boundary (e.g., 0x007F to 0x0080). Real-world code with larger loops and longer linear sequences will have few transitions across the 64-byte address boundaries.
4. Includes oscillator and regulator supply current.
5. IDD can be estimated for frequencies $\leq$ 14 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range, then adding an offset of 84 µA. When using these numbers to estimate I_{DD} for > 14 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.0 V; F = 20 MHz, I_{DD} = 3.6 mA – (25 MHz – 20 MHz) x 0.088 mA/MHz = 3.16 mA assuming the same oscillator setting.
6. Idle IDD can be estimated by taking the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.0 V; F = 5 MHz, Idle I_{DD} = 1.75 mA – (25 MHz – 5 MHz) x 0.067 mA/MHz = 0.41 mA.

5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode

The ADC0 on C8051F980/6 and C8051F990/6 devices is a 300 ksps, 10-bit or 75 ksps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold and programmable window detector. ADC0 also has an autonomous low power Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. It also has a 16-bit accumulator that can automatically oversample and average the ADC results. See Section 5.4 for more details on using the ADC in 12-bit mode. C8051F982 and C8051F988 devices only support the 10-bit mode.

The ADC is fully configurable under software control via Special Function Registers. The ADC0 operates in Single-ended mode and may be configured to measure various different signals using the analog multiplexer described in “5.7. ADC0 Analog Multiplexer” on page 83. The voltage reference for the ADC is selected as described in “5.9. Voltage and Ground Reference Options” on page 88.

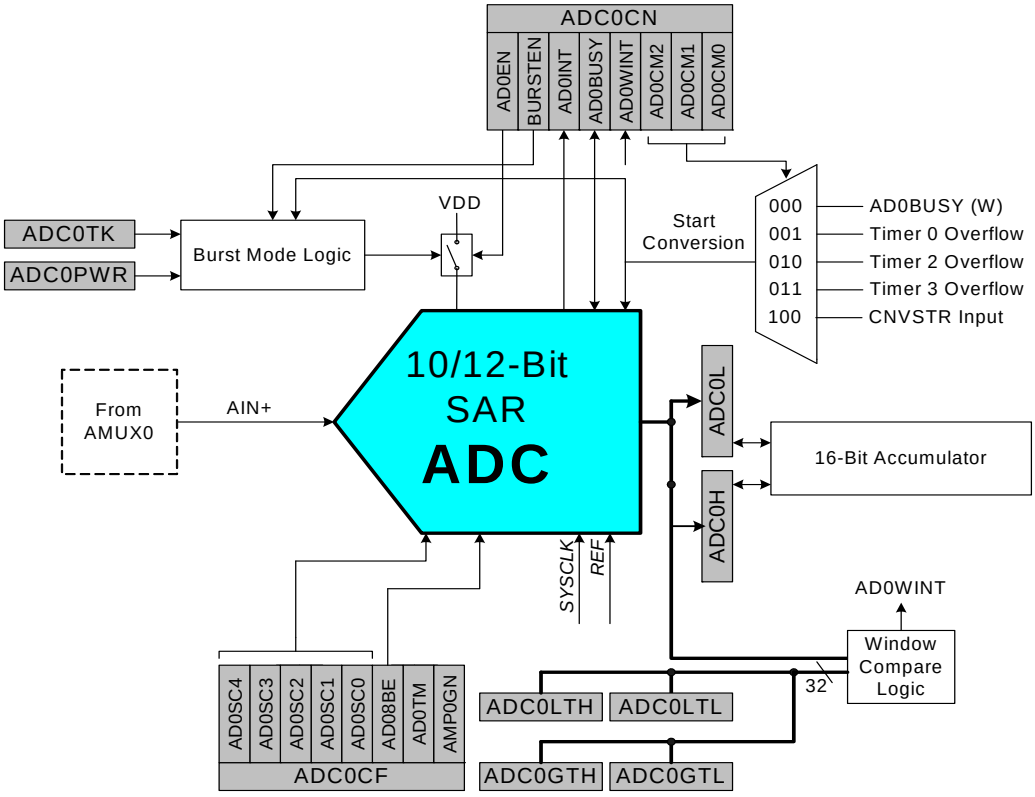

Figure 5.1. ADC0 Functional Block Diagram

5.8.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 4.11 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.9 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

A single-point offset measurement of the temperature sensor is performed on each device during production test. The measurement is performed at 25 °C ±5 °C, using the ADC with the internal high speed reference buffer selected as the Voltage Reference. The direct ADC result of the measurement is stored in the SFR registers TOFFH and TOFFL, shown in SFR Definition 5.13 and SFR Definition 5.14.

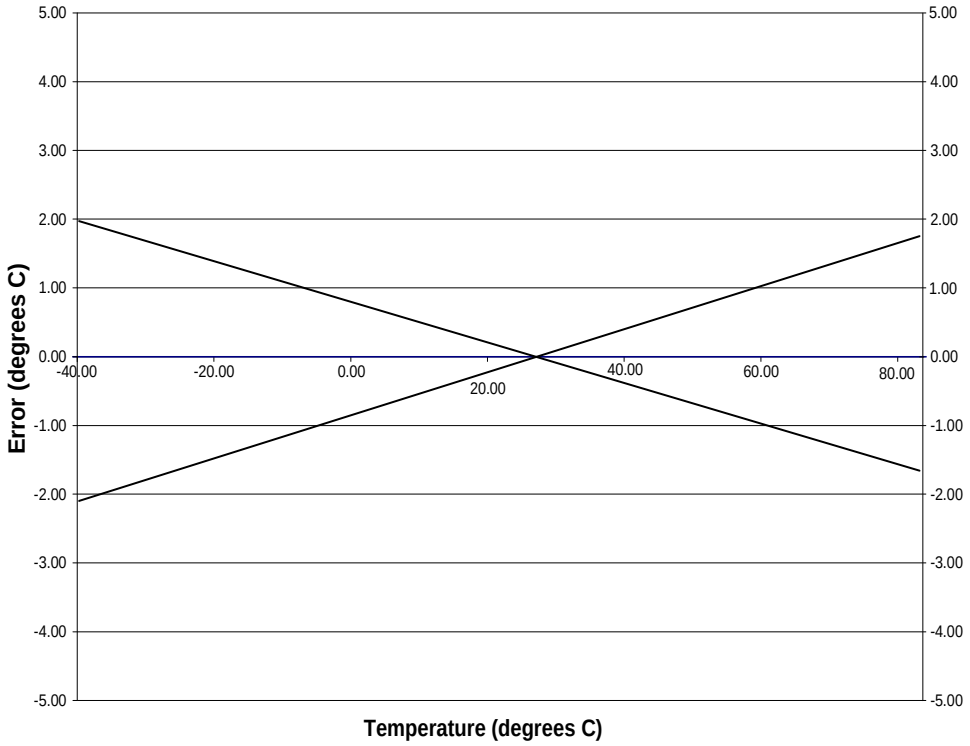

Figure 5.9. Temperature Sensor Error with 1-Point Calibration (V_{REF} = 1.65 V)

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.

8.4. CS0 Multiple Channel Enable

CS0 has the capability of measuring the total capacitance of multiple channels using a single conversion. When the multiple channel feature is enabled (CS0MCEN = 1), Channels selected by CS0SCAN0/1 are internally shorted together and the combined node is selected as the CS0 input. This mode can be used to detect a capacitance change on multiple channels using a single conversion and is useful for implementing “wake-on-multiple channels”.

8.5. CS0 Gain Adjustment

The gain of the CS0 circuit can be adjusted in integer increments from 1x to 8x (8x is the default). High gain gives the best sensitivity and resolution for small capacitors, such as those typically implemented as touch-sensitive PCB features. To measure larger capacitance values, the gain should be lowered accordingly. The bits CS0CG[2:0] in register CS0MD set the gain value.

8.6. Wake from Suspend

CS0 has the capability of waking the device from a low power suspend mode upon detection of a “touch” using the digital comparator. When the CS0SMEN is set to 1, CS0 may also wake up the device after an end of scan event when CS0CM[2:0] are set to 101b or after each conversion when CS0CM[2:0] are set to 110b or 111b. If the accumulate feature is enabled, the device wakes up after all samples have been accumulated. The CS0WOI bit in the CS0MD1 register can be used to configure desire wake from suspend behavior.

8.7. Using CS0 in Applications that Utilize Sleep Mode

To achieve maximum power efficiency, CS0 should be enabled only when taking a conversion and disabled at all other times. CS0 must be disabled by software prior to entering Sleep Mode.

SFR Definition 8.2. CS0CF: Capacitive Sense Configuration

Bit	7	6	5	4	3	2	1	0
Name	CS0SMEN	CS0CM[2:0]			CS0MCEN	CS0ACU[2:0]		
Type	R/W	R/W			R	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAA

Bit	Name	Description
7	CS0SMEN	**CS0 Channel Scan Masking Enable.** 0: The CS0SCAN0 and CS0SCAN1 register contents are ignored. 1: The CS0SCAN0 and CS0SCAN1 registers are used to determine which channels will be included in the scan.
6:4	CS0CM[2:0]	**CS0 Start of Conversion Mode Select.** 000: Conversion initiated on every write of 1 to CS0BUSY. 001: Conversion initiated on overflow of Timer 0. 010: Conversion initiated on overflow of Timer 2. 011: Conversion initiated on overflow of Timer 1. 100: Conversion initiated on overflow of Timer 3. **When CS0SMEN = 0** 101: Reserved. 110: Conversion initiated continuously on the channel selected by CS0MX after writing 1 to CS0BUSY. 111: Conversions initiated continuously on channels from CS0SS to CS0SE after writing 1 to CS0BUSY. **When CS0SMEN = 1** 101: Single Scan Mode, scans the channels selected by CS0SCAN0/1 once. 110: Conversion initiated continuously on the channel selected by CS0MX after writing 1 to CS0BUSY. 111: Auto Scan Mode, continuously scans the channels selected by CS0SCAN0/1.
3	CS0MCEN	**CS0 Multiple Channel Enable.** 0: Multiple channel feature is disabled. 1: Channels selected by CS0SCAN0/1 are internally shorted together and the combined node is selected as the CS0 input. This mode can be used to detect a capacitance change on multiple channels using a single conversion.
2:0	CS0ACU[2:0]	**CS0 Accumulator Mode Select.** 000: Accumulate 1 sample. 001: Accumulate 4 samples. 010: Accumulate 8 samples. 011: Accumulate 16 samples 100: Accumulate 32 samples. 101: Accumulate 64 samples. 11x: Reserved.

SFR Definition 8.7. CS0SS: Capacitive Sense Auto-Scan Start Channel

Bit	7	6	5	4	3	2	1	0
Name				CS0SS[4:0]				
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xDD

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't care
4:0	CS0SS[4:0]	**Starting Channel for Auto-Scan.** Sets the first CS0 channel to be selected by the mux for Capacitive Sense conversion when auto-scan is enabled and active. All channels detailed in CS0MX SFR Definition 8.15 are possible choices for this register. When auto-scan is enabled, a write to CS0SS will also update CS0MX.

SFR Definition 8.8. CS0SE: Capacitive Sense Auto-Scan End Channel

Bit	7	6	5	4	3	2	1	0
Name				CS0SE[4:0]				
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xDE

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't care
4:0	CS0SE[4:0]	**Ending Channel for Auto-Scan.** Sets the last CS0 channel to be selected by the mux for Capacitive Sense conversion when auto-scan is enabled and active. All channels detailed in CS0MX SFR Definition 8.15 are possible choices for this register.

SFR Definition 8.15. CS0MX: Capacitive Sense Mux Channel Select

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	CS0MX[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAB

Bit	Name	Description	
7:5	Reserved	Read = 0000b; Write = 0000b.	
4:0	CS0MX[4:0]	**CS0 Mux Channel Select.** Selects one of the 14 input channels for Capacitive Sense conversion.	
		Value	**Channel**
		0000	P0.0
		0001	P0.1
		0010	P0.2
		0011	P0.3
		0100	P0.4
		0101	P0.5
		0110	P0.6
		0111	P0.7
		1000	P1.0
		1001	P1.1
		1010	P1.2
		1011	P1.3
		1100	P1.4 (24-pin packages only)
		1101	P1.5
		1110	Reserved
		1111	Reserved

13.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If a high priority interrupt preempts a low priority interrupt, the low priority interrupt will finish execution after the high priority interrupt completes. Each interrupt has an associated interrupt priority bit in in the Interrupt Priority and Extended Interrupt Priority registers used to configure its priority level. Low priority is the default.

If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate. See Table 13.1 on page 140 to determine the fixed priority order used to arbitrate between simultaneously recognized interrupts.

13.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

13.6. 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 13.7). 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 PnSKIP (see Section “21.3. Priority Crossbar Decoder” on page 219 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.

15.1. Normal Mode

The MCU is fully functional in normal mode. Figure 15.1 shows the on-chip power distribution to various peripherals. There are two supply voltages powering various sections of the chip: V_{DD} and the 1.8 V internal core supply. All analog peripherals are directly powered from the V_{DD} pin. All digital peripherals and the CIP-51 core are powered from the 1.8 V internal core supply. RAM, PMU0 and the SmaRTClock are always powered directly from the V_{DD} pin in sleep mode and powered from the core supply in all other power modes.

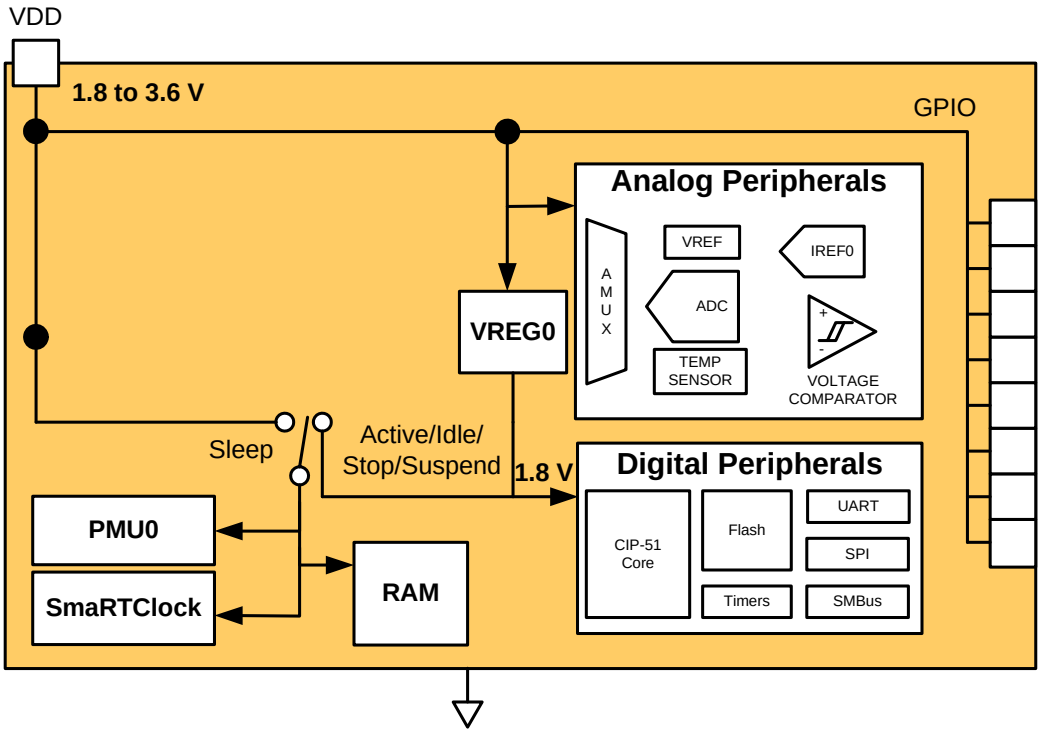

Figure 15.1. C8051F99x-C8051F98x Power Distribution

15.4. Suspend Mode

Setting the Suspend Mode Select bit (PMU0CF.6) causes the system clock to be gated off and all internal oscillators disabled. The system clock source must be set to the low power internal oscillator or the precision oscillator prior to entering Suspend Mode. All digital logic (timers, communication peripherals, interrupts, CPU, etc.) stops functioning until one of the enabled wake-up sources occurs.

The following wake-up sources can be configured to wake the device from Suspend Mode:

- CS0 End of Conversion or End of Scan
- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- Comparator0 Rising Edge

Note: Upon wake-up from suspend mode, PMU0 requires two system clocks in order to update the PMU0CF wake-up flags. All flags will read back a value of '0' during the first two system clocks following a wake-up from suspend mode.

In addition, a noise glitch on $\overline{\text{RST}}$ that is not long enough to reset the device will cause the device to exit suspend. In order for the MCU to respond to the pin reset event, software must not place the device back into suspend mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the /RST pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into suspend mode. A 4.7 kW pullup resistor to VDD is recommend for RST to prevent noise glitches from waking the device.

15.5. Sleep Mode

Setting the Sleep Mode Select bit (PMU0CF.7) turns off the internal 1.8 V regulator (VREG0) and switches the power supply of all on-chip RAM to the VDD pin (see Figure 15.1). Power to most digital logic on the chip is disconnected; only PMU0 and the SmaRTClock remain powered. Analog peripherals remain powered in two-cell mode. Only the Comparators remain functional when the device enters Sleep Mode. All other analog peripherals (CS0, ADC0, IREF0, External Oscillator, etc.) should be disabled prior to entering Sleep Mode. The system clock source must be set to the low power internal oscillator prior to entering Sleep Mode.

Important Note: The precision internal oscillator may potentially lock up after exiting Sleep mode. Systems using Sleep Mode should switch to the low power oscillator prior to entering Sleep Mode:

1. Switch the system clock to the low power oscillator (CLKSEL = 0x04).
2. Turn off the Precision Oscillator (OSCICN &= ~0x80).
3. Enter Sleep.
4. Exit Sleep.
5. Turn on the Precision Oscillator (OSCICN |= 0x80).
6. Switch the system clock to the Precision Oscillator (CLKSEL = 0x00).

GPIO pins configured as digital outputs will retain their output state during sleep mode. In two-cell mode, they will maintain the same current drive capability in sleep mode as they have in normal mode.

GPIO pins configured as digital inputs can be used during sleep mode as wakeup sources using the port match feature. In two-cell mode, they will maintain the same input level specs in sleep mode as they have in normal mode.

C8051F99x-C8051F98x devices support a wakeup request for external devices. Upon exit from sleep mode, the wake-up request signal is driven low, allowing other devices in the system to wake up from their low power modes.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on V_{DD} does not fall below V_{POR}. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from sleep mode.

Important Note: On device reset or upon waking up from Sleep mode, address 0x0000 of external memory may be overwritten by an indeterminate value. The indeterminate value is 0x00 in most situations. A dummy variable should be placed at address 0x0000 in external memory to ensure that the application firmware does not store any data that needs to be retained during sleep or reset at this memory location.

The following wake-up sources can be configured to wake the device from sleep mode:

- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- Comparator0 Rising Edge

The comparator requires a supply voltage of at least 1.8 V to operate properly. On C8051F99x-C8051F98x devices, the POR supply monitor can be disabled to save power by writing 1 to the MONDIS (PMU0MD.5) bit. When the POR supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the POR supply monitor.

Important Note: The POR Supply Monitor should not be disabled if the supply voltage is greater than 2.4 V. The lowest power sleep mode current, 10 nA typical, can only be achieved when the supply voltage is less than 2.4 V. The lowest power sleep mode for voltages above 2.4 V is 50 nA typical with the POR Supply Monitor enabled.

In addition, any falling edge on $\overline{\text{RST}}$ (due to a pin reset or a noise glitch) will cause the device to exit sleep mode. In order for the MCU to respond to the pin reset event, software must not place the device back into sleep mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the $\overline{\text{RST}}$ pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into sleep mode. A 4.7 kΩ pullup resistor to VDD is recommend for $\overline{\text{RST}}$ to prevent noise glitches from waking the device.

15.6. Configuring Wakeup Sources

Before placing the device in a low power mode, one or more wakeup sources should be enabled so that the device does not remain in the low power mode indefinitely. For Idle Mode, this includes enabling any interrupt. For Stop Mode, this includes enabling any reset source or relying on the $\overline{\text{RST}}$ pin to reset the device.

Wake-up sources for suspend and sleep modes are configured through the PMU0CF register. Wake-up sources are enabled by writing 1 to the corresponding wake-up source enable bit. Wake-up sources must be re-enabled each time the device is placed in suspend or sleep mode, in the same write that places the device in the low power mode.

The reset pin is always enabled as a wake-up source. On the falling edge of $\overline{\text{RST}}$, the device will be awaken from sleep mode. The device must remain awake for more than 15 µs in order for the reset to take place.

SFR Definition 22.2. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC0; Bit-Addressable

Bit	Name	Description	Read	Write
7	MASTER	**SMBus Master/Slave Indicator.** This read-only bit indicates when the SMBus is operating as a master.	0: SMBus operating in slave mode. 1: SMBus operating in master mode.	N/A
6	TXMODE	**SMBus Transmit Mode Indicator.** This read-only bit indicates when the SMBus is operating as a transmitter.	0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.	N/A
5	STA	**SMBus Start Flag.**	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO	**SMBus Stop Flag.**	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ	**SMBus Acknowledge Request.**	0: No Ack requested 1: ACK requested	N/A
2	ARBLOST	**SMBus Arbitration Lost Indicator.**	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK	**SMBus Acknowledge.**	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI	**SMBus Interrupt Flag.** This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.	**0: No interrupt pending** 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

22.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received.

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

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

The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 22.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

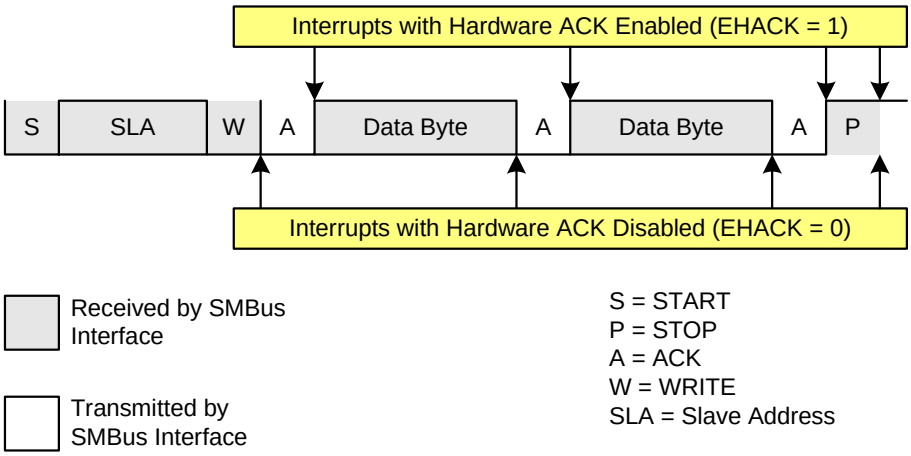

Figure 22.7. Typical Slave Write Sequence

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	Oscillator 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	Oscillator 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 25.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8C

Bit	Name	Function
7:0	TH0[7:0]	**Timer 0 High Byte.** The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 25.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8D

Bit	Name	Function
7:0	TH1[7:0]	**Timer 1 High Byte.** The TH1 register is the high byte of the 16-bit Timer 1.

25.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, SmaRTClock divided by 8 or Comparator 0 output. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bits (T2XCLK[1:0] in TMR2CN), as follows:

T2MH	T2XCLK[1:0]	TMR2H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

T2ML	T2XCLK[1:0]	TMR2L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

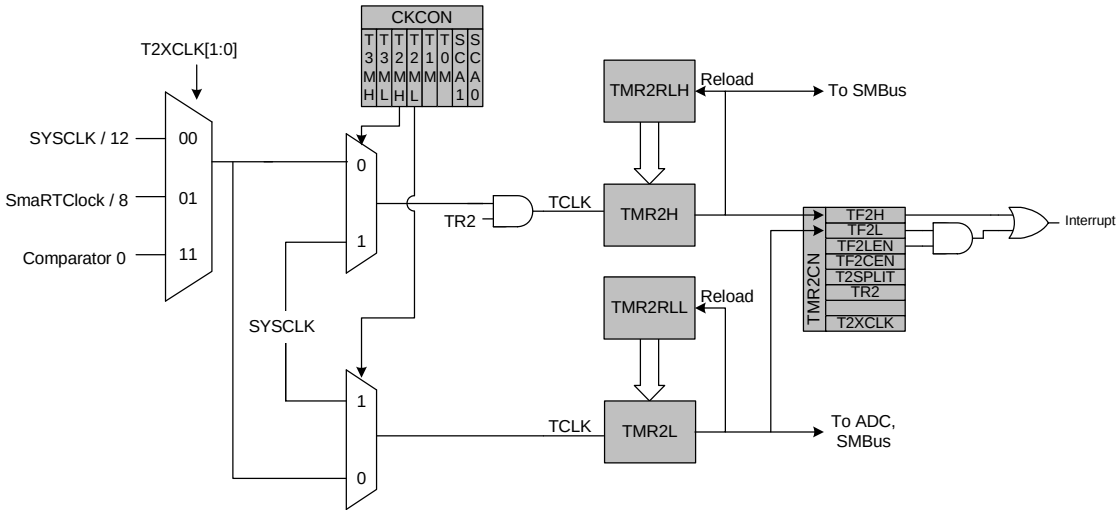

Figure 25.5. Timer 2 8-Bit Mode Block Diagram

26.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

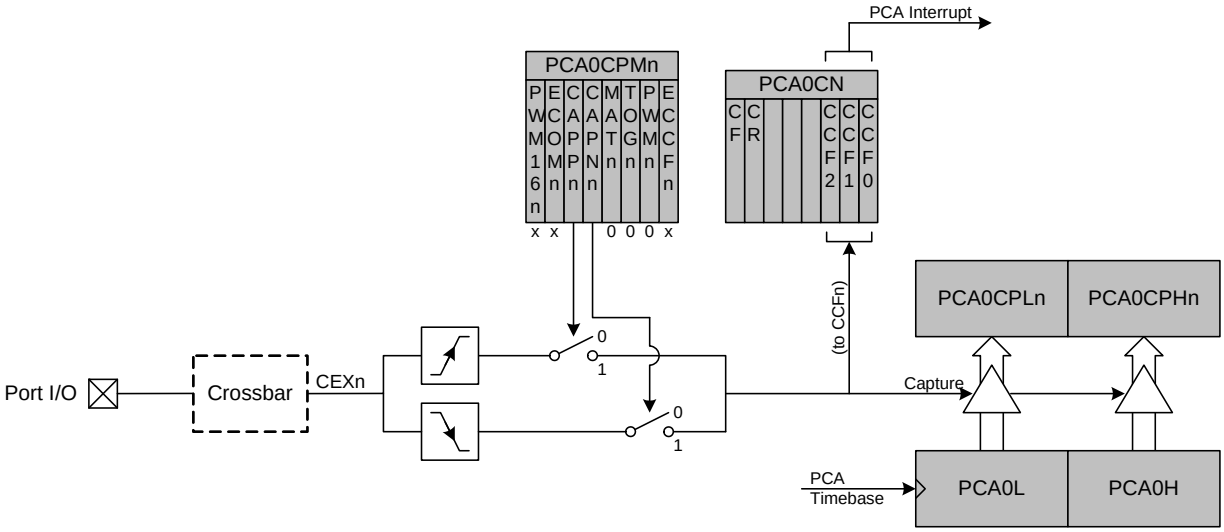

Figure 26.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.