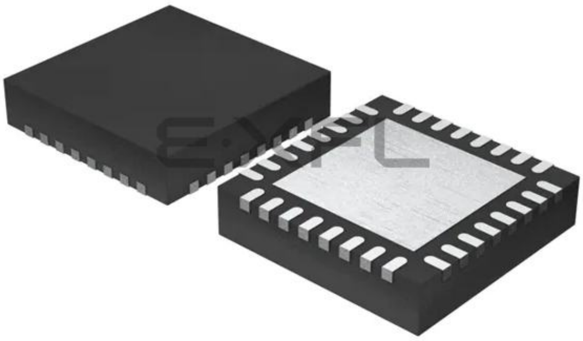


Welcome to E-XFL.COM

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	48MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 21x10b
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/c8051f342-gm

Table 3.2. Index to Electrical Characteristics Tables

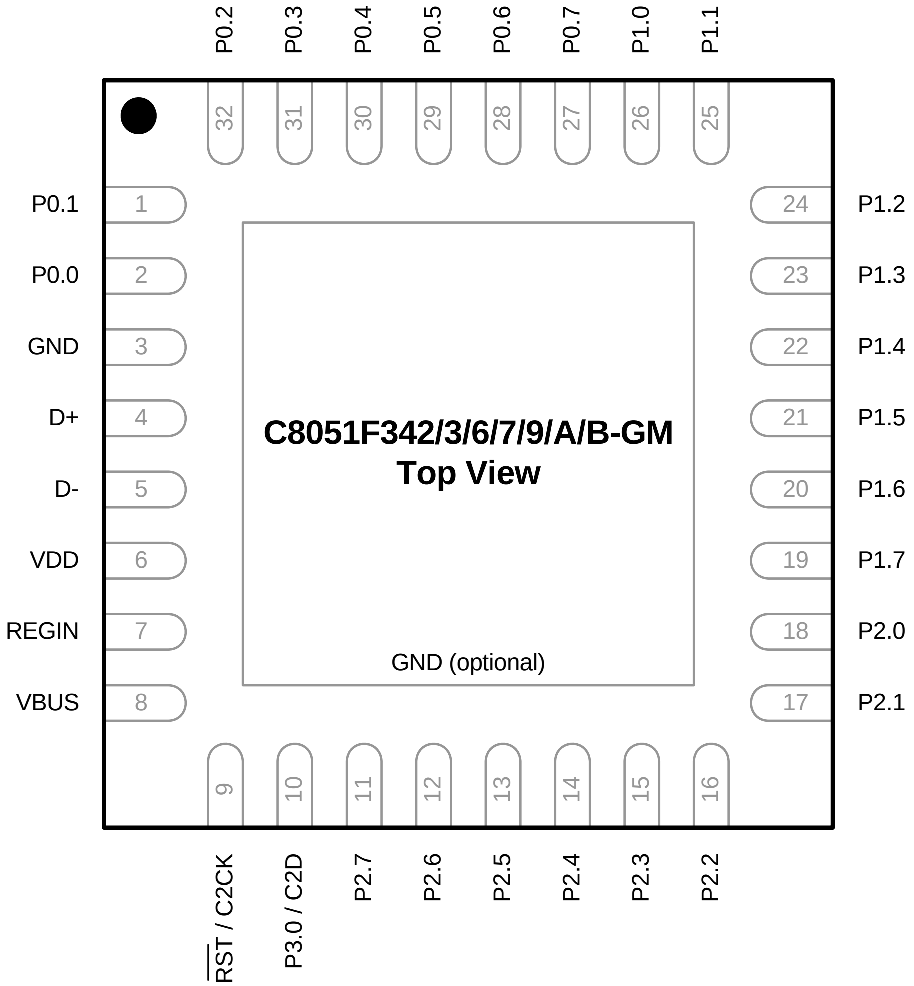


Figure 4.7. QFN-32 Pinout Diagram (Top View)

5.4.2. Window Detector In Differential Mode

Figure 5.8 shows two example window comparisons for right-justified, differential data, with ADC0LTH:ADC0LTL = 0x0040 (+64d) and ADC0GTH:ADC0GTH = 0xFFFF (-1d). In differential mode, the measurable voltage between the input pins is between -VREF and VREF*(511/512). Output codes are represented as 10-bit 2's complement signed integers. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0xFFFF (-1d) < ADC0H:ADC0L < 0x0040 (64d)). In the right example, an AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0xFFFF (-1d) or ADC0H:ADC0L > 0x0040 (+64d)). Figure 5.9 shows an example using left-justified data with equivalent ADC0GT and ADC0LT register settings.

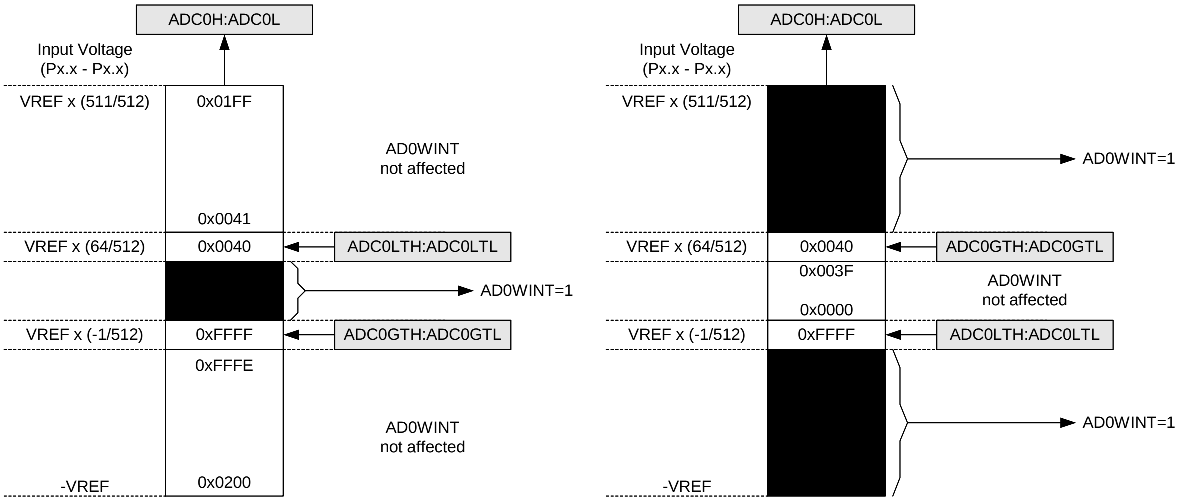


Figure 5.8. ADC Window Compare Example: Right-Justified Differential Data

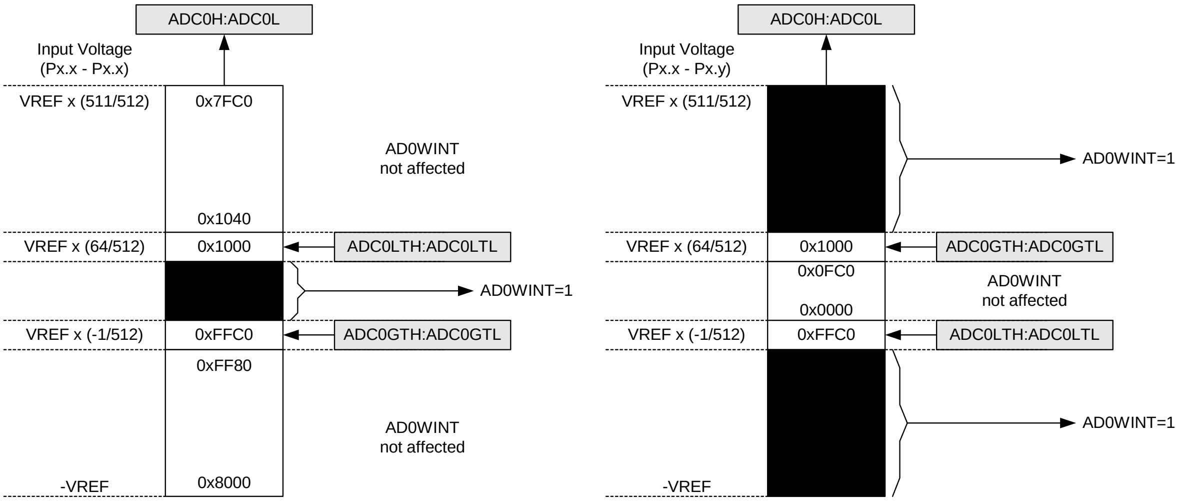


Figure 5.9. ADC Window Compare Example: Left-Justified Differential Data

8. Voltage Regulator (REG0)

C8051F34x devices include a voltage regulator (REG0). When enabled, the REG0 output appears on the V_{DD} pin and can be used to power external devices. REG0 can be enabled/disabled by software using bit REGEN in register REG0CN. See Table 8.1 for REG0 electrical characteristics.

Note that the VBUS signal must be connected to the VBUS pin when using the device in a USB network. The VBUS signal should only be connected to the REGIN pin when operating the device as a bus-powered function. REG0 configuration options are shown in Figure 8.1–Figure 8.4.

8.1. Regulator Mode Selection

REG0 offers a low power mode intended for use when the device is in suspend mode. In this low power mode, the REG0 output remains as specified; however the REG0 dynamic performance (response time) is degraded. See Table 8.1 for normal and low power mode supply current specifications. The REG0 mode selection is controlled via the REGMOD bit in register REG0CN.

8.2. VBUS Detection

When the USB Function Controller is used (see section **Section “16. Universal Serial Bus Controller (USB0)” on page 159**), the VBUS signal should be connected to the VBUS pin. The VBSTAT bit (register REG0CN) indicates the current logic level of the VBUS signal. If enabled, a VBUS interrupt will be generated when the VBUS signal matches the polarity selected by the VBPOL bit in register REG0CN. The VBUS interrupt is level-sensitive, and has no associated interrupt pending flag. The VBUS interrupt will be active as long as the VBUS signal matches the polarity selected by VBPOL. See Table 8.1 for VBUS input parameters.

Important Note: When USB is selected as a reset source, a system reset will be generated when the VBUS signal matches the polarity selected by the VBPOL bit. See **Section “11. Reset Sources” on page 100** for details on selecting USB as a reset source

Table 8.1. Voltage Regulator Electrical Specifications

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range[1]		2.7		5.25	V
Output Voltage (V_{DD})[2]	Output Current = 1 to 100 mA	3.0	3.3	3.6	V
Output Current[2]				100	mA
VBUS Detection Input Low Voltage				1.0	V
VBUS Detection Input High Voltage		3.0			V
Bias Current	Normal Mode (REGMOD = ‘0’) Low Power Mode (REGMOD = ‘1’)		65 35	111 61	µA
Dropout Voltage (V_{DO})[3]			1		mV/mA

Notes:
1. Input range specified for regulation. When an external regulator is used, should be tied to V_{DD}.
2. Output current is total regulator output, including any current required by the C8051F34x.
3. The minimum input voltage is 2.70 V or VDD + V_{DO} (max load), whichever is greater.

9.2. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The CIP-51 memory organization is shown in Figure 9.2 and Figure 9.3.

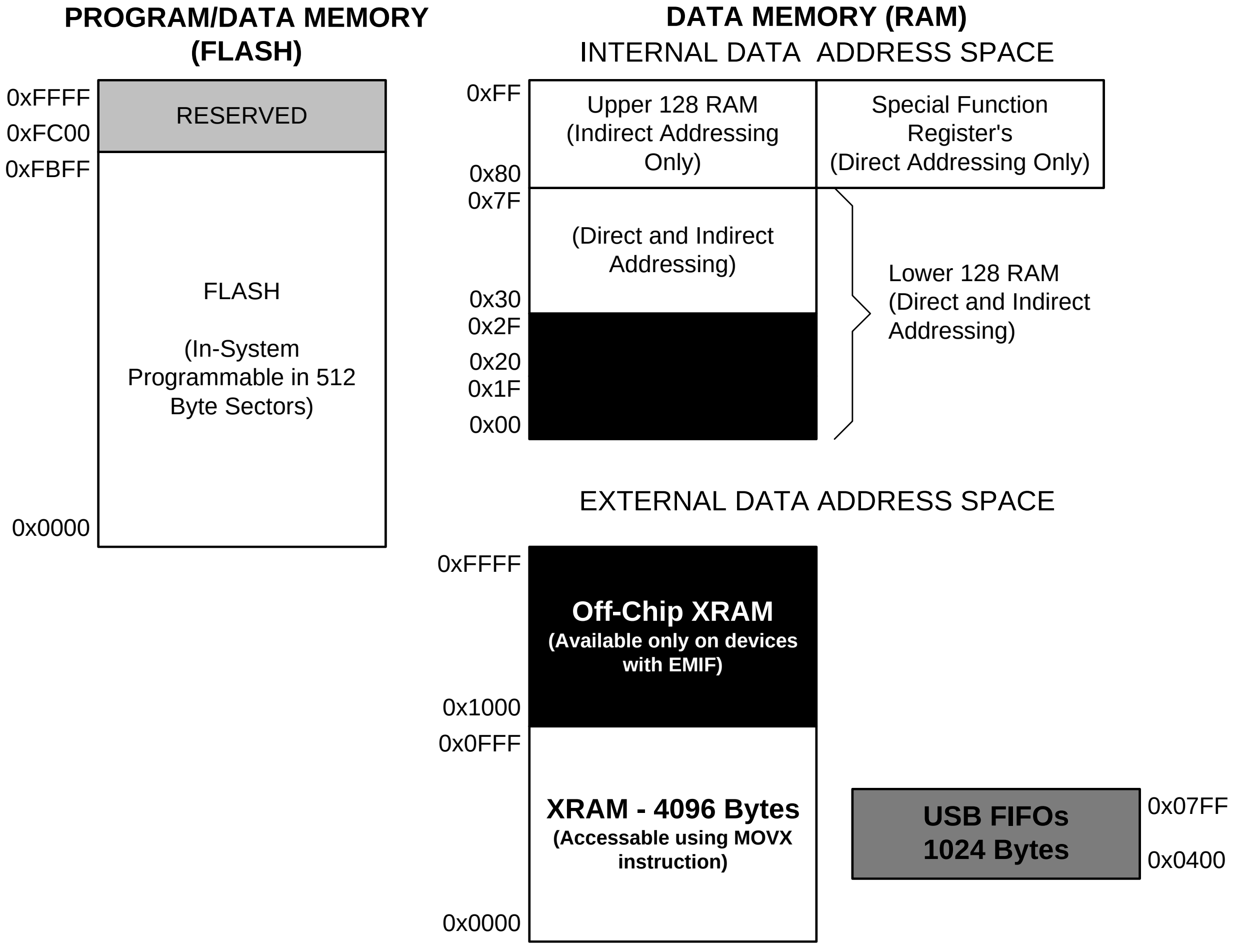

PROGRAM/DATA MEMORY
(FLASH)
0xFFFF
0xFC00
RESERVED
0xFBFF
FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
DATA MEMORY (RAM)
INTERNAL DATA ADDRESS SPACE
0xFF
Upper 128 RAM (Indirect Addressing Only)
Special Function Register's (Direct Addressing Only)
0x80
0x7F
(Direct and Indirect Addressing)
0x30
0x2F
0x20
0x1F
0x00
Lower 128 RAM (Direct and Indirect Addressing)
EXTERNAL DATA ADDRESS SPACE
0xFFFF
Off-Chip XRAM
(Available only on devices with EMIF)
0x1000
0x0FFF
XRAM - 4096 Bytes
(Accessable using MOVX instruction)
0x0000
USB FIFOs
1024 Bytes
0x07FF
0x0400

Figure 9.2. On-Chip Memory Map for 64 kB Devices

SFR Definition 11.2. RSTSRC: Reset Source

R/W	R	R/W	R/W	R	R/W	R/W	R	Reset Value
USBRSF	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xEF

Bit7: USBRSF: USB Reset Flag
0: **Read:** Last reset was not a USB reset; **Write:** USB resets disabled.
1: **Read:** Last reset was a USB reset; **Write:** USB resets enabled.

Bit6: FERROR: Flash Error Indicator.
0: Source of last reset was not a Flash read/write/erase error.
1: Source of last reset was a Flash read/write/erase error.

Bit5: C0RSEF: Comparator0 Reset Enable and Flag.
0: **Read:** Source of last reset was not Comparator0; **Write:** Comparator0 is not a reset source.
1: **Read:** Source of last reset was Comparator0; **Write:** Comparator0 is a reset source (active-low).

Bit4: SWRSF: Software Reset Force and Flag.
0: **Read:** Source of last reset was not a write to the SWRSF bit; **Write:** No Effect.
1: **Read:** Source of last was a write to the SWRSF bit; **Write:** Forces a system reset.

Bit3: WDTRSF: Watchdog Timer Reset Flag.
0: Source of last reset was not a WDT timeout.
1: Source of last reset was a WDT timeout.

Bit2: MCDRSF: Missing Clock Detector Flag.
0: **Read:** Source of last reset was not a Missing Clock Detector timeout; **Write:** Missing Clock Detector disabled.
1: **Read:** Source of last reset was a Missing Clock Detector timeout; **Write:** Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.

Bit1: PORSF: Power-On / V_{DD} Monitor Reset Flag.
This bit is set anytime a power-on reset occurs. Writing this bit selects/deselects the V_{DD} monitor as a reset source. **Note: writing '1' to this bit before the V_{DD} monitor is enabled and stabilized can cause a system reset.** See register VDM0CN (SFR Definition 11.1).
0: **Read:** Last reset was not a power-on or V_{DD} monitor reset; **Write:** V_{DD} monitor is not a reset source.
1: **Read:** Last reset was a power-on or V_{DD} monitor reset; all other reset flags indeterminate; **Write:** V_{DD} monitor is a reset source.

Bit0: PINRSF: HW Pin Reset Flag.
0: Source of last reset was not $\overline{\text{RST}}$ pin.
1: Source of last reset was $\overline{\text{RST}}$ pin.

Note: For bits that act as both reset source enables (on a write) and reset indicator flags (on a read), read-modify-write instructions read and modify the source enable only. This applies to bits: USBRSF, C0RSEF, SWRSF, MCDRSF, PORSF.

12. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 12.1 for complete Flash memory electrical characteristics.

12.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see **Section “23. C2 Interface” on page 271**.

To ensure the integrity of Flash contents, it is strongly recommended that the V_{DD} monitor be left enabled in any system which writes or erases Flash memory from code. It is also crucial to ensure that the FLRT bit in register FLSCL be set to '1' if a clock speed higher than 25 MHz is being used for the device.

12.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 12.2.

12.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE bit (register PSCTL).
Step 8. Clear the PSEE bit (register PSCTl).

SFR Definition 12.3. FLSCL: Flash Scale

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FOSE	Reserved	Reserved	FLRT	Reserved	Reserved	Reserved	Reserved	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xB6

Bits7: FOSE: Flash One-shot Enable
This bit enables the Flash read one-shot. When the Flash one-shot disabled, the Flash sense amps are enabled for a full clock cycle during Flash reads. At system clock frequencies below 10 MHz, disabling the Flash one-shot will increase system power consumption.
0: Flash one-shot disabled.
1: Flash one-shot enabled.

Bits6–5: RESERVED. Read = 00b. Must Write 00b.

Bit 4: FLRT: FLASH Read Time.
This bit should be programmed to the smallest allowed value, according to the system clock speed.
0: SYSCLK <= 25 MHz.
1: SYSCLK <= 48 MHz.

Bits3–0: RESERVED. Read = 0000b. Must Write 0000b.

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	External Oscillator / 4	USBCLK = 101b

External Oscillator	Crystal Oscillator Mode
	24 MHz Crystal
		XOSCMD = 110b
		XFCN = 111b

SFR Definition 14.6. CLKSEL: Clock Select

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	USBCLK			-	CLKSL			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address
								0xA9

Bit 7: Unused. Read = 0b; Write = don’t care.

Bits6–4: USBCLK2–0: USB Clock Select

These bits select the clock supplied to USB0. When operating USB0 in full-speed mode, the selected clock should be 48 MHz. When operating USB0 in low-speed mode, the selected clock should be 6 MHz.

USBCLK	Selected Clock
000	4x Clock Multiplier
001	Internal Oscillator / 2
010	External Oscillator
011	External Oscillator / 2
100	External Oscillator / 3
101	External Oscillator / 4
110	RESERVED
111	RESERVED

Bit3: Unused. Read = 0b; Write = don’t care.

Bits2–0: CLKSL2–0: System Clock Select

These bits select the system clock source. When operating from a system clock of 25 MHz or less, the FLRT bit should be set to ‘0’. When operating with a system clock of greater than 25 MHz (up to 48 MHz), the FLRT bit (FLSCL.4) should be set to ‘1’. See **Section “10. Prefetch Engine” on page 99** for more details.

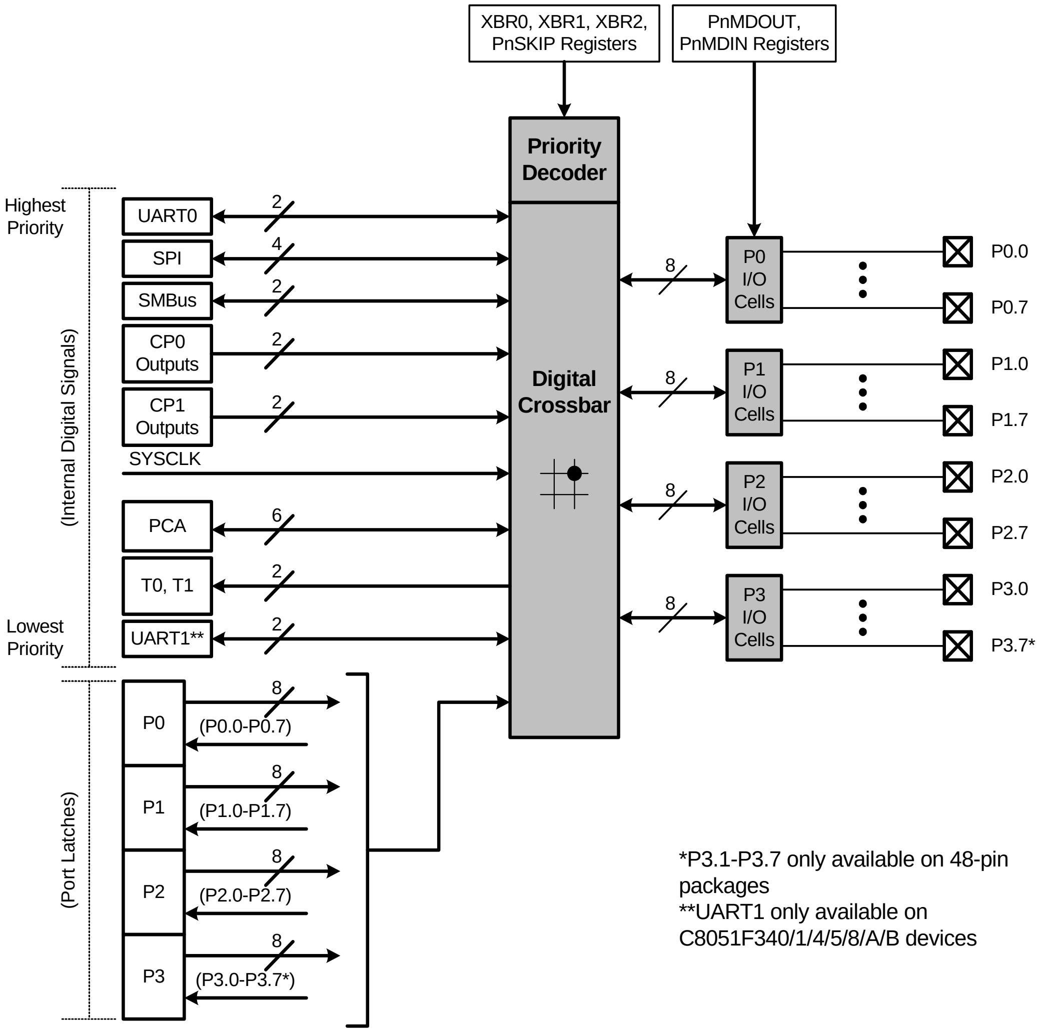
CLKSL	Selected Clock
000	Internal Oscillator (as determined by the IFCN bits in register OSCICN)
001	External Oscillator
010	4x Clock Multiplier / 2
011*	4x Clock Multiplier*
100	Low-Frequency Oscillator
101-111	RESERVED
***Note:** This option is only available on 48 MHz devices.	

15. Port Input/Output

Digital and analog resources are available through 40 I/O pins (48-pin packages) or 25 I/O pins (32-pin packages). Port pins are organized as shown in Figure 15.1. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input; Port pins P0.0-P3.7 can be assigned to one of the internal digital resources as shown in Figure 15.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the Priority Decoder (Figure 15.3 and Figure 15.4). The registers XBR0, XBR1, and XBR2 defined in SFR Definition 15.1, SFR Definition 15.2, and SFR Definition 15.3, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 15.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1,2,3,4). Complete Electrical Specifications for Port I/O are given in Table 15.1 on page 158.



*P3.1-P3.7 only available on 48-pin packages
**UART1 only available on C8051F340/1/4/5/8/A/B devices

Figure 15.1. Port I/O Functional Block Diagram (Port 0 through Port 3)

16.5. FIFO Management

1024 bytes of on-chip XRAM are used as FIFO space for USB0. This FIFO space is split between Endpoints0-3 as shown in Figure 16.3. FIFO space allocated for Endpoints1-3 is configurable as IN, OUT, or both (Split Mode: half IN, half OUT).

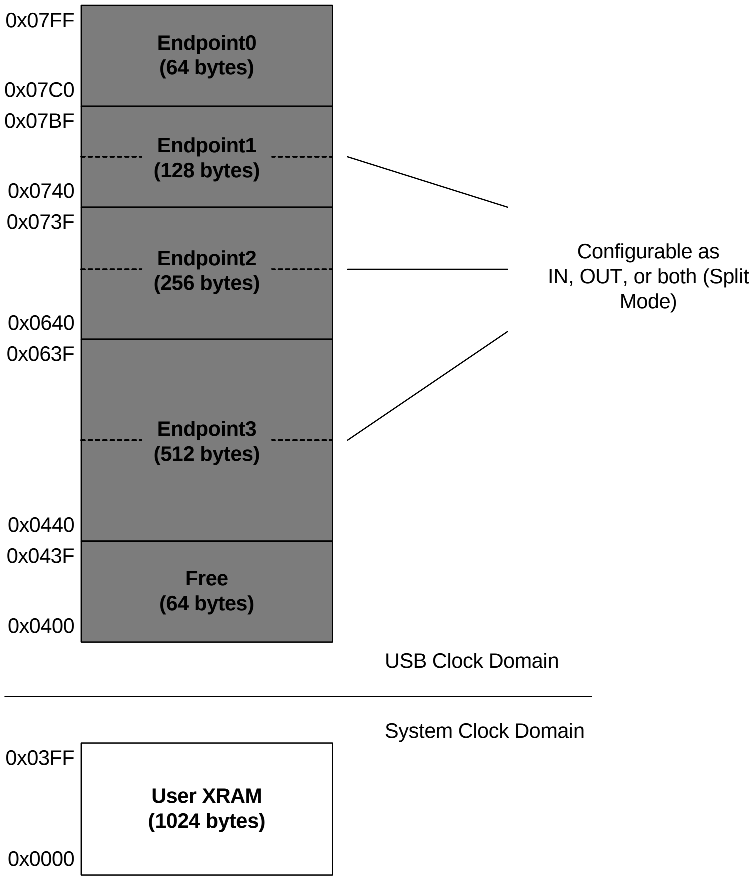


Figure 16.3. USB FIFO Allocation

16.5.1. FIFO Split Mode

The FIFO space for Endpoints1-3 can be split such that the upper half of the FIFO space is used by the IN endpoint, and the lower half is used by the OUT endpoint. For example: if the Endpoint3 FIFO is configured for Split Mode, the upper 256 bytes (0x0540 to 0x063F) are used by Endpoint3 IN and the lower 256 bytes (0x0440 to 0x053F) are used by Endpoint3 OUT.

If an endpoint FIFO is not configured for Split Mode, that endpoint IN/OUT pair's FIFOs are combined to form a single IN *or* OUT FIFO. In this case only one direction of the endpoint IN/OUT pair may be used at a time. The endpoint direction (IN/OUT) is determined by the DIRSEL bit in the corresponding endpoint's EINCSRH register (see SFR Definition 16.20).

USB Register Definition 16.9. FRAMEL: USB0 Frame Number Low

R	R	R	R	R	R	R	R	Reset Value
Frame Number Low								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x0C

Bits7-0: Frame Number Low
This register contains bits7-0 of the last received frame number.

USB Register Definition 16.10. FRAMEH: USB0 Frame Number High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	Frame Number High			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x0D

Bits7-3: Unused. Read = 0. Write = don’t care.
Bits2-0: Frame Number High Byte
This register contains bits10-8 of the last received frame number.

16.8. Interrupts

The read-only USB0 interrupt flags are located in the USB registers shown in USB Register Definition 16.11 through USB Register Definition 16.13. The associated interrupt enable bits are located in the USB registers shown in USB Register Definition 16.14 through USB Register Definition 16.16. A USB0 interrupt is generated when any of the USB interrupt flags is set to ‘1’. The USB0 interrupt is enabled via the EIE1 SFR (see **Section “9.3. Interrupt Handler” on page 88**).

Important Note: Reading a USB interrupt flag register resets all flags in that register to ‘0’.

USB Register Definition 16.11. IN1INT: USB0 IN Endpoint Interrupt

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	IN3	IN2	IN1	EP0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x02

Bits7–4: Unused. Read = 0000b. Write = don’t care.
Bit3: IN3: IN Endpoint 3 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 3 interrupt inactive.
1: IN Endpoint 3 interrupt active.
Bit2: IN2: IN Endpoint 2 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 2 interrupt inactive.
1: IN Endpoint 2 interrupt active.
Bit1: IN1: IN Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 1 interrupt inactive.
1: IN Endpoint 1 interrupt active.
Bit0: EP0: Endpoint 0 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: Endpoint 0 interrupt inactive.
1: Endpoint 0 interrupt active.

USB Register Definition 16.12. OUT1INT: USB0 Out Endpoint Interrupt

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	OUT3	OUT2	OUT1	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x04

Bits7–4: Unused. Read = 0000b. Write = don’t care.
Bit3: OUT3: OUT Endpoint 3 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 3 interrupt inactive.
1: OUT Endpoint 3 interrupt active.
Bit2: OUT2: OUT Endpoint 2 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 2 interrupt inactive.
1: OUT Endpoint 2 interrupt active.
Bit1: OUT1: OUT Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 1 interrupt inactive.
1: OUT Endpoint 1 interrupt active.
Bit0: Unused. Read = 0; Write = don’t care.

The E0CNT register (USB Register Definition 16.18) holds the number of received data bytes in the Endpoint0 FIFO.

Hardware will automatically detect protocol errors and send a STALL condition in response. Firmware may force a STALL condition to abort the current transfer. When a STALL condition is generated, the STSTL bit will be set to ‘1’ and an interrupt generated. The following conditions will cause hardware to generate a STALL condition:

1. The host sends an OUT token during a OUT data phase after the DATAEND bit has been set to ‘1’.
2. The host sends an IN token during an IN data phase after the DATAEND bit has been set to ‘1’.
3. The host sends a packet that exceeds the maximum packet size for Endpoint0.
4. The host sends a non-zero length DATA1 packet during the status phase of an IN transaction.
5. Firmware sets the SDSTL bit (E0CSR.5) to ‘1’.

16.10.1.Endpoint0 SETUP Transactions

All control transfers must begin with a SETUP packet. SETUP packets are similar to OUT packets, containing an 8-byte data field sent by the host. Any SETUP packet containing a command field of anything other than 8 bytes will be automatically rejected by USB0. An Endpoint0 interrupt is generated when the data from a SETUP packet is loaded into the Endpoint0 FIFO. Software should unload the command from the Endpoint0 FIFO, decode the command, perform any necessary tasks, and set the SOPRDY bit to indicate that it has serviced the OUT packet.

16.10.2.Endpoint0 IN Transactions

When a SETUP request is received that requires USB0 to transmit data to the host, one or more IN requests will be sent by the host. For the first IN transaction, firmware should load an IN packet into the Endpoint0 FIFO, and set the INPRDY bit (E0CSR.1). An interrupt will be generated when an IN packet is transmitted successfully. Note that no interrupt will be generated if an IN request is received before firmware has loaded a packet into the Endpoint0 FIFO. If the requested data exceeds the maximum packet size for Endpoint0 (as reported to the host), the data should be split into multiple packets; each packet should be of the maximum packet size excluding the last (residual) packet. If the requested data is an integer multiple of the maximum packet size for Endpoint0, the last data packet should be a zero-length packet signaling the end of the transfer. Firmware should set the DATAEND bit to ‘1’ after loading into the Endpoint0 FIFO the last data packet for a transfer.

Upon reception of the first IN token for a particular control transfer, Endpoint0 is said to be in Transmit Mode. In this mode, only IN tokens should be sent by the host to Endpoint0. The SUEND bit (E0CSR.4) is set to ‘1’ if a SETUP or OUT token is received while Endpoint0 is in Transmit Mode.

Endpoint0 will remain in Transmit Mode until any of the following occur:

1. USB0 receives an Endpoint0 SETUP or OUT token.
2. Firmware sends a packet less than the maximum Endpoint0 packet size.
3. Firmware sends a zero-length packet.

Firmware should set the DATAEND bit (E0CSR.3) to ‘1’ when performing (2) and (3) above.

The SIE will transmit a NAK in response to an IN token if there is no packet ready in the IN FIFO (INPRDY = ‘0’).

USB Register Definition 16.21. EOUTCSRL: USB0 OUT Endpoint Control Low Byte

W	R/W	R/W	R/W	R	R/W	R	R/W	Reset Value
CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x14

Bit7: CLRDT: Clear Data Toggle
Write: Software should write ‘1’ to this bit to reset the OUT endpoint data toggle to ‘0’.
Read: This bit always reads ‘0’.

Bit6: STSTL: Sent Stall
Hardware sets this bit to ‘1’ when a STALL handshake signal is transmitted. This flag must be cleared by software.

Bit5: SDSTL: Send Stall
Software should write ‘1’ to this bit to generate a STALL handshake. Software should write ‘0’ to this bit to terminate the STALL signal. This bit has no effect in ISO mode.

Bit4: FLUSH: FIFO Flush
Writing a ‘1’ to this bit flushes the next packet to be read from the OUT endpoint FIFO. The FIFO pointer is reset and the OPRDY bit is cleared. If the FIFO contains multiple packets, software must write ‘1’ to FLUSH for each packet. Hardware resets the FLUSH bit to ‘0’ when the FIFO flush is complete.

Note: If data for the current packet has already been read from the FIFO, the FLUSH bit should not be used to flush the packet. Instead, the entire data packet should be read from the FIFO manually.

Bit3: DATERR: Data Error
In ISO mode, this bit is set by hardware if a received packet has a CRC or bit-stuffing error. It is cleared when software clears OPRDY. This bit is only valid in ISO mode.

Bit2: OVRUN: Data Overrun
This bit is set by hardware when an incoming data packet cannot be loaded into the OUT endpoint FIFO. This bit is only valid in ISO mode, and must be cleared by software.
0: No data overrun.
1: A data packet was lost because of a full FIFO since this flag was last cleared.

Bit1: FIFOFUL: OUT FIFO Full
This bit indicates the contents of the OUT FIFO. If double buffering is enabled for the endpoint (DBIEN = ‘1’), the FIFO is full when the FIFO contains two packets. If DBIEN = ‘0’, the FIFO is full when the FIFO contains one packet.
0: OUT endpoint FIFO is not full.
1: OUT endpoint FIFO is full.

Bit0: OPRDY: OUT Packet Ready
Hardware sets this bit to ‘1’ and generates an interrupt when a data packet is available. Software should clear this bit after each data packet is unloaded from the OUT endpoint FIFO.

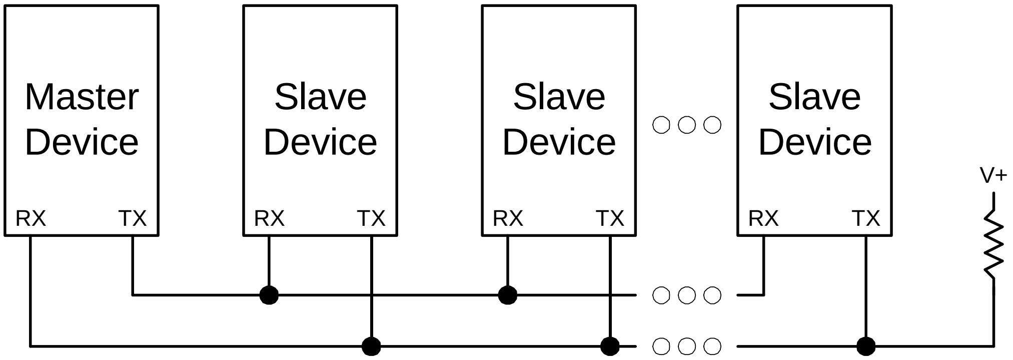


Figure 18.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 21.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level.
1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.3).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level.
1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.2).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

21.3.3. USB Start-of-Frame Capture

When T3CE = ‘1’, Timer 3 will operate in one of two special capture modes. The capture event can be selected between a USB Start-of-Frame (SOF) capture, and a Low-Frequency Oscillator (LFO) Rising Edge capture, using the T3CSS bit. The USB SOF capture mode can be used to calibrate the system clock or external oscillator against the known USB host SOF clock. The LFO rising-edge capture mode can be used to calibrate the internal Low-Frequency Oscillator against the internal High-Frequency Oscillator or an external clock source. When T3SPLIT = ‘0’, Timer 3 counts up and overflows from 0xFFFF to 0x0000. Each time a capture event is received, the contents of the Timer 3 registers (TMR3H:TMR3L) are latched into the Timer 3 Reload registers (TMR3RLH:TMR3RLL). A Timer 3 interrupt is generated if enabled.

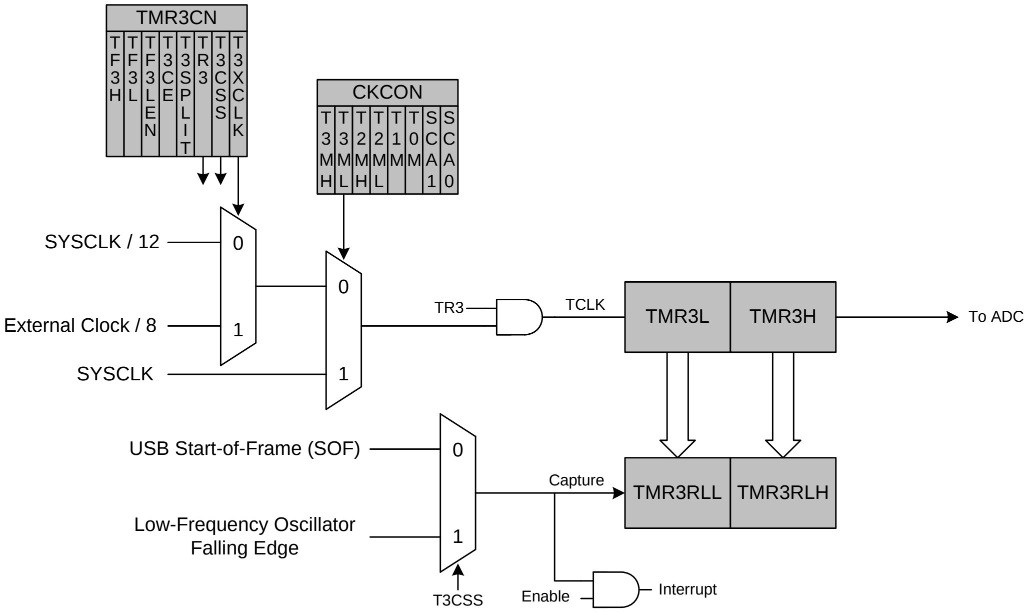


Figure 21.10. Timer 3 Capture Mode (T3SPLIT = ‘0’)

22.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are 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. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

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’.

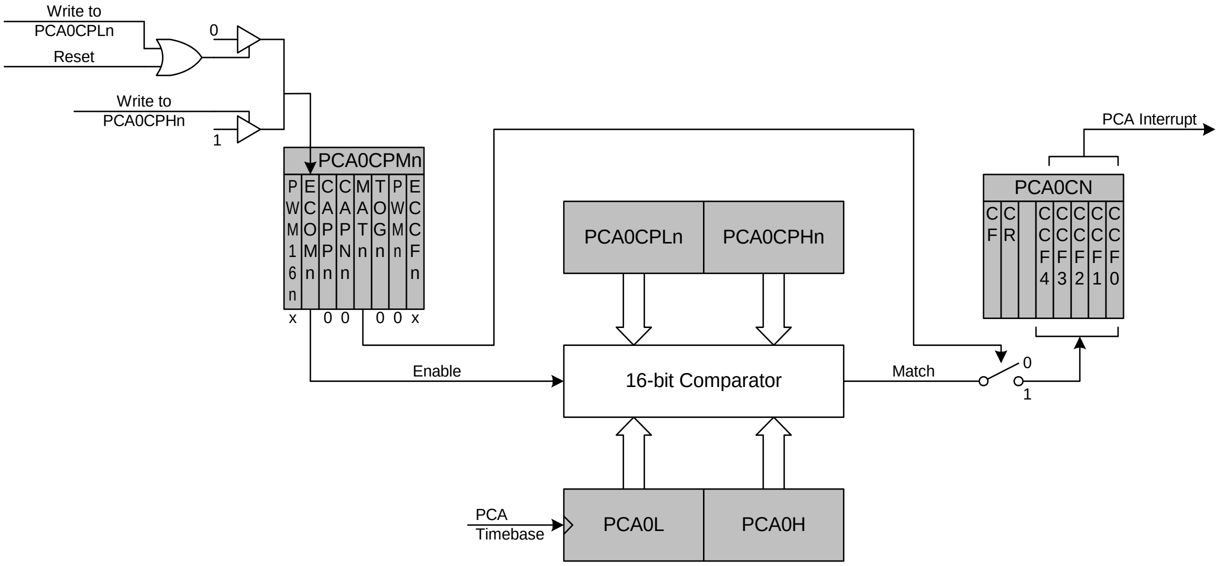


Figure 22.5. PCA Software Timer Mode Diagram

22.2.5. 8-Bit Pulse Width Modulator Mode

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. The duty cycle of the PWM output signal 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 22.8). 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 enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 22.2.

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’.

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Equation 22.2. 8-Bit PWM Duty Cycle

Using Equation 22.2, 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’.

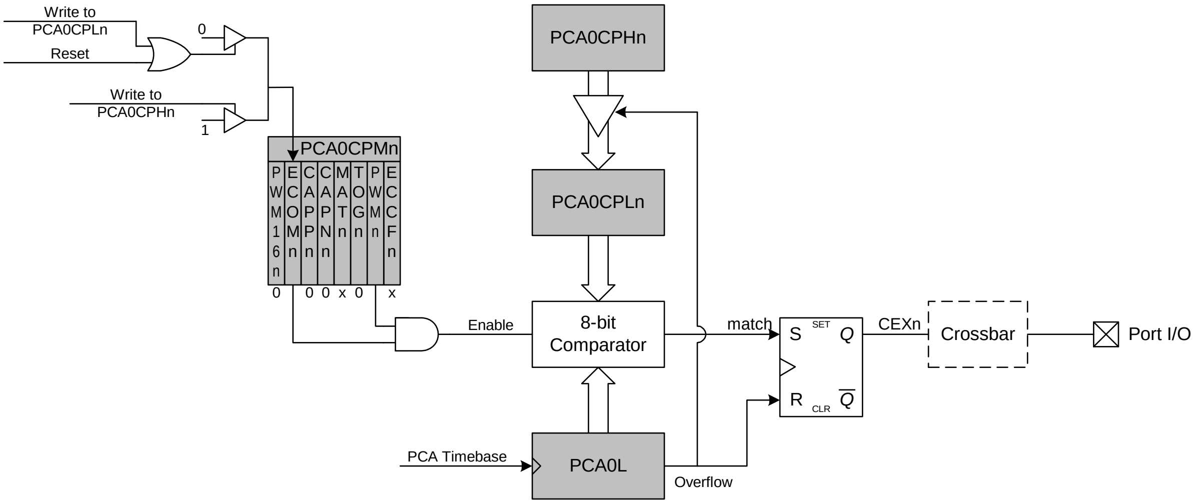


Figure 22.8. PCA 8-Bit PWM Mode Diagram