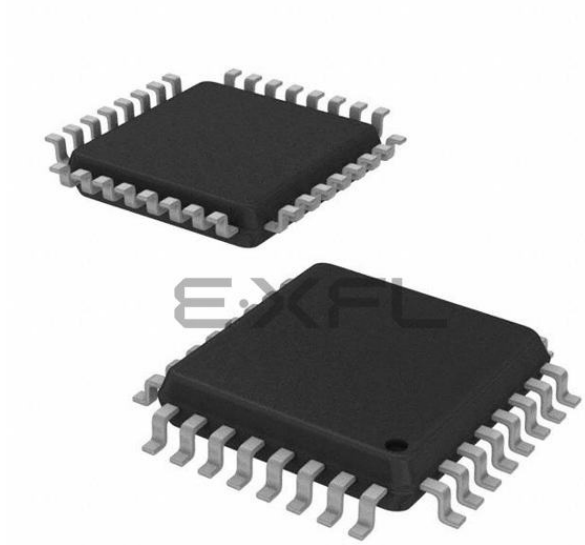


Welcome to E-XFL.COM

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 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f34a-gq

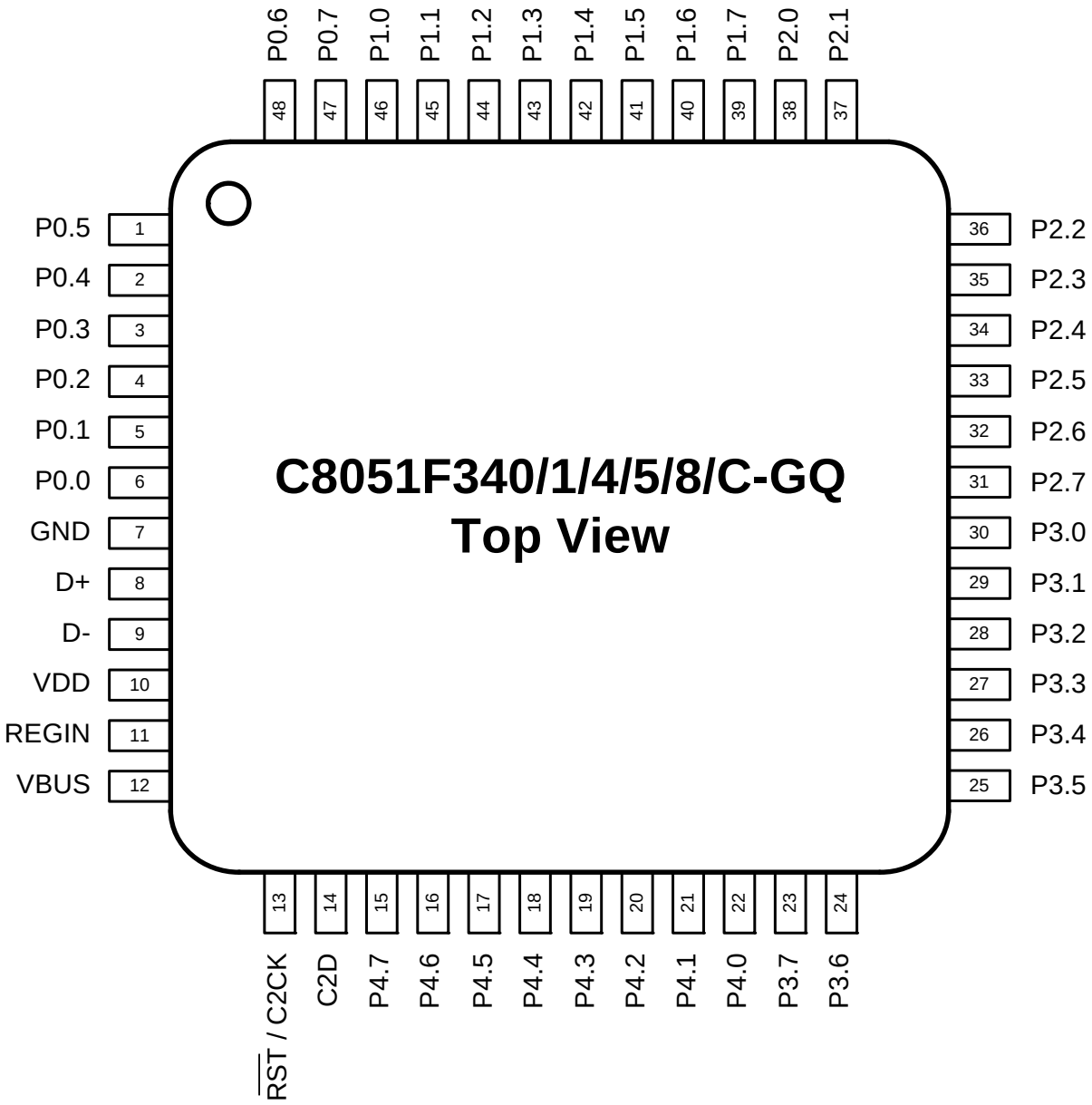


Figure 4.1. TQFP-48 Pinout Diagram (Top View)

5.3.3. Settling Time Requirements

When the ADC0 input configuration is changed (i.e., a different AMUX0 selection is made), a minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the minimum tracking time requirements.

Figure 5.5 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 5.1. When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 5.1 for ADC0 minimum settling time requirements.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 5.1. ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).

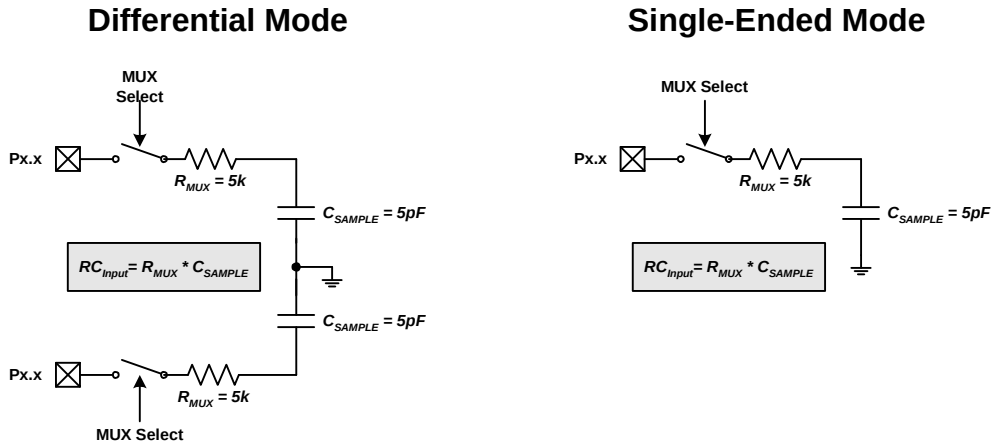


Figure 5.5. ADC0 Equivalent Input Circuits

SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC6

Bits7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC5

Bits7–0: Low byte of ADC0 Less-Than Data Word.

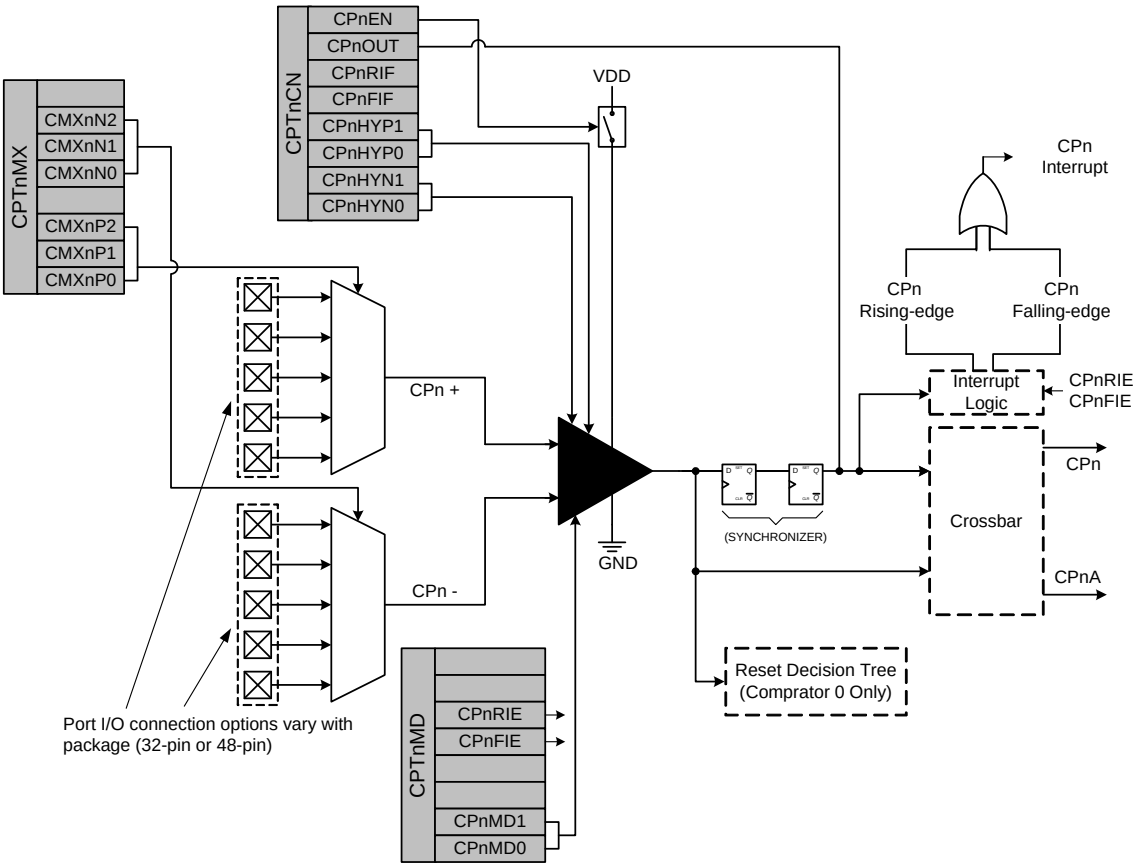


Figure 7.1. Comparator Functional Block Diagram

Comparator outputs can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, Comparator outputs are available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and supply current falls to less than 100 nA. See **Section "15.1. Priority Crossbar Decoder" on page 144** for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Table 7.1.

Comparator response time may be configured in software via the CPTnMD registers (see SFR Definition 7.3 and SFR Definition 7.6). Selecting a longer response time reduces the Comparator supply current. See Table 7.1 for complete timing and supply current specifications.

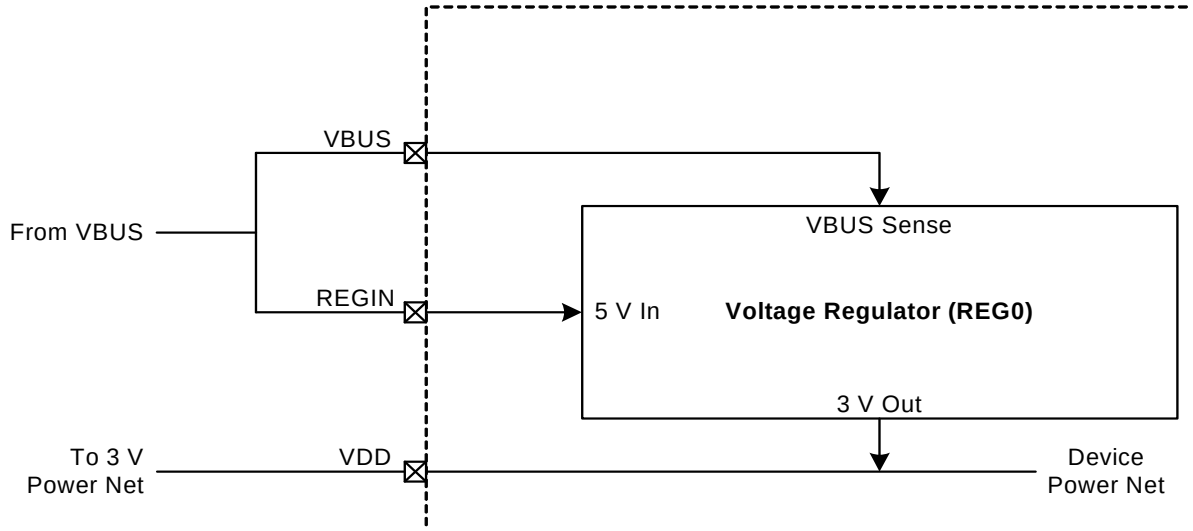


Figure 8.1. REG0 Configuration: USB Bus-Powered

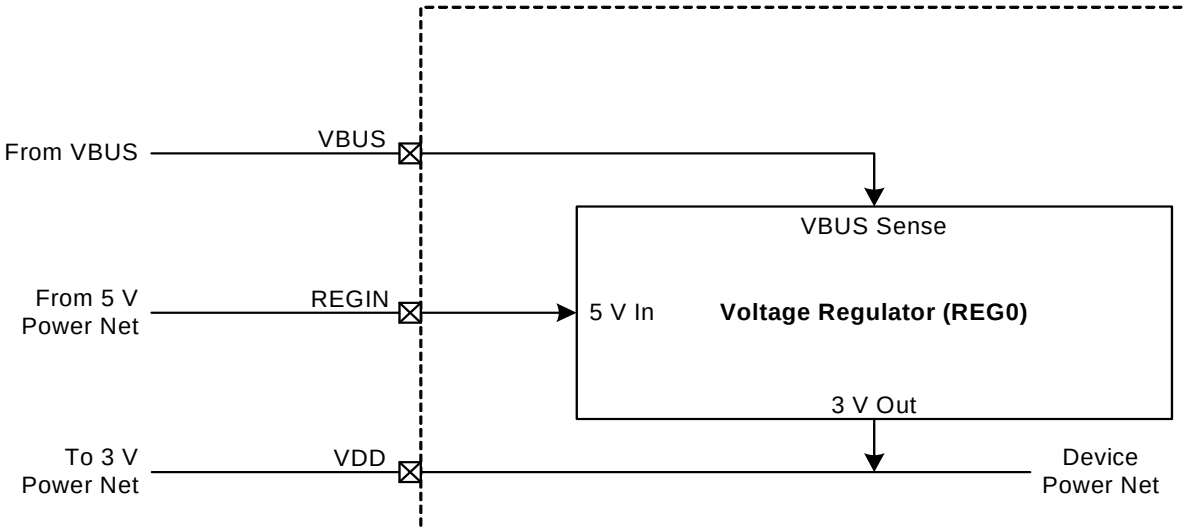


Figure 8.2. REG0 Configuration: USB Self-Powered

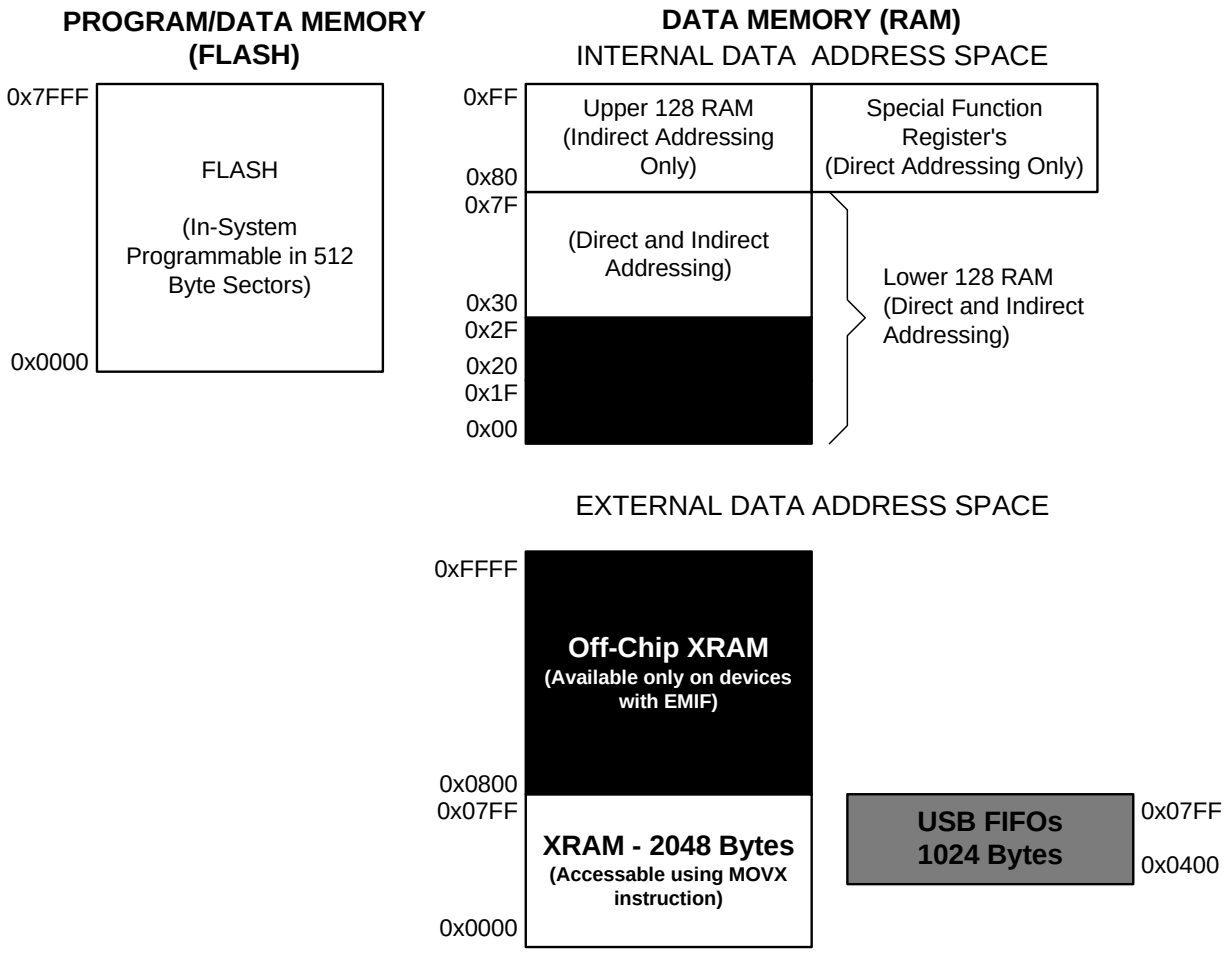


Figure 9.3. On-Chip Memory Map for 32 kB Devices

9.2.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F34x implements 64k or 32k bytes of this program memory space as in-system, re-programmable Flash memory. Note that on the 64k versions of the C8051F34x, addresses above 0xFBFF are reserved.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “12. Flash Memory” on page 107** for further details.

9.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 9.2 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 9.3, for a detailed description of each register.

Table 9.2. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	B	P0MDIN	P1MDIN	P2MDIN	P3MDIN	P4MDIN	EIP1	EIP2
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF	SMOD1	EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	P3SKIP
D0	PSW	REF0CN	SCON1	SBUF1	P0SKIP	P1SKIP	P2SKIP	USB0XCN
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	-	-
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P4
B8	IP	CLKMUL	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	-
B0	P3	OSCXCN	OSCICN	OSCICL	SBRLL1	SBRLH1	FLSCL	FLKEY
A8	IE	CLKSEL	EMI0CN	-	SBCON1	-	P4MDOUT	PFE0CN
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	USB0ADR	USB0DAT
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	EMI0TC	EMI0CF	OSCLCN	PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

SFR Definition 9.8. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB8

Bit7: UNUSED. Read = 1, Write = don't care.

Bit6: PSPI0: Serial Peripheral Interface (SPI0) Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority level.
1: SPI0 interrupt set to high priority level.

Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority level.
1: Timer 2 interrupts set to high priority level.

Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupts set to high priority level.

Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupts set to high priority level.

Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.

Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.

Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

SFR Definition 9.13. IT01CF: INT0/INT1 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xE4

Note: Refer to SFR Definition 21.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: $\overline{\text{INT1}}$ Polarity
0: $\overline{\text{INT1}}$ input is active low.
1: $\overline{\text{INT1}}$ input is active high.

Bits6–4: IN1SL2–0: $\overline{\text{INT1}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN1SL2–0	$\overline{\text{INT1}}$ Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

Bit3: IN0PL: $\overline{\text{INT0}}$ Polarity
0: $\overline{\text{INT0}}$ interrupt is active low.
1: $\overline{\text{INT0}}$ interrupt is active high.

Bits2–0: INT0SL2–0: $\overline{\text{INT0}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar. $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN0SL2–0	$\overline{\text{INT0}}$ Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

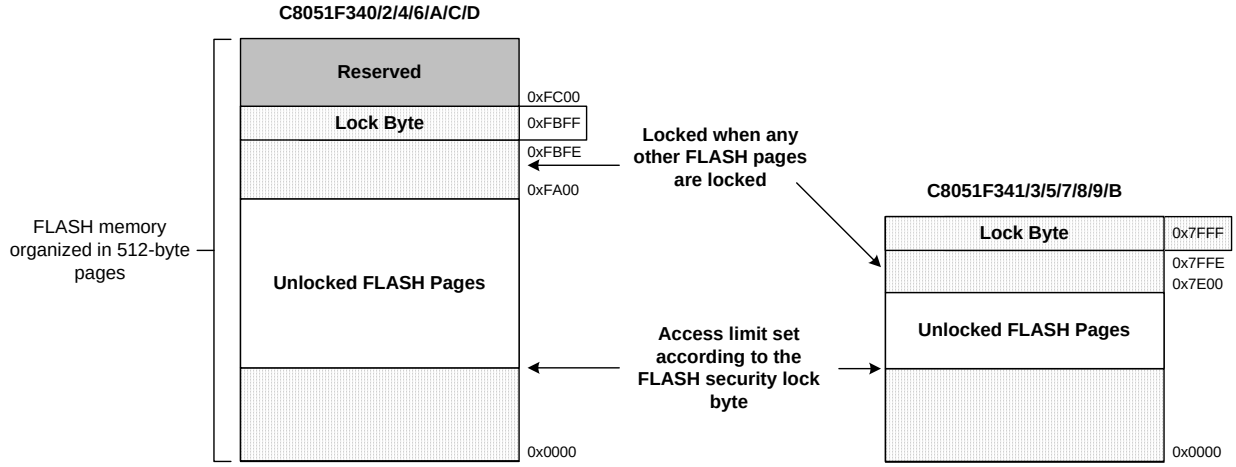


Figure 12.1. Flash Program Memory Map and Security Byte

SFR Definition 13.2. EMI0CF: External Memory Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	USBFAE	-	EMD2	EMD1	EMD0	EALE1	EALE0	00000011
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x85

Bit7: Unused. Read = 0b. Write = don't care.

Bit6: USBFAE: USB FIFO Access Enable.
0: USB FIFO RAM not available through MOVX instructions.
1: USB FIFO RAM available using MOVX instructions. The 1k of USB RAM will be mapped in XRAM space at addresses 0x0400 to 0x07FF. **The USB clock must be active and greater than or equal to twice the SYSCLK (USBCLK $\geq$ 2 x SYSCLK) to access this area with MOVX instructions.**

Bit5: Unused. Read = 0b. Write = don't care.

Bit4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).

Bits3–2: EMD1–0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space.
01: Split Mode without Bank Select: Accesses below the on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM boundary are directed off-chip. 8-bit off-chip MOVX operations use the current contents of the Address High port latches to resolve upper address byte. Note that in order to access off-chip space, EMI0CN must be set to a page that is not contained in the on-chip address space.
10: Split Mode with Bank Select: Accesses below the on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM boundary are directed off-chip. 8-bit off-chip MOVX operations use the contents of EMI0CN to determine the high-byte of the address.
11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.

Bits1–0: EALE1–0: ALE Pulse-Width Select Bits (only has effect when EMD2 = 0).
00: ALE high and ALE low pulse width = 1 SYSCLK cycle.
01: ALE high and ALE low pulse width = 2 SYSCLK cycles.
10: ALE high and ALE low pulse width = 3 SYSCLK cycles.
11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

13.5.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 13.3. See **Section "13.7.1. Non-multiplexed Mode" on page 124** for more information about Non-multiplexed operation.

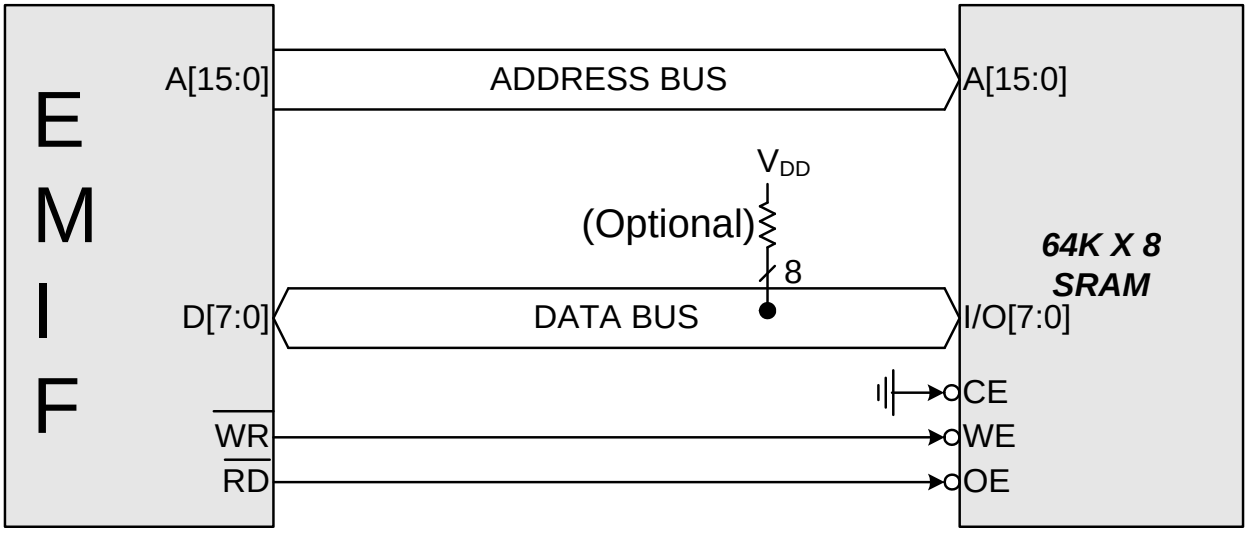


Figure 13.3. Non-multiplexed Configuration Example

13.6. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 13.4, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 13.2). These modes are summarized below. More information about the different modes can be found in **Section "13.7. Timing" on page 122**.

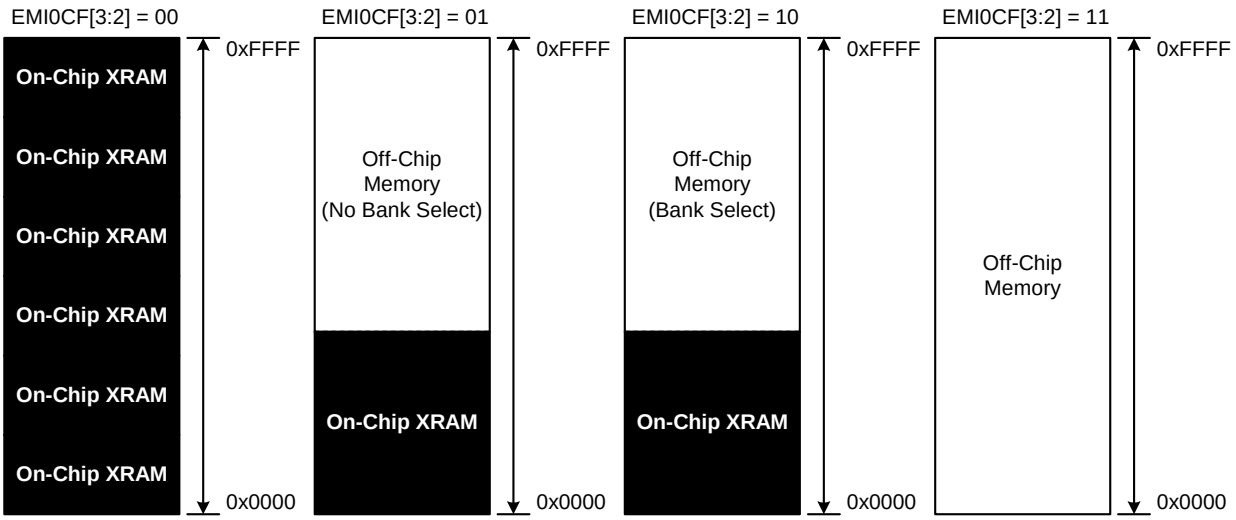


Figure 13.4. EMIF Operating Modes

Table 13.1. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address / Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address / Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address / Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

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').

16.10.3.Endpoint0 OUT Transactions

When a SETUP request is received that requires the host to transmit data to USB0, one or more OUT requests will be sent by the host. When an OUT packet is successfully received by USB0, hardware will set the OPRDY bit (E0CSR.0) to ‘1’ and generate an Endpoint0 interrupt. Following this interrupt, firmware should unload the OUT packet from the Endpoint0 FIFO and set the SOPRDY bit (E0CSR.6) to ‘1’.

If the amount of data required for the transfer exceeds the maximum packet size for Endpoint0, the data will be split into multiple packets. If the requested data is an integer multiple of the maximum packet size for Endpoint0 (as reported to the host), the host will send a zero-length data packet signaling the end of the transfer.

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

Endpoint0 will remain in Receive mode until:

1. The SIE receives a SETUP or IN token.
2. The host sends a packet less than the maximum Endpoint0 packet size.
3. The host sends a zero-length packet.

Firmware should set the DATAEND bit (E0CSR.3) to ‘1’ when the expected amount of data has been received. The SIE will transmit a STALL condition if the host sends an OUT packet after the DATAEND bit has been set by firmware. An interrupt will be generated with the STSTL bit (E0CSR.2) set to ‘1’ after the STALL is transmitted.

space is made available in the FIFO for another incoming byte. If enabled, an interrupt will occur when RI1 is set. RI1 can only be cleared to '0' by software when there is no more information in the FIFO. The recommended procedure to empty the FIFO contents is as follows:

1. Clear RI1 to '0'.
2. Read SBUF1.
3. Check RI1, and repeat at step 1 if RI1 is set to '1'.

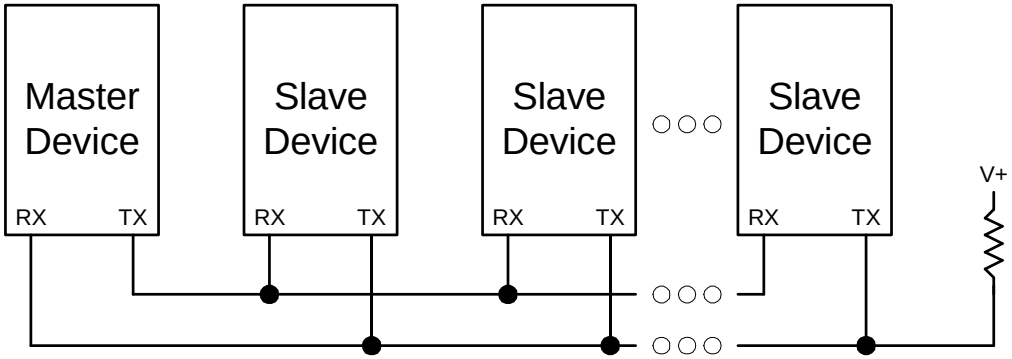
If the extra bit function is enabled (XBE1 = ‘1’) and the parity function is disabled (PE1 = ‘0’), the extra bit for the oldest byte in the FIFO can be read from the RBX1 bit (SCON1.2). If the extra bit function is not enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX1. When the parity function is enabled (PE1 = ‘1’), hardware will check the received parity bit against the selected parity type (selected with S1PT[1:0]) when receiving data. If a byte with parity error is received, the PERR1 flag will be set to ‘1’. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

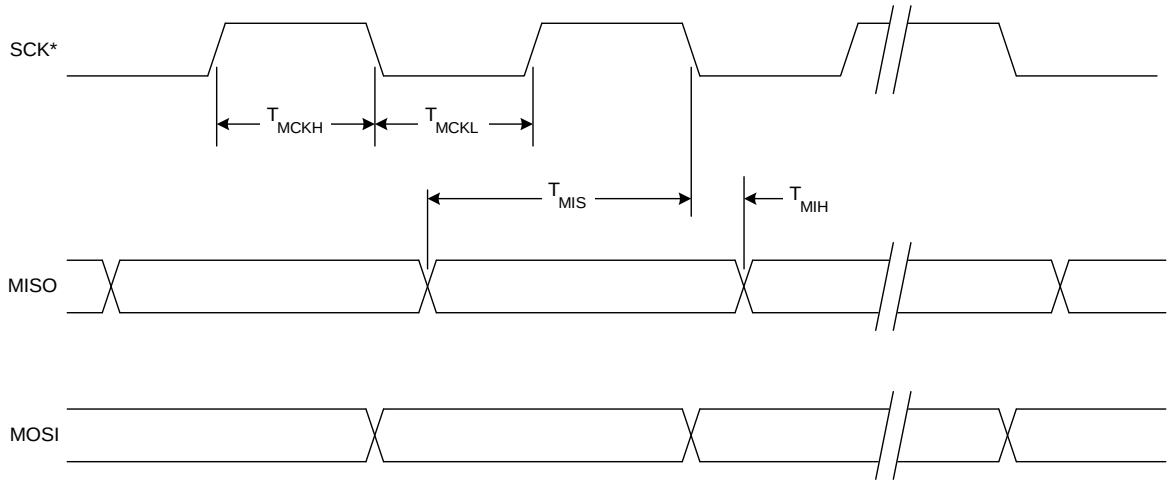
19.3.3. Multiprocessor Communications

UART1 supports multiprocessor communication between a master processor and one or more slave processors by special use of the extra data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its extra bit is logic 1; in a data byte, the extra bit is always set to logic 0.

Setting the MCE1 bit (SMOD1.7) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the extra bit is logic 1 (RBX1 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned address. If the addresses match, the slave will clear its MCE1 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE1 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE1 bit to ignore all transmissions until it receives the next address byte.

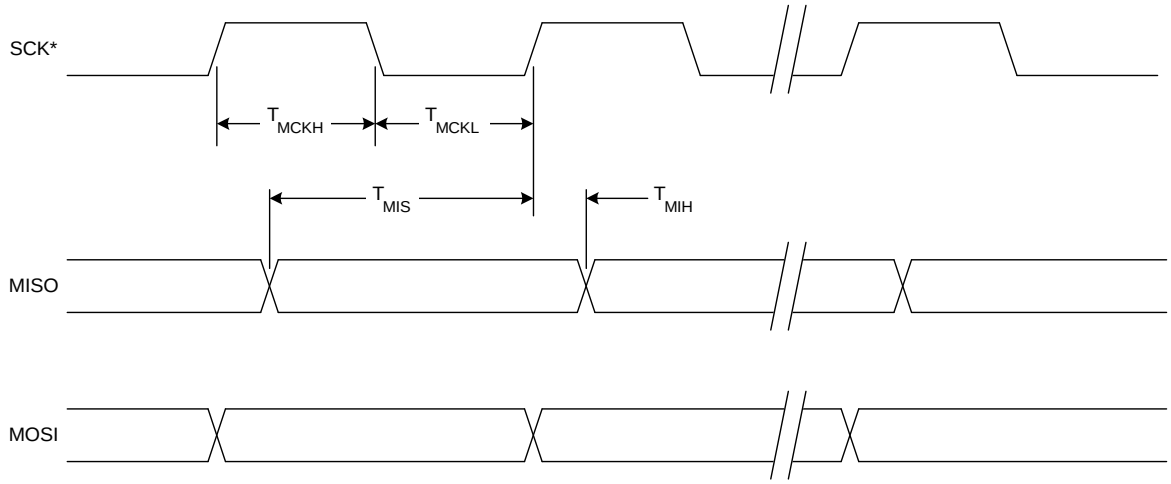
Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).





* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.8. SPI Master Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.9. SPI Master Timing (CKPHA = 1)

22. Programmable Counter Array (PCA0)

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and five 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See **Section "15.1. Priority Crossbar Decoder" on page 144** for details on configuring the Crossbar). The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each mode is described in **Section "22.2. Capture/Compare Modules" on page 257**). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 22.1

Important Note: The PCA Module 4 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. Access to certain PCA registers is restricted while WDT mode is enabled. See **Section 22.3** for details.

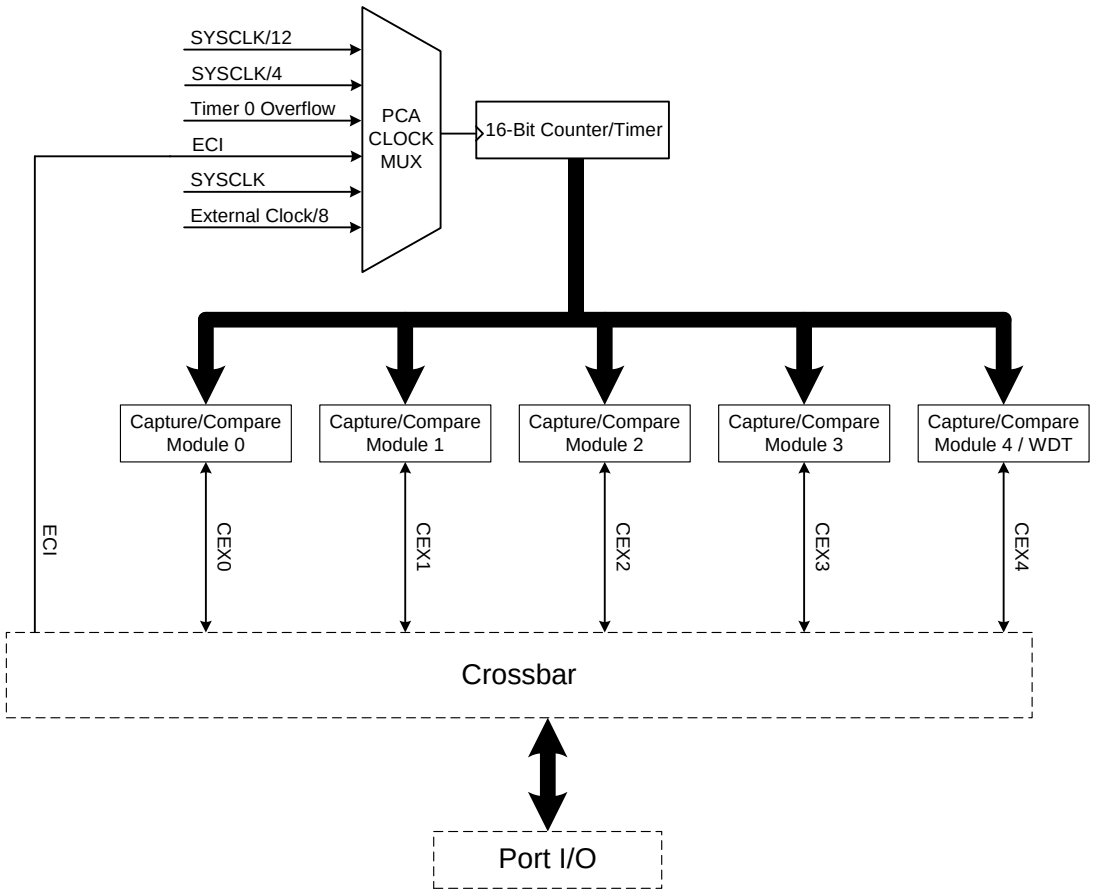


Figure 22.1. PCA Block Diagram