



Welcome to [E-XFL.COM](https://www.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/I ² C), 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-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f34a-gmr

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

11.2.Power-Fail Reset / VDD Monitor	102
11.3.External Reset	103
11.4.Missing Clock Detector Reset	103
11.5.Comparator0 Reset	103
11.6.PCA Watchdog Timer Reset	103
11.7.Flash Error Reset	103
11.8.Software Reset	104
11.9.USB Reset	104
12. Flash Memory	107
12.1.Programming The Flash Memory	107
12.1.1.Flash Lock and Key Functions	107
12.1.2.Flash Erase Procedure	107
12.1.3.Flash Write Procedure	108
12.2.Non-Volatile Data Storage	109
12.3.Security Options	109
13. External Data Memory Interface and On-Chip XRAM	114
13.1.Accessing XRAM	114
13.1.1.16-Bit MOVX Example	114
13.1.2.8-Bit MOVX Example	114
13.2.Accessing USB FIFO Space	115
13.3.Configuring the External Memory Interface	116
13.4.Port Configuration	116
13.5.Multiplexed and Non-multiplexed Selection	119
13.5.1.Multiplexed Configuration	119
13.5.2.Non-multiplexed Configuration	120
13.6.Memory Mode Selection	120
13.6.1.Internal XRAM Only	121
13.6.2.Split Mode without Bank Select	121
13.6.3.Split Mode with Bank Select	122
13.6.4.External Only	122
13.7.Timing	122
13.7.1.Non-multiplexed Mode	124
13.7.2.Multiplexed Mode	127
14. Oscillators	131
14.1.Programmable Internal High-Frequency (H-F) Oscillator	132
14.1.1.Internal H-F Oscillator Suspend Mode	132
14.2.Programmable Internal Low-Frequency (L-F) Oscillator	133
14.2.1.Calibrating the Internal L-F Oscillator	133
14.3.External Oscillator Drive Circuit	135
14.3.1.Clocking Timers Directly Through the External Oscillator	135
14.3.2.External Crystal Example	135
14.3.3.External RC Example	136
14.3.4.External Capacitor Example	136
14.4.4x Clock Multiplier	138
14.5.System and USB Clock Selection	139

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

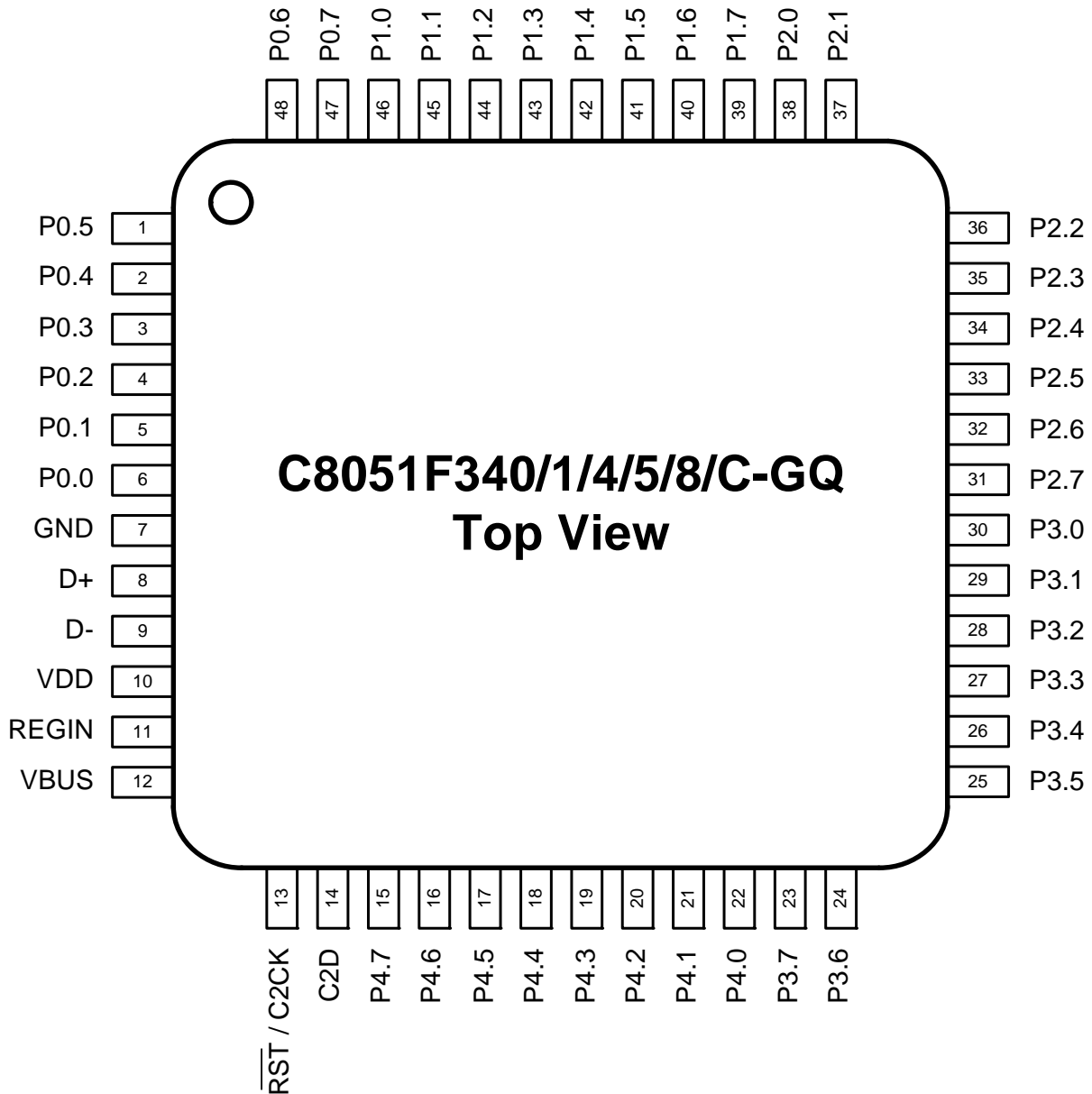


Figure 4.1. TQFP-48 Pinout Diagram (Top View)

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

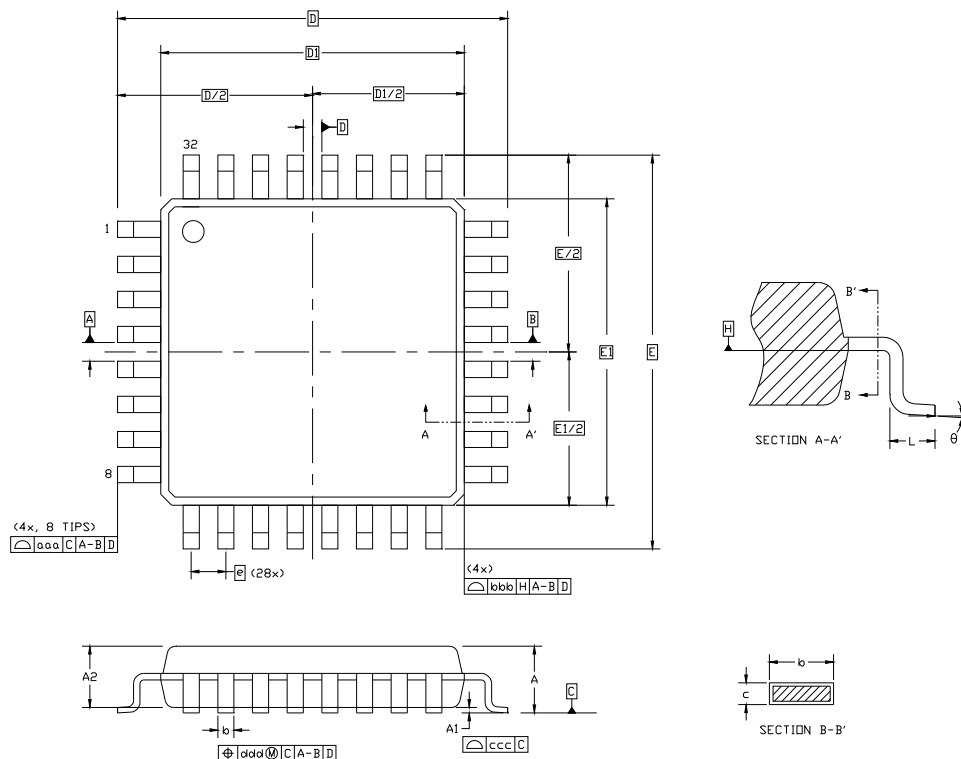


Figure 4.5. LQFP-32 Package Diagram

Table 4.4. LQFP-32 Package Dimensions

Dimension	Min	Nom	Max
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
c	0.09	—	0.20
D	9.00 BSC		
D1	7.00 BSC		
e	0.80 BSC		
E	9.00 BSC		
E1	7.00 BSC		
L	0.45	0.60	0.75
aaa	0.20		
bbb	0.20		
ccc	0.10		
ddd	0.20		
θ	0°	3.5°	7°

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MS-026, variation BBA.
4. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

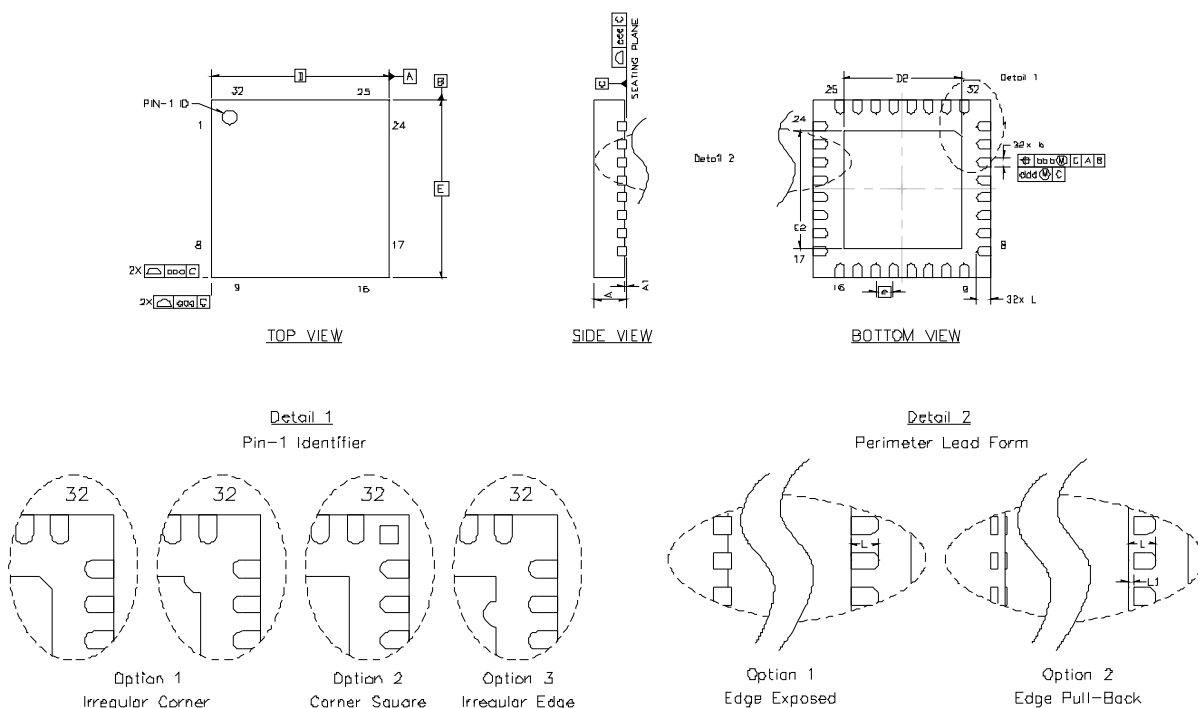


Figure 4.8. QFN-32 Package Drawing

Table 4.6. QFN-32 Package Dimensions

Dimension	Min	Nom	Max
A	0.80	0.9	1.00
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D	5.00 BSC		
D2	3.20	3.30	3.40
e	0.50 BSC		
E	5.00 BSC		
E2	3.20	3.30	3.40
L	0.30	0.40	0.50

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, and L which are tolerated per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

5.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 conversion results to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

The Window Detector registers must be written with the same format (left/right justified, signed/unsigned) as that of the current ADC configuration (left/right justified, single-ended/differential).

SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC4

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

SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC3

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

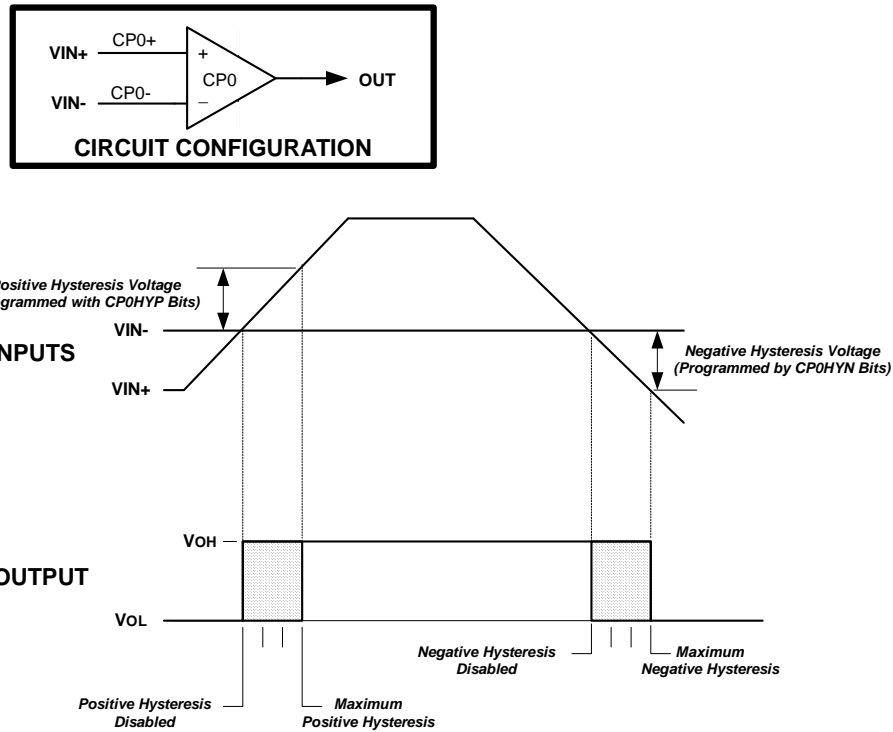


Figure 7.2. Comparator Hysteresis Plot

Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in SFR Definition 7.1 and SFR Definition 7.4). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Figure 7.2, various levels of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CPnHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section “9.3. Interrupt Handler” on page 88.**) The CPnFIF flag is set to ‘1’ upon a Comparator falling-edge, and the CPnRIF flag is set to ‘1’ upon the Comparator rising-edge. Once set, these bits remain set until cleared by software. The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to ‘1’, and is disabled by clearing this bit to ‘0’.

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that for execution time.

Clocks to Execute	1	2	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

Programming and Debugging Support

In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2). Note that the re-programmable Flash can also be read and changed a single byte at a time by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins. C2 details can be found in **Section “23. C2 Interface” on page 271**.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, debugger, and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. An 8051 assembler, linker and evaluation 'C' compiler are included in the Development Kit. Many third party macro assemblers and C compilers are also available, which can be used directly with the IDE.

9.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

9.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take two fewer clock cycles to complete when the branch is not taken as opposed to when the branch is taken. Table 9.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

13.3. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common), and skip the associated pins in the crossbar.
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic ‘1’).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition 13.2.

13.4. Port Configuration

The External Memory Interface appears on Ports 4, 3, 2, and 1 when it is used for off-chip memory access. When the EMIF is used, the Crossbar should be configured to skip over the control lines P1.7 (\overline{WR}), P1.6 (\overline{RD}), and if multiplexed mode is selected P1.3 (ALE) using the P1SKIP register. For more information about configuring the Crossbar, see **Section “Figure 15.1. Port I/O Functional Block Diagram (Port 0 through Port 3)” on page 142.**

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar settings for those pins. See **Section “15. Port Input/Output” on page 142** for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to ‘park’ the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1.**

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

13.5. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMIOCF.4) bit.

13.5.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 13.2.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the 'Q' outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time \overline{RD} or \overline{WR} is asserted.

See Section “13.7.2. Multiplexed Mode” on page 127 for more information.

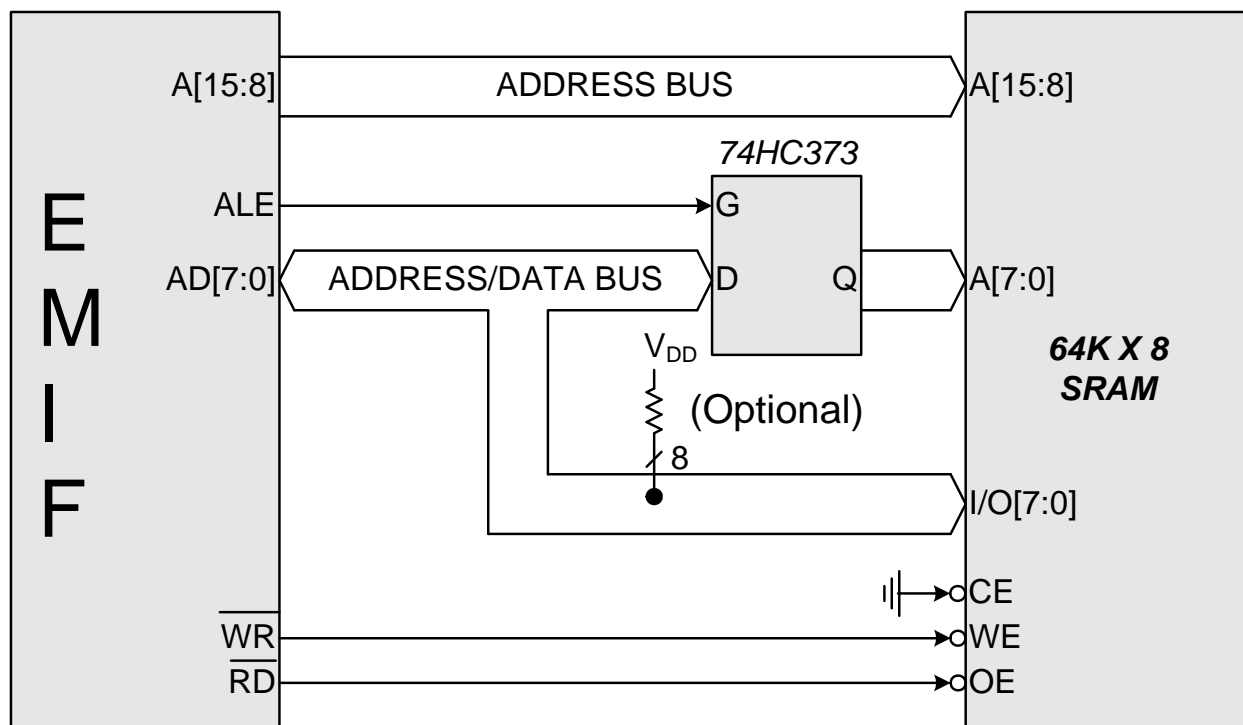


Figure 13.2. Multiplexed Configuration Example

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

13.7.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = '101' or '111'.

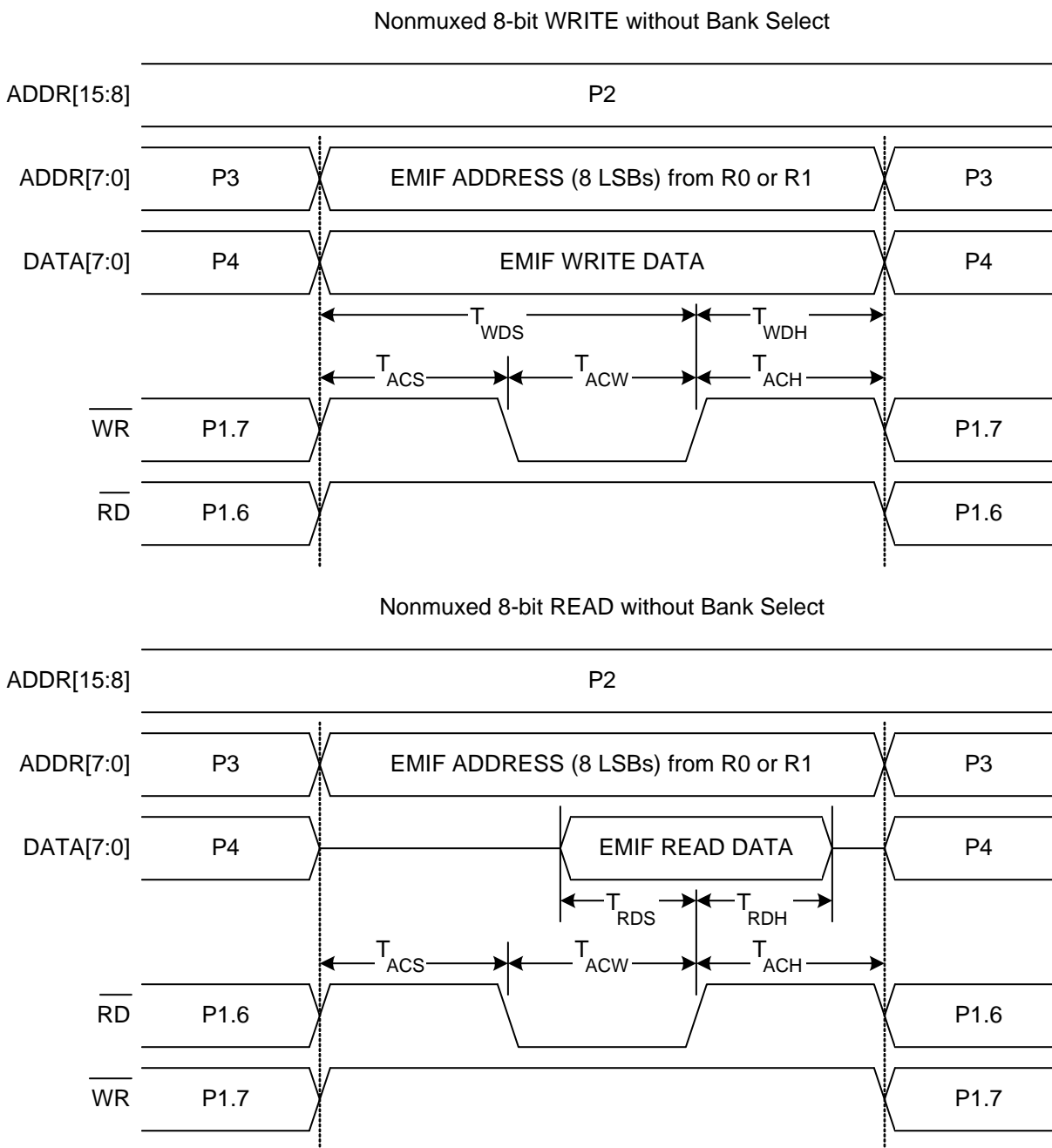


Figure 13.6. Non-multiplexed 8-bit MOVX without Bank Select Timing

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

SFR Definition 15.2. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2
<p>Bit7: WEAKPUD: Port I/O Weak Pull-up Disable. 0: Weak Pull-ups enabled (except for Ports whose I/O are configured as analog input or push-pull output). 1: Weak Pull-ups disabled.</p> <p>Bit6: XBARE: Crossbar Enable. 0: Crossbar disabled; all Port drivers disabled. 1: Crossbar enabled.</p> <p>Bit5: T1E: T1 Enable 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.</p> <p>Bit4: T0E: T0 Enable 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.</p> <p>Bit3: ECIE: PCA0 External Counter Input Enable 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.</p> <p>Bits2–0: PCA0ME: PCA Module I/O Enable Bits. 000: All PCA I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100: CEX0, CEX1, CEX2, CEX3 routed to Port pins. 101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to Port pins. 110: Reserved. 111: Reserved.</p>								

SFR Definition 15.3. XBR2: Port I/O Crossbar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
							URT1E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3
<p>Bits7–1: RESERVED: Always write to 0000000b</p> <p>Bit0: URT1E: UART1 I/O Output Enable (C8051F340/1/4/5/8/A/B Only) 0: UART1 I/O unavailable at Port pins. 1: UART1 TX1, RX1 routed to Port pins.</p>								

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

SFR Definition 15.19. P3SKIP: Port3 Skip

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: 0xDF

Bits7–0: P3SKIP[3:0]: Port3 Crossbar Skip Enable Bits.

These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.

0: Corresponding P3.n pin is not skipped by the Crossbar.

1: Corresponding P3.n pin is skipped by the Crossbar.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.20. P4: Port4 Latch

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC7

Bits7–0: P4.[7:0]

Write - Output appears on I/O pins.

0: Logic Low Output.

1: Logic High Output (high impedance if corresponding P4MDOUT.n bit = 0).

Read - Always reads '0' if selected as analog input in register P4MDIN. Directly reads Port pin when configured as digital input.

0: P4.n pin is logic low.

1: P4.n pin is logic high.

Note: P4 is only available on 48-pin devices.

16.5.2. FIFO Double Buffering

FIFO slots for Endpoints1-3 can be configured for double-buffered mode. In this mode, the maximum packet size is halved and the FIFO may contain two packets at a time. This mode is available for Endpoints1-3. When an endpoint is configured for Split Mode, double buffering may be enabled for the IN Endpoint and/or the OUT endpoint. When Split Mode is not enabled, double-buffering may be enabled for the entire endpoint FIFO. See Table 16.3 for a list of maximum packet sizes for each FIFO configuration.

Table 16.3. FIFO Configurations

Endpoint Number	Split Mode Enabled?	Maximum IN Packet Size (Double Buffer Disabled / Enabled)	Maximum OUT Packet Size (Double Buffer Disabled / Enabled)
0	N/A	64	
1	N	128 / 64	
	Y	64 / 32	64 / 32
2	N	256 / 128	
	Y	128 / 64	128 / 64
3	N	512 / 256	
	Y	256 / 128	256 / 128

16.5.1. FIFO Access

Each endpoint FIFO is accessed through a corresponding FIFOn register. A read of an endpoint FIFOn register unloads one byte from the FIFO; a write of an endpoint FIFOn register loads one byte into the endpoint FIFO. When an endpoint FIFO is configured for Split Mode, a read of the endpoint FIFOn register unloads one byte from the OUT endpoint FIFO; a write of the endpoint FIFOn register loads one byte into the IN endpoint FIFO.

USB Register Definition 16.6. FIFOn: USB0 Endpoint FIFO Access

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FIFODATA								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x20 - 0x23

USB Addresses 0x20–0x23 provide access to the 4 pairs of endpoint FIFOs:

IN/OUT Endpoint FIFO	USB Address
0	0x20
1	0x21
2	0x22
3	0x23

Writing to the FIFO address loads data into the IN FIFO for the corresponding endpoint. Reading from the FIFO address unloads data from the OUT FIFO for the corresponding endpoint.

USB Register Definition 16.18. E0CNT: USB0 Endpoint 0 Data Count

R	R	R	R	R	R	R	R	Reset Value
-	E0CNT							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x16

Bit7: Unused. Read = 0; Write = don't care.
 Bits6–0: E0CNT: Endpoint 0 Data Count
 This 7-bit number indicates the number of received data bytes in the Endpoint 0 FIFO. This number is only valid while bit OPRDY is a '1'.

16.11. Configuring Endpoints1-3

Endpoints1-3 are configured and controlled through their own sets of the following control/status registers: IN registers EINCSSL and EINCSSLH, and OUT registers EOUTSSL and EOUTSSLH. Only one set of endpoint control/status registers is mapped into the USB register address space at a time, defined by the contents of the INDEX register (USB Register Definition 16.4).

Endpoints1-3 can be configured as IN, OUT, or both IN/OUT (Split Mode) as described in **Section 16.5.1**. The endpoint mode (Split/Normal) is selected via the SPLIT bit in register EINCSSLH.

When SPLIT = '1', the corresponding endpoint FIFO is split, and both IN and OUT pipes are available.

When SPLIT = '0', the corresponding endpoint functions as either IN or OUT; the endpoint direction is selected by the DIRSEL bit in register EINCSSLH.

16.12. Controlling Endpoints1-3 IN

Endpoints1-3 IN are managed via USB registers EINCSSL and EINCSSLH. All IN endpoints can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing '1' to the ISO bit in register EINCSSLH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1-3 IN interrupt is generated by any of the following conditions:

1. An IN packet is successfully transferred to the host.
2. Software writes '1' to the FLUSH bit (EINCSSL.3) when the target FIFO is not empty.
3. Hardware generates a STALL condition.

16.12.1. Endpoints1-3 IN Interrupt or Bulk Mode

When the ISO bit (EINCSSLH.6) = '0' the target endpoint operates in Bulk or Interrupt Mode. Once an endpoint has been configured to operate in Bulk/Interrupt IN mode (typically following an Endpoint0 SET_INTERFACE command), firmware should load an IN packet into the endpoint IN FIFO and set the INPRDY bit (EINCSSL.0). Upon reception of an IN token, hardware will transmit the data, clear the INPRDY bit, and generate an interrupt.

18. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section “18.1. Enhanced Baud Rate Generation” on page 206**). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

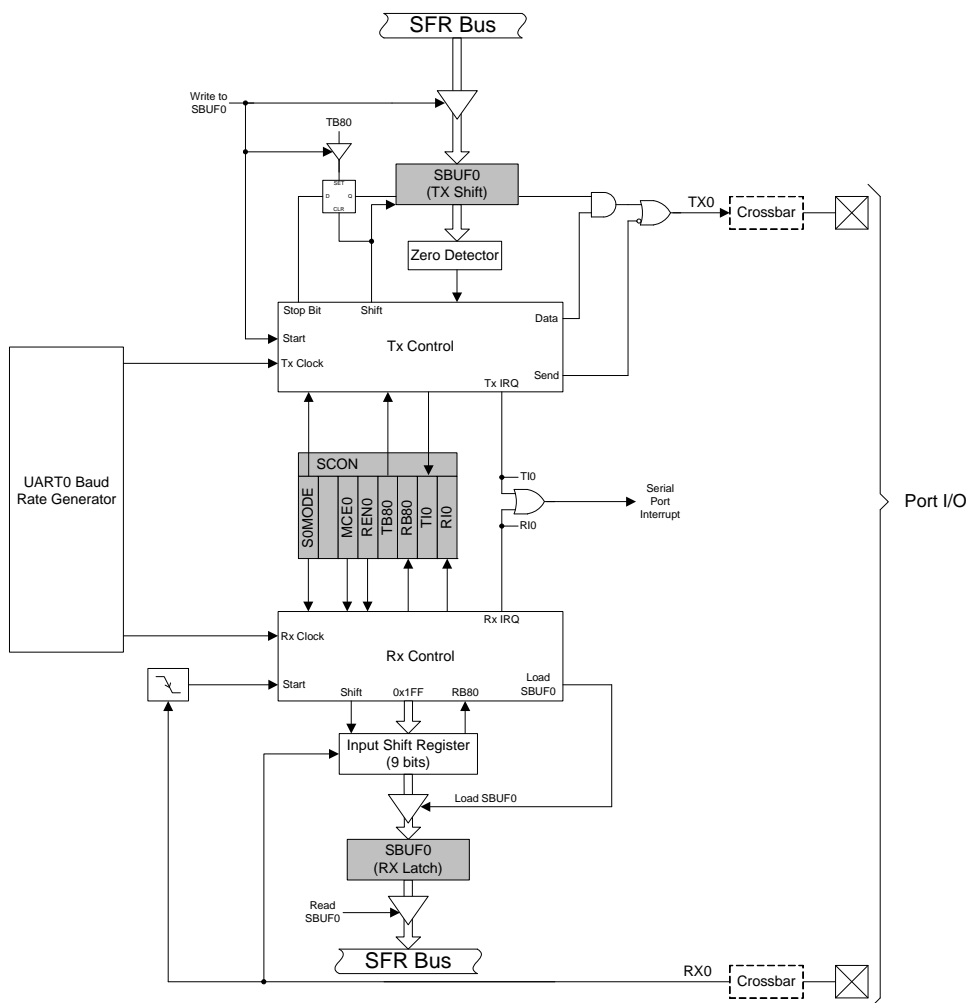


Figure 18.1. UART0 Block Diagram

18.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to '1'. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to '1'. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to '1'.

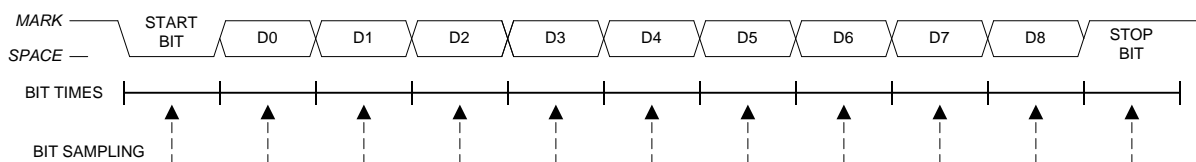


Figure 18.5. 9-Bit UART Timing Diagram

18.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth 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 ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 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 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 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 MCE0 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).

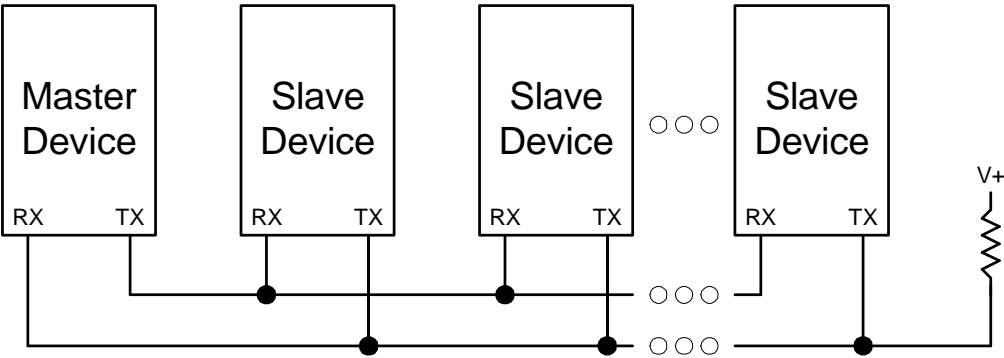


Figure 18.6. UART Multi-Processor Mode Interconnect Diagram

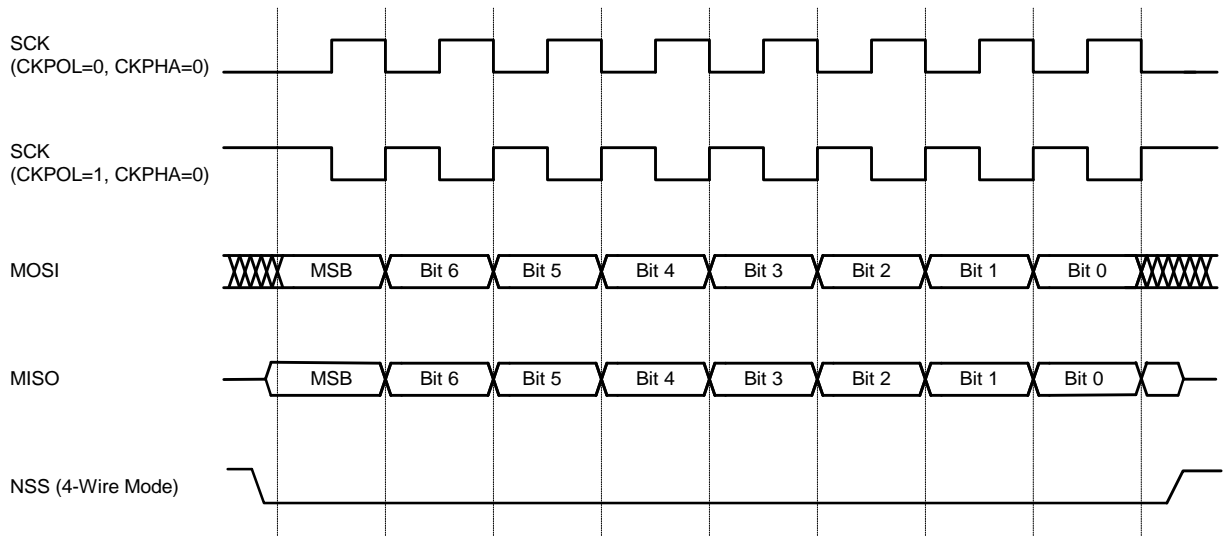


Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

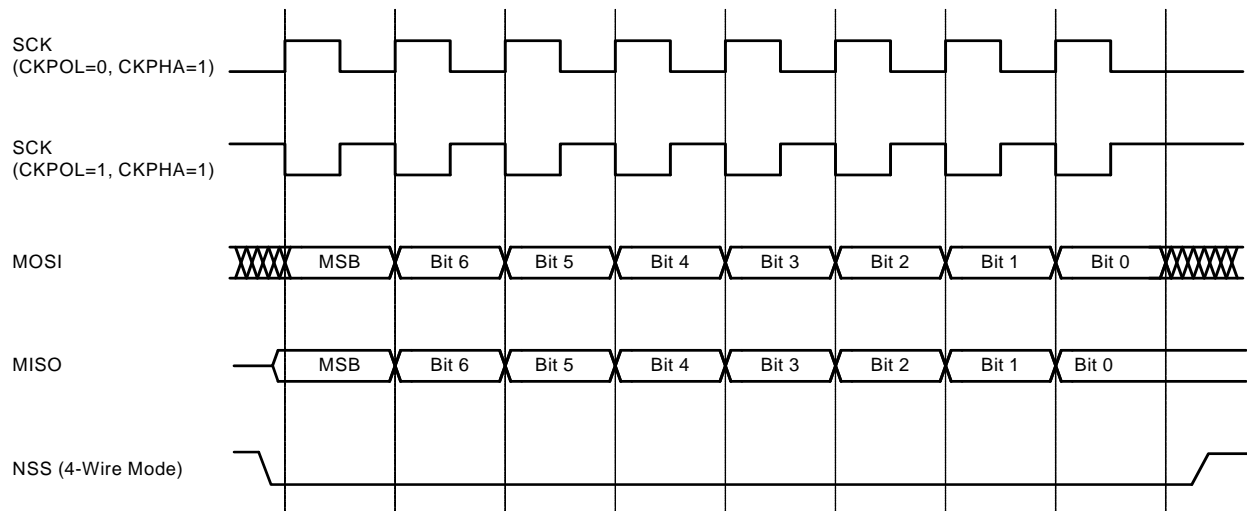
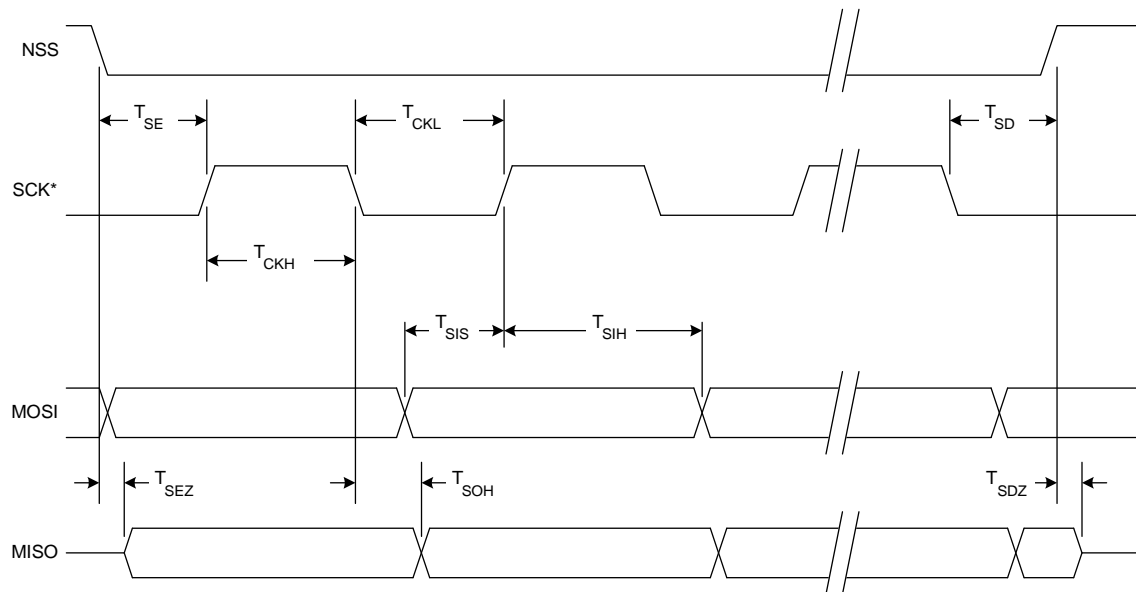
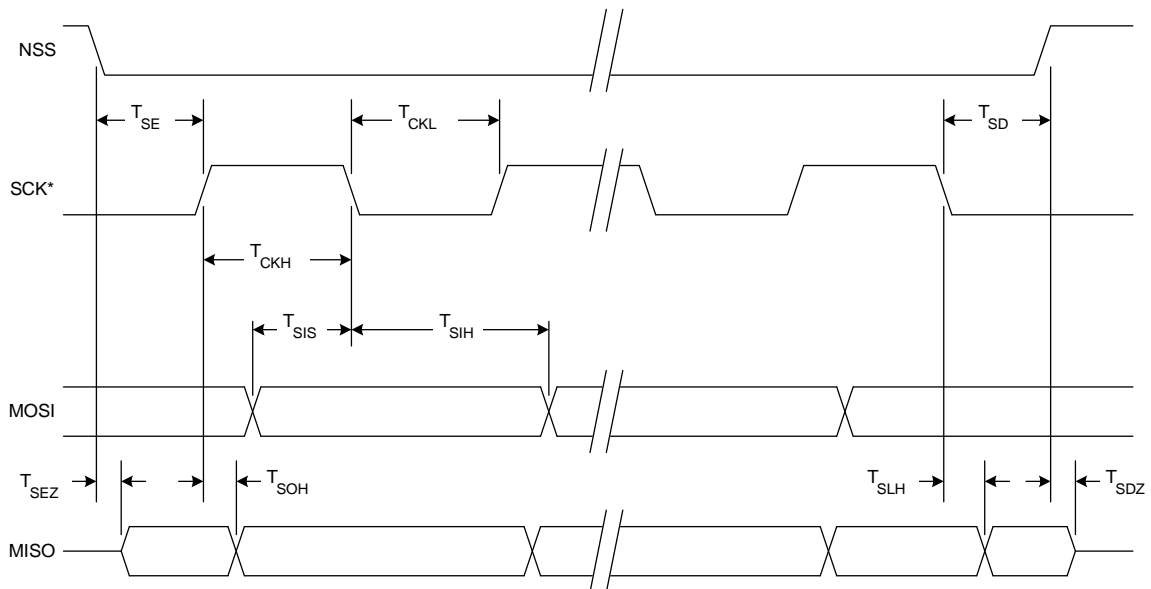


Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)



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

Figure 20.10. SPI Slave Timing (CKPHA = 0)



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

Figure 20.11. SPI Slave Timing (CKPHA = 1)

SFR Definition 22.3. PCA0CPMn: PCA Capture/Compare Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xDA, 0xDB, 0xDC, 0xDD, 0xDE
PCA0CPMn Address: PCA0CPM0 = 0xDA (n = 0), PCA0CPM1 = 0xDB (n = 1), PCA0CPM2 = 0xDC (n = 2), PCA0CPM3 = 0xDD (n = 3), PCA0CPM4 = 0xDE (n = 4)								
Bit7:	PWM16n: 16-bit Pulse Width Modulation Enable. This bit selects 16-bit mode when Pulse Width Modulation mode is enabled (PWMn = 1). 0: 8-bit PWM selected. 1: 16-bit PWM selected.							
Bit6:	ECOMn: Comparator Function Enable. This bit enables/disables the comparator function for PCA module n. 0: Disabled. 1: Enabled.							
Bit5:	CAPPn: Capture Positive Function Enable. This bit enables/disables the positive edge capture for PCA module n. 0: Disabled. 1: Enabled.							
Bit4:	CAPNn: Capture Negative Function Enable. This bit enables/disables the negative edge capture for PCA module n. 0: Disabled. 1: Enabled.							
Bit3:	MATn: Match Function Enable. This bit enables/disables the match function for PCA module n. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1. 0: Disabled. 1: Enabled.							
Bit2:	TOGn: Toggle Function Enable. This bit enables/disables the toggle function for PCA module n. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode. 0: Disabled. 1: Enabled.							
Bit1:	PWMn: Pulse Width Modulation Mode Enable. This bit enables/disables the PWM function for PCA module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode. 0: Disabled. 1: Enabled.							
Bit0:	ECCFn: Capture/Compare Flag Interrupt Enable. This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.							