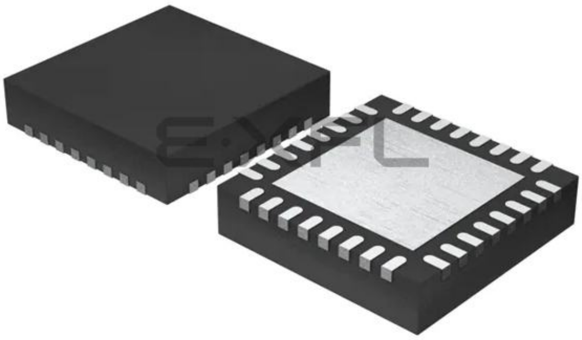


Welcome to E-XFL.COM

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	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K 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/c8051f34b-gmr

5.1. Analog Multiplexer

AMUX0 selects the positive and negative inputs to the ADC. The positive input (AIN+) can be connected to individual Port pins, the on-chip temperature sensor, or the positive power supply (V_{DD}). The negative input (AIN-) can be connected to individual Port pins, VREF, or GND. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; at all other times, ADC0 operates in Differential Mode.** The ADC0 input channels are selected in the AMX0P and AMX0N registers as described in SFR Definition 5.1 and SFR Definition 5.2.

The conversion code format differs between Single-ended and Differential modes. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.0). When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from ‘0’ to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to ‘0’.

Input Voltage (Single-Ended)	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When in Differential Mode, conversion codes are represented as 10-bit signed 2’s complement numbers. Inputs are measured from –VREF to VREF x 511/512. Example codes are shown below for both right-justified and left-justified data. For right-justified data, the unused MSBs of ADC0H are a sign-extension of the data word. For left-justified data, the unused LSBs in the ADC0L register are set to ‘0’.

Input Voltage (Differential)	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 511/512	0x01FF	0x7FC0
VREF x 256/512	0x0100	0x4000
0	0x0000	0x0000
–VREF x 256/512	0xFF00	0xC000
–VREF	0xFE00	0x8000

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to ‘0’ the corresponding bit in register PnMDIN (for n = 0,1,2,3). To force the Crossbar to skip a Port pin, set to ‘1’ the corresponding bit in register PnSKIP (for n = 0,1,2). See **Section “15. Port Input/ Output” on page 142** for more Port I/O configuration details.

5.3. Modes of Operation

ADC0 has a maximum conversion speed of 200 ksps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register (system clock divided by (AD0SC + 1) for $0 \leq AD0SC \leq 31$).

5.3.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a ‘1’ to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 1 overflow
5. A rising edge on the CNVSTR input signal
6. A Timer 3 overflow

Writing a ‘1’ to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. Note that when Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See **Section “21. Timers” on page 235** for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as a Port pin. When the CNVSTR input is used as the ADC0 conversion source, the associated Port pin should be skipped by the Digital Crossbar. To configure the Crossbar to skip a pin, set the corresponding bit in the PnSKIP register to ‘1’. See **Section “15. Port Input/Output” on page 142** for details on Port I/O configuration.

7. Comparators

C8051F34x devices include two on-chip programmable voltage Comparators. A block diagram of the comparators is shown in Figure 7.1, where “n” is the comparator number (0 or 1). The two Comparators operate identically with the following exceptions: (1) Their input selections differ, and (2) Comparator0 can be used as a reset source. For input selection details, refer to SFR Definition 7.2 and SFR Definition 7.5.

Each Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous signal is available even when the system clock is not active. This allows the Comparators to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator outputs may be configured as open drain or push-pull (see **Section “15.2. Port I/O Initialization” on page 147**). Comparator0 may also be used as a reset source (see **Section “11.5. Comparator0 Reset” on page 103**).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 7.5). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 negative input.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section “15.3. General Purpose Port I/O” on page 150**).

SFR Definition 7.6. CPT1MD: Comparator1 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9C

Bits7–6: UNUSED. Read = 00b, Write = don’t care.
Bit5: CP1RIE: Comparator1 Rising-Edge Interrupt Enable.
0: Comparator1 rising-edge interrupt disabled.
1: Comparator1 rising-edge interrupt enabled.
Bit4: CP1FIE: Comparator1 Falling-Edge Interrupt Enable.
0: Comparator1 falling-edge interrupt disabled.
1: Comparator1 falling-edge interrupt enabled.
Bits1–0: CP1MD1–CP1MD0: Comparator1 Mode Select.
These bits select the response time for Comparator1.

Mode	CP1MD1	CP1MD0	CP1 Response Time*
0	0	0	Fastest Response
1	0	1	
2	1	0	
3	1	1	Lowest Power

* See Table 7.1 for response time parameters.

Table 9.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2

Table 9.3. Special Function Registers

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

Register	Address	Description	Page
ACC	0xE0	Accumulator	87
ADC0CF	0xBC	ADC0 Configuration	50
ADC0CN	0xE8	ADC0 Control	51
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	52
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	52
ADC0H	0xBE	ADC0 High	50
ADC0L	0xBD	ADC0 Low	50
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	53
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	53
AMX0N	0xBA	AMUX0 Negative Channel Select	49
AMX0P	0xBB	AMUX0 Positive Channel Select	48
B	0xF0	B Register	88
CKCON	0x8E	Clock Control	241
CLKMUL	0xB9	Clock Multiplier	138
CLKSEL	0xA9	Clock Select	140
CPT0CN	0x9B	Comparator0 Control	62
CPT0MD	0x9D	Comparator0 Mode Selection	64
CPT0MX	0x9F	Comparator0 MUX Selection	63
CPT1CN	0x9A	Comparator1 Control	65
CPT1MD	0x9C	Comparator1 Mode Selection	67
CPT1MX	0x9E	Comparator1 MUX Selection	66
DPH	0x83	Data Pointer High	86
DPL	0x82	Data Pointer Low	86
EIE1	0xE6	Extended Interrupt Enable 1	93
EIE2	0xE7	Extended Interrupt Enable 2	95
EIP1	0xF6	Extended Interrupt Priority 1	94
EIP2	0xF7	Extended Interrupt Priority 2	95
EMI0CN	0xAA	External Memory Interface Control	117
EMI0CF	0x85	External Memory Interface Configuration	118
EMI0TC	0x84	External Memory Interface Timing	123
FLKEY	0xB7	Flash Lock and Key	112
FLSCL	0xB6	Flash Scale	113
IE	0xA8	Interrupt Enable	91
IP	0xB8	Interrupt Priority	92
IT01CF	0xE4	INT0/INT1 Configuration	96
OSCICL	0xB3	Internal Oscillator Calibration	133
OSCICN	0xB2	Internal Oscillator Control	132
OSCLCN	0x86	Internal Low-Frequency Oscillator Control	134
OSCXCN	0xB1	External Oscillator Control	137
P0	0x80	Port 0 Latch	150
P0MDIN	0xF1	Port 0 Input Mode Configuration	150
P0MDOUT	0xA4	Port 0 Output Mode Configuration	151
P0SKIP	0xD4	Port 0 Skip	151
P1	0x90	Port 1 Latch	152

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 $\overline{RST}$ pin.
1: Source of last reset was $\overline{RST}$ 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.

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 (EMI0CF.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.

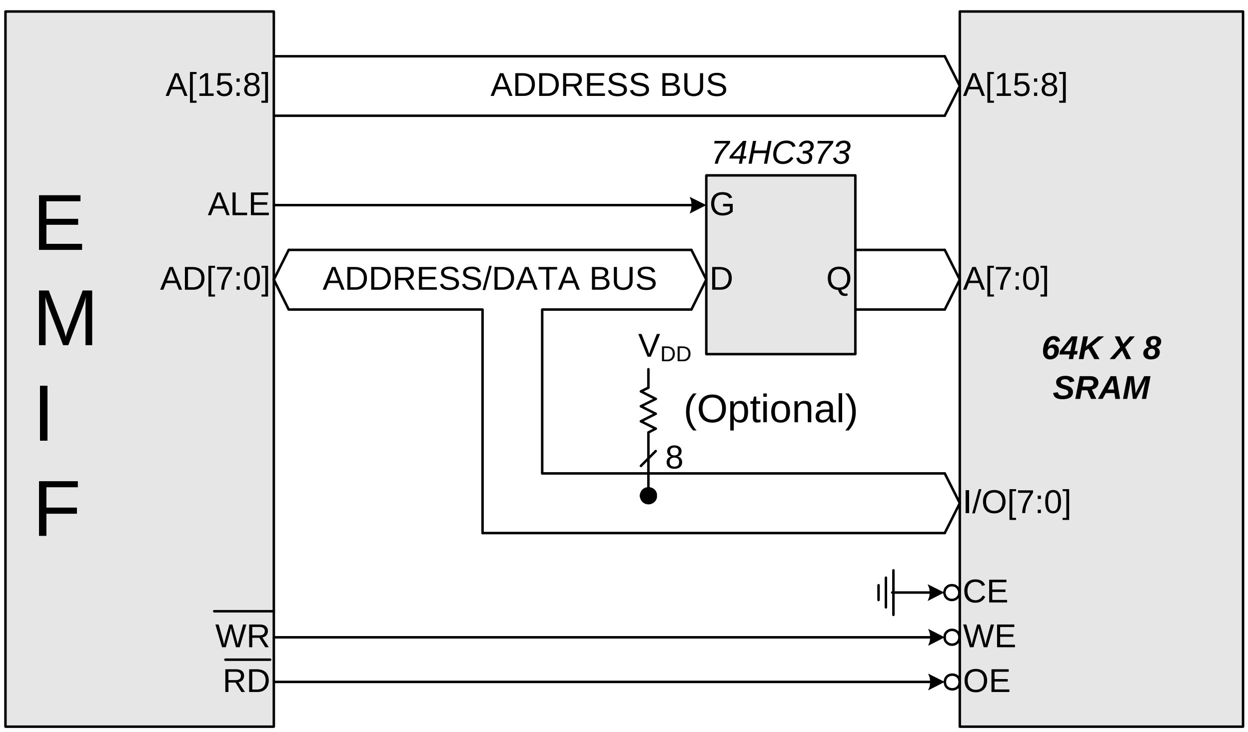


Figure 13.2. Multiplexed Configuration Example

14.1. Programmable Internal High-Frequency (H-F) Oscillator

All C8051F34x devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be programmed via the OSCICL register shown in SFR Definition 14.2. The OSCICL register is factory calibrated to obtain a 12 MHz internal oscillator frequency. Electrical specifications for the precision internal oscillator are given in Table 14.1 on page 141. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset.

14.1.1. Internal H-F Oscillator Suspend Mode

The internal high-frequency oscillator may be placed in Suspend mode by writing ‘1’ to the SUSPEND bit in register OSCICN. In Suspend mode, the internal H-F oscillator is stopped until a non-idle USB event is detected (**Section 16**) or VBUS matches the polarity selected by the VBPOL bit in register REG0CN (**Section 8.2**). Note that the USB transceiver can still detect USB events when it is disabled.

SFR Definition 14.1. OSCICN: Internal H-F Oscillator Control

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	SUSPEND	-	-	-	IFCN1	IFCN0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bit7: IOSCEN: Internal H-F Oscillator Enable Bit.
0: Internal H-F Oscillator Disabled.
1: Internal H-F Oscillator Enabled.

Bit6: IFRDY: Internal H-F Oscillator Frequency Ready Flag.
0: Internal H-F Oscillator is not running at programmed frequency.
1: Internal H-F Oscillator is running at programmed frequency.

Bit5: SUSPEND: Force Suspend
Writing a ‘1’ to this bit will force the internal H-F oscillator to be stopped. The oscillator will be re-started on the next non-idle USB event (i.e., RESUME signaling) or VBUS interrupt event (see SFR Definition 8.1).

Bits4–2: UNUSED. Read = 000b, Write = don't care.

Bits1–0: IFCN1–0: Internal H-F Oscillator Frequency Control.
00: SYSCLK derived from Internal H-F Oscillator divided by 8.
01: SYSCLK derived from Internal H-F Oscillator divided by 4.
10: SYSCLK derived from Internal H-F Oscillator divided by 2.
11: SYSCLK derived from Internal H-F Oscillator divided by 1.

14.4. 4x Clock Multiplier

The 4x Clock Multiplier allows a 12 MHz oscillator to generate the 48 MHz clock required for Full Speed USB communication (see **Section “16.4. USB Clock Configuration” on page 166**). A divided version of the Multiplier output can also be used as the system clock. C8051F340/1/2/3 devices can use the 48 MHz Clock Multiplier output as system clock. See Table 3.1, “Global DC Electrical Characteristics,” on page 25 for system clock frequency specifications. See **Section 14.5** for details on system clock and USB clock source selection.

The 4x Clock Multiplier is configured via the CLKMUL register. The procedure for configuring and enabling the 4x Clock Multiplier is as follows:

1. Reset the Multiplier by writing 0x00 to register CLKMUL.
2. Select the Multiplier input source via the MULSEL bits.
3. Enable the Multiplier with the MULEN bit (CLKMUL | = 0x80).
4. Delay for >5 µs.
5. Initialize the Multiplier with the MULINIT bit (CLKMUL | = 0xC0).
6. Poll for MULRDY => ‘1’.

Important Note: When using an external oscillator as the input to the 4x Clock Multiplier, the external source must be enabled and stable before the Multiplier is initialized. See Section 14.5 for details on selecting an external oscillator source.

SFR Definition 14.5. CLKMUL: Clock Multiplier Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
MULEN	MULINIT	MULRDY	-	-	-	MULSEL		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address 0xB9

Bit7: MULEN: Clock Multiplier Enable
0: Clock Multiplier disabled.
1: Clock Multiplier enabled.

Bit6: MULINIT: Clock Multiplier Initialize
This bit should be a ‘0’ when the Clock Multiplier is enabled. Once enabled, writing a ‘1’ to this bit will initialize the Clock Multiplier. The MULRDY bit reads ‘1’ when the Clock Multiplier is stabilized.

Bit5: MULRDY: Clock Multiplier Ready
This read-only bit indicates the status of the Clock Multiplier.
0: Clock Multiplier not ready.
1: Clock Multiplier ready (locked).

Bits4–2: Unused. Read = 000b; Write = don’t care.

Bits1–0: MULSEL: Clock Multiplier Input Select
These bits select the clock supplied to the Clock Multiplier.

MULSEL	Selected Clock
00	Internal Oscillator
01	External Oscillator
10	External Oscillator / 2
11	RESERVED

15.1. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 15.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to the VREF signal, external oscillator pins (XTAL1, XTAL2), the ADC's external conversion start signal (CNVSTR), EMIF control signals, and any selected ADC or Comparator inputs. The PnSKIP registers may also be used to skip pins to be used as GPIO. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 15.3 shows all the possible pins available to each peripheral. Figure 15.4 shows the Crossbar Decoder priority with no Port pins skipped. Figure 15.5 shows a Crossbar example with pins P0.2, P0.3, and P1.0 skipped.

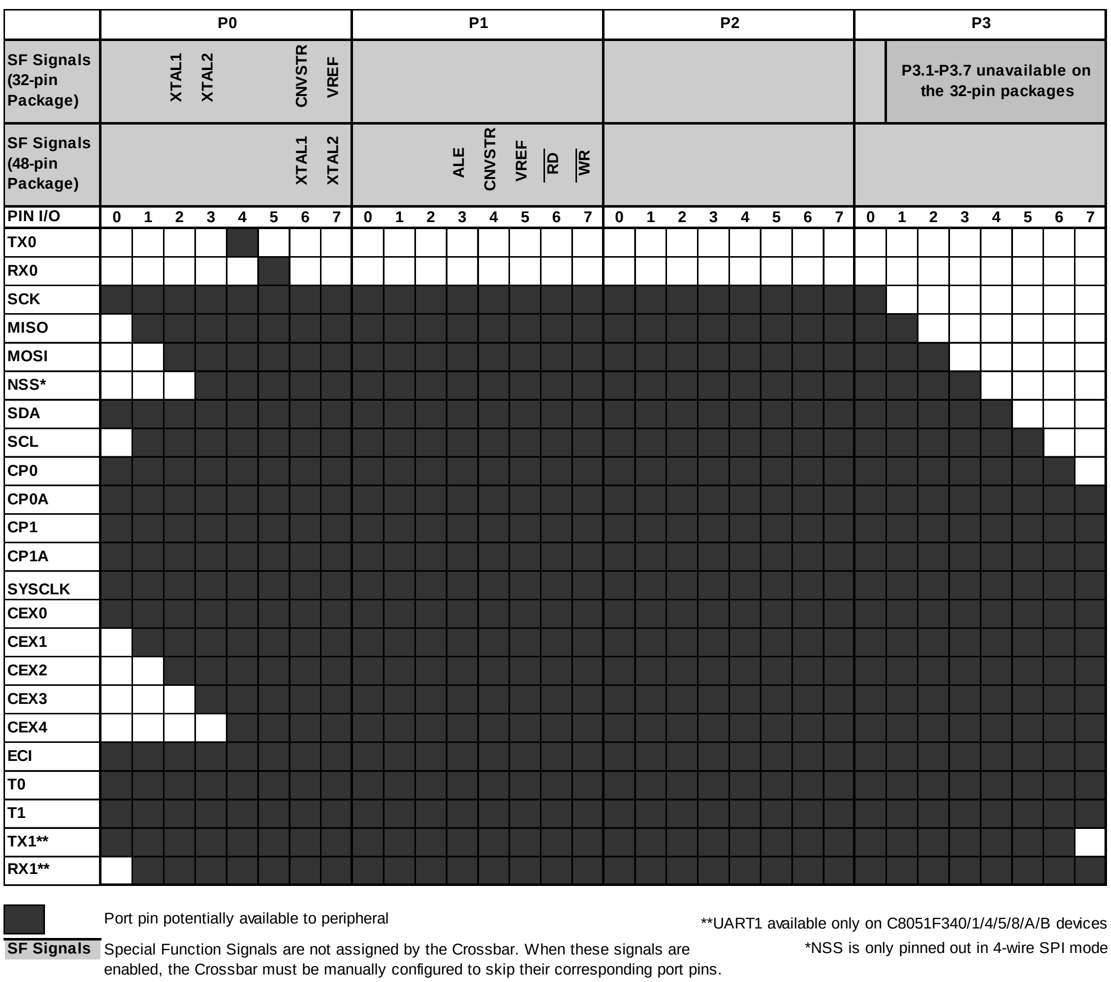


Figure 15.3. Peripheral Availability on Port I/O Pins

SFR Definition 15.1. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1

Bit7: CP1AE: Comparator1 Asynchronous Output Enable
0: Asynchronous CP1 unavailable at Port pin.
1: Asynchronous CP1 routed to Port pin.

Bit6: CP1E: Comparator1 Output Enable
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

Bit5: CP0AE: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.

Bit4: CP0E: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit3: SYSCKE: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK output routed to Port pin.

Bit2: SMB0E: SMBus I/O Enable
0: SMBus I/O unavailable at Port pins.
1: SMBus I/O routed to Port pins.

Bit1: SPI0E: SPI I/O Enable
0: SPI I/O unavailable at Port pins.
1: SPI I/O routed to Port pins.

Bit0: URT0E: UART0 I/O Output Enable
0: UART0 I/O unavailable at Port pins.
1: UART0 TX0, RX0 routed to Port pins P0.4 and P0.5.

SFR Definition 16.3. USB0DAT: USB0 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
USB0DAT								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x97

This SFR is used to indirectly read and write USB0 registers.

Write Procedure:
1. Poll for BUSY (USB 0ADR.7) => '0'.
2. Load the target USB0 register address into the USBADDR bits in register USB0ADR.
3. Write data to USB0DAT.
4. Repeat (Step 2 may be skipped when writing to the same USB0 register).

Read Procedure:
1. Poll for BUSY (USB 0ADR.7) => '0'.
2. Load the target USB0 register address into the USBADDR bits in register USB0ADR.
3. Write '1' to the BUSY bit in register USB0ADR (steps 2 and 3 can be performed in the same write).
4. Poll for BUSY (USB 0ADR.7) => '0'.
5. Read data from USB0DAT.
6. Repeat from Step 2 (Step 2 may be skipped when reading the same USB0 register; Step 3 may be skipped when the AUTORD bit (USB0ADR.6) is logic 1).

19.3. Configuration and Operation

UART1 provides standard asynchronous, full duplex communication. It can operate in a point-to-point serial communications application, or as a node on a multi-processor serial interface. To operate in a point-to-point application, where there are only two devices on the serial bus, the MCE1 bit in SMOD1 should be cleared to ‘0’. For operation as part of a multi-processor communications bus, the MCE1 and XBE1 bits should both be set to ‘1’. In both types of applications, data is transmitted from the microcontroller on the TX1 pin, and received on the RX1 pin. The TX1 and RX1 pins are configured using the crossbar and the Port I/O registers, as detailed in **Section “15. Port Input/Output” on page 142**.

In typical UART communications, The transmit (TX) output of one device is connected to the receive (RX) input of the other device, either directly or through a bus transceiver, as shown in Figure 19.5.

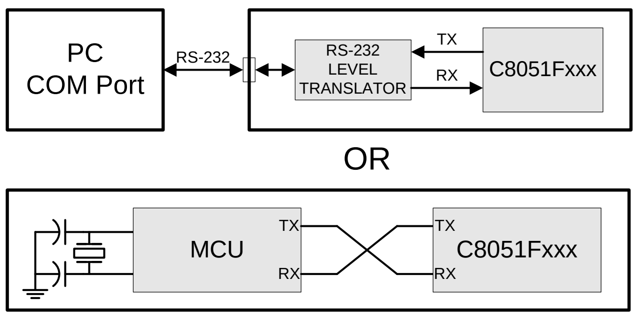


Figure 19.5. Typical UART Interconnect Diagram

19.3.1. Data Transmission

Data transmission is double-buffered, and begins when software writes a data byte to the SBUF1 register. Writing to SBUF1 places data in the Transmit Holding Register, and the Transmit Holding Register Empty flag (THRE1) will be cleared to ‘0’. If the UARTs shift register is empty (i.e., no transmission is in progress) the data will be placed in the shift register, and the THRE1 bit will be set to ‘1’. If a transmission is in progress, the data will remain in the Transmit Holding Register until the current transmission is complete. The TI1 Transmit Interrupt Flag (SCON1.1) will be set at the end of any transmission (the beginning of the stop-bit time). If enabled, an interrupt will occur when TI1 is set.

If the extra bit function is enabled (XBE1 = ‘1’) and the parity function is disabled (PE1 = ‘0’), the value of the TBX1 (SCON1.3) bit will be sent in the extra bit position. When the parity function is enabled (PE1 = ‘1’), hardware will generate the parity bit according to the selected parity type (selected with S1PT[1:0]), and append it to the data field. Note: when parity is enabled, the extra bit function is not available.

19.3.2. Data Reception

Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to logic 1. After the stop bit is received, the data byte will be stored in the receive FIFO if the following conditions are met: the receive FIFO (3 bytes deep) must not be full, and the stop bit(s) must be logic 1. In the event that the receive FIFO is full, the incoming byte will be lost, and a Receive FIFO Overrun Error will be generated (OVR1 in register SCON1 will be set to logic 1). If the stop bit(s) were logic 0, the incoming data will not be stored in the receive FIFO. If the reception conditions are met, the data is stored in the receive FIFO, and the RI1 flag will be set. Note: when MCE1 = ‘1’, RI1 will only be set if the extra bit was equal to ‘1’. Data can be read from the receive FIFO by reading the SBUF1 register. The SBUF1 register represents the oldest byte in the FIFO. After SBUF1 is read, the next byte in the FIFO is immediately loaded into SBUF1, and 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).

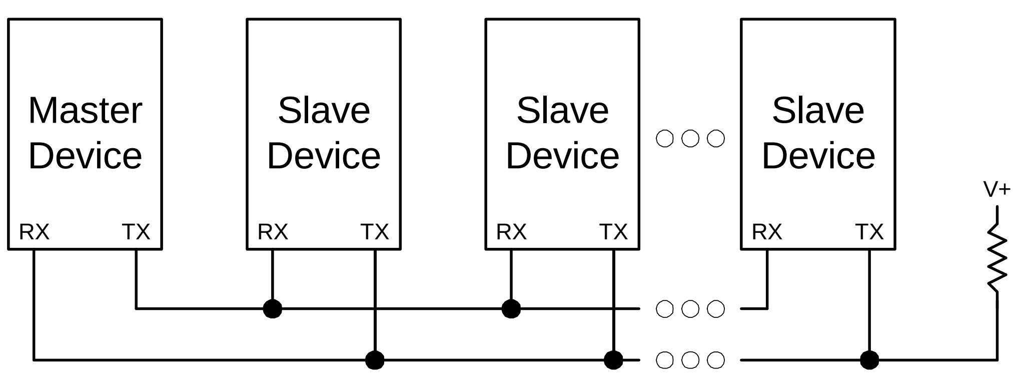


Figure 19.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 19.1. SCON1: UART1 Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
OVR1	PERR1	THRE1	REN1	TBX1	RBX1	TI1	RI1	00100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD2

Bit7: OVR1: Receive FIFO Overrun Flag.
This bit is used to indicate a receive FIFO overrun condition.
0: Receive FIFO Overrun has not occurred.
1: Receive FIFO Overrun has occurred (an incoming character was discarded due to a full FIFO).
This bit must be cleared to ‘0’ by software.

Bit6: PERR1: Parity Error Flag.
When parity is enabled, this bit is used to indicate that a parity error has occurred. It is set to ‘1’ when the parity of the oldest byte in the FIFO does not match the selected Parity Type.
0: Parity Error has not occurred.
1: Parity Error has occurred.
This bit must be cleared to ‘0’ by software.

Bit5: THRE1: Transmit Holding Register Empty Flag.
0: Transmit Holding Register not Empty - do not write to SBUF1.
1: Transmit Holding Register Empty - it is safe to write to SBUF1.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO.
0: UART1 reception disabled.
1: UART1 reception enabled.

Bit3: TBX1: Extra Transmission Bit.
The logic level of this bit will be assigned to the extra transmission bit when XBE1 is set to ‘1’. This bit is not used when Parity is enabled.

Bit2: RBX1: Extra Receive Bit.
RBX1 is assigned the value of the extra bit when XBE1 is set to ‘1’. If XBE1 is cleared to ‘0’, RBX1 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled.

Bit1: TI1: Transmit Interrupt Flag.
Set to a ‘1’ by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART1 interrupt is enabled, setting this bit causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI1: Receive Interrupt Flag.
Set to ‘1’ by hardware when a byte of data has been received by UART1 (set at the STOP bit sampling time). When the UART1 interrupt is enabled, setting this bit to ‘1’ causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software. Note that RI1 will remain set to '1' as long as there is still data in the UART FIFO. After the last byte has been shifted from the FIFO to SBUF1, RI1 can be cleared.

SFR Definition 19.2. SMOD1: UART1 Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MCE1	S1PT1	S1PT0	PE1	S1DL1	S1DL0	XBE1	SBL1	00001100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE5

Bit7: MCE1: Multiprocessor Communication Enable.
0: RI will be activated if stop bit(s) are ‘1’.
1: RI will be activated if stop bit(s) and extra bit are ‘1’ (extra bit must be enabled using XBE1).
Note: This function is not available when hardware parity is enabled.

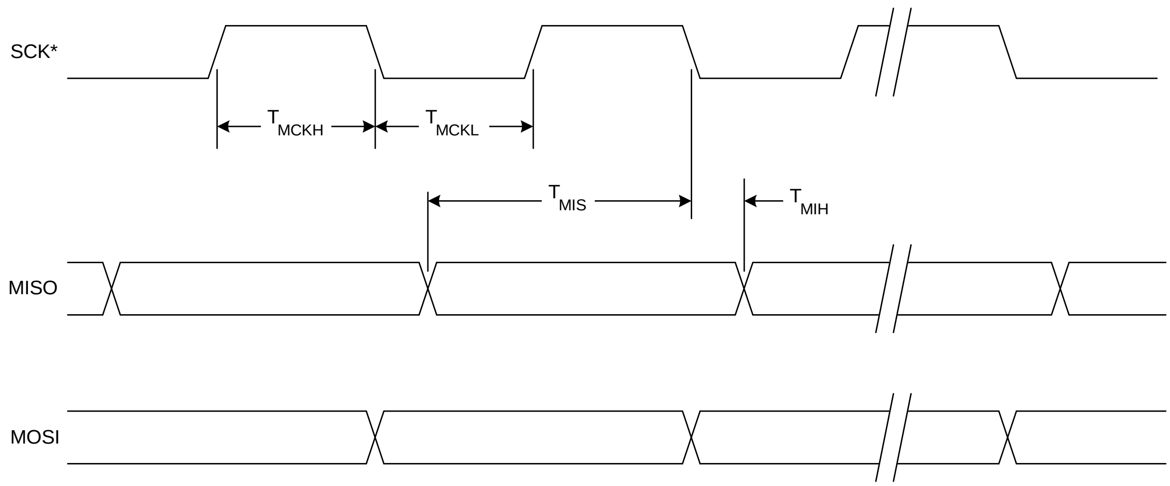
Bits6–5: S1PT[1:0]: Parity Type.
00: Odd
01: Even
10: Mark
11: Space

Bit4: PE1: Parity Enable.
This bit activates hardware parity generation and checking. The parity type is selected by bits S1PT1-0 when parity is enabled.
0: Hardware parity is disabled.
1: Hardware parity is enabled.

Bits3–2: S1DL[1:0]: Data Length.
00: 5-bit data
01: 6-bit data
10: 7-bit data
11: 8-bit data

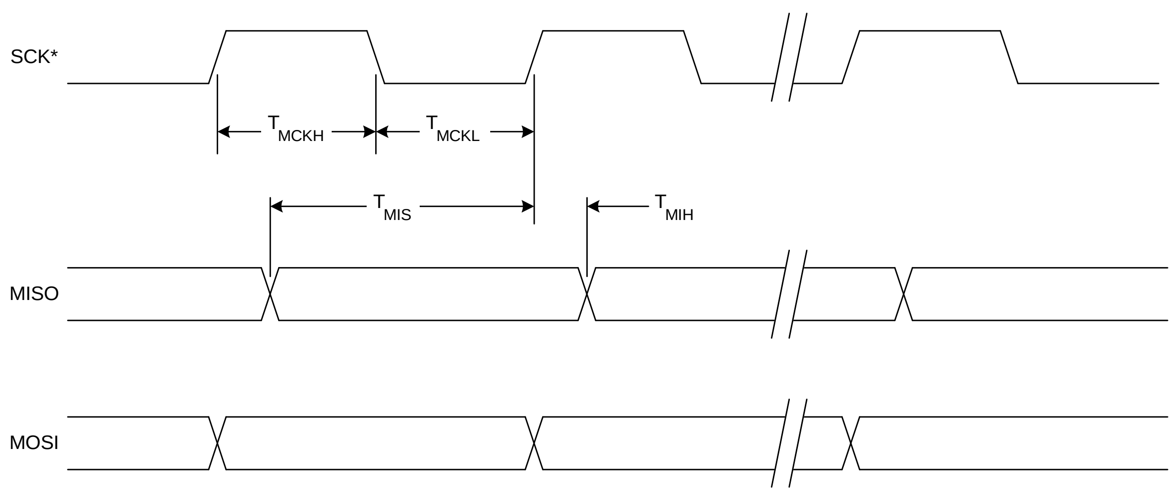
Bit1: XBE1: Extra Bit Enable
When enabled, the value of TBX1 will be appended to the data field.
0: Extra Bit Disabled.
1: Extra Bit Enabled.

Bit0: SBL1: Stop Bit Length
0: Short - Stop bit is active for one bit time.
1: Long - Stop bit is active for two bit times (data length = 6, 7, or 8 bits), or 1.5 bit times (data length = 5 bits).



* 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)

23.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming functions may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK ($\overline{\text{RST}}$) and C2D (P3.0) pins. Note that the C2D pin is shared on the 32-pin packages only (C8051F342/3/6/7/9/A/B). In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 23.1.

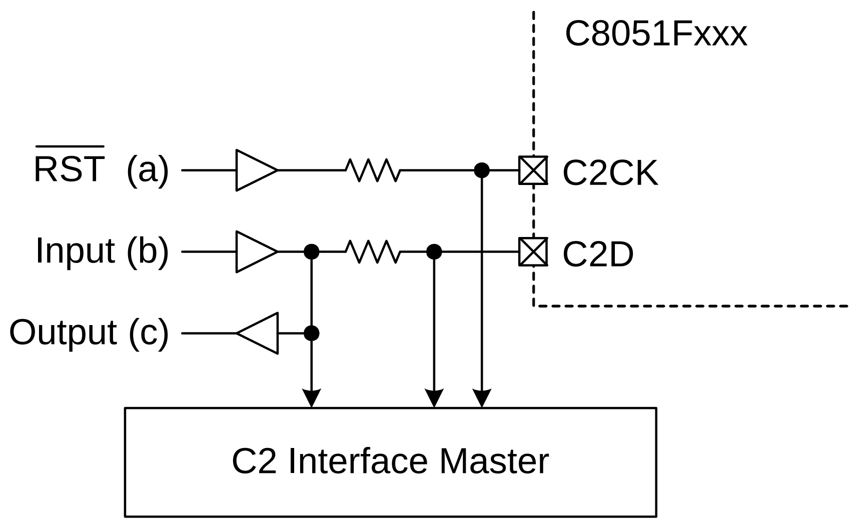


Figure 23.1. Typical C2 Pin Sharing

The configuration in Figure 23.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The $\overline{\text{RST}}$ pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.