

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	48 MIPS
Connectivity	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 ~ 5.25V
Data Converters	A/D 21x10b
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/c8051f381-gq

C8051F380/1/2/3/4/5/6/7/C

SFR Definition 19.2. OSCICL: Internal H-F Oscillator Calibration	145
SFR Definition 19.3. OSCICN: Internal H-F Oscillator Control	146
SFR Definition 19.4. CLKMUL: Clock Multiplier Control	147
SFR Definition 19.5. OSCLCN: Internal L-F Oscillator Control	148
SFR Definition 19.6. OSCXCN: External Oscillator Control	152
SFR Definition 20.1. XBR0: Port I/O Crossbar Register 0	159
SFR Definition 20.2. XBR1: Port I/O Crossbar Register 1	160
SFR Definition 20.3. XBR2: Port I/O Crossbar Register 2	161
SFR Definition 20.4. P0: Port 0	162
SFR Definition 20.5. P0MDIN: Port 0 Input Mode	162
SFR Definition 20.6. P0MDOUT: Port 0 Output Mode	163
SFR Definition 20.7. P0SKIP: Port 0 Skip	163
SFR Definition 20.8. P1: Port 1	164
SFR Definition 20.9. P1MDIN: Port 1 Input Mode	164
SFR Definition 20.10. P1MDOUT: Port 1 Output Mode	165
SFR Definition 20.11. P1SKIP: Port 1 Skip	165
SFR Definition 20.12. P2: Port 2	166
SFR Definition 20.13. P2MDIN: Port 2 Input Mode	166
SFR Definition 20.14. P2MDOUT: Port 2 Output Mode	167
SFR Definition 20.15. P2SKIP: Port 2 Skip	167
SFR Definition 20.16. P3: Port 3	168
SFR Definition 20.17. P3MDIN: Port 3 Input Mode	168
SFR Definition 20.18. P3MDOUT: Port 3 Output Mode	169
SFR Definition 20.19. P3SKIP: Port 3 Skip	169
SFR Definition 20.20. P4: Port 4	170
SFR Definition 20.21. P4MDIN: Port 4 Input Mode	170
SFR Definition 20.22. P4MDOUT: Port 4 Output Mode	171
SFR Definition 21.1. USB0XCN: USB0 Transceiver Control	174
SFR Definition 21.2. USB0ADR: USB0 Indirect Address	176
SFR Definition 21.3. USB0DAT: USB0 Data	177
USB Register Definition 21.4. INDEX: USB0 Endpoint Index	179
USB Register Definition 21.5. CLKREC: Clock Recovery Control	180
USB Register Definition 21.6. FIFOn: USB0 Endpoint FIFO Access	182
USB Register Definition 21.7. FADDR: USB0 Function Address	183
USB Register Definition 21.8. POWER: USB0 Power	185
USB Register Definition 21.9. FRAMEL: USB0 Frame Number Low	186
USB Register Definition 21.10. FRAMEH: USB0 Frame Number High	186
USB Register Definition 21.11. IN1INT: USB0 IN Endpoint Interrupt	187
USB Register Definition 21.12. OUT1INT: USB0 OUT Endpoint Interrupt	188
USB Register Definition 21.13. CMINT: USB0 Common Interrupt	189
USB Register Definition 21.14. IN1IE: USB0 IN Endpoint Interrupt Enable	190
USB Register Definition 21.15. OUT1IE: USB0 OUT Endpoint Interrupt Enable	191
USB Register Definition 21.16. CMIE: USB0 Common Interrupt Enable	192
USB Register Definition 21.17. E0CSR: USB0 Endpoint0 Control	195
USB Register Definition 21.18. E0CNT: USB0 Endpoint0 Data Count	196

Table 5.14. USB Transceiver Electrical Characteristics

$V_{DD} = 3.0 \text{ V to } 3.6 \text{ V}$, $-40 \text{ to } +85 \text{ }^{\circ}\text{C}$ unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Transmitter					
Output High Voltage (V_{OH})		2.8	—	—	V
Output Low Voltage (V_{OL})		—	—	0.8	V
VBUS Detection Input Low Voltage		—	—	1.0	V
VBUS Detection Input High Voltage		3.0	—	—	V
Output Crossover Point (V_{CRS})		1.3	—	2.0	V
Output Impedance (Z_{DRV})	Driving High	—	38	—	W
	Driving Low	—	38	—	W
Pull-up Resistance (R_{PU})	Full Speed (D+ Pull-up) Low Speed (D– Pull-up)	1.425	1.5	1.575	k Ω
Output Rise Time (T_R)	Low Speed	75	—	300	ns
	Full Speed	4	—	20	ns
Output Fall Time (T_F)	Low Speed	75	—	300	ns
	Full Speed	4	—	20	ns
Receiver					
Differential Input Sensitivity (V_{DI})	(D+) – (D–)	0.2	—	—	V
Differential Input Common Mode Range (V_{CM})		0.8	—	2.5	V
Input Leakage Current (I_L)	Pullups Disabled	—	<1.0	—	μA
Note: Refer to the USB Specification for timing diagrams and symbol definitions.					

6.4.1. Window Detector Example

Figure 6.6 shows two example window comparisons for right-justified, single-ended data, with $ADC0LTH:ADC0LTL = 0x0080$ (128d) and $ADC0GTH:ADC0GTL = 0x0040$ (64d). The input voltage can range from 0 to $VREF \times (1023/1024)$ with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an $AD0WINT$ interrupt will be generated if the $ADC0$ conversion word ($ADC0H:ADC0L$) is within the range defined by $ADC0GTH:ADC0GTL$ and $ADC0LTH:ADC0LTL$ (if $0x0040 < ADC0H:ADC0L < 0x0080$). In the right example, an $AD0WINT$ interrupt will be generated if the $ADC0$ conversion word is outside of the range defined by the $ADC0GT$ and $ADC0LT$ registers (if $ADC0H:ADC0L < 0x0040$ or $ADC0H:ADC0L > 0x0080$). Figure 6.7 shows an example using left-justified data with the same comparison values.

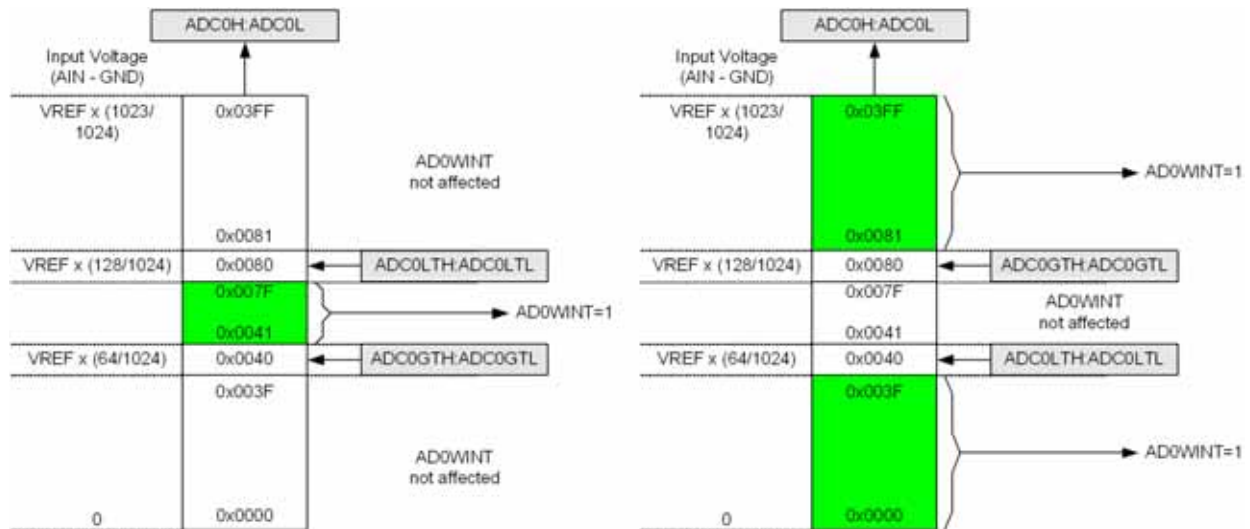


Figure 6.6. ADC Window Compare Example: Right-Justified Data

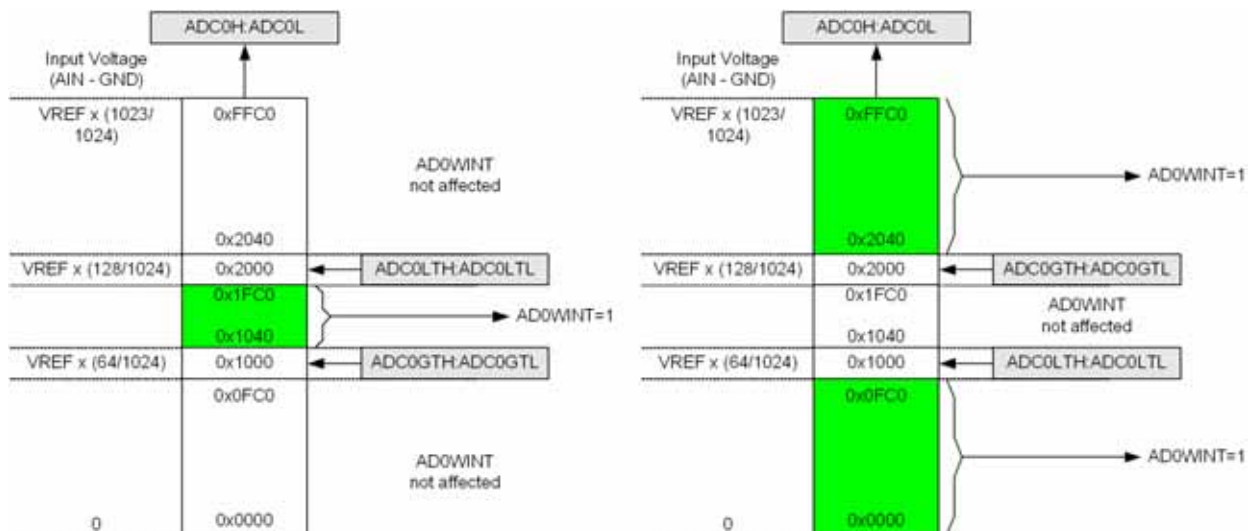


Figure 6.7. ADC Window Compare Example: Left-Justified Data

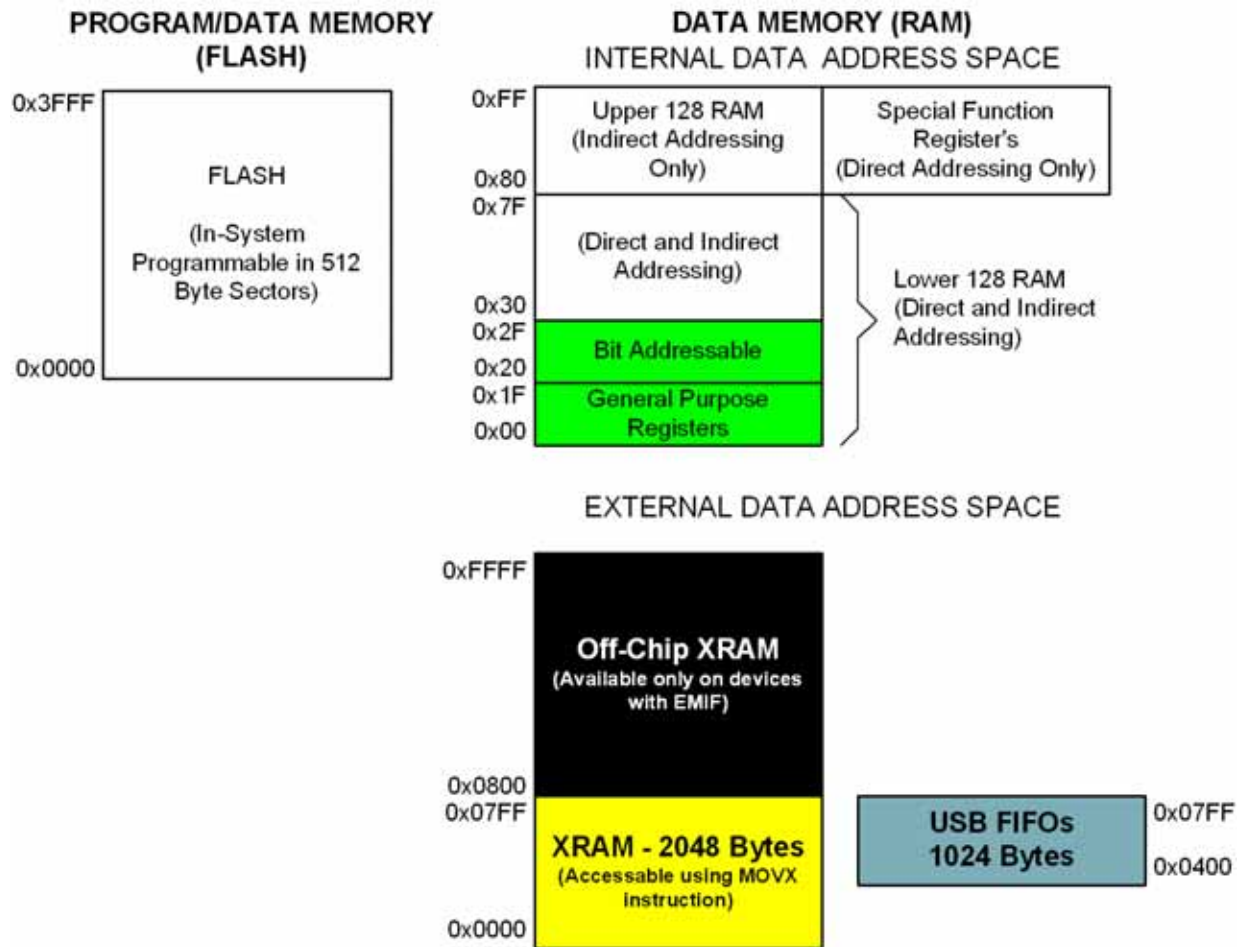


Figure 13.3. On-Chip Memory Map for 16 kB Devices (C8051F38C)

13.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F380/1/2/3/4/5/6/7/C implements 64 kB, 32 kB, or 16 kB of this program memory space as in-system, re-programmable Flash memory. Note that on the C8051F380/1/4/5 (64 kB version), 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 "18. Flash Memory" on page 135 for further details.

13.2. Data Memory

The CIP-51 includes 256 of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

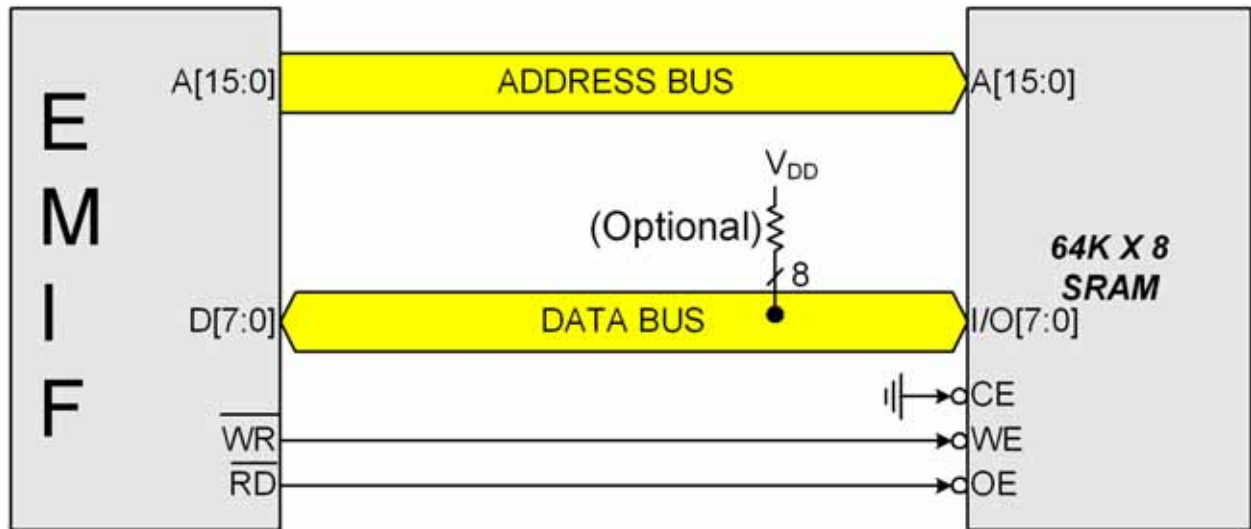


Figure 14.3. Non-multiplexed Configuration Example

C8051F380/1/2/3/4/5/6/7/C

14.7.2.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 001 or 011

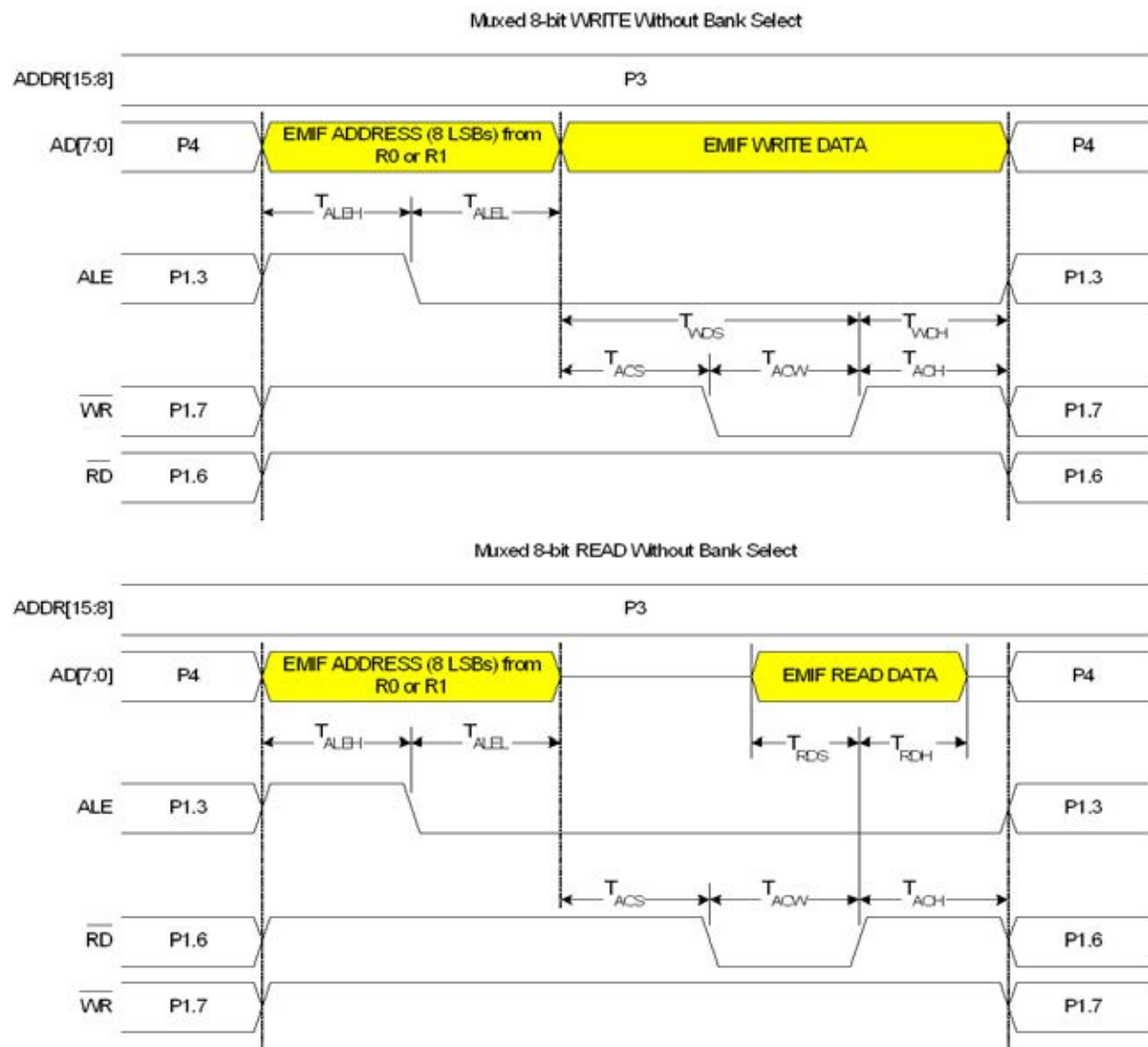


Figure 14.9. Multiplexed 8-bit MOVX without Bank Select Timing

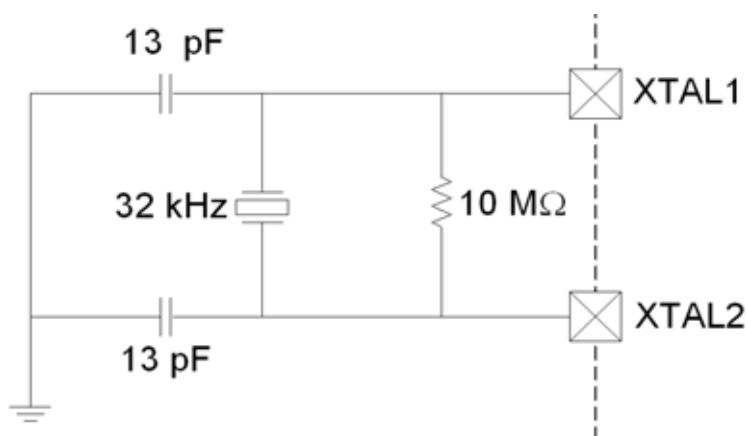


Figure 19.2. External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency (see SFR Definition 19.6).

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock is valid and running. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is:

1. Configure XTAL1 and XTAL2 for analog I/O.
2. Disable the XTAL1 and XTAL2 digital output drivers by writing 1s to the appropriate bits in the Port Latch register.
3. Configure and enable the external oscillator.
4. Wait at least 1 ms.
5. Poll for $XTLVLD \geq 1$.
6. Switch the system clock to the external oscillator.

SFR Definition 21.1. USB0XCN: USB0 Transceiver Control

Bit	7	6	5	4	3	2	1	0
Name	PREN	PHYEN	SPEED	PHYTST[1:0]		DFREC	Dp	Dn
Type	R/W	R/W	R/W	R/W		R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD7; SFR Page = All Pages

Bit	Name	Function
7	PREN	Internal Pull-up Resistor Enable. The location of the pull-up resistor (D+ or D-) is determined by the SPEED bit. 0: Internal pull-up resistor disabled (device effectively detached from USB network). 1: Internal pull-up resistor enabled when VBUS is present (device attached to the USB network).
6	PHYEN	Physical Layer Enable. 0: USB0 physical layer Transceiver disabled (suspend). 1: USB0 physical layer Transceiver enabled (normal).
5	SPEED	USB0 Speed Select. This bit selects the USB0 speed. 0: USB0 operates as a Low Speed device. If enabled, the internal pull-up resistor appears on the D- line. 1: USB0 operates as a Full Speed device. If enabled, the internal pull-up resistor appears on the D+ line.
4:3	PHYTST[1:0]	Physical Layer Test Bits. 00: Mode 0: Normal (non-test mode) (D+ = X, D- = X) 01: Mode 1: Differential 1 Forced (D+ = 1, D- = 0) 10: Mode 2: Differential 0 Forced (D+ = 0, D- = 1) 11: Mode 3: Single-Ended 0 Forced (D+ = 0, D- = 0)
2	DFREC	Differential Receiver Bit The state of this bit indicates the current differential value present on the D+ and D- lines when PHYEN = 1. 0: Differential 0 signalling on the bus. 1: Differential 1 signalling on the bus.
1	Dp	D+ Signal Status. This bit indicates the current logic level of the D+ pin. 0: D+ signal currently at logic 0. 1: D+ signal currently at logic 1.
0	Dn	D- Signal Status. This bit indicates the current logic level of the D- pin. 0: D- signal currently at logic 0. 1: D- signal currently at logic 1.

C8051F380/1/2/3/4/5/6/7/C

USB Register Definition 21.20. EINCSSL: USB0 IN Endpoint Control Low

Bit	7	6	5	4	3	2	1	0
Name		CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY
Type	R	W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x11

Bit	Name	Description	Write	Read
7	Unused	Read = 0b. Write = don't care.		
6	CLRDT	Clear Data Toggle Bit.	Software should write 1 to this bit to reset the IN Endpoint data toggle to 0.	This bit always reads 0.
5	STSTL	Sent Stall Bit. Hardware sets this bit to 1 when a STALL handshake signal is transmitted. The FIFO is flushed, and the INPRDY bit cleared. This flag must be cleared by software.		
4	SDSTL	Send Stall. Software should write 1 to this bit to generate a STALL handshake in response to an IN token. Software should write 0 to this bit to terminate the STALL signal. This bit has no effect in ISO mode.		
3	FLUSH	FIFO Flush Bit. Writing a 1 to this bit flushes the next packet to be transmitted from the IN Endpoint FIFO. The FIFO pointer is reset and the INPRDY bit is cleared. If the FIFO contains multiple packets, software must write 1 to FLUSH for each packet. Hardware resets the FLUSH bit to 0 when the FIFO flush is complete.		
2	UNDRUN	Data Underrun Bit. The function of this bit depends on the IN Endpoint mode: ISO: Set when a zero-length packet is sent after an IN token is received while bit INPRDY = 0. Interrupt/Bulk: Set when a NAK is returned in response to an IN token. This bit must be cleared by software.		
1	FIFONE	FIFO Not Empty. 0: The IN Endpoint FIFO is empty. 1: The IN Endpoint FIFO contains one or more packets.		
0	INPRDY	In Packet Ready. Software should write 1 to this bit after loading a data packet into the IN Endpoint FIFO. Hardware clears INPRDY due to any of the following: 1) A data packet is transmitted. 2) Double buffering is enabled (DBIEN = 1) and there is an open FIFO packet slot. 3) If the endpoint is in Isochronous Mode (ISO = 1) and ISOUD = 1, INPRDY will read 0 until the next SOF is received. Note: An interrupt (if enabled) will be generated when hardware clears INPRDY as a result of a packet being transmitted.		

21.13. Controlling Endpoints1-3 OUT

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

An Endpoint1-3 OUT interrupt may be generated by the following:

1. Hardware sets the OPRDY bit (EINCSRL.0) to 1.
2. Hardware generates a STALL condition.

21.13.1. Endpoints1-3 OUT Interrupt or Bulk Mode

When the ISO bit (EOUTCSRH.6) = 0 the target endpoint operates in Bulk or Interrupt mode. Once an endpoint has been configured to operate in Bulk/Interrupt OUT mode (typically following an Endpoint0 SET_INTERFACE command), hardware will set the OPRDY bit (EOUTCSRL.0) to 1 and generate an interrupt upon reception of an OUT token and data packet. The number of bytes in the current OUT data packet (the packet ready to be unloaded from the FIFO) is given in the EOUTCNTH and EOUTCNTL registers. In response to this interrupt, firmware should unload the data packet from the OUT FIFO and reset the OPRDY bit to 0.

A Bulk or Interrupt pipe can be shut down (or Halted) by writing 1 to the SDSTL bit (EOUTCSRL.5). While SDSTL = 1, hardware will respond to all OUT requests with a STALL condition. Each time hardware generates a STALL condition, an interrupt will be generated and the STSTL bit (EOUTCSRL.6) set to 1. The STSTL bit must be reset to 0 by firmware.

Hardware will automatically set OPRDY when a packet is ready in the OUT FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for two packets to be ready in the OUT FIFO at a time. In this case, hardware will set OPRDY to 1 immediately after firmware unloads the first packet and resets OPRDY to 0. A second interrupt will be generated in this case.

21.13.2. Endpoints1-3 OUT Isochronous Mode

When the ISO bit (EOUTCSRH.6) is set to 1, the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO OUT mode, the host will send exactly one data per USB frame; the location of the data packet within each frame may vary, however. Because of this, it is recommended that double buffering be enabled for ISO OUT endpoints.

Each time a data packet is received, hardware will load the received data packet into the endpoint FIFO, set the OPRDY bit (EOUTCSRL.0) to 1, and generate an interrupt (if enabled). Firmware would typically use this interrupt to unload the data packet from the endpoint FIFO and reset the OPRDY bit to 0.

If a data packet is received when there is no room in the endpoint FIFO, an interrupt will be generated and the OVRUN bit (EOUTCSRL.2) set to 1. If USB0 receives an ISO data packet with a CRC error, the data packet will be loaded into the endpoint FIFO, OPRDY will be set to 1, an interrupt (if enabled) will be generated, and the DATAERR bit (EOUTCSRL.3) will be set to 1. Software should check the DATAERR bit each time a data packet is unloaded from an ISO OUT endpoint FIFO.

USB Register Definition 21.25. EOUTCNTH: USB0 OUT Endpoint Count High

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

USB Register Address = 0x17

Bit	Name	Function
7:2	Unused	Read = 000000b. Write = don't care.
1:0	EOCH[1:0]	OUT Endpoint Count High Byte. EOCH holds the upper 2-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = 1.

Table 22.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (Continued)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
	0010				Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—

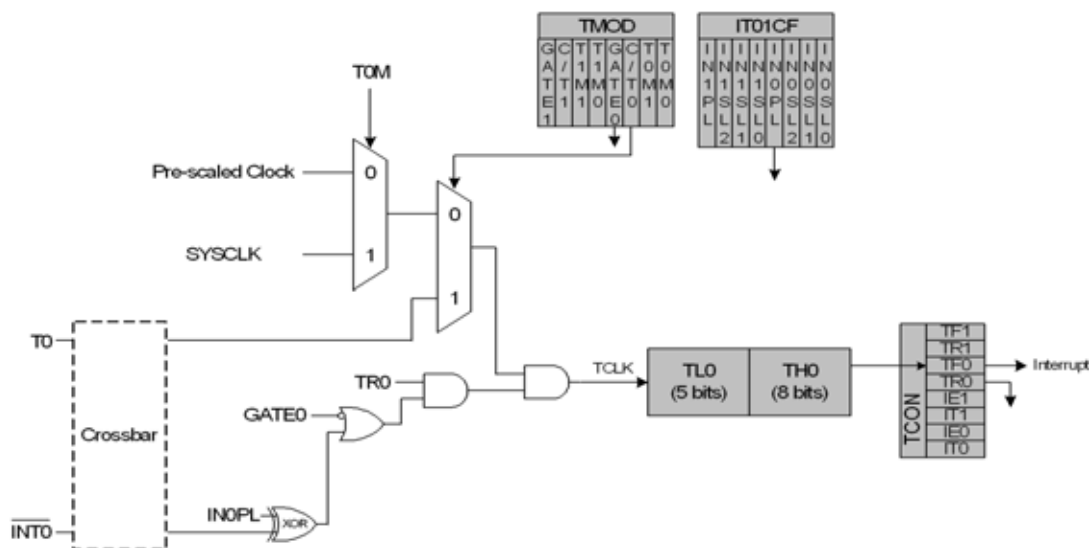


Figure 26.1. T0 Mode 0 Block Diagram

26.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

26.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 16.7 for details on the external input signals $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$).



In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates or overflow conditions for other peripherals. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

SFR Definition 26.9. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2CSS	T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; SFR Page = 0; Bit-Addressable

Bit	Name	Function
7	TF2H	Timer 2 High Byte Overflow Flag. Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	Timer 2 Low Byte Overflow Flag. Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	Timer 2 Low Byte Interrupt Enable. When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	Timer 2 Low-Frequency Oscillator Capture Enable. When set to 1, this bit enables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL.
3	T2SPLIT	Timer 2 Split Mode Enable. When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.
2	TR2	Timer 2 Run Control. Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	T2CSS	Timer 2 Capture Source Select. This bit selects the source of a capture event when bit T2CE is set to 1. 0: Capture source is USB SOF event. 1: Capture source is falling edge of Low-Frequency Oscillator.
0	T2XCLK	Timer 2 External Clock Select. This bit selects the external clock source for Timer 2. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

C8051F380/1/2/3/4/5/6/7/C

SFR Definition 26.25. TMR5RLL: Timer 5 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCA; SFR Page = F

Bit	Name	Function
7:0	TMR5RLL[7:0]	Timer 5 Reload Register Low Byte. TMR5RLL holds the low byte of the reload value for Timer 5.

SFR Definition 26.26. TMR5RLH: Timer 5 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCB; SFR Page = F

Bit	Name	Function
7:0	TMR5RLH[7:0]	Timer 5 Reload Register High Byte. TMR5RLH holds the high byte of the reload value for Timer 5.

SFR Definition 26.27. TMR5L: Timer 5 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC; SFR Page = F

Bit	Name	Function
7:0	TMR5L[7:0]	Timer 5 Low Byte. In 16-bit mode, the TMR5L register contains the low byte of the 16-bit Timer 5. In 8-bit mode, TMR5L contains the 8-bit low byte timer value.

27.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 4. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH4) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 4 operates as a watchdog timer (WDT). The Module 4 high byte is compared to the PCA counter high byte; the Module 4 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

27.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2–CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 4 is forced into software timer mode.
- Writes to the Module 4 mode register (PCA0CPM4) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH4 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH4. Upon a PCA0CPH4 write, PCA0H plus the offset held in PCA0CPL4 is loaded into PCA0CPH4 (See Figure 27.10).

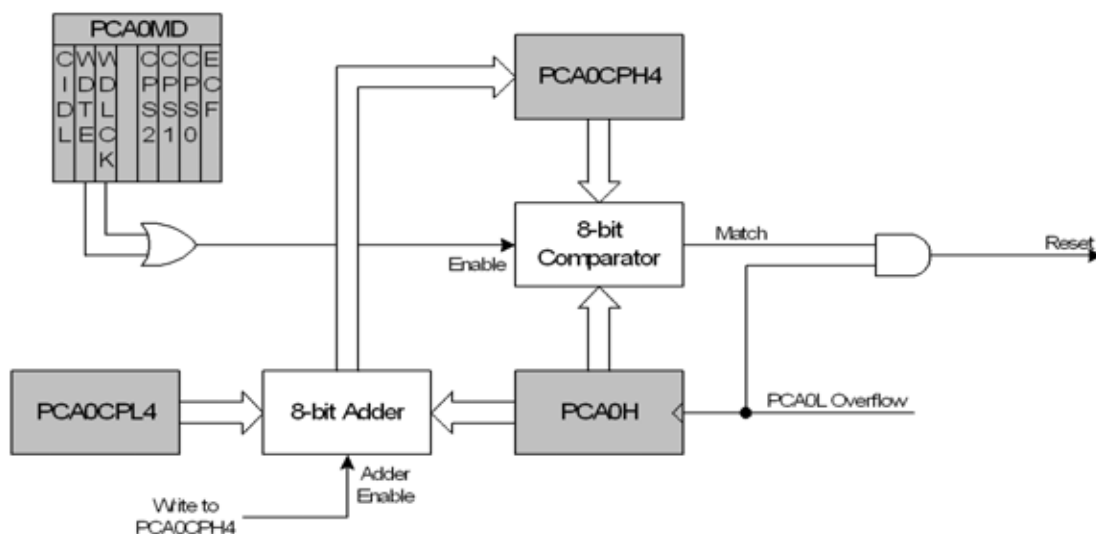


Figure 27.10. PCA Module 4 with Watchdog Timer Enabled

C2 Register Definition 28.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	Flash Programming Control Register. This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 28.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xAD

Bit	Name	Function	
7:0	FPDAT[7:0]	C2 Flash Programming Data Register. This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	Command
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase

C8051F380/1/2/3/4/5/6/7/C

28.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming 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 pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 28.1.

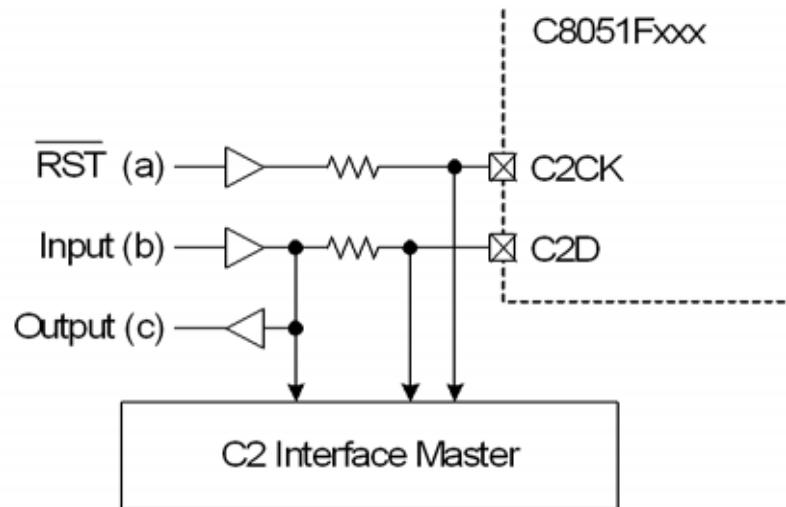


Figure 28.1. Typical C2 Pin Sharing

The configuration in Figure 28.1 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.