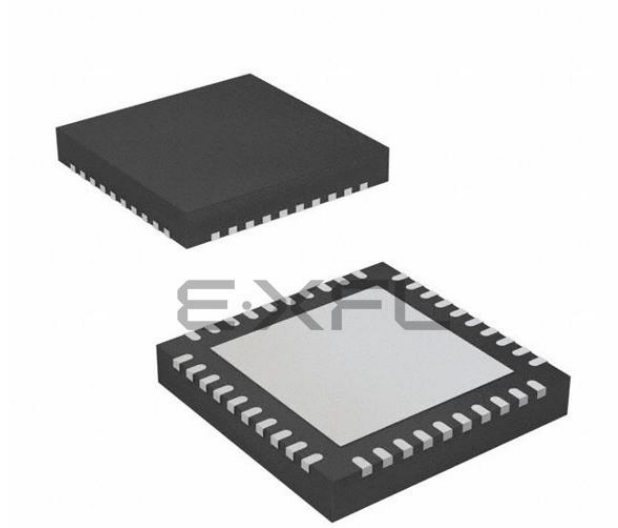

Welcome to E-XFL.COM

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	33
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f509-im

SFR Definition 6.5. ADC0H: ADC0 Data Word MSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBE; SFR Page = 0x00

Bit	Name	Function
7:0	ADC0H[7:0]	**ADC0 Data Word High-Order Bits.** For AD0LJST = 0 and AD0RPT as follows: 00: Bits 3–0 are the upper 4 bits of the 12-bit result. Bits 7–4 are 0000b. 01: Bits 4–0 are the upper 5 bits of the 14-bit result. Bits 7–5 are 000b. 10: Bits 5–0 are the upper 6 bits of the 15-bit result. Bits 7–6 are 00b. 11: Bits 7–0 are the upper 8 bits of the 16-bit result. For AD0LJST = 1 (AD0RPT must be 00): Bits 7–0 are the most-significant bits of the ADC0 12-bit result.

SFR Definition 6.6. ADC0L: ADC0 Data Word LSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBD; SFR Page = 0x00

Bit	Name	Function
7:0	ADC0L[7:0]	**ADC0 Data Word Low-Order Bits.** For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the ADC0 Accumulated Result. For AD0LJST = 1 (AD0RPT must be '00'): Bits 7–4 are the lower 4 bits of the 12-bit result. Bits 3–0 are 0000b.

SFR Definition 9.3. CPT1CN: Comparator1 Control

Bit	7	6	5	4	3	2	1	0
Name	CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP[1:0]		CP1HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9D; SFR Page = 0x00

Bit	Name	Function
7	CP1EN	**Comparator1 Enable Bit.** 0: Comparator1 Disabled. 1: Comparator1 Enabled.
6	CP1OUT	**Comparator1 Output State Flag.** 0: Voltage on CP1+ < CP1–. 1: Voltage on CP1+ > CP1–.
5	CP1RIF	**Comparator1 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.
4	CP1FIF	**Comparator1 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.
3:2	CP1HYP[1:0]	**Comparator1 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP1HYN[1:0]	**Comparator1 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

9.1. Comparator Multiplexer

C8051F50x/F51x devices include an analog input multiplexer for each of the comparators to connect Port I/O pins to the comparator inputs. The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 9.5). The CMX0P3–CMX0P0 bits select the Comparator0 positive input; the CMX0N3–CMX0N0 bits select the Comparator0 negative input. Similarly, the Comparator1 inputs are selected in the CPT1MX register using the CMX1P3-CMX1P0 bits and CMX1N3-CMX1N0 bits. The same pins are available to both multiplexers at the same time and can be used by both comparators simultaneously.

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 “20.6. Special Function Registers for Accessing and Configuring Port I/O” on page 191).

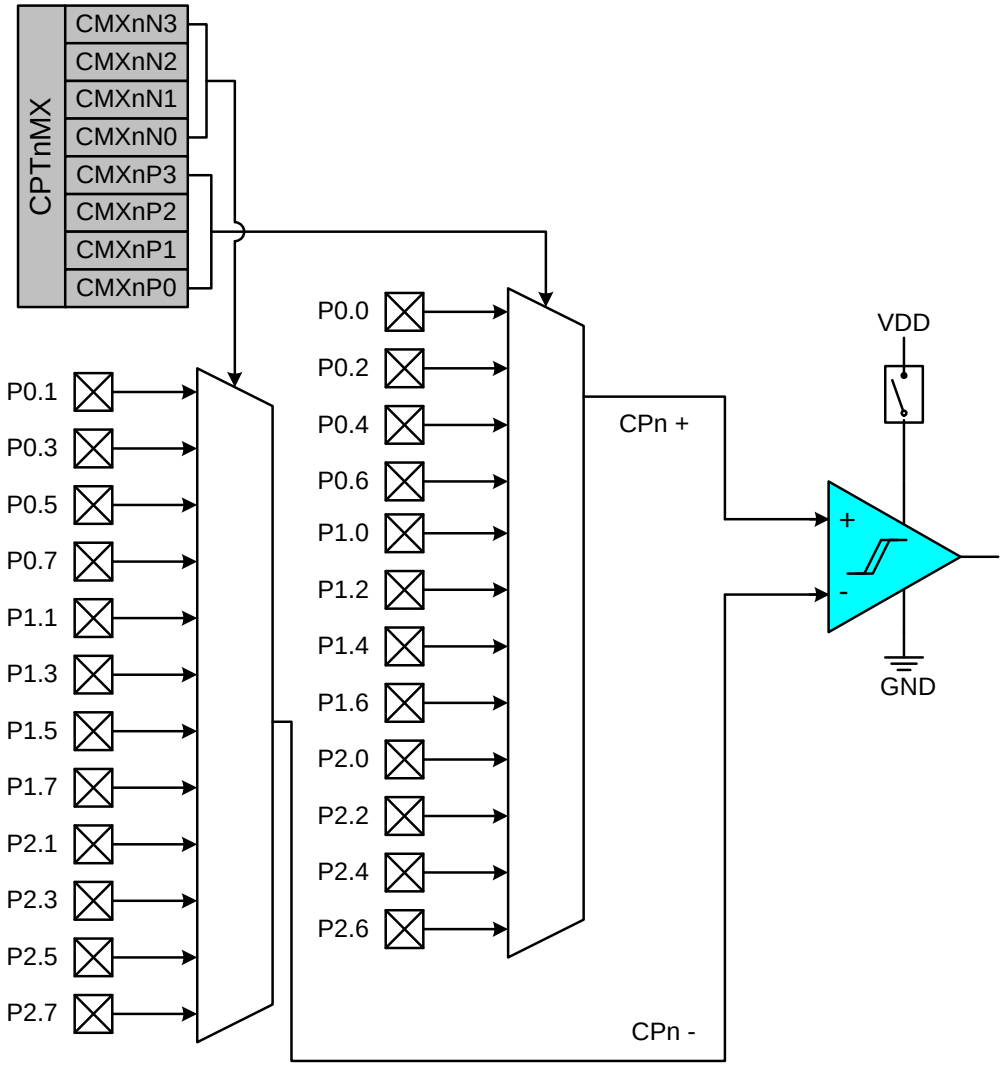

Figure 9.3. Comparator Input Multiplexer Block Diagram

Table 11.1. CIP-51 Instruction Set Summary (Prefetch-Enabled)(Continued)

Mnemonic	Description	Bytes	Clock Cycles
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	2
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	4-7*
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
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2

Note: Certain instructions take a variable number of clock cycles to execute depending on instruction alignment and the FLRT setting (SFR Definition 15.3).

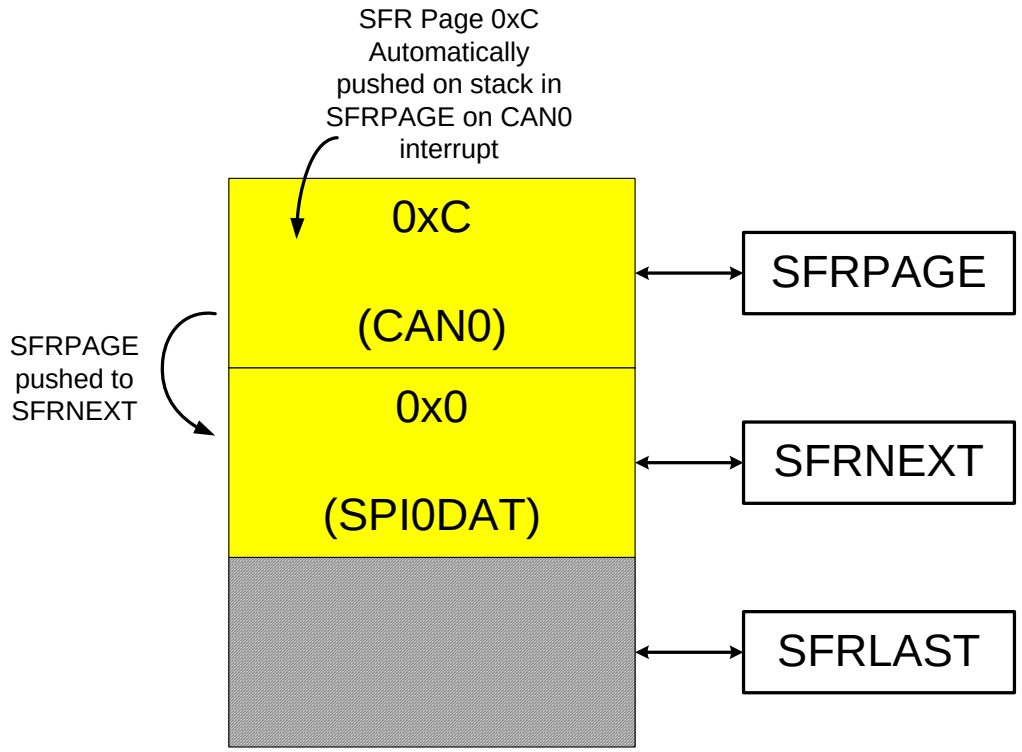

Figure 13.3. SFR Page Stack After CAN0 Interrupt Occurs

While in the CAN0 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a high priority interrupt, while the CAN0 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR Page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 0x0C for CAN0) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x00 for SPI0DAT) is pushed down to the SFRLAST register, the "bottom" of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 13.4.

14. Interrupts

The C8051F50x/F51x devices include an extended interrupt system supporting a total of 18 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE, EIE1, or EIE2). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Note: Any instruction that clears a bit to disable an interrupt should be immediately followed by an instruction that has two or more opcode bytes. Using EA (global interrupt enable) as an example:

```
// in 'C':
EA = 0; // clear EA bit.
EA = 0; // this is a dummy instruction with two-byte opcode.

; in assembly:
CLR EA ; clear EA bit.
CLR EA ; this is a dummy instruction with two-byte opcode.
```

For example, if an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears a bit to disable an interrupt source), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the enable bit will return a 0 inside the interrupt service routine. When the bit-clearing opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

14.1. MCU Interrupt Sources and Vectors

The C8051F50x/F51x MCUs support 18 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 14.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 15.2. FLKEY: Flash Lock and Key

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

SFR Address = 0xB7; SFR Page = All Pages

Bit	Name	Function
7:0	FLKEY[7:0]	**Flash Lock and Key Register.** Write: This register provides a lock and key function for Flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a Flash write or erase operation is attempted while these operations are disabled, the Flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to Flash, it can intentionally lock the Flash by writing a non-0xA5 value to FLKEY from software. Read: When read, bits 1–0 indicate the current Flash lock state. 00: Flash is write/erase locked. 01: The first key code has been written (0xA5). 10: Flash is unlocked (writes/erases allowed). 11: Flash writes/erases disabled until the next reset.

17.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until V_{DD} settles above V_{RST}. A delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 17.2. plots the power-on and V_{DD} monitor reset timing.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000) software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} monitor is enabled following a power-on reset.

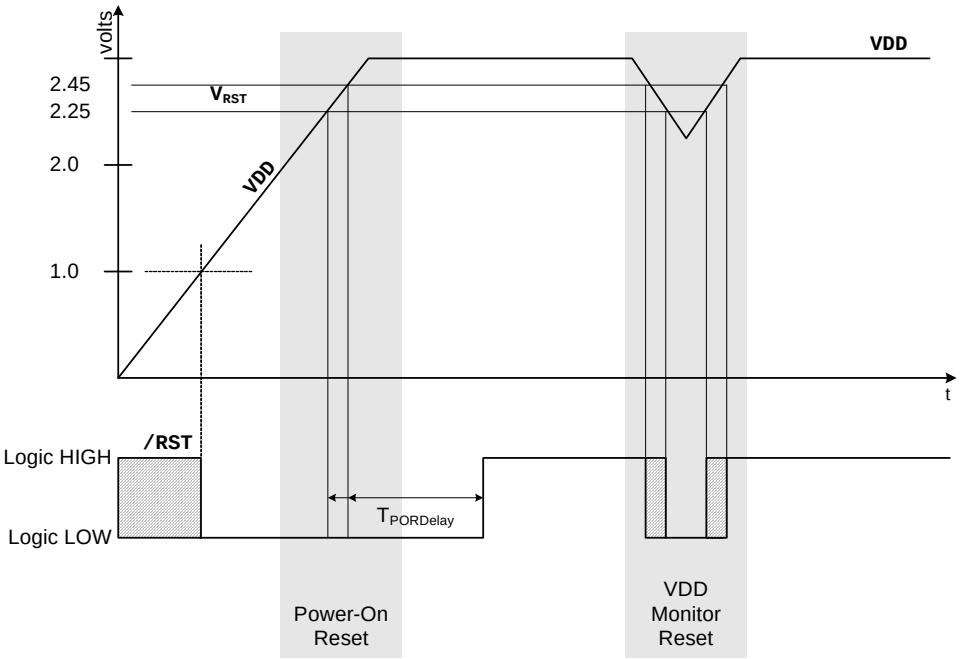

Figure 17.2. Power-On and V_{DD} Monitor Reset Timing

17.2. Power-Fail Reset/V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST}, the power supply monitor will drive the $\overline{\text{RST}}$ pin low and hold the CIP-51 in a reset state (see Figure 17.2). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The V_{DD} monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by code and a software reset is performed, the V_{DD} monitor will still be disabled after the reset. **To protect the integrity of Flash contents, the V_{DD} monitor must be enabled to the higher setting (VDMLVL = 1) and selected as a reset source if software contains routines which erase or write Flash memory. If the V_{DD} monitor is not enabled and set to the high level, any erase or write performed on Flash memory will cause a Flash Error device reset.**

18.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 101 or 111

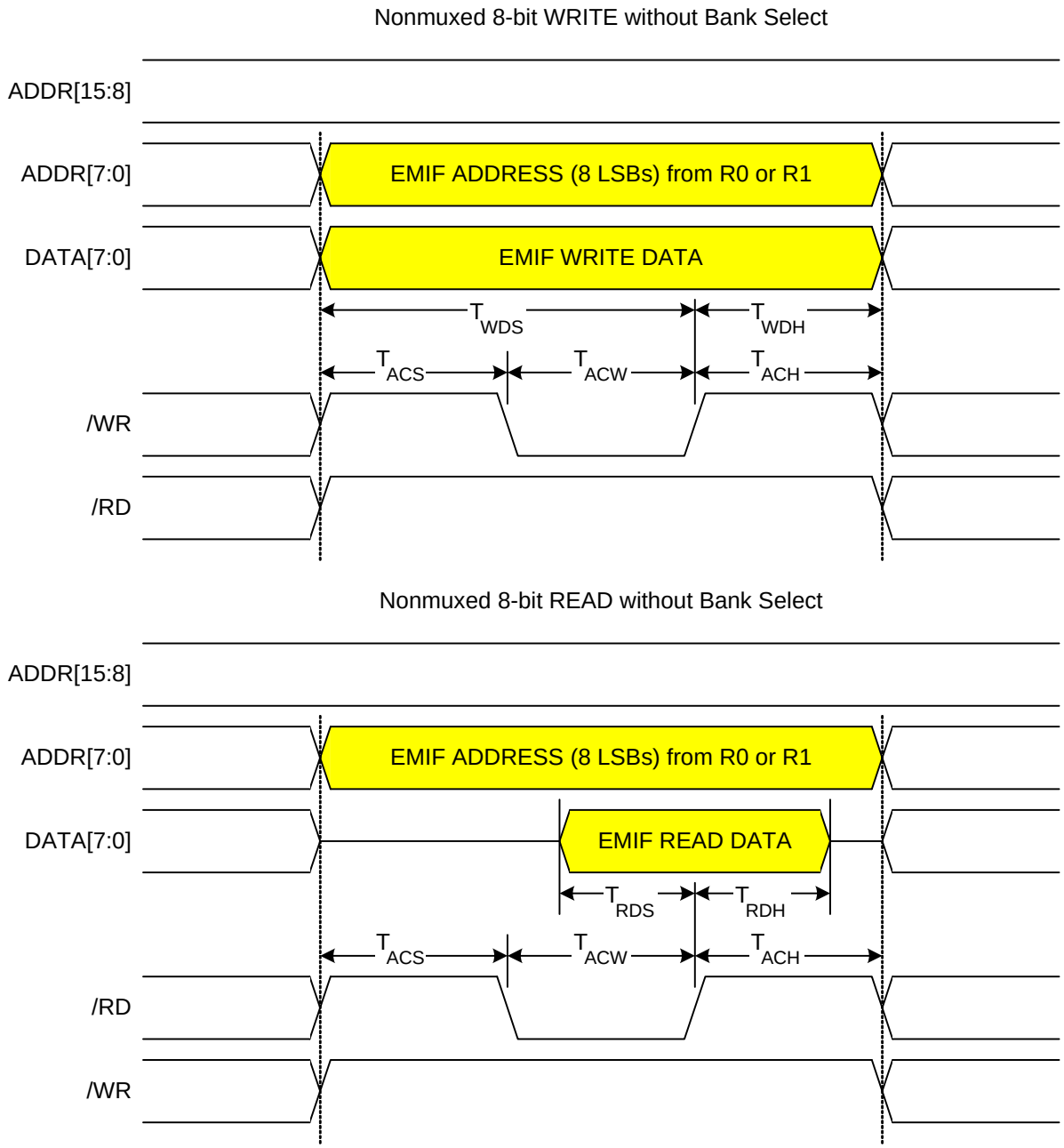

Figure 18.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

18.6.2.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 001 or 011

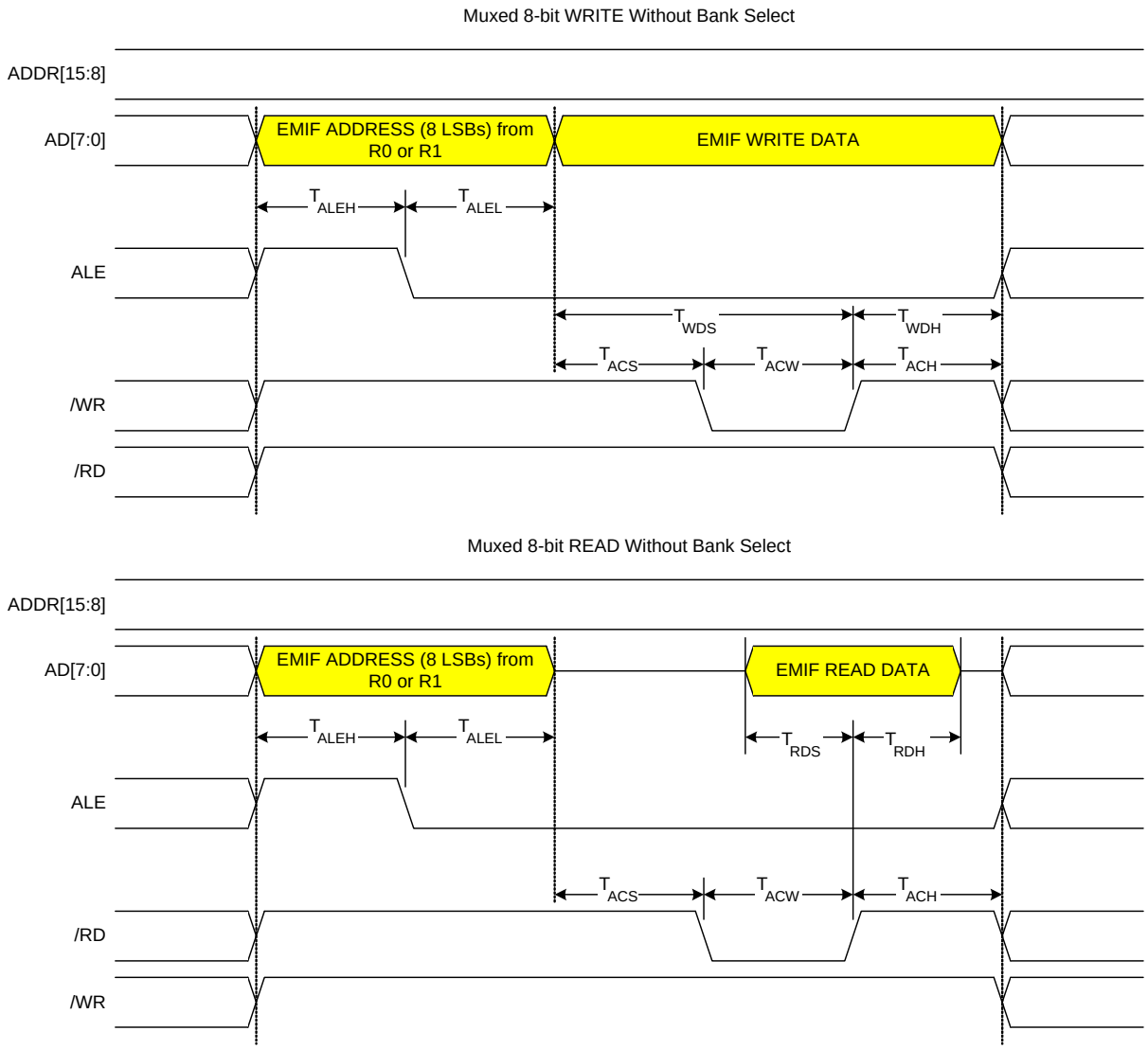

Figure 18.8. Multiplexed 8-bit MOVX without Bank Select Timing

Table 18.3. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns
***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

20.1.3. Interfacing Port I/O in a Multi-Voltage System

All Port I/O are capable of interfacing to digital logic operating at a supply voltage higher than VDD and less than 5.25 V. Connect the VIO pin to the voltage source of the interface logic.

20.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P3.7 can be assigned to various analog, digital, and external interrupt functions. P4.0-P4.7 can be assigned to only digital functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

20.2.1. Assigning Port I/O Pins to Analog Functions

Table 20.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 20.1 shows the potential mapping of Port I/O to each analog function.

Table 20.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P3.7*	ADC0MX, PnSKIP
Comparator0 or Compartor1 Input	P0.0–P2.7	CPT0MX, CPT1MX, PnSKIP
Voltage Reference (VREF0)	P0.0	REF0CN, PnSKIP
External Oscillator in Crystal Mode (XTAL1)	P0.2	OSCXCN, PnSKIP
External Oscillator in RC, C, or Crystal Mode (XTAL2)	P0.3	OSCXCN, PnSKIP
***Note:** P3.1–P3.7 are only available on the 48-pin and 40-pin packages		

20.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 20.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 20.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, CAN0, LIN0, CP0, CP0A, CP1, CP1A, SYSCLK, PCA0 (CEX0-5 and ECI), T0 or T1.	Any Port pin available for assignment by the Crossbar. This includes P0.0–P4.7* pins which have their PnSKIP bit set to 0. **Note:** The Crossbar will always assign UART0 pins to P0.4 and P0.5 and always assign CAN0 to P0.6 and P0.7.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0–P4.7*	P0SKIP, P1SKIP, P2SKIP, P3SKIP
***Note:** P3.1–P3.7 and P4.0 are only available on the 48-pin and 40pin packages. P4.1–P4.7 are only available on the 48-pin package. A skip register is not available for P4.		

SFR Definition 20.10. P3MASK: Port 3 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P3MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAF; SFR Page = 0x00

Bit	Name	Function
7:0	P3MASK[7:0]	**Port 1 Mask Value.** Selects P3 pins to be compared to the corresponding bits in P3MAT. 0: P3.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P3.n pin logic value is compared to P3MAT.n.
Note: P3.1–P3.7 are only available on the 48-pin and 40-pin packages		

SFR Definition 20.11. P3MAT: Port 3 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P3MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xAE; SFR Page = 0x00

Bit	Name	Function
7:0	P3MAT[7:0]	**Port 3 Match Value.** Match comparison value used on Port 3 for bits in P3MAT which are set to 1. 0: P3.n pin logic value is compared with logic LOW. 1: P3.n pin logic value is compared with logic HIGH.
Note: P3.1–P3.7 are only available on the 48-pin and 40-pin packages		

SFR Definition 22.1. CAN0CFG: CAN Clock Configuration

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

SFR Address = 0x92; SFR Page = 0x0C

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = Don't Care.
1:0	SYSDIV[1:0]	**CAN System Clock Divider Bits.** The CAN controller clock is derived from the CIP-51 system clock. The CAN controller clock must be less than or equal to 25 MHz. 00: CAN controller clock = System Clock/1. 01: CAN controller clock = System Clock/2. 10: CAN controller clock = System Clock/4. 11: CAN controller clock = System Clock/8.

overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

23.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. Note that a clock source is required for free timeout detection, even in a slave-only implementation.

23.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

SMBus interrupts are generated for each data byte or slave address that is transferred. The point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e. sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e. receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See Section 23.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section 23.4.2; Table 23.4 provides a quick SMB0CN decoding reference.

23.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

$$\text{Baud Rate} = \frac{\text{SYSCLK}}{(65536 - (\text{SBRLH0:SBRLL0}))} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

Equation 24.1. UART0 Baud Rate

A quick reference for typical baud rates and clock frequencies is given in Table 24.1.

Table 24.1. Baud Rate Generator Settings for Standard Baud Rates

	Target Baud Rate (bps)	Actual Baud Rate (bps)	Baud Rate Error	Oscillator Divide Factor	SB0PS[1:0] (Prescaler Bits)	Reload Value in SBRLH0:SBRLL0
SYSCLK = 48	230400	230769	0.16%	208	11	0xFF98
	115200	115385	0.16%	416	11	0xFF30
	57600	57554	0.08%	834	11	0xFE5F
	28800	28812	0.04%	1666	11	0xFCBF
	14400	14397	0.02%	3334	11	0xF97D
	9600	9600	0.00%	5000	11	0xF63C
	2400	2400	0.00%	20000	11	0xD8F0
	1200	1200	0.00%	40000	11	0xB1E0
SYSCLK = 24	230400	230769	0.16%	104	11	0xFFCC
	115200	115385	0.16%	208	11	0xFF98
	57600	57692	0.16%	416	11	0xFF30
	28800	28777	0.08%	834	11	0xFE5F
	14400	14406	0.04%	1666	11	0xFCBF
	9600	9600	0.00%	2500	11	0xFB1E
	2400	2400	0.00%	10000	11	0xEC78
	1200	1200	0.00%	20000	11	0xD8F0
SYSCLK = 12	230400	230769	0.16%	52	11	0xFFE6
	115200	115385	0.16%	104	11	0xFFCC
	57600	57692	0.16%	208	11	0xFF98
	28800	28846	0.16%	416	11	0xFF30
	14400	14388	0.08%	834	11	0xFE5F
	9600	9600	0.00%	1250	11	0xFD8F
	2400	2400	0.00%	5000	11	0xF63C
	1200	1200	0.00%	10000	11	0xEC78

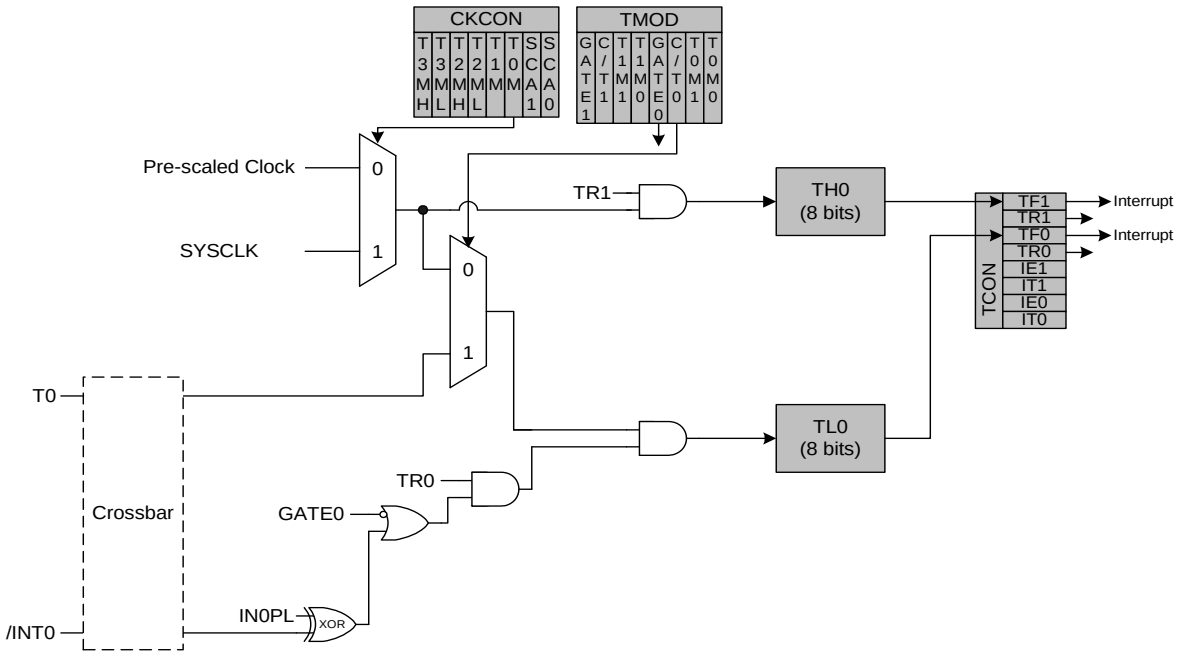

Figure 26.3. T0 Mode 3 Block Diagram

SFR Definition 26.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x89; SFR Page = All Pages

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 14.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 14.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

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 (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.

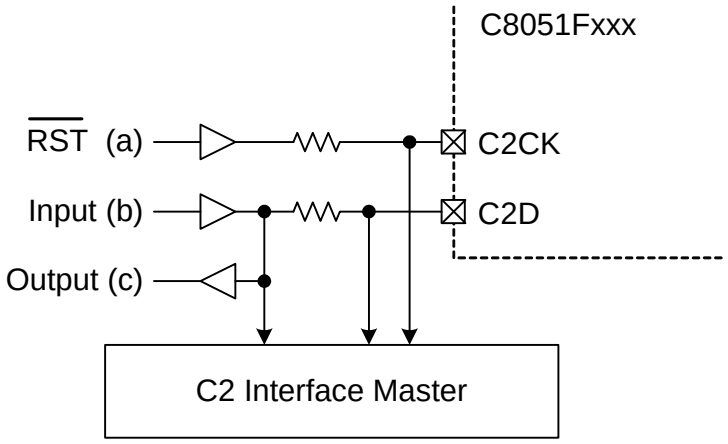

Figure 28.1. Typical C2 Pin Sharing

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 RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.