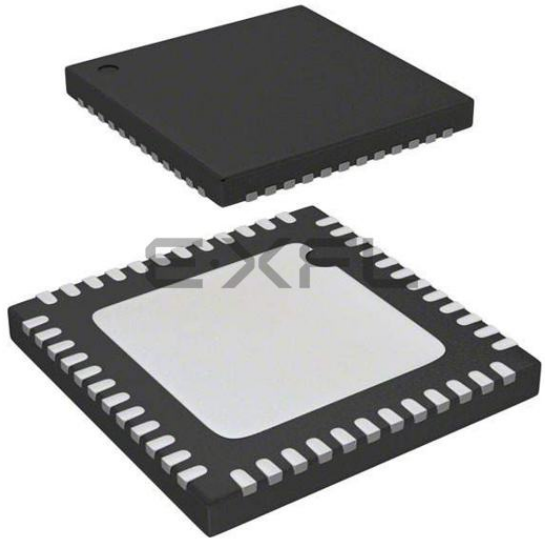

Welcome to E-XFL.COM

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, Temp Sensor, WDT
Number of I/O	39
Program Memory Size	15KB (15K x 8)
Program Memory Type	FLASH
EEPROM Size	32 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f704-gmr

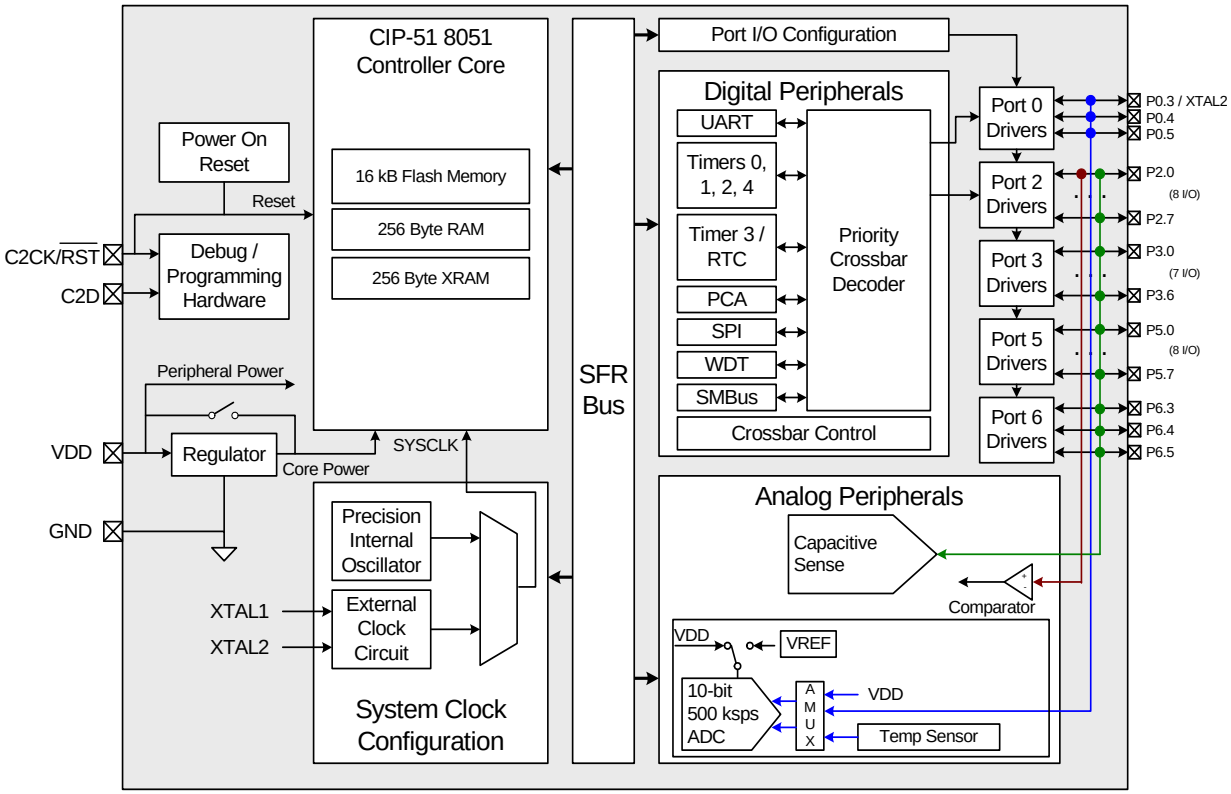

Figure 1.7. C8051F716 Block Diagram

7. QFN-32 Package Specifications

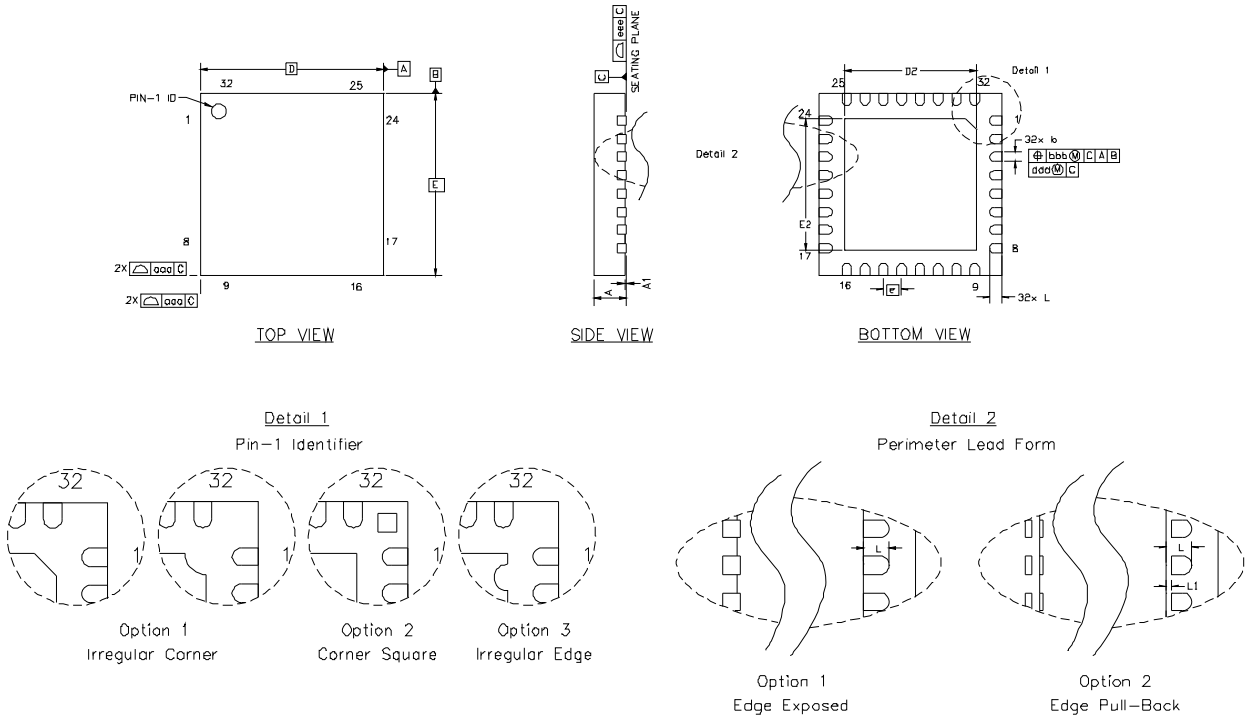

Figure 7.1. QFN-32 Package Drawing

Table 7.1. QFN-32 Package Dimensions

Dimension	Min	Typ	Max
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D		5.00 BSC.	
D2	3.50	3.60	3.70
e		0.50 BSC.	
E		5.00 BSC.	

Dimension	Min	Typ	Max
E2	3.50	3.60	3.70
L	0.30	0.35	0.40
L1	0.00	—	0.10
aaa		0.15	
bbb		0.10	
ddd		0.05	
eee		0.08	

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, L and L1 which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

10.5. ADC0 Analog Multiplexer

ADC0 on the C8051F700/2/4/6/8 and C8051F710/2/4/6 uses an analog input multiplexer to select the positive input to the ADC. Any of the following may be selected as the positive input: Port 0 or Port 1 I/O pins, the on-chip temperature sensor, or the positive power supply (V_{DD}). The ADC0 input channel is selected in the ADC0MX register described in SFR Definition 10.9.

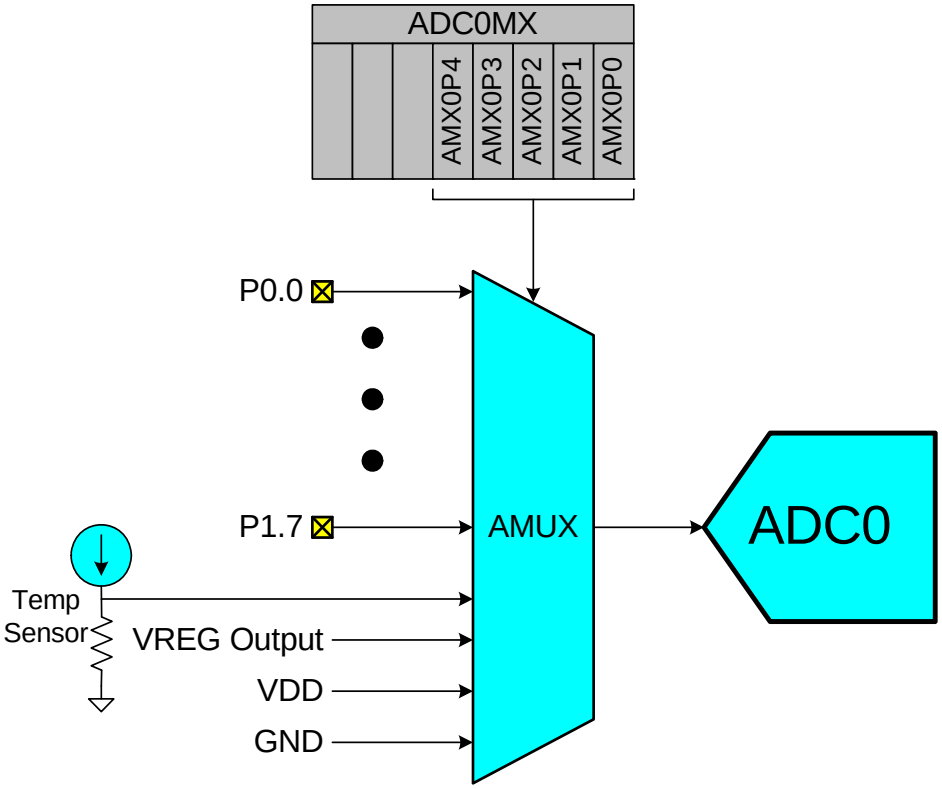

Figure 10.6. ADC0 Multiplexer Block Diagram

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 the corresponding bit in register PnMDIN to 0. To force the Crossbar to skip a Port pin, set the corresponding bit in register PnSKIP to 1. See Section “28. Port Input/Output” on page 180 for more Port I/O configuration details.

11. Temperature Sensor

An on-chip temperature sensor is included on the C8051F700/2/4/6/8 and C8051F710/2/4/6 which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 11.1. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 12.1. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 9.12 for the slope and offset parameters of the temperature sensor.

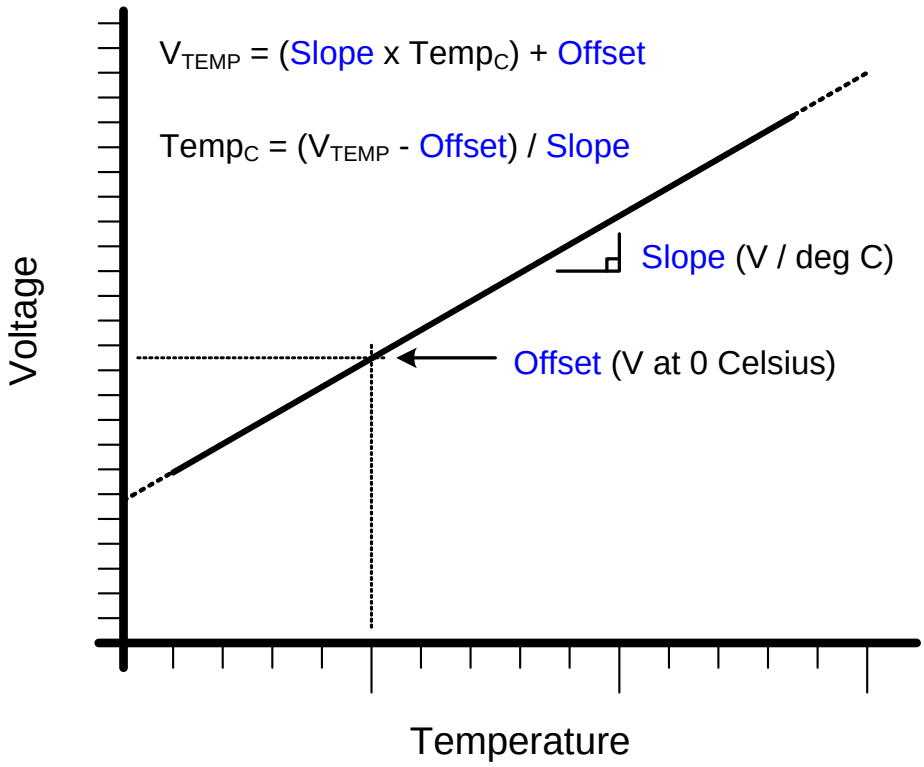

Figure 11.1. Temperature Sensor Transfer Function

11.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the ADC's input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 0 °C.

Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.

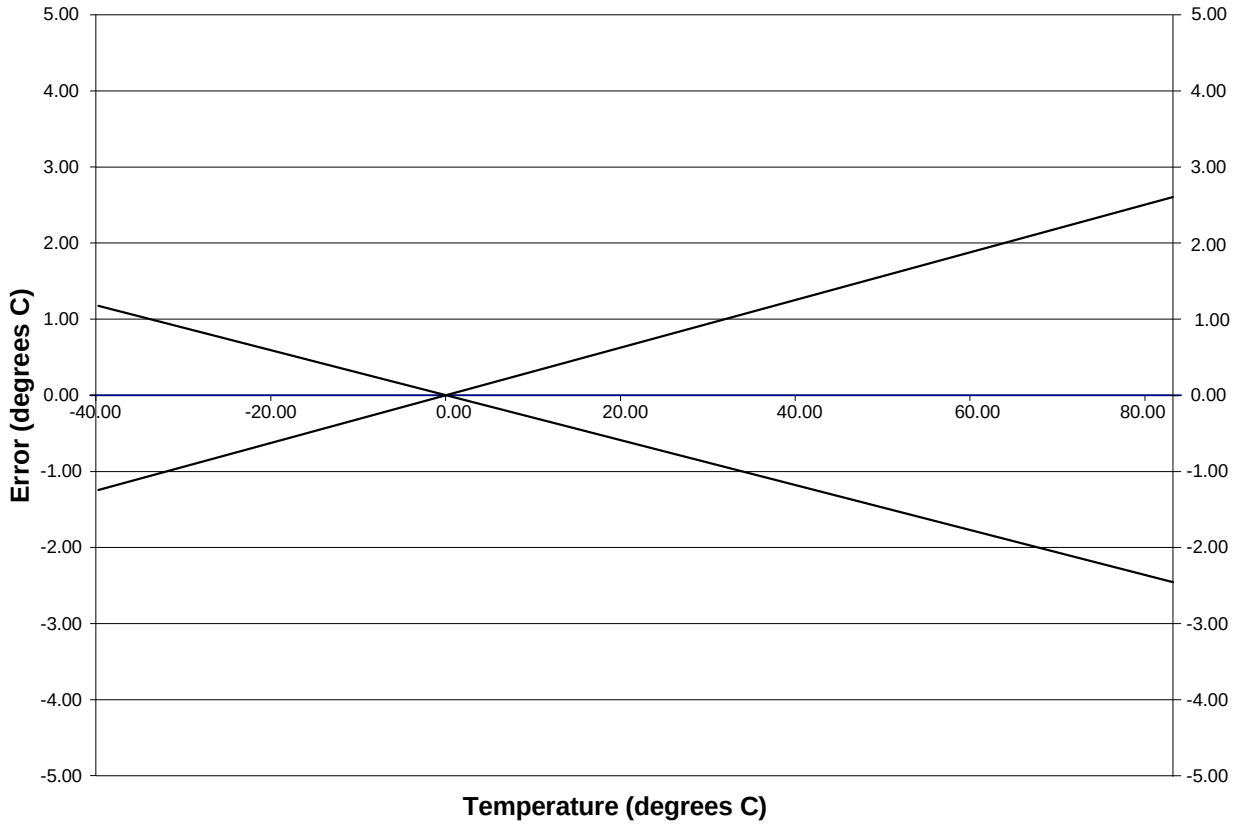

Figure 11.2. Temperature Sensor Error with 1-Point Calibration at 0 Celsius

SFR Definition 15.12. CS0MX: Capacitive Sense Mux Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CS0UC		CS0MX[5:0]					
Type	R/W	R/W	R/W					
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9C; SFR Page = 0

Bit	Name	Description
7	CS0UC	**CS0 Unconnected.** Disconnects CS0 from all port pins, regardless of the selected channel. 0: CS0 connected to port pins 1: CS0 disconnected from port pins
6	Reserved	Write = 0b
5:0	CS0MX[5:0]	**CS0 Mux Channel Select.** Selects one of the 38 input channels for Capacitive Sense conversion.

Value	64-pin	48-pin	32-pin	24-pin	Value	64-pin	48-pin	32-pin	24-pin
000000	P2.0	P2.0	P2.0	P2.0	**010011**	P4.3	P4.3	—	P4.3
000001	P2.1	P2.1	P2.1	P2.1	**010100**	P4.4	—	—	P4.4
000010	P2.2	P2.2	P2.2	P2.2	**010101**	P4.5	—	—	P4.5
000011	P2.3	P2.3	P2.3	P2.3	**010110**	P4.6	—	—	P4.6
000100	P2.4	P2.4	P2.4	P2.4	**010111**	P4.7	—	—	P4.7
000101	P2.5	P2.5	P2.5	P2.5	**011000**	P5.0	P5.0	P5.0	—
000110	P2.6	P2.6	P2.6	P2.6	**011001**	P5.1	P5.1	P5.1	—
000111	P2.7	P2.7	P2.7	P2.7	**011010**	P5.2	P5.2	P5.2	—
001000	P3.0	—	P3.0	—	**011011**	P5.3	P5.3	P5.3	—
001001	P3.1	—	P3.1	—	**011100**	P5.4	P5.4	P5.4	—
001010	P3.2	—	P3.2	—	**011101**	P5.5	P5.5	P5.5	—
001011	P3.3	—	P3.3	—	**011110**	P5.6	P5.6	P5.6	—
001100	P3.4	P3.4	P3.4	—	**011111**	P5.7	P5.7	P5.7	—
001101	P3.5	P3.5	P3.5	—	**100000**	P6.0	—	—	—
001110	P3.6	P3.6	P3.6	—	**100001**	P6.1	—	—	—
001111	P3.7	P3.7	—	—	**100010**	P6.2	—	—	—
010000	P4.0	P4.0	—	P4.0	**100011**	P6.3	P6.3	P6.3	—
010001	P4.1	P4.1	—	P4.1	**100100**	P6.4	P6.4	P6.4	P6.4
010010	P4.2	P4.2	—	P4.2	**100101**	P6.5	P6.5	P6.5	P6.5

18.4.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 18.2. See Section “18.6.1. Non-Multiplexed Mode” on page 120 for more information about Non-multiplexed operation.

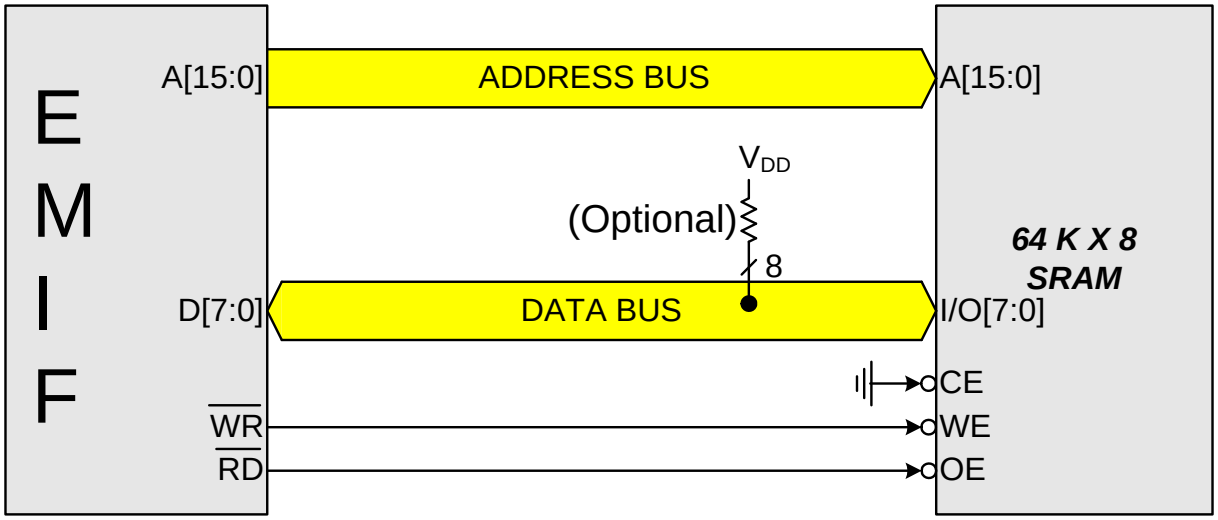

Figure 18.2. Non-multiplexed Configuration Example

18.6.1. Non-Multiplexed Mode

18.6.1.1. 16-bit MOVX: EMI0CF[4:2] = 101, 110, or 111

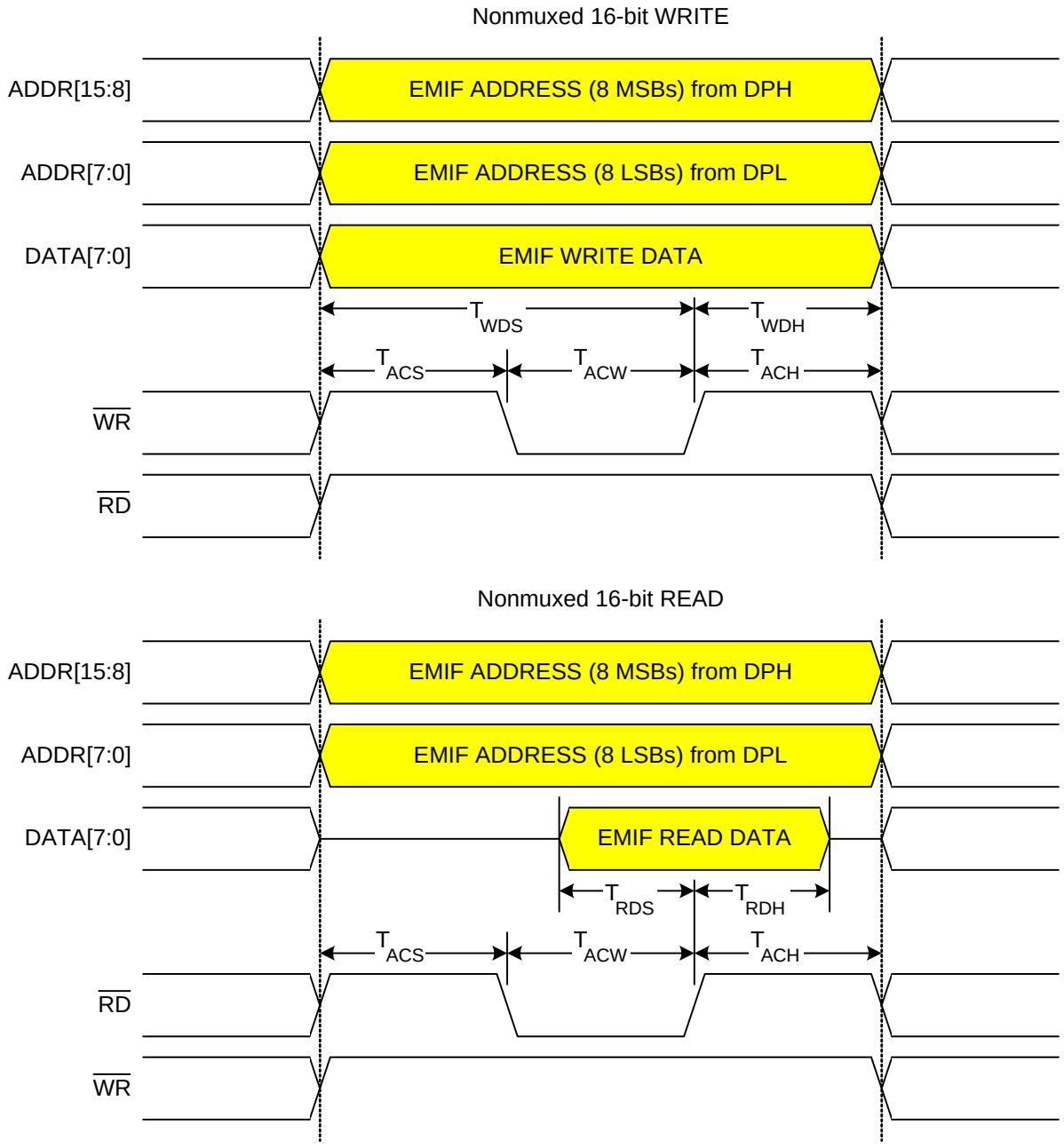

Figure 18.4. Non-multiplexed 16-bit MOVX Timing

Table 18.1. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	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	T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	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).				

Table 20.2. Special Function Registers (Continued)

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

Register	Address	Page	Description	Page
CS0CN	0x9A	0	CS0 Control	88
CS0DH	0xAA	0	CS0 Data High	90
CS0DL	0xA9	0	CS0 Data Low	90
CS0CF	0x9E	0	CS0 Configuration	89
CS0MD1	0xAD	0	CS0 Mode 1	94
CS0MD2	0xBE	F	CS0 Mode 2	95
CS0MX	0x9C	0	CS0 Mux	97
CS0PM	0x9F	F	CS0 Pin Monitor	93
CS0SE	0x93	F	Auto Scan End Channel	91
CS0SS	0x92	F	Auto Scan Start Channel	91
CS0THH	0x97	0	CS0 Digital Compare Threshold High	92
CS0THL	0x96	0	CS0 Digital Compare Threshold Low	92
DERIVID	0xEC	F	Derivative Identification	128
DPH	0x83	All Pages	Data Pointer High	104
DPL	0x82	All Pages	Data Pointer Low	104
EEADDR	0xB6	All Pages	EEPROM Byte Address	156
EECNTL	0xC5	F	EEPROM Control	158
EEDATA	0xD1	All Pages	EEPROM Byte Data	157
EEKEY	0xC6	F	EEPROM Protect Key	159
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	142
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	143
EIP1	0xCE	F	Extended Interrupt Priority 1	144
EIP2	0xCF	F	Extended Interrupt Priority 2	145
EMI0CF	0xC7	F	EMIF Configuration	114
EMI0CN	0xAA	F	EMIF Control	113
EMI0TC	0xEE	F	EMIF Timing Control	119
FLKEY	0xB7	All Pages	Flash Lock And Key	154
HWID	0xC4	F	Hardware Identification	128
IE	0xA8	All Pages	Interrupt Enable	140
IP	0xB8	All Pages	Interrupt Priority	141
IT01CF	0xE4	F	INT0/INT1 Configuration	147
OSCICL	0xBF	F	Internal Oscillator Calibration	173
OSCICN	0xA9	F	Internal Oscillator Control	174
OSCXCN	0xB5	F	External Oscillator Control	176
P0	0x80	All Pages	Port 0 Latch	195
P0DRV	0xF9	F	Port 0 Drive Strength	197
P0MASK	0xF4	0	Port 0 Mask	192
P0MAT	0xF3	0	Port 0 Match	193

6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
7. Clear the PSWE and PSEE bits.
8. Restore previous interrupt state.

Steps 4–6 must be repeated for each 512-byte page to be erased.

Note: Flash security settings may prevent erasure of some Flash pages, such as the reserved area and the page containing the lock bytes. For a summary of Flash security settings and restrictions affecting Flash erase operations, please see Section “22.3. Security Options” on page 149.

22.1.3. Flash Write Procedure

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.**

The recommended procedure for writing a single byte in Flash is as follows:

1. Save current interrupt state and disable interrupts.
2. Ensure that the Flash byte has been erased (has a value of 0xFF).
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
8. Clear the PSWE bit.
9. Restore previous interrupt state.

Steps 5–7 must be repeated for each byte to be written.

Note: Flash security settings may prevent writes to some areas of Flash, such as the reserved area. For a summary of Flash security settings and restrictions affecting Flash write operations, please see Section “22.3. Security Options” on page 149.

22.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction.

Note: MOVX read instructions always target XRAM.

22.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, and erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock all Flash pages, starting at page 0, by writing a non-0xFF value to the lock byte. **Note that writing a non-0xFF value to the lock byte will lock all pages of FLASH from reads, writes, and erases, including the page containing the lock byte.**

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on

25. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source. Program execution begins at location 0x0000.

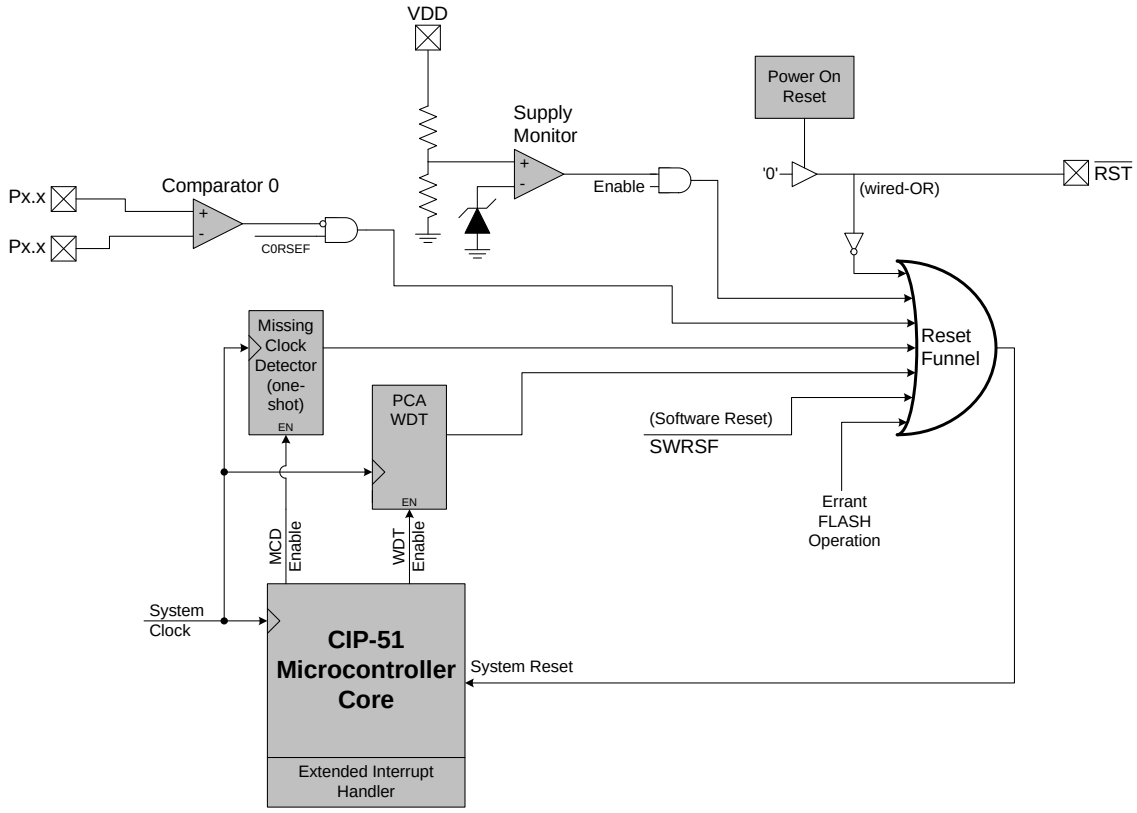

Figure 25.1. Reset Sources

28.1. Port I/O Modes of Operation

Port pins P0.0 - P6.5 use the Port I/O cell shown in Figure 28.2. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN registers. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled. Until the crossbar is enabled (XBARE = 1), both the high and low port I/O drive circuits are explicitly disabled on all crossbar pins.

28.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC input, Capacitive Sense input, external oscillator input/output, VREF output, or AGND connection should be configured for analog I/O (PnMDIN.n = 0). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog I/O will always read back a value of 0.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital I/O may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

28.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external event trigger functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the VDD or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the VDD supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption, and they may be globally disabled by setting WEAKPUD to 1. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

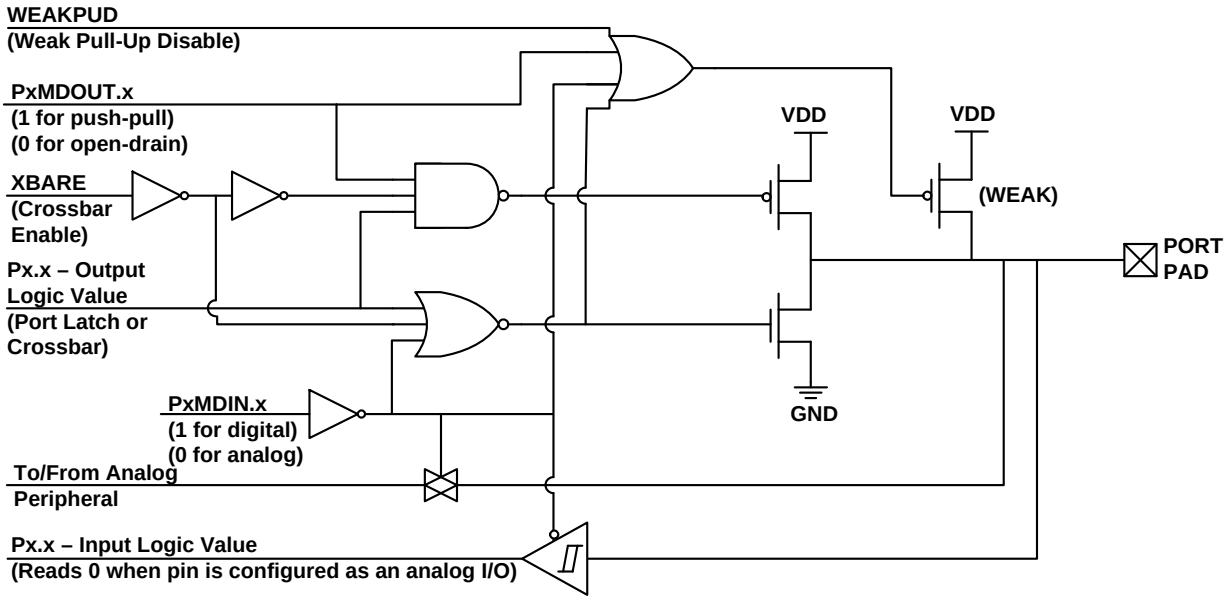

Figure 28.2. Port I/O Cell Block Diagram

SFR Definition 28.27. P4MDIN: Port 4 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF5; SFR Page = F

Bit	Name	Function
7:0	P4MDIN[7:0]	**Analog Configuration Bits for P4.7–P4.0 (respectively).** Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P4.n pin is configured for analog mode. 1: Corresponding P4.n pin is not configured for analog mode.

SFR Definition 28.28. P4MDOUT: Port 4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9A; SFR Page = F

Bit	Name	Function
7:0	P4MDOUT[7:0]	**Output Configuration Bits for P4.7–P4.0 (respectively).** These bits are ignored if the corresponding bit in register P4MDIN is logic 0. 0: Corresponding P4.n Output is open-drain. 1: Corresponding P4.n Output is push-pull.

30.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 30.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 30.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

30.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

30.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 30.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 30.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 30.5 for SMBus status decoding using the SMB0CN register.

Table 30.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (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 Receiver	0010	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
						Reschedule failed transfer	1	0	X	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		0	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
						Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

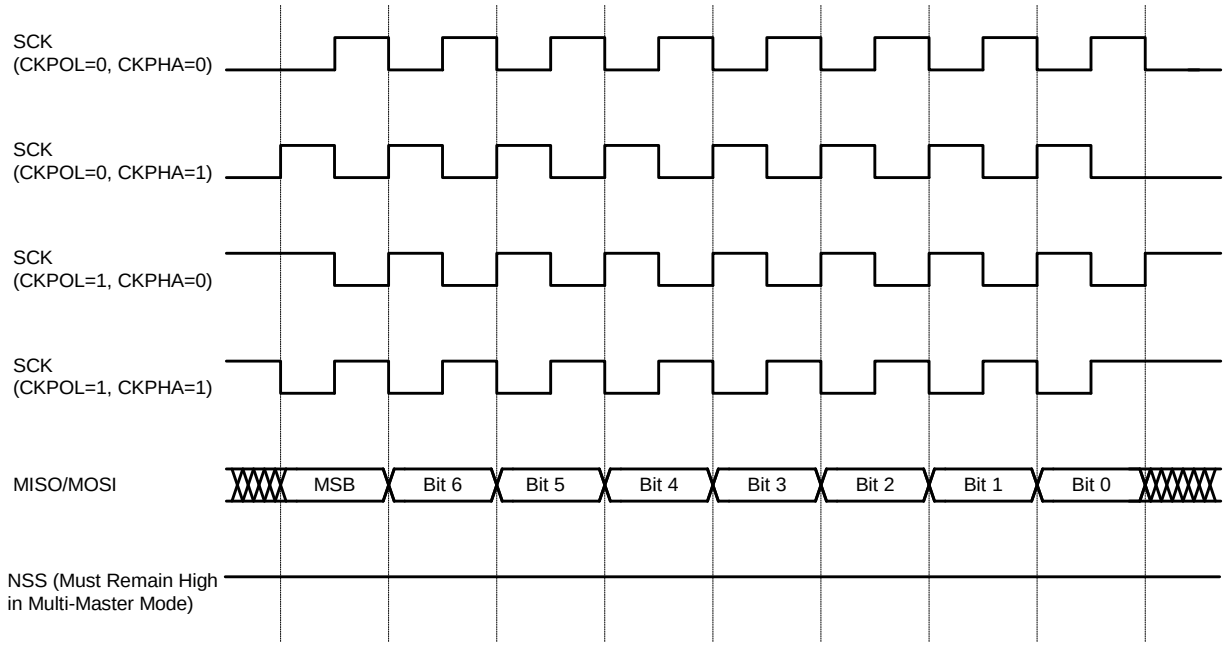

Figure 31.5. Master Mode Data/Clock Timing

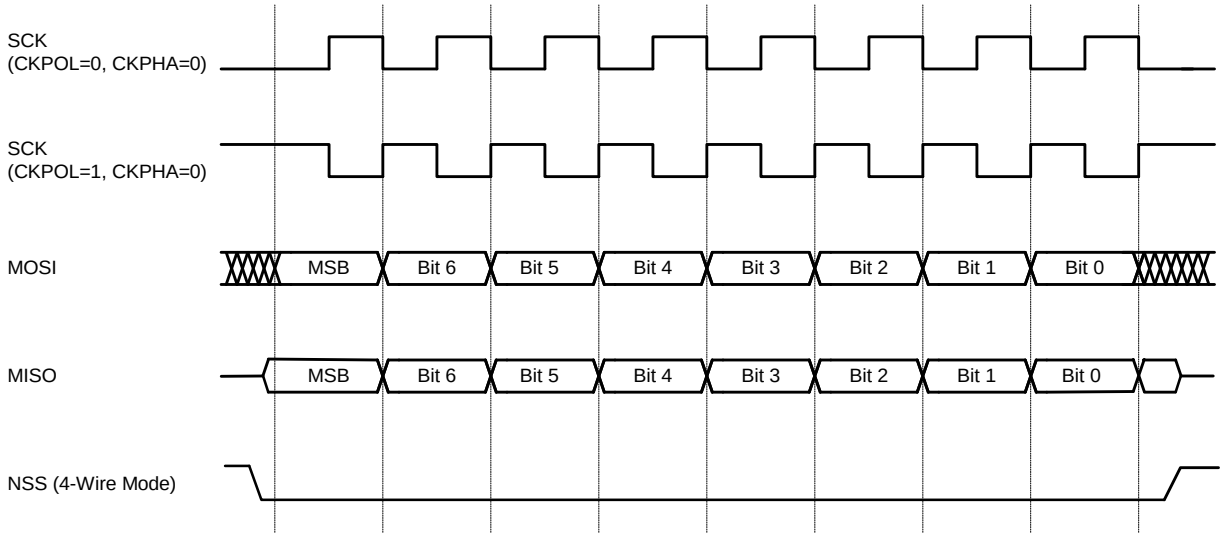

Figure 31.6. Slave Mode Data/Clock Timing (CKPHA = 0)

SFR Definition 31.3. SPI0CKR: SPI0 Clock Rate

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

SFR Address = 0xA2; SFR Page = F

Bit	Name	Function
7:0	SCR[7:0]	**SPI0 Clock Rate.** These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 31.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA3; SFR Page = 0

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

34.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

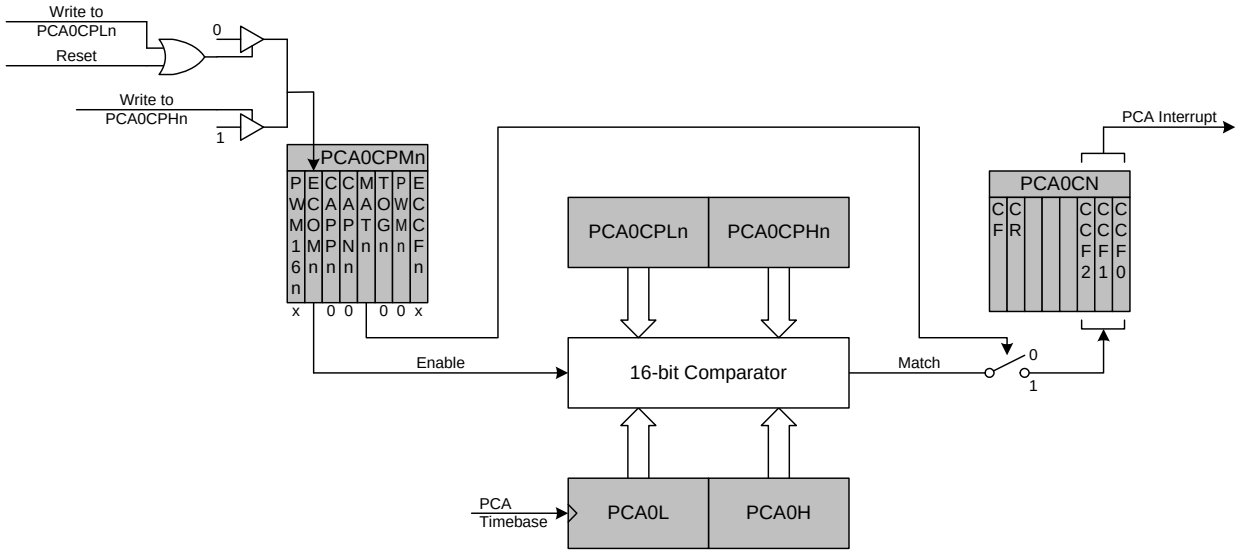

Figure 34.5. PCA Software Timer Mode Diagram